

# Contents

Foreword..... vii

## Invited Presentations

**Peter Grunwald** ..... 1  
*The Catch-Up Phenomenon in Bayesian Inference*

**Robin Hanson** ..... 3  
*Combinatorial Prediction Markets*

**Dan Klein** ..... 5  
*Unsupervised Learning for Natural Language Processing*

**Gabor Lugosi** ..... 7  
*Concentration Inequalities*

## Unsupervised, Semi-Supervised and Active Learning

**Kamalika Chaudhuri and Satish Rao** ..... 9  
*Learning Mixtures of Product Distributions Using Correlations and Independence*

**Kamalika Chaudhuri and Satish Rao** ..... 21  
*Beyond Gaussians: Spectral Methods for Learning Mixtures of Heavy-Tailed Distributions*

**Shai Ben-David, Tyler Lu and David Pal** ..... 33  
*Does Unlabeled Data Provably Help? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning*

**Maria-Florina Balcan, Steve Hanneke and Jennifer Wortman** ..... 45  
*The True Sample Complexity of Active Learning*

## On-Line Learning

**Elad Hazan and Satyen Kale** ..... 57  
*Extracting Certainty from Uncertainty: Regret Bounded by Variation in Costs*

<b>Kosuke Ishibashi, Kohei Hatano and Masayuki Takeda .....</b>	<b>69</b>
<i>Online Learning of Maximum <math>p</math>-Norm Margin Classifiers with Bias</i>	
<b>Subhash Khot and Ashok Kumar Ponnuswami .....</b>	<b>81</b>
<i>Minimizing Wide Range Regret with Time Selection Functions</i>	
<b>Other Directions</b>	
<b>Nir Ailon and Mehryar Mohri.....</b>	<b>87</b>
<i>An Efficient Reduction of Ranking to Classification</i>	
<b>Michael Kearns and Jennifer Wortman.....</b>	<b>99</b>
<i>Learning from Collective Behavior</i>	
<b>Bharath Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Gert Lanckriet and Bernhard Schölkopf.....</b>	<b>111</b>
<i>Injective Hilbert Space Embeddings of Probability Measures</i>	
<b>Complexity and Boolean Functions</b>	
<b>Sung-Soon Choi, Kyomin Jung and Jeong Han Kim .....</b>	<b>123</b>
<i>Almost Tight Upper Bound for Finding Fourier Coefficients of Bounded Pseudo-Boolean Functions</i>	
<b>Robert Holte, Steffen Lange, Sandra Zilles and Martin Zinkevich.....</b>	<b>135</b>
<i>Teaching Dimensions based on Cooperative Learning</i>	
<b>Vitaly Feldman .....</b>	<b>147</b>
<i>On the Power of Membership Queries in Agnostic Learning</i>	
<b>Complexity and Boolean Functions</b>	
<b>Thorsten Doliwa, Michael Kallweit and Hans Ulrich Simon.....</b>	<b>157</b>
<i>Dimension and Margin Bounds for Reflection-invariant Kernels</i>	
<b>Dana Angluin, James Aspnes, Jiang Chen, David Eisenstat and Lev Reyzin.....</b>	<b>169</b>
<i>Learning Acyclic Probabilistic Circuits Using Test Paths</i>	
<b>Linda Sellie .....</b>	<b>181</b>
<i>Learning Random Monotone DNF Under the Uniform Distribution</i>	
<b>Eric Blais, Ryan O'Donnell and Karl Wimmer .....</b>	<b>193</b>
<i>Polynomial Regression under Arbitrary Product Distributions</i>	

## Generalization and Statistics

<b>Alon Zakai and Yaacov Ritov</b> .....	<b>205</b>
<i>How Local Should a Learning Method Be?</i>	
<b>Yiming Ying and Colin Campbell</b> .....	<b>217</b>
<i>Learning Coordinate Gradients with Multi-Task Kernels</i>	
<b>Vladimir Koltchinskii and Ming Yuan</b> .....	<b>229</b>
<i>Sparse Recovery in Large Ensembles of Kernel Machines</i>	

## On-Line Learning and Bandits

<b>Amy Greenwald, Zheng Li and Warren Schudy</b> .....	<b>239</b>
<i>More Efficient Internal-Regret-Minimizing Algorithms</i>	
<b>Giovanni Cavallanti, Nicolo' Cesa-Bianchi and Claudio Gentile</b> .....	<b>251</b>
<i>Linear Algorithms for Online Multitask Classification</i>	
<b>Jacob Abernethy, Elad Hazan and Alexander Rakhlin</b> .....	<b>263</b>
<i>Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization</i>	

## Other Directions

<b>Wouter M. Koolen and Steven De Rooij</b> .....	<b>275</b>
<i>Combining Expert Advice Efficiently</i>	
<b>Maria-Florina Balcan, Avrim Blum and Nathan Srebro</b> .....	<b>287</b>
<i>Improved Guarantees for Learning via Similarity Functions</i>	
<b>J. Hyam Rubinstein and Benjamin I. P. Rubinstein</b> .....	<b>299</b>
<i>Geometric &amp; Topological Representations of Maximum Classes with Applications to Sample Compression</i>	
<b>Shai Shalev-Shwartz and Yoram Singer</b> .....	<b>311</b>
<i>On the Equivalence of Weak Learnability and Linear Separability: New Relaxations and Efficient Boosting Algorithms</i>	

## **Bandits and Reinforcement Learning**

- Andrey Bernstein and Nahum Shimkin**..... 323  
*Adaptive Aggregation for Reinforcement Learning with Efficient Exploration:  
Deterministic Domains*
- Peter Bartlett, Varsha Dani, Thomas Hayes, Sham Kakade, Alexander Rakhlin and  
Ambuj Tewari** ..... 335  
*High-Probability Regret Bounds for Bandit Online Linear Optimization*
- Aleksandrs Slivkins and Eli Upfal**..... 343  
*Adapting to a Changing Environment: the Brownian Restless Bandits*
- Varsha Dani, Thomas P. Hayes and Sham M. Kakade**..... 355  
*Stochastic Linear Optimization under Bandit Feedback*

## **Unsupervised and Semi-Supervised Learning**

- Ohad Shamir and Naftali Tishby** ..... 367  
*Model Selection and Stability in k-means Clustering*
- Shai Ben-David and Ulrike von Luxburg** ..... 379  
*Relating Clustering Stability to Properties of Cluster Boundaries*
- Kamalika Chaudhuri and Andrew McGregor**..... 391  
*Finding Metric Structure in Information Theoretic Clustering*
- Karthik Sridharan and Sham M. Kakade**..... 403  
*An Information Theoretic Framework for Multi-view Learning*

## **Online Learning**

- Jacob Abernethy, Peter Bartlett, Alexander Rakhlin and Ambuj Tewari**..... 415  
*Optimal Strategies and Minimax Lower Bounds for Online Convex Games*
- Robert D. Kleinberg, Alexandru Niculescu-Mizil and Yogeshwer Sharma**..... 425  
*Regret Bounds for Sleeping Experts and Bandits*
- Jacob Abernethy, Manfred K. Warmuth and Joel Yellin**..... 437  
*When Random Play is Optimal Against an Adversary*
- Andras Gyorgy, Gabor Lugosi and Gyorgy Ottucsak** ..... 447  
*On-line Sequential Bin Packing*

## Generalization and Statistics

<b>Shuheng Zhou, John Lafferty and Larry Wasserman</b> .....	<b>455</b>
<i>Time Varying Undirected Graphs</i>	
<b>Constantine Caramanis and Shie Mannor</b> .....	<b>467</b>
<i>Learning in the Limit with Adversarial Disturbances</i>	
<b>Liwei Wang</b> .....	<b>479</b>
<i>On the Margin Explanation of Boosting Algorithms</i>	
<b>Aarti Singh and Robert Nowack and Clayton Scott</b> .....	<b>491</b>
<i>Adaptive Hausdorff Estimation of Density Level Sets</i>	
<b>Satyaki Mahalanabis and Daniel Stefankovic</b> .....	<b>503</b>
<i>Density Estimation in Linear Time</i>	

## Open Problems

<b>Vitaly Feldman and Leslie G. Valiant</b> .....	<b>513</b>
<i>The Learning Power of Evolution</i>	
<b>Parikshit Gopalan and Adam Kalai and Adam R. Klivans</b> .....	<b>515</b>
<i>A Query Algorithm for Agnostically Learning DNF?</i>	
<b>Adam M. Smith and Manfred K. Warmuth</b> .....	<b>517</b>
<i>Learning Rotations</i>	
<b>Author Index</b> .....	<b>519</b>



---

## Foreword

---

This volume contains papers presented at the 21st Annual Conference on Learning Theory (previously known as the Conference on Computational Learning Theory) held in Helsinki, Finland from July 9-12, 2008. The technical program contained 44 papers selected from 126 submissions, three open problems selected from among five contributed, and four invited lectures that were joint with UAI. The invited lectures were given by Peter Grünwald on “The Catch-Up Phenomenon in Bayesian Inference,” by Robin Hanson on “Combinatorial Prediction Markets,” by Dan Klein on “Unsupervised Learning for Natural Language Processing,” and by Gabor Lugosi on “Concentration Inequalities.” The abstracts of these lectures are included in this volume.

The Mark Fulk award is presented annually for the best paper co-authored by a student. This year the Mark Fulk award was supported in part by the *Machine Learning Journal*, which also supported two further awards. Thus three student papers were selected for prizes. The Mark Fulk Award was awarded to Maria-Florina Balcan, Steve Hanneke and Jennifer Wortman for their paper “The True Sample Complexity of Active Learning.” The two Machine Learning Journal Best Paper Awards were awarded to Jacob Abernathy for his paper “Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization” (co-authored by Elad Hazan and Alexander Rakhlin), and to Alexandru Niculescu-Mizil and Yogeshwer Sharma for their paper “Regret Bounds for Sleeping Experts and Bandits” (co-authored with Robert Kleinberg).

This year witnessed many COLT submissions and a very strong program of papers. The selected papers cover a wide range of topics including clustering, unsupervised and semi-supervised learning, active learning, boosting, online learning, bandit problems and reinforcement learning, complexity-theoretic aspects of learning, generalization and statistical learning, kernel methods, and other topics.

We would like to thank the many people who made COLT 2008 a success. We thank the members of the Program Committee for COLT 2008: Dana Angluin (Yale University), Jean-Yves Audibert (Ecole Nationale des Ponts), Peter Auer (University of Leoben), Peter Bartlett (UC Berkeley), Mikhail Belkin (Ohio State University), Shai Ben-David (University of Waterloo), Stéphane Boucheron (Université Paris-Diderot), Nader Bshouty (Technion), Sanjoy Dasgupta (UC San Diego), Ran El-Yaniv (Technion), Vitaly Feldman (IBM Research), Sham M. Kakade (Toyota Technology Institute), Adam Kalai (Georgia Tech), Vladimir Koltchinskii (Georgia Tech), Sanjay Jain (National University of Singapore), John Langford (Yahoo! Research), Ping Li (Cornell University), Shie Mannor (McGill University), Mehryar Mohri (New York University), Massimiliano Pontil (University College, London), Rob Schapire (Princeton University), Shai Shalev-Shwartz (Hebrew University), Alex Smola (National ICT Australia), Nati Srebro (Toyota Technological Institute), Ingo Steinwart (Los Alamos National Laboratory), Nicolas Vayatis, (Ecole Normale Supérieure de Cachan), Volodya Vovk (Royal Holloway, University of London), and Bob Williamson (Australian National University). We are very grateful to all of them for their careful and thorough reviewing and for the detailed discussions that ensured a strong program for the conference. We thank the many sub-reviewers who assisted the Program Committee; unfortunately space constraints prevent us from including the long list of all their names, so we must ask them to accept our thanks anonymously.

We give special thanks to Jyrki Kivinen (University of Helsinki) who served as the Local Chair of COLT 2008. We thank Kati Kervinen for general administrative support of the conference, and Sanna Kettunen for his work in publicizing the conference. We thank Greger Lindén for creating and maintaining the

conference website, and Microsoft Research for providing the CMT software that was used in the Program Committee deliberations. We thank Nicolò Cesa-Bianchi for helping to organize the conference in his role as head of the COLT steering committee. We thank Ran Gilad-Bachrach for his work in updating and maintaining the [www.learningtheory.org](http://www.learningtheory.org) website. We also thank the ICML and UAI conference organizers for ensuring a smooth co-location of the three conferences, including overlap with UAI.

Finally, we would like to thank the Federation of Finnish Learned Societies, Google, Helsinki Institute for Information Technology, IBM, the *Machine Learning Journal*, the University of Helsinki, and Yahoo! for their support and sponsorship of the conference.

April 2008

Rocco Servedio and Tong Zhang  
COLT 2008 Program Chairs

---

# The Catch-Up Phenomenon in Bayesian Inference

---

**Peter Grünwald**  
CWI, Amsterdam, The Netherlands  
Peter.Grunwald@cwi.nl

## Abstract

Standard Bayesian model selection/averaging sometimes learn too slowly: there exist other learning methods that lead to better predictions based on less data. We give a novel analysis of this "catch-up" phenomenon. Based on this analysis, we propose the switching method, a modification of Bayesian model averaging that never learns slower, but sometimes learns much faster than Bayes. The method is related to expert-tracking algorithms developed in the COLT literature, and has time complexity comparable to Bayes.

The switching method resolves a long-standing debate in statistics, known as the AIC-BIC dilemma: model selection/averaging methods like BIC, Bayes, and MDL are consistent (they eventually infer the correct model) but, when used for prediction, the rate at which predictions improve can be suboptimal. Methods like AIC and leave-one-out cross-validation are inconsistent but typically converge at the optimal rate. Our method is the first that provably achieves both. Experiments with nonparametric density estimation confirm that these large-sample theoretical results also hold in practice in small samples.



---

# Combinatorial Prediction Markets

---

**Robin Hanson**

Research Associate, Future of Humanity Institute at Oxford University  
Associate Professor of Economics, George Mason University  
rhanson@gmu.edu

## **Abstract**

Several hundred organizations are now using prediction markets to forecast sales, project completion dates, and more. This number has been doubling annually for several years. Most, however, are simple prediction markets, with one market per number forecast, and several traders per market. In contrast, a single combinatorial prediction market lets a few traders manage an entire combinatorial space of forecasts. For millions of numbers or less, implementation is easy, and lab experiments have confirmed feasibility and accuracy. For larger spaces, however, many open computational problems remain.



---

# Unsupervised Learning for Natural Language Processing

---

**Dan Klein**

University of California, Berkeley  
klein@cs.berkeley.edu

## **Abstract**

Given the abundance of text data, unsupervised approaches are very appealing for natural language processing. We present three latent variable systems which achieve state-of-the-art results in domains previously dominated by fully supervised systems. For syntactic parsing, we describe a grammar induction technique which begins with coarse syntactic structures and iteratively refines them in an unsupervised fashion. The resulting coarse-to-fine grammars admit efficient coarse-to-fine inference schemes and have produced the best parsing results in a variety of languages. For coreference resolution, we describe a discourse model in which entities are shared across documents using a hierarchical Dirichlet process. In each document, entities are repeatedly rendered into mention strings by a sequential model of attentional state and anaphoric constraint. Despite being fully unsupervised, this approach is competitive with the best supervised approaches. Finally, for machine translation, we present a model which learns translation lexicons from non-parallel corpora. Alignments between word types are modeled by a prior over matchings. Given any fixed alignment, a joint density over word vectors derives from probabilistic canonical correlation analysis. This approach is capable of discovering high-precision translations, even when the underlying corpora and languages are divergent.



---

# Concentration inequalities

---

**Gábor Lugosi**

ICREA and Department of Economics, Pompeu Fabra University  
Barcelona, Spain  
gabor.lugosi@gmail.com

## Abstract

In this talk by concentration inequalities we mean inequalities that bound the deviations of a function of independent random variables from its mean. Due to their generality and elegance, many such results have served as standard tools in a variety of areas, including statistical learning theory, probabilistic combinatorics, and the geometry of Banach spaces. To illustrate some of the basic ideas, we start by showing simple ways of bounding the variance of a general function of several independent random variables. We show how to use these inequalities on a few key quantities in statistical learning theory. In the past two decades several techniques have been introduced to improve such variance inequalities to exponential tail inequalities. We focus on a particularly elegant and effective method, the so-called "entropy method", based on logarithmic Sobolev inequalities and their modifications. Similar ideas appear in a variety of areas of mathematics, including discrete and Gaussian isoperimetric problems, and estimation of mixing times of Markov chains. We intend to shed some light to some of these connections. In particular, we mention some closely related results on influences of variables of Boolean functions, phase transitions, and threshold phenomena.



---

# Learning Mixtures of Product Distributions using Correlations and Independence

---

**Kamalika Chaudhuri**

Information Theory and Applications, UC San Diego  
kamalika@soe.ucsd.edu

**Satish Rao**

Computer Science Division, UC Berkeley  
satishr@cs.berkeley.edu

## Abstract

We study the problem of learning mixtures of distributions, a natural formalization of clustering. A mixture of distributions is a collection of distributions  $\mathcal{D} = \{D_1, \dots, D_T\}$ , and *mixing weights*,  $\{w_1, \dots, w_T\}$  such that  $\sum_i w_i = 1$ . A sample from a mixture is generated by choosing  $i$  with probability  $w_i$  and then choosing a sample from distribution  $D_i$ . The problem of learning the mixture is that of finding the parameters of the distributions comprising  $\mathcal{D}$ , given only the ability to sample from the mixture. In this paper, we restrict ourselves to learning mixtures of product distributions.

The key to learning the mixtures is to find a *few* vectors, such that points from different distributions are sharply separated upon projection onto these vectors. Previous techniques use the vectors corresponding to the top few directions of highest variance of the mixture. Unfortunately, these directions may be directions of high noise and not directions along which the distributions are separated. Further, skewed mixing weights amplify the effects of noise, and as a result, previous techniques only work when the separation between the input distributions is large relative to the imbalance in the mixing weights.

In this paper, we show an algorithm which successfully learns mixtures of distributions with a separation condition that depends only logarithmically on the skewed mixing weights. In particular, it succeeds for a separation between the centers that is  $\Theta(\sigma\sqrt{T\log\Lambda})$ , where  $\sigma$  is the maximum directional standard deviation of any distribution in the mixture,  $T$  is the number of distributions, and  $\Lambda$  is polynomial in  $T, \sigma, \log n$  and the imbalance in the mixing

weights. For our algorithm to succeed, we require a *spreading condition*, that the distance between the centers be *spread* across  $\Theta(T\log\Lambda)$  coordinates. Additionally, with arbitrarily small separation, *i.e.*, even when the separation is not enough for clustering, with enough samples, we can approximate the subspace containing the centers. Previous techniques failed to do so in polynomial time for non-spherical distributions regardless of the number of samples, unless the separation was large with respect to the maximum directional variance  $\sigma$  and polynomially large with respect to the imbalance of mixing weights. Our algorithm works for *Binary Product Distributions* and *Axis-Aligned Gaussians*. The spreading condition above is implied by the separation condition for binary product distributions, and is necessary for algorithms that rely on linear correlations.

Finally, when a stronger version of our spreading condition holds, our algorithm performs successful clustering when the separation between the centers is only  $\Theta(\sigma_*\sqrt{T\log\Lambda})$ , where  $\sigma_*$  is the maximum directional standard deviation in the subspace containing the centers of the distributions.

## 1 Introduction

Clustering, the problem of grouping together data points in high dimensional space using a similarity measure, is a fundamental problem of statistics with numerous applications in a wide variety of fields. A natural model for clustering is that of *learning mixtures of distributions*. A mixture of distributions is a collection of distributions  $\mathcal{D} = \{D_1, \dots, D_T\}$ , and *mixing weights*,  $\{w_1, \dots, w_T\}$  such that  $\sum_i w_i = 1$ . A sample from a mixture is generated by choosing  $i$  with probability  $w_i$  and choosing a sample from distribution  $D_i$ . The problem of learning the mixture is that of finding the parameters of the distributions comprising  $\mathcal{D}$ , given only the ability to sample from the mixture.

If the distributions  $D_1, \dots, D_T$  are very close to each other, then even if we knew the parameters of the distributions, it would be impossible to classify the points correctly with high confidence. Therefore, Dasgupta [Das99] introduced the notion of a *separation condition*, which is a promise that each pair of distributions is sufficiently different according to some measure. Given points from a mixture of distributions and a separation condition, the goal is to find the parameters of the mixture  $\mathcal{D}$ , and cluster all but a small fraction of the points correctly. A commonly used separation measure is the distance between the centers of the distributions parameterized by the maximum directional variance,  $\sigma$ , of any distribution in the mixture.

A common approach to learning the mixtures and therefore, clustering the high-dimensional cloud of points is to find a *few* interesting vectors, such that points from different distributions are sharply separated upon projection onto these vectors. Various distance-based methods [AK01, Llo82, DLR77] are then applied to cluster in the resulting low-dimensional subspace. The state-of-the-art, in practice, is to use the vectors corresponding to the top few directions of *highest variance* of the mixture and to hope that it contains most of the separation between the centers. This is computed by a *Singular Value Decomposition*(SVD) of the matrix of samples. This approach has been theoretically analyzed by [VW02] for spherical distributions, and for more general distributions in [KSV05, AM05]. The latter show that the maximum variance directions are indeed the interesting directions when the separation is  $\Theta(\frac{\sigma}{\sqrt{w_{\min}}})$ , where  $w_{\min}$  is the smallest mixing weight of any distribution.

This is the best possible result for SVD-based approaches; the directions of maximum variance may well not be the directions in which the centers are separated, but instead may be the directions of very high noise, as illustrated in Figure 1(b). This problem is exacerbated when the mixing weights  $w_i$  are skewed – because a distribution with low mixing weight diminishes the contribution to the variance along a direction that separates

the centers.

This bound is suboptimal for two reasons. Although mixtures with skewed mixing weights arise naturally in practice (see [PSD00] for an example), given enough samples, mixing weights have no bearing on the separability of distributions. Consider two mixtures  $\mathcal{D}'$  and  $\mathcal{D}''$  of distributions  $D_1$  and  $D_2$ : in  $\mathcal{D}'$ ,  $w_1 = w_2 = 1/2$ , and in  $\mathcal{D}''$ ,  $w_1 = 1/4$  and  $w_2 = 3/4$ . Given enough computational resources, if we can learn  $\mathcal{D}'$  from 50 samples, we should be able to learn  $\mathcal{D}''$  from 100 samples. This does not necessarily hold for SVD-based methods. Secondly, regardless of  $\sigma$ , an algorithm, which has prior knowledge of the subspace containing the centers of the distributions, should be able to learn the mixture when the separation is proportional to  $\sigma_*$ , the maximum directional standard deviation of any distribution in the subspace containing the centers. An example in which  $\sigma$  and  $\sigma_*$  are significantly different is shown in Figure 1(b).

In this paper, we study the problem of learning mixtures of *product distributions*. A product distribution over  $\mathbf{R}^n$  is one in which each coordinate is distributed independently of any others. In practice, mixtures of product distributions have been used as mathematical models for data and learning mixtures of product distributions specifically has been studied [FM99, FOS05, FOS06, DHKS05] – see the Related Work section for examples and details. However, even under this seemingly restrictive assumption, providing an efficient algorithm that does better than the bounds of [AM05, KSV05] turns out to be quite challenging. The main challenge is to find a low-dimensional subspace that contains most of the separation between the centers; although the independence assumption can (sometimes) help us identify which coordinates contribute to the distance between some pair of centers, the problem of actually finding the low-dimensional space still requires more involved techniques.

In this paper, we present an algorithm for learning mixtures of product distributions, which is stable in the presence of skewed mixing weights, and, under certain conditions, in the presence of high variance outside the subspace containing the centers. In particular, the dependence of the separation required by our algorithm on skewed mixing weights is only logarithmic. Additionally, with arbitrarily small separation, (*i.e.*, even when the separation is not enough for classification), with enough samples, we can approximate the subspace containing the centers. Previous techniques failed to do so for non-spherical distributions regardless of the number of samples, unless the separation was sufficiently large. Our algorithm works for binary product distributions and axis-aligned Gaussians. We require that the distance between the centers be *spread* across  $\Theta(T \log \Lambda)$  coordinates, where  $\Lambda$  depends polynomially on the max-

imum distance between centers and  $w_{min}$ . For our algorithm to classify the samples correctly, we further need the separation between centers to be  $\Theta(\sigma\sqrt{T\log\Lambda})$ .

In addition, if a stronger version of the spreading condition is satisfied, then our algorithm requires a separation of only  $\Theta(\sigma_*\sqrt{T\log\Lambda})$  to ensure correct classification of the samples. The stronger spreading condition, discussed in more detail later, ensures that when we split the coordinates randomly into two sets, the maximum directional variance of any distribution in the mixture along the projection of the subspace containing the centers into the subspaces spanned by the coordinate vectors in each set, is comparable to  $\sigma_*^2$ .

In summary, compared to [AM05, KSV05], our algorithm is much (exponentially) less susceptible to the imbalance in mixture weights and, when the stronger spreading condition holds, to high variance noise outside the subspace containing the centers. However, our algorithm requires a spreading condition and coordinate-independence, while [AM05, KSV05] are more general. We note that for perfectly spherical distributions, the results of [VW02] are better than our results – however, these results do not apply even for distributions with bounded eccentricity. Finally unlike the results of [Das99, AK01, DS00], which require the separation to grow polynomially with dimension, our separation only grows logarithmically with the dimension.

Our algorithm is based upon two key insights. The first insight is that if the centers are separated along several coordinates, then many of these coordinates are *correlated* with each other. To exploit this observation, we choose half the coordinates randomly, and search the space of this half for directions of high variance. We use the remaining half of coordinates to *filter* the found directions. If a found direction separates the centers, it is likely to have some correlation with coordinates in the remaining half, and therefore is preserved by the filter. If, on the other hand, the direction found is due to noise, coordinate independence ensures that there will be no correlation with the second half of coordinates, and therefore such directions get filtered away.

The second insight is that the tasks of searching for and filtering the directions can be simultaneously accomplished via a singular value decomposition of the matrix of covariances between the two halves of coordinates. In particular, we show that the top few directions of maximum variance of the covariance matrix approximately capture the subspace containing the centers. Moreover, we show that the covariance matrix has low singular value along any noise direction. By combining these ideas, we obtain an algorithm that is almost insensitive to mixing weights, a property essential for applications like population stratification [CHRZ07], and which can be implemented using the heavily optimized and thus, efficient, SVD procedure, and which works

with a separation condition closer to the information theoretic bound.

## Related Work

The first provable results for learning mixtures of Gaussians are due to Dasgupta [Das99] who shows how to learn mixtures of spherical Gaussians with a separation of  $\Theta(\sigma\sqrt{n})$  in an  $n$ -dimensional space. An EM based algorithm by Dasgupta and Schulman [DS00] was shown to apply to more situations, and with a separation of  $\Theta(\sigma n^{1/4})$ . Arora and Kannan [AK01] show how to learn mixtures of distributions of arbitrary Gaussians whose centers are separated by  $\Theta(n^{1/4}\sigma)$ . Their results apply to many other situations, for example, *concentric* Gaussians with sufficiently different variance.

The first result that removed the dependence on  $n$  in the separation requirement was that of Vempala and Wang [VW02] who use SVD to learn mixtures of spherical Gaussians with  $O(\sigma T^{1/4})$  separation. They project to a subspace of dimension  $T$  using an SVD and use a distance based method in the low dimensional space. If the separation is not enough for classification, [VW02] can also find, given enough samples, a subspace approximating the subspace containing the centers. While the results of [VW02] are independent of the imbalance on mixing weights, they apply only to perfectly spherical Gaussians, and cannot be extended to Gaussians with bounded eccentricity. In further work Kannan, Salmasian, and Vempala [KSV05] and Achlioptas and McSherry [AM05] show how to cluster general Gaussians using SVD. While these results are weaker than ours, they apply to a mixture of general Gaussians, axis-aligned or not. We note that their analysis also applies to binary product distributions again with polynomial dependence on the imbalance in mixing weights<sup>1</sup>. In contrast, our separation requirement is  $\Omega(\sigma_*\sqrt{T\log\Lambda})$ , *i.e.*, is logarithmically dependent on the mixing weights and dimension and the maximum variance in noise directions.

There is also ample literature on specifically learning mixtures of product distributions. Freund and Mansour [FM99] show an algorithm which generates distributions that are  $\epsilon$ -close to a mixture of two product distributions over  $\{0, 1\}^n$  in time polynomial in  $n$  and  $1/\epsilon$ . Feldman, O’Donnell, and Servedio show how to generate distributions that are  $\epsilon$ -close to a mixture of  $T$  product distributions [FOS05] and axis-aligned Gaussians [FOS06]. Like [FM99], they have no separation requirements, but their algorithm takes  $n^{O(T^3)}$  time. Dasgupta *et. al* [DHKS05] provide an algorithm for learning mixtures of heavy-tailed product distributions which works with a separation of  $\Theta(R\sqrt{T})$ , where  $R$  is the maximum half-radius of any distribution in the mixture.

<sup>1</sup>They do not directly address binary product distributions in their paper, but their techniques apply.

While their separation requirement does not depend polynomially on  $\frac{1}{w_{\min}}$ , their algorithm runs in time exponential in  $\Theta(\frac{n}{w_{\min}})$ . They also require a slope, which is comparable to our spreading condition. Chaudhuri *et al.* [CHRZ07] show an iterative algorithm for learning mixtures of two product distributions that implicitly uses the notion of co-ordinate independence to filter out noise directions. However, the algorithm heavily uses the two distribution restriction to find the appropriate directions, and does not work when  $T > 2$ .

More broadly, the problem of analyzing mixture models data has received a great deal of attention in statistics, see for example, [MB88, TSM85], and has numerous applications. We present three applications where data is modelled as a mixture of product distributions. First, the problem of population stratification in population genetics has been posed as learning mixtures of binary product distributions in [SRH07]. In their work, the authors develop an MCMC method for addressing the problem and their software embodiment is widely used. A second application is in speech recognition [Rey95, PFK02], which models acoustic features at a specific time point as a mixture of axis-aligned Gaussians. A third application is the widely used Latent Dirichlet Allocation model [BNJ03]. Here, documents are modelled as distributions over topics which, in turn, are distributions over words. Subsequent choices of topics and words are assumed to be *independent*. (For words, this is referred to as the “bag of words” assumption.) [BNJ03] develops variational techniques that provide interesting results for various corpora. Interestingly, the same model was used by Kleinberg and Sandler [KS04] to model user preferences for purchasing goods (users correspond to documents, topics to categories, and words to goods). Their algorithm, which provides provably good performance in this model, also uses SVD-like clustering algorithms as a subroutine.

Our clustering method also involves a Canonical Correlations Analysis of the samples, which seems to have connections with multiview learning [KF07] and co-training [AT98].

## Discussion

**The Spreading Condition.** The spreading condition loosely states that the distance between each pair of centers is spread along about  $\Theta(T \log \Lambda)$  coordinates. We demonstrate by an example, that a spread of  $\Omega(T)$ , is a natural limit for all methods that use linear correlations between coordinates, such as our methods and SVD based methods [VW02, KSV05, AM05]. We present, as an example, two distributions: a mixture  $\mathcal{D}_1$  of  $T$  binary product distributions, and a single binary product distribution  $\mathcal{D}_2$ , which have exactly the same covariance matrix. Our example is based on the Hadamard code, in which a codeword for a  $k$ -bit message is  $2^k$  bits long, and includes a parity bit for each subset of the bits of

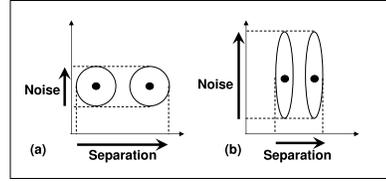


Figure 1: (a) Spherical Gaussians: Direction of maximum variance is the direction separating the centers (b) Arbitrary Gaussians: Direction of maximum variance is a noise direction.

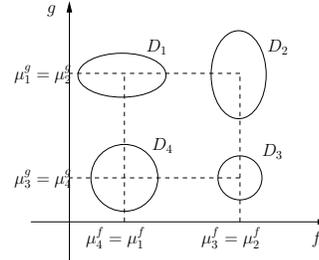


Figure 2: An Example where All Covariances are 0

the message. The distributions comprising  $\mathcal{D}_1$  are defined as follows. Each of the  $T = 2^k$  centers is a codeword for a  $k$ -bit string appended by a string of length  $n - k$  in which each coordinate has value  $1/2$ . Notice that the last  $n - k$  bits are noise. Thus, the centers are separated by  $T/2$  coordinates.  $\mathcal{D}_2$  is the uniform distribution over the  $n$ -dimensional hypercube. As there are no linear correlations between any two bits in the Hadamard code, the covariance of  $\mathcal{D}_1$  along any two directions is 0, and each direction has the same variance. As this is also the case for  $\mathcal{D}_2$ , any SVD-based correlation-based algorithm will fail to distinguish between the two mixtures. We also note that learning binary product distributions with minimum separation 2 and average separation  $1 + \frac{1}{2} \log T$  would allow one to learn parities of  $\log T$  variables with noise. Finally, we note that when the spreading condition fails, one has only a few coordinates that contain most of the distance between centers. One could enumerate the set of possible coordinates to deal with this case, and is exponential in  $T \log n \log \Lambda$ . [FOS05] on the other hand takes time exponential in  $T^3 \log n$ , and works with no separation requirement.

## 2 A Summary of Our Results

We begin with some preliminary definitions about distributions drawn over  $n$  dimensional spaces. We use

$f, g, \dots$  to range over coordinates, and  $i, j, \dots$  to range over distributions. For any  $x \in \mathbf{R}^n$ , we write  $x^f$  for the  $f$ -th coordinate of  $x$ . For any subspace  $\mathcal{H}$  (resp. vector  $v$ ), we use  $\mathcal{H}$  (resp.  $\bar{v}$ ) to denote the orthogonal complement of  $\mathcal{H}$  (resp.  $v$ ). For a subspace  $\mathcal{H}$  and a vector  $v$ , we write  $\mathbf{P}_{\mathcal{H}}(v)$  for the projection of  $v$  onto the subspace  $\mathcal{H}$ . For any vector  $x$ , we use  $\|x\|$  for the Euclidean norm of  $x$ . For any two vectors  $x$  and  $y$ , we use  $\langle x, y \rangle$  for the dot-product of  $x$  and  $y$ .

**Mixtures of Distributions.** A *mixture of distributions*  $\mathcal{D}$ , is a collection of distributions,  $\{D_1, \dots, D_T\}$ , over points in  $\mathbf{R}^n$ , and a set of mixing weights  $w_1, \dots, w_T$  such that  $\sum_i w_i = 1$ . In the sequel,  $n$  is assumed to be much larger than  $T$ . In a product distribution over  $\mathbf{R}^n$ , each coordinate is distributed independently of the others. When working with a mixture of binary product distributions, we assume that the  $f$ -th coordinate of a point drawn from distribution  $D_i$  is 1 with probability  $\mu_i^f$ , and 0 with probability  $1 - \mu_i^f$ . When working with a mixture of axis-aligned Gaussian distributions, we assume that the  $f$ -th coordinate of a point drawn from distribution  $D_i$  is distributed as a Gaussian with mean  $\mu_i^f$  and standard deviation  $\sigma_i^f$ .

**Centers.** We define the *center* of a distribution  $i$  as the vector  $\mu_i$ , and the *center of mass of the mixture* as the vector  $\bar{\mu}$  where  $\bar{\mu}^f$  is the mean of the mixture for the coordinate  $f$ . We write  $\mathcal{C}$  for the subspace containing  $\mu_1, \dots, \mu_T$ .

**Directional Variance.** We define  $\sigma^2$  as the maximum variance of any distribution in the mixture along any direction. We define  $\sigma_*^2$  as the maximum variance of any distribution in the mixture along any direction in the subspace containing the centers of the distributions. We write  $\sigma_{\max}^2$  as the maximum variance of the entire mixture in any direction. This may be more than  $\sigma^2$  due to contribution from the separation between the centers.

**Spread.** We say that a unit vector  $v$  in  $\mathbf{R}^n$  has spread  $\mathcal{S}$  if  $\sum_f (v^f)^2 \geq \mathcal{S} \cdot \max_f (v^f)^2$ .

**Distance.** Given a subspace  $\mathcal{K}$  of  $\mathbf{R}^n$  and two points  $x, y$  in  $\mathbf{R}^n$ , we write  $d_{\mathcal{K}}(x, y)$  for the square of the Euclidean distance between  $x$  and  $y$  projected along the subspace  $\mathcal{K}$ .

**The Spreading Condition and Effective Distance.** The spreading condition tells us that the distance between each  $\mu_i$  and  $\mu_j$  should not be concentrated along a few coordinates. One way to ensure this is to demand that for all  $i, j$ , the vector  $\mu_i - \mu_j$  has high spread. This is comparable to the slope condition used in [DHKS05].

However, we do not need such a strong condition for dealing with mixtures with imbalanced mixing weights. Our *spreading condition* therefore demands that for each pair of centers  $\mu_i, \mu_j$ , the norm of the vector  $\mu_i - \mu_j$  high, even if we ignore the contribution of the top few

(about  $T \log T$ ) coordinates. Due to technicalities in our proofs, the number of coordinates we can ignore needs to depend (logarithmically) on this distance.

We therefore define the spreading condition as follows. We define parameters  $c_{ij}$  and a parameter  $\Lambda$  as :  $\Lambda > \frac{\sigma_{\max} T \log^2 n}{w_{\min} \cdot (\min_{i,j} c_{ij}^2)}$  and  $c_{ij}$  is the maximum value such that there are  $49T \log \Lambda$  coordinates  $f$  with  $|\mu_i^f - \mu_j^f| > c_{ij}$ . We note that  $\Lambda$  is bounded by a polynomial in  $T, \sigma_*, 1/w_{\min}, 1/c_{ij}$  and logarithmic in  $n$ .

We define  $c_{\min}$  to be the minimum over all pairs  $i, j$  of  $c_{ij}$ . Given a pair of centers  $i$  and  $j$ , let  $\Delta_{ij}$  be the set of coordinates  $f$  such that  $|\mu_i^f - \mu_j^f| > c_{ij}$ , and let  $\nu_{ij}$  be defined as:  $\nu_{ij}^f = \mu_i^f - \mu_j^f$ , if  $f \in \Delta_{ij}$ , and  $\nu_{ij}^f = c_{ij}$  otherwise. We define  $\bar{d}(\mu_i, \mu_j)$ , the effective distance between  $\mu_i$  and  $\mu_j$  to be the square of the  $L_2$  norm of  $\nu_{ij}$ . In contrast, the square of the norm of the vector  $\mu_i - \mu_j$  is the actual distance between centers  $\mu_i$  and  $\mu_j$ , and is always greater than or equal to the effective distance between  $\mu_i$  and  $\mu_j$ . Moreover, given  $i$  and  $j$  and the subspace  $\mathcal{K}$ , we define  $\bar{d}_{\mathcal{K}}(\mu_i, \mu_j)$  as the square of the norm of the vector  $\nu_{ij}$  projected onto the subspace  $\mathcal{K}$ .

Under these definitions, our spreading condition now requires that  $\bar{d}(\mu_i, \mu_j) \geq 49c_{ij}^2 T \log \Lambda$  and our stronger spreading condition requires that every vector in  $\mathcal{C}$  has spread  $32T \log \frac{\sigma}{\sigma_*}$ .

**A Formal Statement of our Results.** Our main contribution is Algorithm CORR-CLUSTER, a correlation based algorithm for learning mixtures of binary product distributions and axis-aligned Gaussians. The input to the algorithm is a set of samples from a mixture of distributions, and the output is a clustering of the samples.

The main component of Algorithm CORR-CLUSTER is Algorithm CORR-SUBSPACE, which, given samples from a mixture of distributions, computes an approximation to the subspace containing the centers of the distributions. The motivation for approximating the latter space is as follows. In the  $T$ -dimensional subspace containing the centers of the distributions, the distance between each pair of centers  $\mu_i$  and  $\mu_j$  is the same as their distance in  $\mathbf{R}^n$ ; however, because of the low dimensionality, the magnitude of the noise is small. Therefore, provided the centers of the distributions are sufficiently separated, projection onto this subspace will sharply separate samples from different distributions. SVD-based algorithms [VW02, AM05, KSV05] attempt to approximate this subspace by the top  $T$  singular vectors of the matrix of samples. However, for product distributions, our Algorithm CORR-SUBSPACE can approximate this subspace correctly under more restrictive separation conditions.

The properties of Algorithms CORR-SUBSPACE and

CORR-CLUSTER are formally summarized in Theorem 1 and Theorem 2 respectively.

**Theorem 1 (Spanning centers)** *Suppose we are given a mixture of distributions  $\mathcal{D} = \{D_1, \dots, D_T\}$ , with mixing weights  $w_1, \dots, w_T$ . Then with at least constant probability, the subspace  $\mathcal{K}$  of dimension at most  $2T$  output by Algorithm CORR-SUBSPACE has the following properties.*

1. *If, for all  $i$  and  $j$ ,  $\bar{d}(\mu_i, \mu_j) \geq 49c_{ij}^2 T \log \Lambda$ , then, for all pairs  $i, j$ ,*

$$d_{\mathcal{K}}(\mu_i, \mu_j) \geq \frac{99}{100}(\bar{d}(\mu_i, \mu_j) - 49T c_{ij}^2 \log \Lambda)$$

2. *If, in addition, every vector in  $\mathcal{C}$  has spread  $32T \log \frac{\sigma}{\sigma_*}$ , then, with at least constant probability, the maximum directional variance in  $\mathcal{K}$  of any distribution  $D_i$  in the mixture is at most  $11\sigma_*^2$ .*

*The number of samples required by Algorithm CORR-SUBSPACE is polynomial in  $\frac{\sigma}{\sigma_*}$ ,  $T$ ,  $n, \sigma$  and  $\frac{1}{w_{\min}}$ , and the algorithm runs in time polynomial in  $n$ ,  $T$ , and the number of samples.*

The subspace  $\mathcal{K}$  computed by Algorithm CORR-SUBSPACE approximates the subspace containing the centers of the distributions in the sense that the distance between each pair of centers  $\mu_i$  and  $\mu_j$  is high along  $\mathcal{K}$ . Theorem 1 states that Algorithm CORR-SUBSPACE computes an approximation to the subspace containing the centers of the distributions, provided the spreading condition is satisfied. If the strong spreading condition is satisfied as well, then the maximum variance of each  $D_i$  along  $\mathcal{K}$  is also close to  $\sigma_*^2$ .

Note that in Theorem 1, there is no absolute lower bound required on the distance between any pair of centers. This means that, so long as the spreading condition is satisfied, and there are sufficiently many samples, even if the distance between the centers is not large enough for correct classification, we can compute an approximation to the subspace containing the centers of the distributions. We also note that although we show that Algorithm CORR-SUBSPACE succeeds with constant probability, we can make this probability higher at the expense of a more restrictive spreading condition, or by running the algorithm multiple times.

**Theorem 2 (Clustering)** *Suppose we are given a mixture of distributions  $\mathcal{D} = \{D_1, \dots, D_T\}$ , with mixing weights  $w_1, \dots, w_T$ . Then, Algorithm CORR-CLUSTER has the following properties.*

1. *If for all  $i$  and  $j$ ,  $\bar{d}(\mu_i, \mu_j) \geq 49T c_{ij}^2 \log \Lambda$ , and for all  $i, j$  we have:*

$$\bar{d}(\mu_i, \mu_j) > 59\sigma^2 T (\log \Lambda + \log n)$$

*(for axis-aligned Gaussians)*

$$\bar{d}(\mu_i, \mu_j) > 59T (\log \Lambda + \log n)$$

*(for binary product distributions)*

*then with probability  $1 - \frac{1}{n}$  over the samples and with constant probability over the random choices made by the algorithm, Algorithm CORR-CLUSTER computes a correct clustering of the sample points.*

2. *For axis-aligned Gaussians, if every vector in  $\mathcal{C}$  has spread at least  $32T \log \frac{\sigma}{\sigma_*}$ , and for all  $i, j$ :*

$$\bar{d}(\mu_i, \mu_j) \geq 150\sigma_*^2 T (\log \Lambda + \log n)$$

*then, with constant probability over the randomness in the algorithm, and with probability  $1 - \frac{1}{n}$  over the samples, Algorithm CORR-CLUSTER computes a correct clustering of the sample points.*

*Algorithm CORR-CLUSTER runs in time polynomial in  $n$  and the number of samples required by Algorithm CORR-CLUSTER is polynomial in  $\frac{\sigma}{\sigma_*}$ ,  $T$ ,  $n$ ,  $\sigma$  and  $\frac{1}{w_{\min}}$ .*

We note that because we are required to do classification here, we do require an absolute lower bound on the distance between each pair of centers in Theorem 2.

The second theorem follows from the first and the distance concentration Lemmas of [AM05] as described in detail in Chapter 3 of [Cha07]. The Lemmas show that once the points are projected onto the subspace computed in Theorem 1, a distance-based clustering method suffices to correctly cluster the points.

**A Note on the Stronger Spreading Condition.** The motivation for requiring the stronger spreading condition is as follows. Our algorithm splits the coordinates randomly into two sets  $\mathcal{F}$  and  $\mathcal{G}$ . If  $\mathcal{C}_{\mathcal{F}}$  and  $\mathcal{C}_{\mathcal{G}}$  denote the restriction of  $\mathcal{C}$  to the coordinates in  $\mathcal{F}$  and  $\mathcal{G}$  respectively, then our algorithm requires that the maximum directional variance of any distribution in the mixture is close to  $\sigma_*$  in  $\mathcal{C}_{\mathcal{F}}$  and  $\mathcal{C}_{\mathcal{G}}$  respectively. Notice that this does not follow from the fact that the maximum directional variance along  $\mathcal{C}$  is  $\sigma_*^2$ : suppose  $\mathcal{C}$  is spanned by  $(0.1, 0.1, 1, 1)$  and  $(0.1, 0.1, -1, 1)$ , variances of  $D_1$  along the axes are  $(10, 10, 1, 1)$ , and  $\mathcal{F}$  is  $\{1, 2\}$ . Then,  $\sigma_*^2$  is about 2.8, while the variance of  $D_1$  along  $\mathcal{C}_{\mathcal{F}}$  is 10. However, as Lemma 9 shows, the required condition is ensured by the strong spreading condition.

However, in general, the maximum directional variance of any  $D_i$  in the mixture along  $\mathcal{C}_{\mathcal{F}}$  and  $\mathcal{C}_{\mathcal{G}}$  may still be close to  $\sigma_*^2$ , even though strong spreading condition is far from being met. For example: if  $\mathcal{C}$  is the space spanned by the first  $T$  coordinate vectors  $e_1, \dots, e_T$ , then with probability  $1 - \frac{1}{2^T}$ , the maximum variance along  $\mathcal{C}_{\mathcal{F}}$  and  $\mathcal{C}_{\mathcal{G}}$  is also  $\sigma_*^2$ .

### 3 Algorithm CORR-CLUSTER

Our clustering algorithm follows the same basic framework as the SVD-based algorithms of [VW02, KSV05, AM05]. The input to the algorithm is a set  $S$  of samples,

and the output is a pair of clusterings of the samples according to source distribution.

**CORR-CLUSTER( $S$ )**

1. Partition  $S$  into  $S_A$  and  $S_B$  uniformly at random.
2. Compute:  $\mathcal{K}_A = \text{Corr-Subspace}(S_A)$ ,  $\mathcal{K}_B = \text{Corr-Subspace}(S_B)$
3. Project each point in  $S_B$  (resp.  $S_A$ ) on the subspace  $\mathcal{K}_A$  (resp.  $\mathcal{K}_B$ ).
4. Use a distance-based clustering algorithm [AK01] to partition the points in  $S_A$  and  $S_B$  after projection.

The first step in the algorithm is to use Algorithm CORR-SUBSPACE to find a  $O(T)$ -dimensional subspace  $\mathcal{K}$  which is an approximation to the subspace containing the centers of the distributions. Next, the samples are projected onto  $\mathcal{K}$  and a distance-based clustering algorithm is used to find the clusters.

We note that in order to preserve independence the samples we project onto  $\mathcal{K}$  should be distinct from the ones we use to compute  $\mathcal{K}$ . A clustering of the complete set of points can then be computed by partitioning the samples into two sets  $A$  and  $B$ . We use  $A$  to compute  $\mathcal{K}_A$ , which is used to cluster  $B$  and vice-versa.

We now present our algorithm which computes a basis for the subspace  $\mathcal{K}$ . With slight abuse of notation we use  $\mathcal{K}$  to denote the set of vectors that form the basis for the subspace  $\mathcal{K}$ . The input to CORR-SUBSPACE is a set  $S$  of samples, and the output is a subspace  $\mathcal{K}$  of dimension at most  $2T$ .

**Algorithm CORR-SUBSPACE:**

**Step 1: Initialize and Split** Initialize the basis  $\mathcal{K}$  with the empty set of vectors. Randomly partition the coordinates into two sets,  $\mathcal{F}$  and  $\mathcal{G}$ , each of size  $n/2$ . Order the coordinates as those in  $\mathcal{F}$  first, followed by those in  $\mathcal{G}$ .

**Step 2: Sample** Translate each sample point so that the center of mass of the set of sample points is at the origin. Let  $F$  (respectively  $G$ ) be the matrix which contains a row for each sample point, and a column for each coordinate in  $\mathcal{F}$  (respectively  $\mathcal{G}$ ). For each matrix, the entry at row  $x$ , column  $f$  is the value of the  $f$ -th coordinate of the sample point  $x$  divided by  $\sqrt{|S|}$ .

**Step 3: Compute Singular Space** For the matrix  $F^T G$ , compute  $\{v_1, \dots, v_T\}$ , the top  $T$  left singular vectors,  $\{y_1, \dots, y_T\}$ , the top  $T$  right singular vectors, and  $\{\lambda_1, \dots, \lambda_T\}$ , the top  $T$  singular values.

**Step 4: Expand Basis** For each  $i$ , we abuse notation and use  $v_i$  ( $y_i$  respectively) to denote the vector obtained by concatenating  $v_i$  with the 0 vector in

$n/2$  dimensions (0 vector in  $n/2$  dimensions concatenated with  $y_i$  respectively). For each  $i$ , if the singular value  $\lambda_i$  is more than a threshold  $\tau = O\left(\frac{w_{\min} c_{\mathcal{G}}^2}{T \log^2 n} \cdot \sqrt{\log \Lambda}\right)$ , we add  $v_i$  and  $y_i$  to  $\mathcal{K}$ .

**Step 5: Output** Output the set of vectors  $\mathcal{K}$ .

The main idea behind our algorithm is to use half the coordinates to compute a subspace which approximates the subspace containing the centers, and the remaining half to validate that the subspace computed is indeed a good approximation. We critically use the coordinate independence property of product distributions to make this validation possible.

## 4 Analysis of Algorithm CORR-CLUSTER

This section is devoted to proving Theorems 1, and 2. We use the following notation.

**Notation.** We write  $\mathcal{F}$ -space (resp.  $\mathcal{G}$ -space) for the  $n/2$  dimensional subspace of  $\mathbf{R}^n$  spanned by the coordinate vectors  $\{e_f \mid f \in \mathcal{F}\}$  (resp.  $\{e_g \mid g \in \mathcal{G}\}$ ). We write  $\mathcal{C}$  for the subspace spanned by the set of vectors  $\mu_i$ . We write  $\mathcal{C}_{\mathcal{F}}$  for the space spanned by the set of vectors  $\mathbf{P}_{\mathcal{F}}(\mu_i)$ . We write  $\mathbf{P}_{\mathcal{F}}(\mathcal{C}_{\mathcal{F}})$  for the orthogonal complement of  $\mathcal{C}_{\mathcal{F}}$  in the  $\mathcal{F}$ -space. Moreover, we write  $\mathcal{C}_{\mathcal{F} \cup \mathcal{G}}$  for the subspace of dimension  $2T$  spanned by the union of a basis of  $\mathcal{C}_{\mathcal{F}}$  and a basis of  $\mathcal{C}_{\mathcal{G}}$ . Next, we define a key ingredient of the analysis.

**Covariance Matrix.** Let  $N$  be a large number. We define  $\hat{F}$  (resp.  $\hat{G}$ ), the *perfect sample matrix* with respect to  $\mathcal{F}$  (resp.  $\mathcal{G}$ ) as the  $N \times n/2$  matrix whose rows from  $(w_1 + \dots + w_{i-1})N + 1$  through  $(w_1 + \dots + w_i)N$  are equal to the vector  $\mathbf{P}_{\mathcal{F}}(\mu_i)/\sqrt{N}$  (resp.  $\mathbf{P}_{\mathcal{G}}(\mu_i)/\sqrt{N}$ ). For a coordinate  $f$ , let  $X_f$  be a random variable which is distributed as the  $f$ -th coordinate of the mixture  $\mathcal{D}$ . As the entry in row  $f$  and column  $g$  in the matrix  $\hat{F}^T \hat{G}$  is equal to  $\text{Cov}(X_f, X_g)$ , the covariance of  $X_f$  and  $X_g$ , we call the matrix  $\hat{F}^T \hat{G}$  the *covariance matrix* of  $\mathcal{F}$  and  $\mathcal{G}$ .

**Proof Structure.** The overall structure of our proof is as follows. First, we show that the centers of the distributions in the mixture have a high projection on the subspace of highest correlation between the coordinates. To do this, we first assume, in Section 4.1 that the input to the algorithm in Step 2 are the perfect sample matrices  $\hat{F}$  and  $\hat{G}$ . Of course, we cannot directly feed in the matrices  $\hat{F}$ ,  $\hat{G}$ , as the values of the centers are not known in advance. Next, we show in Section 4.2 that this holds even when the matrices  $F$  and  $G$  in Step 2 of Algorithm CORR-SUBSPACE are obtained by sampling. In Section 4.3, we combine these two results and prove Theorem 1. Finally, using results on distance concentration from [AM05, AK01], we complete the analysis by proving Theorem 2.

#### 4.1 The Perfect Sample Matrix

The goal of this section is to prove Lemmas 3 and 7, which establish a relationship between directions of high correlation of the covariance matrix constructed from the perfect sample matrix, and directions which contain a lot of separation between centers. Lemma 3 shows that a direction which contains a lot of effective distance between some pair of centers, is also a direction of high correlation.

Lemma 7 shows that a direction  $v \in \mathbf{P}_{\mathcal{F}}(\bar{\mathcal{C}}_{\mathcal{F}})$ , which is perpendicular to the space containing the centers, is a direction with 0 correlation. In addition, we show in Lemma 8, another property of the perfect sample matrix – the covariance matrix constructed from the perfect sample matrix has rank at most  $T$ . We conclude this section by showing in Lemma 9 that when every vector in  $\mathcal{C}$  has high spread, the directional variance of any distribution in the mixture along  $\mathcal{F}$ -space or  $\mathcal{G}$ -space is of the order of  $\sigma_*^2$ .

We begin by showing that if a direction  $v$  contains a lot of the distance between the centers, then, for most ways of splitting the coordinates, the magnitude of the covariance of the mixture along the projection of  $v$  on  $\mathcal{F}$ -space and the projection of  $v$  on  $\mathcal{G}$ -space is high. In other words, the projections of  $v$  along  $\mathcal{F}$ -space and  $\mathcal{G}$ -space are directions of high correlation.

**Lemma 3** *Let  $v$  be any vector in  $\mathcal{C}_{\mathcal{F} \cup \mathcal{G}}$  such that for some  $i$  and  $j$ ,  $\bar{d}_v(\mu_i, \mu_j) \geq 49Tc_{ij}^2 \log \Lambda$ . If  $v_{\mathcal{F}}$  and  $v_{\mathcal{G}}$  are the normalized projections of  $v$  to  $\mathcal{F}$ -space and  $\mathcal{G}$ -space respectively, then, with probability at least  $1 - \frac{1}{T}$  over the splitting step, for all such  $v$ ,  $v_{\mathcal{F}}^{\mathbf{T}} \hat{F}^{\mathbf{T}} \hat{G} v_{\mathcal{G}} \geq \tau$  where  $\tau = O\left(\frac{w_{\min} c_{ij}^2}{T \log^2 n} \cdot \sqrt{\log \Lambda}\right)$ .*

A detailed proof, presented in [Cha07], is omitted due to lack of space. However, the main ingredient of the proof is Lemma 4.

**Lemma 4** *Let  $v$  be a fixed vector in  $\mathcal{C}$  such that for some  $i$  and  $j$ ,  $\bar{d}_v(\mu_i, \mu_j) \geq 49Tc_{ij}^2 \log \Lambda$ . If  $v_{\mathcal{F}}$  and  $v_{\mathcal{G}}$  are the projections of  $v$  to  $\mathcal{F}$ -space and  $\mathcal{G}$ -space respectively, then, with probability at least  $1 - \Lambda^{-2T}$  over the splitting step,  $v_{\mathcal{F}}^{\mathbf{T}} \hat{F}^{\mathbf{T}} \hat{G} v_{\mathcal{G}} \geq 2\tau$  where  $\tau = O\left(\frac{w_{\min} c_{ij}^2}{T \log^2 n} \cdot \sqrt{\log \Lambda}\right)$ .*

Let  $\hat{F}_v$  ( $\hat{G}_v$  respectively) be the  $s \times n/2$  matrix obtained by projecting each row of  $\hat{F}$  (respectively  $\hat{G}$ ) on  $v_{\mathcal{F}}$  (respectively  $v_{\mathcal{G}}$ ). Then,

$$\begin{aligned} & v_{\mathcal{F}}^{\mathbf{T}} \hat{F}_v^{\mathbf{T}} \hat{G}_v v_{\mathcal{G}} \\ &= \sum_i w_i \langle v_{\mathcal{F}}, \mathbf{P}_{v_{\mathcal{F}}}(\mu_i - \bar{\mu}) \rangle \langle v_{\mathcal{G}}, \mathbf{P}_{v_{\mathcal{G}}}(\mu_i - \bar{\mu}) \rangle \\ &= v_{\mathcal{F}}^{\mathbf{T}} \hat{F}^{\mathbf{T}} \hat{G} v_{\mathcal{G}} \end{aligned}$$

Moreover, for any pair of vectors  $x$  in  $\mathcal{F}$ -space and  $y$  in  $\mathcal{G}$ -space such that  $\langle x, v_{\mathcal{F}} \rangle = 0$  and  $\langle y, v_{\mathcal{G}} \rangle = 0$ ,

$$x^{\mathbf{T}} \hat{F}_v^{\mathbf{T}} \hat{G}_v y = \sum_i w_i \langle x, \mathbf{P}_{v_{\mathcal{F}}}(\mu_i - \bar{\mu}) \rangle \langle y, \mathbf{P}_{v_{\mathcal{G}}}(\mu_i - \bar{\mu}) \rangle = 0$$

Therefore,  $\hat{F}_v^{\mathbf{T}} \hat{G}_v$  has rank at most 1.

The proof strategy for Lemma 4 is to show that if  $d_v(\mu_i, \mu_j)$  is large then the matrix  $\hat{F}_v^{\mathbf{T}} \hat{G}_v$  has high norm. We require the following notation. For each coordinate  $f$  we define a  $T$ -dimensional vector  $z_f$  as

$$z_f = [\sqrt{w_1} \mathbf{P}_v(\mu_1^f - \bar{\mu}^f), \dots, \sqrt{w_T} \mathbf{P}_v(\mu_T^f - \bar{\mu}^f)]$$

Notice that for any two coordinates  $f, g$ :

$$\langle z_f, z_g \rangle = \mathbf{Cov}(\mathbf{P}_v(X_f), \mathbf{P}_v(X_g))$$

, computed over the entire mixture. We also observe that

$$\sum_f \|z_f\|^2 = \sum_i w_i \cdot d_v(\mu_i, \bar{\mu})$$

The RHS of this equality is the weighted sum of the squares of the Euclidean distances between the centers of the distributions and the center of mass. By the triangle inequality, this quantity is at least  $49w_{\min} c_{ij}^2 T \log \Lambda$ . We also a couple of technical lemmas – Lemmas 5 and 6, which are stated below. The proofs of these lemmas are omitted due to lack of space, but can be found in [Cha07].

**Lemma 5** *Let  $A$  be a set of coordinates with cardinality more than  $144T^2 \log \Lambda$  such that for each  $f \in A$ ,  $\|z_f\|$  is equal and  $\sum_{f \in A} \|z_f\|^2 = D$ . Then, (1)*

$$\sum_{f, g \in A, f \neq g} \langle z_f, z_g \rangle^2 \geq \frac{D^2}{288T^2 \log \Lambda}$$

and (2) with probability  $1 - \Lambda^{-2T}$  over the splitting of coordinates in Step 1,

$$\sum_{f \in \mathcal{F} \cap A, g \in \mathcal{G} \cap A} \langle z_f, z_g \rangle^2 \geq \frac{D^2}{1152T^2 \log \Lambda}$$

**Lemma 6** *Let  $A$  be a set of coordinates such that for each  $f \in A$ ,  $\|z_f\|$  is equal and  $\sum_{f \in A} \|z_f\|^2 = D$ . If  $48T \log \Lambda + T < |A| \leq 144T^2 \log \Lambda$ , then (1)*

$$\sum_{f, g \in A, f \neq g} \langle z_f, z_g \rangle^2 \geq \frac{D^2}{1152T^4 \log \Lambda}$$

and (2) with probability  $1 - \Lambda^{-2T}$  over the splitting in Step 1,

$$\sum_{f \in \mathcal{F} \cap A, g \in \mathcal{G} \cap A} \langle z_f, z_g \rangle^2 \geq \frac{D^2}{4608T^4 \log \Lambda}$$

**Proof:**(Of Lemma 4) From the definition of effective distance, if the condition:  $\bar{d}_v(\mu_i, \mu_j) > 49c_{ij}^2 T \log \Lambda$  holds then there are at least  $49T \log \Lambda$  vectors  $z_f$  with total squared norm at least  $98w_{\min}c_{ij}^2 T \log \Lambda$ . In the sequel we will scale down each vector  $z_f$  with norm greater than  $c_{ij}\sqrt{w_{\min}}$  so that its norm is exactly  $c_{ij}\sqrt{w_{\min}}$ . We divide the vectors into  $\log n$  groups as follows: group  $B_k$  contains vectors which have norm between  $\frac{c_{ij}\sqrt{w_{\min}}}{2^k}$  and  $\frac{c_{ij}\sqrt{w_{\min}}}{2^{k-1}}$ .

We will call a vector *small* if its norm is less than  $\frac{\sqrt{w_{\min}c_{ij}}}{2\sqrt{\log n}}$ , and otherwise, we call the vector *big*. We observe that there exists a set of vector  $B$  with the following properties: (1) the cardinality of  $B$  is more than  $49T \log \Lambda$ , (2) the total sum of squares of the norm of the vectors in  $B$  is greater than  $\frac{49T \log \Lambda w_{\min}c_{ij}^2}{\log n}$ , and, (3) the ratio of the norms of any two vectors in  $B$  is at most  $2\sqrt{\log n}$ .

**Case 1:** Suppose there exists a group  $B_k$  of small vectors the squares of whose norms sum to a value greater than  $\frac{49T w_{\min}c_{ij}^2 \log \Lambda}{\log n}$ . By definition, such a group has more than  $49T \log \Lambda$  vectors, and the ratio is at most 2.

**Case 2:** Otherwise, there are at least  $49T \log \Lambda$  big vectors. By definition, the sum of the squares of their norms exceeds  $\frac{49T w_{\min}c_{ij}^2 \log \Lambda}{\log n}$ . Due to the scaling, the ratio is at most  $2\sqrt{\log n}$ .

We scale down the vectors in  $B$  so that each vector has squared norm  $\frac{w_{\min}c_{ij}^2}{2^k}$  in case 1, and, squared norm  $\frac{w_{\min}c_{ij}^2}{4 \log n}$  in case 2. Due to (2) and (3), the total squared norm of the scaled vectors is at least  $\frac{49T w_{\min}c_{ij}^2 \log \Lambda}{4 \log^2 n}$ .

Due to (1), we can now apply Lemmas 5 and 6 on the vectors to conclude that for some constant  $a_1$ , with probability  $1 - \Lambda^{-2T}$ ,

$$\sum_{f \in \mathcal{F}, g \in \mathcal{G}} \langle z_f, z_g \rangle^2 \geq a_1 \cdot \left( \frac{w_{\min}^2 c_{ij}^4 \log \Lambda}{T^2 \log^4 n} \right)$$

The above sum is the square of the Frobenius norm  $|\hat{F}_v^T \hat{G}_v|_{\mathbf{F}}$  of the matrix  $\hat{F}_v^T \hat{G}_v$ . Since  $\hat{F}_v^T \hat{G}_v$  has rank at most 1, and the maximum singular value of a rank 1 matrix is its Frobenius norm [GL96], plugging in  $\tau = O\left(\frac{w_{\min}c_{ij}^2}{T \log^2 n} \cdot \sqrt{\log \Lambda}\right)$  completes the proof.  $\square$

Next we show that a vector  $x \in \mathbf{P}_{\mathcal{F}}(\bar{\mathcal{C}}_{\mathcal{F}})$  is a direction of 0 correlation. A similar statement holds for a vector  $y \in \mathbf{P}_{\mathcal{G}}(\bar{\mathcal{C}}_{\mathcal{G}})$ .

**Lemma 7** *If at Step 2 of Algorithm CORR-SUBSPACE, the values of  $F$  and  $G$  are respectively  $\hat{F}$  and  $\hat{G}$ , and for some  $k$ , the top  $k$ -th left singular vector is  $v_k$  and the corresponding singular value  $\lambda_k$  is more than  $\tau$ , then for any vector  $x$  in  $\mathbf{P}_{\mathcal{F}}(\bar{\mathcal{C}}_{\mathcal{F}})$ ,  $\langle v_k, x \rangle = 0$ .*

**Proof:** We first show that for any  $x$  in  $\mathbf{P}_{\mathcal{F}}(\bar{\mathcal{C}}_{\mathcal{F}})$ , and any  $y$ ,  $x^T \hat{F}^T \hat{G} y = 0$ .

$$x^T \hat{F}^T \hat{G} y = \sum_{i=1}^T w_i \langle \mathbf{P}_{\mathcal{F}}(\mu_i), x \rangle \cdot \langle \mathbf{P}_{\mathcal{G}}(\mu_i), y \rangle$$

Since  $x$  is in  $\mathbf{P}_{\mathcal{F}}(\bar{\mathcal{C}}_{\mathcal{F}})$ ,  $\langle \mathbf{P}_{\mathcal{F}}(\mu_i), x \rangle = 0$ , for all  $i$ , and hence  $x^T \hat{F}^T \hat{G} y = 0$  for all  $x$  in  $\mathbf{P}_{\mathcal{F}}(\bar{\mathcal{C}}_{\mathcal{F}})$ . We now prove the Lemma by induction on  $k$ .

**Base case ( $k = 1$ ).** Let  $v_1 = u_1 + x_1$ , where  $u_1 \in \mathcal{C}_{\mathcal{F}}$  and  $x_1 \in \mathbf{P}_{\mathcal{F}}(\bar{\mathcal{C}}_{\mathcal{F}})$ . Let  $y_1$  be the top right singular vector of  $\hat{F}^T \hat{G}$ , and let  $|x_1| > 0$ . Then,  $v_1^T \hat{F}^T \hat{G} y_1 = u_1^T \hat{F}^T \hat{G} y_1$ , and  $u_1/|u_1|$  is a vector of norm 1 such that  $\frac{1}{|u_1|} u_1^T \hat{F}^T \hat{G} y_1 > v_1^T \hat{F}^T \hat{G} y_1$ , which contradicts the fact that  $v_1$  is the top left singular vector of  $\hat{F}^T \hat{G}$ .

**Inductive case.** Let  $v_k = u_k + x_k$ , where  $u_k \in \mathcal{C}_{\mathcal{F}}$  and  $x_k \in \mathbf{P}_{\mathcal{F}}(\bar{\mathcal{C}}_{\mathcal{F}})$ . Let  $y_k$  be the top  $k$ -th right singular vector of  $\hat{F}^T \hat{G}$ , and let  $|x_k| > 0$ . We first show that  $u_k$  is orthogonal to each of the vectors  $v_1, \dots, v_{k-1}$ . Otherwise, suppose there is some  $j$ ,  $1 \leq j \leq k-1$ , such that  $\langle u_k, v_j \rangle \neq 0$ . Then,  $\langle v_k, v_j \rangle = \langle x_k, v_j \rangle + \langle u_k, v_j \rangle = \langle u_k, v_j \rangle \neq 0$ . This contradicts the fact that  $v_k$  is a left singular vector of  $\hat{F}^T \hat{G}$ . Therefore,  $v_k^T \hat{F}^T \hat{G} y_k = u_k^T \hat{F}^T \hat{G} y_k$ , and  $u_k/|u_k|$  is a vector of norm 1, orthogonal to  $v_1, \dots, v_{k-1}$  such that  $\frac{1}{|u_k|} u_k^T \hat{F}^T \hat{G} y_k > v_k^T \hat{F}^T \hat{G} y_k$ . This contradicts the fact that  $v_k$  is the top  $k$ -th left singular vector of  $\hat{F}^T \hat{G}$ . The Lemma follows.  $\square$

**Lemma 8** *The covariance matrix  $\hat{F}^T \hat{G}$  has rank at most  $T$ .*

The proof is omitted due to space constraints.

Finally, we show that if the spread of every vector in  $\mathcal{C}$  is high, then with high probability over the splitting of coordinates in Step 1 of Algorithm CORR-SUBSPACE, the maximum directional variances of any distribution  $D_i$  in  $\mathcal{C}_{\mathcal{F}}$  and  $\mathcal{C}_{\mathcal{G}}$  are high. This means that there is enough information in both  $\mathcal{F}$ -space and  $\mathcal{G}$ -space for correctly clustering the distributions through distance concentration.

**Lemma 9** *If every vector  $v \in \mathcal{C}$  has spread at least  $32T \log \frac{\sigma}{\sigma_*}$ , then, with constant probability over the splitting of coordinates in Step 1 of Algorithm CORR-SUBSPACE, the maximum variance along any direction in  $\mathcal{C}_{\mathcal{F}}$  or  $\mathcal{C}_{\mathcal{G}}$  is at most  $5\sigma_*^2$ .*

**Proof:**(Of Lemma 9) Let  $v$  and  $v'$  be two unit vectors in  $\mathcal{C}$ , and let  $v_{\mathcal{F}}$  (resp.  $v'_{\mathcal{F}}$ ) and  $v_{\mathcal{G}}$  (resp.  $v'_{\mathcal{G}}$ ) denote the normalized projections of  $v$  (resp.  $v'$ ) on  $\mathcal{F}$ -space and  $\mathcal{G}$ -space respectively. If  $\|v_{\mathcal{F}} - v'_{\mathcal{F}}\| < \frac{\sigma_*}{\sigma}$ , then, the

directional variance of any  $D_i$  in the mixture along  $v'_\mathcal{F}$  can be written as:

$$\begin{aligned} & \mathbf{E}[\langle v'_\mathcal{F}, x - \mathbf{E}[x] \rangle^2] \\ &= \mathbf{E}[\langle v_\mathcal{F}, x - \mathbf{E}[x] \rangle^2] + \mathbf{E}[\langle v'_\mathcal{F} - v_\mathcal{F}, x - \mathbf{E}[x] \rangle^2] \\ & \quad + 2\mathbf{E}[\langle v_\mathcal{F}, x - \mathbf{E}[x] \rangle \langle v'_\mathcal{F} - v_\mathcal{F}, x - \mathbf{E}[x] \rangle] \\ &\leq \mathbf{E}[\langle v_\mathcal{F}, x - \mathbf{E}[x] \rangle^2] + \|v_\mathcal{F} - v'_\mathcal{F}\|^2 \sigma^2 \end{aligned}$$

Thus, the directional variance of any distribution in the mixture along  $v'$  is at most the directional variance along  $v$ , plus an additional  $\sigma_*^2$ . Therefore, to show this lemma, we need to show that if  $v$  is any vector on a  $\frac{\sigma_*}{\sigma}$ -cover of  $\mathcal{C}$ , then with high probability over the splitting of coordinates in Step 1 of Algorithm CORR-SUBSPACE, the directional variances of any  $D_i$  in the mixture along  $v_\mathcal{F}$  and  $v_\mathcal{G}$  are at most  $4\sigma_*^2$ .

We show this in two steps. First we show that for any  $v$  in a  $\frac{\sigma_*}{\sigma}$ -cover of  $\mathcal{C}$ ,  $\frac{1}{4} \leq \sum_{f \in \mathcal{F}} (v^f)^2 \leq \frac{3}{4}$ . Then, we show that this condition means that for this vector  $v$ , the maximum directional variances along  $v_\mathcal{F}$  and  $v_\mathcal{G}$  are at most  $4\sigma_*^2$ .

Let  $v$  be any fixed unit vector in  $\mathcal{C}$ . We first show that with probability  $1 - \left(\frac{\sigma_*}{\sigma}\right)^{2T}$  over the splitting of coordinates in Step 1 of Algorithm CORR-SUBSPACE,  $\frac{1}{4} \leq \sum_{f \in \mathcal{F}} (v^f)^2 \leq \frac{3}{4}$ . To show this bound, we apply the Method of Bounded Difference [PD05]. Since we split the coordinates into  $\mathcal{F}$  and  $\mathcal{G}$  uniformly at random,  $\mathbf{E}[\sum_{f \in \mathcal{F}} (v^f)^2] = \frac{1}{2}$ . Let  $\gamma_f$  be the change in  $\sum_{f \in \mathcal{F}} (v^f)^2$  when the inclusion or exclusion of coordinate  $f$  in the set  $\mathcal{F}$  changes. Then,  $\gamma_f = (v^f)^2$  and  $\gamma = \sum_f \gamma_f^2$ . Since the spread of vector  $v$  is at least  $32T \log \frac{\sigma}{\sigma_*}$ ,  $\gamma = \sum_f (v^f)^4 \leq \frac{1}{32T \log \frac{\sigma}{\sigma_*}}$ , and from the Method of Bounded Differences,

$$\begin{aligned} \Pr\left[\left|\sum_{f \in \mathcal{F}} (v^f)^2 - \mathbf{E}\left[\sum_{f \in \mathcal{F}} (v^f)^2\right]\right| > \frac{1}{4}\right] &\leq e^{-1/32\gamma} \\ &\leq \left(\frac{\sigma_*}{\sigma}\right)^{2T} \end{aligned}$$

By taking an union bound over all  $v$  on a  $\frac{\sigma_*}{\sigma}$ -cover of  $\mathcal{C}$ , we deduce that for any such  $v$ ,  $\frac{1}{4} \leq \sum_{f \in \mathcal{F}} (v^f)^2 \leq \frac{3}{4}$ .

Since the maximum directional variance of any distribution  $D_i$  in the mixture in  $\mathcal{C}$  is at most  $\sigma_*^2$ ,  $\sum_f (v^f)^2 (\sigma_i^f)^2 \leq \sigma_*^2$ . Therefore the maximum variance along  $v_\mathcal{F}$  as well as  $v_\mathcal{G}$  can be computed as:

$$\frac{1}{\|v_\mathcal{F}\|^2} \sum_{f \in \mathcal{F}} (v^f)^2 (\sigma_i^f)^2 \leq \frac{1}{\|v_\mathcal{F}\|^2} \sum_f (v^f)^2 (\sigma_i^f)^2 \leq 4\sigma_*^2$$

The lemma follows.  $\square$

## 4.2 Working with Real Samples

In this section, we show that given sufficient samples, the properties of the matrix  $F^T G$ , where  $F$  and  $G$  are

generated by sampling in Step 2 of Algorithm CORR-CLUSTER are very close to the properties of the matrix  $\hat{F}^T \hat{G}$ . The lemmas are stated below. The proofs are omitted due to space constraints, but can be found in [Cha07]. The proofs use the Method of Bounded Differences (when the input is a mixture of binary product distributions) and the Gaussian Concentration of Measure Inequality (for axis-aligned Gaussians).

The central lemma of this section is Lemma 10, which shows that, if there are sufficiently many samples, for any set of  $2m$  vectors,  $\{v_1, \dots, v_m\}$  and  $\{y_1, \dots, y_m\}$ ,  $\sum_k v_k^T F^T G y_k$  and  $\sum_k v_k^T \hat{F}^T \hat{G} y_k$  are very close. This lemma is then used to prove Lemmas 11 and 12. Lemma 11 shows that the top few singular vectors of  $F^T G$  output by Algorithm CORR-SUBSPACE have very low projection on  $\mathbf{P}_\mathcal{F}(\bar{\mathcal{C}}_\mathcal{F})$  or  $\mathbf{P}_\mathcal{G}(\bar{\mathcal{C}}_\mathcal{G})$ . Lemma 12 shows that the rank of the matrix  $F^T G$  is almost  $T$ , in the sense that the  $T + 1$ -th singular value of this matrix is very low.

**Lemma 10** *Let  $U = \{u_1, \dots, u_m\}$ ,  $Y = \{y_1, \dots, y_m\}$  be any two sets of orthonormal vectors, and let  $F$  and  $G$  be the matrices generated by sampling in Step 2 of the algorithm. If the number of samples  $|S|$  is greater than  $\Omega\left(\frac{m^3 n^2 \log n \log(\sigma_{\max}/\delta)}{\delta^2}\right)$  (for Binary Product Distributions), and  $\Omega(\max(a_1, a_2))$  (for Axis-Aligned Gaussians), where  $a_1 = \frac{\sigma^4 m^4 n^2 \log^2 n \log^2(\sigma_{\max}/\delta)}{\delta^2}$ , and  $a_2 = \frac{\sigma^2 \sigma_{\max}^2 m^3 n \log n \log(\sigma_{\max}/\delta)}{\delta^2}$ , then, with probability at least  $1 - 1/n$ ,*

$$\left| \sum_k u_k^T (F^T G - \mathbf{E}[F^T G]) y_k \right| \leq \delta$$

**Lemma 11** *Let  $F$  and  $G$  be the matrices generated by sampling in Step 2 of the algorithm, and let  $v_1, \dots, v_m$  be the vectors output by the algorithm in Step 4. If the number of samples  $|S|$  is greater than*

$\Omega\left(\frac{m^3 n^2 \log n (\log \Lambda + \log \frac{1}{\epsilon})}{\tau^2 \epsilon^4}\right)$  (for Binary Product Distributions), and  $\max(a_1, a_2)$  (for Axis-Aligned Gaussians)

where  $a_1 = \frac{\sigma^4 m^4 n^2 \log^2 n \log^2(\Lambda/\epsilon)}{\tau^2 \epsilon^4}$ , and

$a_2 = \frac{\sigma^2 \sigma_{\max}^2 m^3 n \log n \log(\Lambda/\epsilon)}{\tau^2 \epsilon^4}$ , then, for each  $k$ , and any  $x$  in  $\mathbf{P}_\mathcal{F}(\bar{\mathcal{C}}_\mathcal{F})$ ,  $\langle v_k, x \rangle \leq \epsilon$ .

**Lemma 12** *Let  $F$  and  $G$  be the matrices generated by sampling in Step 2 of Algorithm CORR-SUBSPACE. If the number of samples  $|S|$  is greater than*

$\Omega\left(\frac{T^3 n^2 \log n \log \Lambda}{\tau^2}\right)$  (for binary product distributions) and

$\Omega\left(\max\left(\frac{\sigma^4 T^4 n^2 \log^2 \log \Lambda}{\tau^2}, \frac{\sigma_{\max}^2 \sigma^2 T^3 n \log n \log \Lambda}{\tau^2}\right)\right)$  for axis-

aligned Gaussians, then,  $\lambda_{T+1}$ , the  $T + 1$ -th singular value of the matrix  $F^T G$  is at most  $\tau/8$ .

### 4.3 The Combined Analysis

In this section, we combine the lemmas proved in Sections 4.1 and 4.2 to prove Theorem 1.

We begin with a lemma which shows that if every vector in  $\mathcal{C}$  has spread  $32T \log \frac{\sigma}{\sigma_*}$ , then the maximum directional variance in  $\mathcal{K}$ , the space output by Algorithm CORR-SUBSPACE, is at most  $11\sigma_*^2$ .

**Lemma 13** *Let  $\mathcal{K}$  be the subspace output by the algorithm, and let  $v$  be any vector in  $\mathcal{K}$ . If every vector in  $\mathcal{C}$  has spread  $32T \log \frac{\sigma}{\sigma_*}$ , and the number of samples  $|S|$  is greater than*

$$\Omega \left( \max \left( \frac{\sigma^6 T^4 n^2 \log^2 \log \Lambda}{\tau^2 \sigma_*^4}, \frac{\sigma_{\max}^2 \sigma^4 T^3 n \log n \log \Lambda}{\tau^2 \sigma_*^4} \right) \right) \text{ then}$$

for any  $i$  the maximum variance of  $D_i$  along  $v$  is at most  $11\sigma_*^2$ .

The proof is omitted due to space constraints, and can be found in [Cha07].

The above Lemmas are now combined to prove Theorem 1.

**Proof:**(Of Theorem 1)

Suppose  $\mathcal{K} = \mathcal{K}_L \cup \mathcal{K}_R$ , where  $\mathcal{K}_L = \{v_1, \dots, v_m\}$ , the top  $m$  left singular vectors of  $F^T G$  and  $\mathcal{K}_R = \{y_1, \dots, y_m\}$  are the corresponding right singular vectors. We abuse notation and use  $v_k$  to denote the vector  $v_k$  concatenated with a vector consisting of  $n/2$  zeros, and use  $y_k$  to denote the vector consisting of  $n/2$  zeros concatenated with  $y_k$ . Moreover, we use  $\mathcal{K}$ ,  $\mathcal{K}_L$ , and  $\mathcal{K}_R$  interchangeably to denote sets of vectors and the subspace spanned by those sets of vectors.

We show that with probability at least  $1 - \frac{1}{T}$  over the splitting step, there exists no vector  $v \in \mathcal{C}_{\mathcal{F} \cup \mathcal{G}}$  such that (1)  $v$  is orthogonal to the space spanned by the vectors  $\mathcal{K}$  and (2) there exists some pair of centers  $i$  and  $j$  such that  $\bar{d}_v(\mu_i, \mu_j) > 49T c_{ij}^2 \log \Lambda$ . For contradiction, suppose there exists such a vector  $v$ .

Then, if  $v_{\mathcal{F}}$  and  $v_{\mathcal{G}}$  denote the normalized projections of  $v$  onto  $\mathcal{F}$ -space and  $\mathcal{G}$ -space respectively, from Lemma 3,  $v_{\mathcal{F}}^T \hat{F}^T G v_{\mathcal{G}} \geq \tau$  with probability at least  $1 - \frac{1}{T}$  over the splitting step. From Lemma 10, if the number of samples  $|S|$  is greater than  $\Omega \left( \frac{T^3 n^2 \log n \log \Lambda}{\tau^2} \right)$  for binary product distributions, and if  $|S|$  is greater than  $\Omega \left( \max \left( \frac{\sigma^4 n^2 \log^2 \log \Lambda}{\tau^2}, \frac{\sigma_{\max}^2 n \log n \log \Lambda}{\tau^2} \right) \right)$  for axis-aligned Gaussians,  $v_{\mathcal{F}}^T F^T G v_{\mathcal{G}} \geq \frac{\tau}{2}$  with at least constant probability. Since  $v$  is orthogonal to the space spanned by  $\mathcal{K}$ ,  $v_{\mathcal{F}}$  is orthogonal to  $\mathcal{K}_L$  and  $v_{\mathcal{G}}$  is orthogonal to  $\mathcal{K}_R$ . As  $\lambda_{m+1}$  is the maximum value of  $x^T F^T G y$  over all vectors  $x$  orthogonal to  $\mathcal{K}_L$  and  $y$  orthogonal to  $\mathcal{K}_R$ ,  $\lambda_{m+1} \geq \frac{\tau}{2}$ , which is a contradiction. Moreover, from Lemma 12,  $\lambda_{T+1} < \frac{\tau}{8}$ , and hence  $m \leq T$ .

Let us construct an orthonormal series of vectors  $v_1, \dots, v_m, \dots$  which are *almost* in  $\mathcal{C}_{\mathcal{F}}$  as follows.

$v_1, \dots, v_m$  are the vectors output by Algorithm CORR-SUBSPACE. We inductively define  $v_l$  as follows. Suppose for each  $k$ ,  $v_k = u_k + x_k$ , where  $u_k \in \mathcal{C}_{\mathcal{F}}$  and  $x_k \in \mathcal{P}_{\mathcal{F}}(\bar{\mathcal{C}}_{\mathcal{F}})$ . Let  $u_l$  be a unit vector in  $\mathcal{C}_{\mathcal{F}}$  which is perpendicular to  $u_1, \dots, u_{l-1}$ . Then,  $v_l = u_l$ . By definition, this vector is orthogonal to  $u_1, \dots, u_{l-1}$ . In addition, for any  $k \neq l$ ,  $\langle v_l, v_k \rangle = \langle u_l, u_k \rangle + \langle u_l, x_k \rangle = 0$ , and  $v_l$  is also orthogonal to  $v_1, \dots, v_{l-1}$ . Moreover, if  $\epsilon < \frac{1}{100T}$ ,  $u_1, \dots, u_m$  are linearly independent, and we can always find  $\dim(\mathcal{C}_{\mathcal{F}})$  such vectors. Similarly, we construct a set of vectors  $y_1, y_2, \dots$ . Let us call the combined set of vectors  $\mathcal{C}^*$ .

We now show that if there are sufficient samples,  $d_{\bar{\mathcal{C}}^*}(\mu_i, \mu_j) \leq c_{ij}^2$ . Note that for any unit vector  $v^*$  in  $\mathcal{C}^*$ , and any unit  $x \in \bar{\mathcal{C}}_{\mathcal{F} \cup \mathcal{G}}$ ,  $\langle v, x \rangle \leq m\epsilon$ . Also, note that for any  $u_k$  and  $u_l$ ,  $k \neq l$ ,  $|\langle u_k, u_l \rangle| \leq \epsilon^2$ , and  $\|u_k\|^2 \geq 1 - \epsilon^2$ . Let  $v = \sum_k \alpha_k u_k$  be any unit vector in  $\mathcal{C}_{\mathcal{F} \cup \mathcal{G}}$ . Then,  $1 = \|v\|^2 = \sum_{k,k'} \alpha_k \alpha_{k'} \langle u_k, u_{k'} \rangle \geq \sum_k \alpha_k^2 \|u_k\|^2 - \Omega(T^2 \epsilon^2)$ .

The projection of  $v$  on  $\mathcal{C}^*$  can be written as:

$$\begin{aligned} \sum_k \langle v, v_k \rangle^2 &= \sum_k \langle v, u_k \rangle^2 \\ &= \sum_k \sum_l \alpha_l^2 \langle u_k, u_l \rangle^2 + 2 \sum_{l,l'} \alpha_l \alpha_{l'} \langle u_k, u_l \rangle \langle u_k, u_{l'} \rangle \\ &\geq \sum_k \alpha_k^2 \|u_k\|^4 - T^3 \epsilon^4 \geq 1 - \Omega(T^2 \epsilon^2) \end{aligned}$$

The last step follows because for each  $k$ ,  $\|u_k\|^2 \geq 1 - \epsilon^2$ . If the number of samples  $|S|$  is greater than  $\Omega \left( \frac{m^3 n^2 \log n (\log \Lambda + \log 100T)}{\tau^2 T^4} \right)$  (for Binary Product Distributions), and  $\max \left( \frac{\sigma^4 m^4 n^2 \log^2 n \log^2 (100T\Lambda)}{\tau^2 T^4}, \frac{\sigma_{\max}^2 \sigma^2 m^3 n \log \log (100T\Lambda)}{\tau^2 T^4} \right)$  (for axis-aligned Gaussians), then,  $\epsilon < 1/100T$ . Therefore,

$$d_{\bar{\mathcal{C}}^*}(\mu_i, \mu_j) \leq \frac{1}{100} d(\mu_i, \mu_j)$$

For any  $i$  and  $j$ ,

$d(\mu_i, \mu_j) = d_{\mathcal{K}}(\mu_i, \mu_j) + d_{\mathcal{C}^* \setminus \mathcal{K}}(\mu_i, \mu_j) + d_{\bar{\mathcal{C}}^*}(\mu_i, \mu_j)$   
Since vectors  $v_{m+1}, \dots$  and  $y_{m+1}, \dots$ , all belong to  $\mathcal{C}_{\mathcal{F} \cup \mathcal{G}}$  (as well as  $\mathcal{C}^* \setminus \mathcal{K}$ , there exists no  $v \in \mathcal{C}^* \setminus \mathcal{K}$  with the Conditions (1) and (2) in the previous paragraph, and  $\bar{d}_{\mathcal{C}_{\mathcal{F} \cup \mathcal{G}} \setminus \mathcal{K}}(\mu_i, \mu_j) \leq 49T c_{ij}^2 \log \Lambda$ . That is, the actual distance between  $\mu_i$  and  $\mu_j$  in  $\mathcal{C}_{\mathcal{F} \cup \mathcal{G}} \setminus \mathcal{K}$  (as well as  $\mathcal{C}^* \setminus \mathcal{K}$ ) is at most the contribution to  $d(\mu_i, \mu_j)$  from the top  $49T c_{ij}^2 \log \Lambda$  coordinates, and the contribution to  $d(\mu_i, \mu_j)$  from  $\mathcal{K}$  and  $\bar{\mathcal{C}}^*$  is at least the contribution from the rest of the coordinates. Since  $d_{\bar{\mathcal{C}}^*}(\mu_i, \mu_j) \leq \frac{1}{100} d(\mu_i, \mu_j)$ , the distance between  $\mu_i$  and  $\mu_j$  in  $\mathcal{K}$  is at least  $\frac{99}{100} \bar{d}(\mu_i, \mu_j) - 49T \log \Lambda c_{ij}^2$ . The first part of the theorem follows.

The second part of the theorem follows directly from Lemma 13.  $\square$

## References

- [AK01] S. Arora and R. Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of 33rd ACM Symposium on Theory of Computing*, pages 247–257, 2001.
- [AM05] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 458–469, 2005.
- [AT98] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of Conference on Learning Theory*, 1998.
- [BNJ03] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, (3):993–1022, January 2003.
- [Cha07] K. Chaudhuri. *Learning Mixtures of Distributions*. PhD thesis, University of California, Berkeley, 2007. UCB/EECS-2007-124.
- [CHRZ07] K. Chaudhuri, E. Halperin, S. Rao, and S. Zhou. A rigorous analysis of population stratification with limited data. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2007.
- [Das99] S. Dasgupta. Learning mixtures of gaussians. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pages 634–644, 1999.
- [DHKS05] A. Dasgupta, J. Hopcroft, J. Kleinberg, and M. Sandler. On learning mixtures of heavy-tailed distributions. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science*, pages 491–500, 2005.
- [DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39, pages 1–38, 1977.
- [DS00] S. Dasgupta and L. Schulman. A two-round variant of em for gaussian mixtures. In *Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.
- [FM99] Y. Freund and Y. Mansour. Estimating a mixture of two product distributions. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, Morgan Kaufmann Publishers, 1999.
- [FOS05] J. Feldman, R. O’Donnell, and R. Servedio. Learning mixtures of product distributions over discrete domains. In *Proceedings of FOCS*, 2005.
- [FOS06] J. Feldman, R. O’Donnell, and R. Servedio. Learning mixtures of gaussians with no separation assumptions. In *Proceedings of COLT*, 2006.
- [GL96] G. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.
- [KF07] S. Kakade and D. Foster. Multi-view regression via canonical correlation analysis. In *Proc. of Conference on Learning Theory*, 2007.
- [KS04] Jon M. Kleinberg and Mark Sandler. Using mixture models for collaborative filtering. In *STOC*, pages 569–578, 2004.
- [KSV05] R. Kannan, H. Salmasian, and S. Vempala. The spectral method for general mixture models. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.
- [Llo82] S.P. Lloyd. Least squares quantization in pcm. *IEEE Trans. on Information Theory*, 1982.
- [MB88] G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988.
- [PD05] A. Panconesi and D. Dubhashi. Concentration of measure for the analysis of randomised algorithms. Draft, 2005.
- [PFK02] C. Pal, B. Frey, and T. Kristjansson. Noise robust speech recognition using Gaussian basis functions for non-linear likelihood function approximation. In *ICASSP ’02: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–405–I–408, 2002.
- [PSD00] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:954–959, June 2000.
- [Rey95] D. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Communications*, 1995.
- [SRH07] Srinath Sridhar, Satish Rao, and Eran Halperin. An efficient and accurate graph-based approach to detect population substructure. In *RECOMB*, 2007.
- [TSM85] D.M. Titterton, A.F.M. Smith, and U.E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.
- [VW02] V. Vempala and G. Wang. A spectral algorithm of learning mixtures of distributions. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 113–123, 2002.

---

# Beyond Gaussians: Spectral Methods for Learning Mixtures of Heavy-Tailed Product Distributions

---

**Kamalika Chaudhuri**

Information Theory and Applications, UC San Diego  
kamalika@soe.ucsd.edu

**Satish Rao**

Computer Science Division, UC Berkeley  
satishr@cs.berkeley.edu

## Abstract

We study the problem of learning mixtures of distributions, a natural formalization of clustering. A mixture of distributions is a collection of distributions  $\mathcal{D} = \{D_1, \dots, D_T\}$  and weights  $w_1, \dots, w_T$ . A sample from a mixture is drawn by selecting  $D_i$  with probability  $w_i$  and then selecting a sample from  $D_i$ . The goal, in learning a mixture, is to learn the parameters of the distributions comprising the mixture, given only samples from the mixture.

In this paper, we focus on learning mixtures of heavy-tailed product distributions, which was studied by [DHKS05]. The challenge in learning such mixtures is that the techniques developed for learning mixture-models, such as spectral methods and distance concentration, do not apply. The previous algorithm for this problem was due to [DHKS05], which achieved performance comparable to the algorithms of [AM05, KSV05, CR08] given a mixture of Gaussians, but took time exponential in the dimension. We provide an algorithm which has the same performance, but runs in polynomial time.

Our main contribution is an embedding which transforms a mixture of heavy-tailed product distributions into a mixture of distributions over the hypercube in a higher dimension, while still maintaining separability. Combining this embedding with standard spectral techniques results in algorithms that can learn mixtures of heavy-tailed distributions with separation comparable to the guarantees of [DHKS05]. Our algorithm runs in time polynomial in the dimension, number of clusters, and imbalance in the weights.

## 1 Introduction

We study the problem of learning mixtures of distributions, a natural formalization of clustering. A *mixture of distributions* is a collection of  $T$  distributions  $\mathcal{D} = \{D_1, \dots, D_T\}$  over  $\mathbf{R}^n$  and mixing weights  $w_1, \dots, w_T$  such that  $\sum_{i=1}^T w_i = 1$ . A sample from a mixture is drawn by first selecting  $i$  with probability  $w_i$ , and then choosing a random sample from  $D_i$ . The goal, in learning a mixture, is to learn the parameters of the distributions comprising the mixture, and to classify the samples according to source distribution, given only the ability to sample from the mixture.

Learning mixtures of distributions frequently arise in many applications in machine learning, and a fair amount of empirical work has been devoted to the problem. On the theoretical side, all work (except for the work of [DHKS05]) has focussed on learning mixtures of distributions with one of the following characteristics: either the distributions in question have exponentially-decaying tails, for example, mixtures of Gaussians [Das99, DS00, AM05, KSV05, AK01, VW02], or they have severely bounded range, for example, mixtures of binary product distributions [FOS05, CR08]. In the latter case, the bounds deteriorate with the maximum range of values taken by any coordinate of a sample drawn from the mixture.

In this paper, we focus our attention to learning mixtures of more general distributions. In particular, we study learning mixtures of heavy-tailed product distributions, which was introduced by Dasgupta *et. al* [DHKS05].

If the distributions comprising a mixture are very close together, in the sense that they have a high overlap in probability mass, then, even if we knew the parameters of the distributions comprising the mixture, the samples would be hard to classify. To address this, Dasgupta [Das99] introduced the notion of a *separation condition*. A separation condition is a promise that the distributions comprising a mixture are sufficiently different according to some measure, and the goal of the algorithm is to learn correctly a mixture which obeys a certain separation condition. Naturally, the less stringent a separation condition is, the harder it is to learn a mixture, and therefore, a line of theoretical research

has focussed on learning mixtures of distributions under less and less restrictive separation conditions. For mixtures of Gaussians, the common measure of separation used is the minimum distance between the means of any two distributions in the mixture, parameterized by the maximum directional standard deviation of any distribution in the mixture. However, this is not a good measure for the type of distributions considered here, as the directional standard deviation may be infinite; following [DHKS05], we therefore use as a measure of separation the minimum distance between the *medians* of any two distributions in the mixture, as parameterized by the maximum  $\frac{3}{4}$ -radius. Recall that given  $0 < \beta \leq 1$ , the  $\beta$ -radius of a one-dimensional distribution  $D$  with median  $m(D)$  is the minimum number  $R_\beta$  such that the probability mass of  $D$  in the interval  $[m(D) - R_\beta, m(D) + R_\beta]$  is at least  $\beta$ .

The major challenge in learning mixtures of heavy-tailed distributions is that none of the tools developed in the literature for learning mixtures of Gaussians or binary product distributions work when the mixture consists of more general distributions. The key ingredients of such algorithms for learning mixtures are: (1) a singular value decomposition of part [CR08] or whole [VW02, KSV05, AM05] of the covariance matrix of the samples and (2) distance-thresholding based clustering algorithms. Singular value decompositions of the covariance matrix do not converge if the distributions have infinite variance. Even for mixtures of distributions with finite variance, distance concentration, which works on the principle that two samples from the same distribution are closer in space than two samples from different distributions, does not work unless the distributions have light tails or a very small range. The previous algorithm for the problem is due to [DHKS05], which learns mixtures of heavy-tailed distributions with performance comparable to the performance of algorithms in [AM05, KSV05, CR08] given a mixture of Gaussians; however, it involves an exhaustive search over all partitions of  $\Omega(n)$  samples, where  $n$  is the number of dimensions, and hence takes time exponential in the dimension.

In this paper, we show a general procedure for transforming mixtures of heavy-tailed product distributions into mixtures which are more well-behaved, while preserving the separability of the distributions in the mixture. In particular, we provide an efficiently computable embedding from  $\mathbf{R}^n$  to  $\{0, 1\}^{O(n^{3/2})}$ . Our embedding, when applied to a mixture of heavy-tailed product distributions which have certain conditions comparable to those in [DHKS05], produces a mixture of distributions in  $\{0, 1\}^{O(n^{3/2})}$  with centers that are far apart. In addition, we show that the resulting mixture has good properties such that standard algorithms for learning mixtures of binary product distributions – such as the SVD-based algorithms of [AM05, KSV05] and the correlations-based algorithm of [CR08] can be applied to learn it, leading to efficient algorithms for learning mixtures of

heavy-tailed product distributions.

More specifically, our results are as follows. Given a mixture of general product distributions, such that each distribution is symmetric about its median, and has  $\frac{3}{4}$ -radius upper-bounded by  $R$ , our embedding transforms it into a mixture of distributions over  $\{0, 1\}^{O(n^{3/2})}$ , while preserving the distance between the centers in a certain sense which is explained in Theorem 1. We can now apply either SVD-based clustering algorithms [KSV05, AM05], and in this case, for success with probability  $1 - \delta$ , we require that (a) the separation between the medians of distributions  $D_i$  and  $D_j$  be  $\Omega(R(w_i^{-1/2} + w_j^{-1/2}) + R\sqrt{T \log \frac{nT}{\delta}})$  and (b) this separation be spread across  $\Omega((w_i^{-1/2} + w_j^{-1/2})^2 + T \log \frac{nT}{\delta})$  coordinates. Alternatively, we can apply the correlations-based algorithm of [CR08] on the transformed mixture, to get a logarithmic dependence on the mixing weights. In this case, to learn the mixture with probability  $1 - \delta$ , we require that (a) the minimum distance between the medians of any two distributions in the mixture to be  $\Omega(R\sqrt{T \log \Lambda} + R\sqrt{T \log(nT/\delta)})$  and (b) that this separation to be spread across  $\Omega(T \log \Lambda + T \log(nT/\delta))$  coordinates, where  $\Lambda$  is polynomial in  $n, T$  and  $\frac{1}{w_{\min}}$ .

We note that conditions comparable to all these four conditions are required by [DHKS05] for learning mixtures of heavy-tailed distributions; our work improves on their results by providing a polynomial-time algorithm for the problem, as opposed to an exponential-time algorithm. In addition, we also do not need the restriction, needed by [DHKS05], that the probability density function should be decreasing with distance from the median. We also note that the guarantees of our algorithms are comparable to the guarantees of [AM05, KSV05, CR08] when the input is a mixture of axis-aligned Gaussians.

## Our Techniques

An initial approach for converting a mixture of general product distributions to a mixture of distributions with better properties is to remove the *outlier points*, which lie very far from the other samples. However, for the types of distributions we consider, a sample may be an outlier along each coordinate with constant ( $1/4$ ) probability, and since there are  $n$  coordinates, with high probability, every point is an outlier. Another approach could be to try to round the outlier points along each dimension; however, since the different mixture components may have different mixing weights, given samples from the mixture, it is hard to determine which of the samples are outliers along a specific coordinate.

To address these issues, we use techniques from metric embeddings [Ind01]. The main idea behind our embedding is to use many random *cutting points* to divide the real line into intervals of length  $\Omega(R)$ ; points which fall into the even intervals are then mapped to 0 and those which fall into the odd intervals are mapped to 1. Although this process does not preserve distances be-

tween all pairs of points, we show that this succeeds in separating the centers of two distributions which have medians that are far apart compared to their  $3/4$ -radius  $R$ . Our techniques are related to techniques in metric-embedding [Ind01]; however, so far as we know, this is the first time they have been applied to learning mixtures of distributions. Combining our embedding with existing standard algorithms for learning mixtures of distributions, we get efficient algorithms for learning mixtures of heavy-tailed distributions.

## 2 Related Work

### Heavy-Tailed Mixtures

The work most related to ours is the work of Dasgupta, Hopcroft, Kleinberg and Sandler [DHKS05]. Dasgupta *et. al* [DHKS05] introduced the problem of learning mixtures of heavy-tailed distributions and the notion of using the distance between the medians, parameterized by the half-radius, as a measure of separation between such distributions. Their work deals with the class of all product distributions in which the distribution of each coordinate has the following properties: (a) symmetry around the median (b) decreasing probability density with distance from the median and (c)  $\frac{1}{2}$ -radius upper bounded by  $R'$ . In contrast, we require the distribution of each coordinate to be symmetric about its median and have  $\frac{3}{4}$ -radius upper bounded by  $R$ , and do not require the second assumption of [DHKS05].

[DHKS05] provide two algorithms for learning such mixtures. First, they provide an algorithm which requires a separation of  $\Omega(R' \sqrt{\frac{T}{\delta}})$  and a spreading condition that the distance between the medians of any two distributions in the mixture should be spread over  $\Theta(T/\delta)$  coordinates, to classify a  $1 - \delta$  fraction of the samples correctly. This algorithm works by performing an exhaustive search over all partitions of  $\Theta(\frac{n \log(nT)}{w_{\min}})$  samples, and therefore has a running time exponential in  $\Theta(\frac{n \log(nT)}{w_{\min}})$ . In contrast, our algorithms work with similar separation and spreading conditions, and only take time polynomial in  $n$ .

Second, they provide an algorithm which works with a stronger separation requirement of  $\Omega(R' \sqrt{n})$  and a spreading condition that the distance between the medians of any two distributions in the mixture be spread over  $\Theta(T/\delta)$  coordinates. Typically, for such problems, the dimension  $n$  is much larger than the number of clusters  $T$ , and hence the separation needed here is much larger than the separation needed by the previous algorithm and our algorithms. This algorithm works by performing an exhaustive search over all partitions of  $\Theta(\frac{\log(nT)}{w_{\min}})$  samples, and therefore has a running time exponential in  $\Theta(\frac{\log(nT)}{w_{\min}})$ . Since  $w_{\min}$  is at most  $\frac{1}{T}$ , this may be polynomial in  $n$  but remains exponential in  $T$ . In contrast, the running times of our algorithms are polynomial in  $n$ ,  $T$ , and  $\frac{1}{w_{\min}}$ , and for distributions in which the  $\frac{3}{4}$ -radius is comparable with the half-radius,

our algorithms work with separation and spreading constraints comparable to algorithm (1) of [DHKS05].

[DHKS05] also works with a second class of distributions, which have mildly decaying tails. In this case, they provide an algorithm which clusters correctly  $1 - \delta$  fraction of the samples in time exponential in  $n$ , so long as the separation between any two distributions is  $\Omega(R'T^{5/2}/\delta^2)$ .

### Other Mixture Models

There has been a long line of theoretical work on learning mixtures of Gaussians. For this problem, the separation condition is usually expressed in terms of  $n$ , the number of dimensions,  $\sigma$ , the maximum directional standard deviation of any distribution in the mixture, and  $T$ , the number of clusters. In [Das99], Dasgupta provided an algorithm which learns mixtures of spherical Gaussians when the centers of each pair of distributions is separated by  $\Omega(\sigma\sqrt{n})$ . In [DS00], Dasgupta and Schulman provided an algorithm which applied to more situations and required a separation of  $\Omega(\sigma n^{1/4})$ . [AK01] showed how to learn mixtures of arbitrary Gaussians with a separation of  $\Omega(\sigma n^{1/4})$  using distance concentration. In addition to the usual separation between the centers, their results apply to other situations, for example, to concentric Gaussians with sufficiently different variance.

The first algorithm that removed the dependence on  $n$  was due to Vempala and Wang [VW02], who gave a singular value decomposition based algorithm for learning mixtures of spherical Gaussians with a separation of  $\Omega(T^{1/4}\sigma)$ . Their algorithm applies a singular value decomposition of the matrix of samples to compute a  $T$ -dimensional subspace which approximates the subspace containing the centers, and then uses distance concentration to cluster the samples projected on this low-dimensional space. In further work, [KSV05] and [AM05] showed how to use singular value decomposition based algorithms to learn mixtures of general Gaussians when the separation between the centers of distributions  $D_i$  and  $D_j$  is

$\Omega(\sigma(w_i^{-1/2} + w_j^{-1/2}) + \sigma\sqrt{T \log(\frac{T}{\delta})})$ . The algorithm of [AM05] was shown to apply to  $f$ -convergent and  $g$ -concentrated distributions, with bounds that vary with the nature of the distributions. Their algorithm also applies to product distributions on binary vectors. However, their algorithm does not apply to distributions with infinite variance. Even for distributions with finite variance, unless the distribution has rapidly decaying tails, their algorithm yields poor guarantees, proportional to the maximum range of the distribution of each coordinate.

More recently, [CR08] show an algorithm which, under certain conditions, learns mixtures of binary product distributions and axis-aligned Gaussians when the centers are separated by

$\Omega(\sigma_*(\sqrt{T \log \Lambda} + \sqrt{T \log(\frac{T}{\delta})}))$  where  $\sigma_*$  is the max-

imum directional variance in the space containing the centers, and  $\Lambda$  is polynomial in  $n, T$  and  $\frac{1}{w_{\min}}$ . Their algorithm also does not work for distributions with infinite variance and yields poor guarantees for mixtures of heavy-tailed product distributions.

### 3 A Summary of our Results

We begin with some definitions about distributions over high-dimensional spaces.

**Mixture of Distributions.** A mixture of distributions is a collection of distributions  $\mathcal{D} = \{D_1, \dots, D_T\}$  and mixing weights  $w_1, \dots, w_T$  such that  $\sum_{i=1}^T w_i = 1$ . A sample from a mixture is drawn by selecting  $D_i$  with probability  $w_i$  and then choosing a sample from  $D_i$ .

**Median.** We say that a distribution  $D$  on  $\mathbf{R}$  has median  $m(D)$  if the probability that a sample drawn from  $D$  is less than or equal to  $m(D)$  is  $1/2$ . We say that a distribution  $D$  on  $\mathbf{R}^n$  has median  $m(D) = (m_1, \dots, m_n)$  if the projection of  $D$  on the  $f$ -th coordinate axis has median  $m_f$ , for  $1 \leq f \leq n$ . For a distribution  $D$ , we write  $m(D)$  to denote the median of  $D$ .

**Center.** We say that a distribution  $D$  on  $\mathbf{R}^n$  has center  $(c_1, \dots, c_n)$  if the projection of  $D$  on the  $f$ -th coordinate axis has expectation  $c_f$ , for  $1 \leq f \leq n$ .

**$\beta$ -Radius.** For  $0 < \beta \leq 1$ , the  $\beta$ -Radius of a distribution  $D$  on  $\mathbf{R}$  with median  $m(D)$  is the smallest  $R_\beta$  such that

$$\Pr_{x \sim D} [m(D) - R_\beta \leq x \leq m(D) + R_\beta] \geq \beta$$

**Effective Distance.** To better describe our results, we need to define the concept of *effective distance*. The effective distance between two points  $x$  and  $y$  in  $\mathbf{R}^n$  at scale  $R$ , denoted by  $d_R(x, y)$  is defined as:

$$d_R(x, y) = \sqrt{\sum_{f=1}^n \min(R^2, (x^f - y^f)^2)}$$

The effective distance between two points  $x$  and  $y$  at scale  $R$  is thus high if many coordinates contribute to the distance between the points.

**Notation.** We use subscripts  $i, j$  to index over distributions in the mixture and subscripts  $f, g$  to index over coordinates in  $\mathbf{R}^n$ . Moreover, we use subscripts  $(f, k), \dots$  to index over coordinates in the transformed space. We use  $R$  to denote the maximum  $\frac{3}{4}$ -radius of any coordinate of any distribution in the mixture. For each distribution  $D_i$  in the mixture, and each coordinate  $f$ , we use  $D_i^f$  to denote the projection of  $D_i$  on the  $f$ -th coordinate axis. For any  $i$ , we use  $\tilde{D}_i$  to denote the distribution induced by applying our embedding on  $D_i$ . Similarly, for any  $i$  and any  $f$ , we use  $\tilde{D}_i^f$  to denote the distribution induced by applying our embedding on  $D_i^f$ . Moreover, we use  $\tilde{\mu}_i$  to denote the center of  $\tilde{D}_i$  and  $\tilde{\mu}_i^f$  to denote the center of  $\tilde{D}_i^f$ .

We use  $\|x\|$  to denote the  $L_2$  norm of a vector  $x$ . We use  $n$  to denote the number of dimensions and  $s$  to denote the number of samples. For a point  $x$ , and subspace  $\mathcal{H}$ , we use  $\mathbf{P}_{\mathcal{H}}(x)$  to denote the projection of  $x$  on  $\mathcal{H}$ .

#### 3.1 Our Results

The main contribution of this paper is an embedding from  $\mathbf{R}^n$  to  $\{0, 1\}^{n'}$ , where  $n' > n$ . The embedding has the property that samples from two product distributions on  $\mathbf{R}^n$  which have medians that are far apart map to samples from distributions on  $\{0, 1\}^{n'}$  with centers which are also far apart. In particular, let  $\mathcal{D} = \{D_1, \dots, D_T\}$  be a mixture of product distributions such that each coordinate  $f$  of each distribution  $D_i$  in the mixture satisfies the following properties:

1. *Symmetry* about the median.
2.  $\frac{3}{4}$ -radius upper bounded by  $R$ .

In particular, this allows the distribution of each coordinate to have infinite variance. Then the properties of our embedding can be summarized by the following theorems.

**Theorem 1** *Suppose we are given access to samples from a mixture of product distributions  $\mathcal{D} = \{D_1, \dots, D_T\}$  over  $\mathbf{R}^n$  such that for every  $i$  and  $f$ ,  $D_i^f$  satisfies properties (1) and (2). Moreover, let for any  $i$ ,  $\tilde{\mu}_i$  denote the center of the distribution  $\tilde{D}_i$  obtained by applying our embedding  $\Phi$  on  $D_i$ . If, for some constant  $c_1$ ,*

$$d_R(m(D_i), m(D_j)) \geq c_1 R$$

, then, there exists a constant  $c_2$ , such that

$$\|\tilde{\mu}_i - \tilde{\mu}_j\| \geq c_2 n^{1/4} T^{1/2} (\log n \log T)^{1/2} \times \frac{d_R(m(D_i), m(D_j))}{R}$$

with probability  $1 - \frac{1}{n}$  over the randomness in computing  $\Phi$ . Moreover, for any  $i$ , any  $k, k'$  and any  $f \neq f'$ , coordinates  $(f, k)$  and  $(f', k')$  of  $\tilde{D}_i$  are independently distributed.

Our embedding can be combined with the SVD-based clustering algorithms of [KSV05, AM05] to provide an efficient algorithm for learning mixtures of heavy-tailed distributions. The resulting clustering algorithm has the following guarantees.

**Theorem 2** *Suppose we are given access to samples from a mixture of product distributions  $\mathcal{D} = \{D_1, \dots, D_T\}$  over  $\mathbf{R}^n$  such that for every  $i$  and  $f$ ,  $D_i^f$  satisfies properties (1) and (2). If, for some constant  $c_3$ ,*

$$d_R(m(D_i), m(D_j)) \geq c_3 R (w_i^{-1/2} + w_j^{-1/2} + \sqrt{T \log \frac{nT}{\delta}})$$

Then, Algorithm HT-SVD clusters the samples correctly with probability  $1 - \delta$  over the samples, and with probability  $1 - \frac{1}{n}$  over the randomness in the algorithm. The algorithm runs in time polynomial in  $n$  and  $T$ , and the number of samples required by the algorithm is  $\tilde{O}\left(\frac{n^{3/2}T}{w_{\min}}\right)$ .

Alternatively, we can also combine our algorithm with the more recent correlation-based clustering algorithm of [CR08]. The result is an efficient algorithm with the following guarantees.

**Theorem 3** *Suppose we are given  $s$  samples from a mixture of product distributions  $\mathcal{D} = \{D_1, \dots, D_T\}$  over  $\mathbf{R}^n$  such that for every  $i$  and  $f$ ,  $D_i^f$  satisfies properties (1) and (2). If, for some constant  $c_3$ ,*

$$d_R(m(D_i), m(D_j)) \geq c_3 R (\sqrt{T \log \Lambda} + \sqrt{T \log \frac{nT}{\delta}})$$

where  $\Lambda = \Theta\left(\frac{T\sqrt{n} \log^2 n}{w_{\min}}\right)$ . Then,

Algorithm HT-CORRELATIONS clusters the samples correctly with probability  $1 - \delta$  over the samples, and with at least constant probability over the randomness in the algorithm. The algorithm runs in time polynomial in  $n$  and  $T$ , and the number of samples required by the algorithm is polynomial in  $n$ ,  $T$ , and  $\frac{1}{w_{\min}}$ .

The condition imposed on the centers of the distributions states that every pair of centers is sufficiently far apart in space, and the distance between every pair of centers is spread across  $\Omega\left(T \log \Lambda + T \log \frac{nT}{\delta}\right)$  coordinates.

### 3.2 Discussions

**Symmetry.** Our embedding still seems to work when the distributions do not have perfect symmetry, but satisfy an approximate symmetry condition. However, we illustrate by an example that we need at least a weak version of the symmetry condition for our embedding to work. Let  $D_1$  and  $D_2$  be the following distributions over  $\mathbf{R}$ , where  $M$  is a very large number. For  $D_1$  the probability density function is:

$$\begin{aligned} f_1(x) &= \frac{3}{8R}, \quad -R \leq x \leq R \\ &= \frac{1}{8MR}, \quad MR \leq x \leq 2MR \\ &= \frac{1}{8MR}, \quad -2MR \leq x \leq -MR \end{aligned}$$

The density function for  $D_2$  is:

$$\begin{aligned} f_2(x) &= \frac{3}{8R}, \quad -R \leq x \leq R \\ &= \frac{1}{4MR}, \quad -2MR \leq x \leq -MR \end{aligned}$$

We note that although the medians of  $D_1$  and  $D_2$  are  $R/3$  distance apart, the overlap in their probability mass

in any interval of size  $2R$  is very high. Therefore, since our embedding relies on the fact that two distributions which have medians that are far apart, and  $\frac{3}{4}$ -radius bounded by  $R$ , have low overlap in probability mass in a region of size  $\Omega(R)$  around the median, it does not work for distributions like  $D_1$  and  $D_2$ .

**Spreading Condition.** We note that our spreading condition, while similar to the *slope* requirement of [DHKS05], is weaker; while they require the total contribution to the distance between any two medians from all the coordinates to be large with respect to the contribution from the maximum coordinate, we only require that the contribution come from a few coordinates, regardless of what the maximum contribution from a coordinate is.

## 4 Embedding Distributions onto the Hamming Cube

In this section, we describe an embedding which maps points in  $\mathbf{R}^n$  to points on a Hamming Cube of higher dimension. The embedding has the following property. If for any  $i$  and  $j$ ,  $D_i$  and  $D_j$  are product distributions on  $\mathbf{R}^n$  with properties (1) and (2) such that their medians are far apart, then, the distributions induced on the Hamming cube by applying the embedding on points from  $D_i$  and  $D_j$  respectively also have centers which are far apart.

The building blocks of our embedding are embeddings  $\{\Phi_f\}$ , one for each coordinate,  $f$  in  $\{1, \dots, n\}$ . The final embedding  $\Phi$  is a concatenation of the maps  $\Phi_f$  for  $1 \leq f \leq n$ . We describe more precisely how to put together the maps  $\Phi_f$  in Section 4.3; for now, we focus on the individual embeddings  $\Phi_f$ .

Each embedding  $\Phi_f$ , in its turn, is a concatenation of two embeddings. The first one ensures that, for any  $i$  and  $j$ , if  $D_i^f$  and  $D_j^f$  are two distributions with properties (1) and (2) such that  $|m(D_i^f) - m(D_j^f)|$  is smaller than (or in the same range as)  $R$ , then, the expected distance between the centers of the distributions induced by applying the embedding on points from  $D_i^f$  and  $D_j^f$  is  $\Omega\left(\frac{|m(D_i^f) - m(D_j^f)|}{R}\right)$ . Unfortunately, this embedding does not provide good guarantees when  $|m(D_i^f) - m(D_j^f)|$  is large with respect to  $R$ . To address this, we use our second embedding, which guarantees that when  $|m(D_i^f) - m(D_j^f)|$  is large with respect to  $R$ , the centers of the two distributions induced by applying the embedding on points from  $D_i^f$  and  $D_j^f$  are at least constant distance apart. By concatenating these two embeddings, we ensure that in either case, the centers of the induced distributions obtained by applying  $\Phi_f$  on  $D_i^f$  and  $D_j^f$  are far apart.

### 4.1 Embedding Distributions with Small Separation

In this section, we describe an embedding with the following property. If, for any  $i, j$ , and  $f$ ,  $D_i^f$  and  $D_j^f$  have

properties (1) and (2) and  $|m(D_i^f) - m(D_j^f)| < 8R$ , then the distance between the centers of the distributions induced by applying  $\psi$  to points generated from  $D_i^f$  and  $D_j^f$ , is proportional to  $\frac{|m(D_i^f) - m(D_j^f)|}{8R}$ .

The embedding is as follows. Given a parameter  $R_1$ , and  $r \in [0, R_1)$ , we define, for a point  $x \in \mathbf{R}$ ,

$$\begin{aligned} \psi_r(x) &= 0, \text{ if } \lfloor \frac{x-r}{R_1} \rfloor \text{ is even} \\ &= 1, \text{ otherwise} \end{aligned}$$

In other words, we divide the real line into intervals of length  $R_1$  and assign label 0 to the even intervals and label 1 to the odd intervals. The value of  $\psi_r(x)$  is then the label of the interval containing  $x - r$ .

The properties of this embedding can be summarized as follows.

**Theorem 4** For any  $i, j$ , and  $f$ , if  $D_i^f$  and  $D_j^f$  have properties (1) and (2), and if  $r$  is drawn uniformly at random from  $[0, R_1)$  and  $R_1 > 2R + 3|m(D_i^f) - m(D_j^f)|$ , then,

$$\begin{aligned} \mathbf{E}[|\Pr_{x \sim D_i^f}[\psi_r(x) = 0] - \Pr_{x \sim D_j^f}[\psi_r(x) = 0]|] \\ \geq \frac{|m(D_i^f) - m(D_j^f)|}{2R_1} \end{aligned}$$

Here the expectation is taken over the distribution of  $r$ .

**Notation** For  $i = 1, \dots, T$ , we write  $\varphi_i^f$  as the probability density function of distribution  $D_i^f$  centered at 0, and  $F_i^f$  as the cumulative density function of distribution  $D_i^f$  centered at 0. For a real number  $r \in [0, R_1)$ , and for  $i = 1, \dots, T$ , we define

$$\alpha_i^f(r) = \sum_{\lambda=-\infty}^{\infty} (F_i^f(r + (2\lambda + 1)R_1) - F_i^f(r + 2\lambda R_1))$$

More specifically,  $\alpha_i^f(r)$  is the sum of the probability mass of the distribution  $D_i$  in the even intervals when the shift is  $r$ , which is again the probability that a point drawn from  $D_i$  is mapped to 0 by the embedding  $\psi_r$ . In the sequel, we use  $\Delta$  to denote  $|m(D_i^f) - m(D_j^f)|$ . We also assume without loss of generality that  $m(D_j^f) \leq m(D_i^f)$ , and  $m(D_i^f) = 0$ . Then, the left-hand side of the equation in Theorem 4 can be written as follows.

$$\begin{aligned} \mathbf{E}[|\Pr_{x \sim D_i^f}[\psi_r(x) = 0] - \Pr_{x \sim D_j^f}[\psi_r(x) = 0]|] \\ = \frac{1}{R_1} \int_{r=-R_1/2}^{R_1/2} |\alpha_j^f(r + \Delta) - \alpha_i^f(r)| \mathbf{d}r \quad (1) \end{aligned}$$

The proof of Theorem 4 follows in two steps. First, we show that if  $D_i^f$  were a shifted version of  $D_j^f$ , a slightly stronger version of Theorem 4 would hold. This is shown in Lemma 5. Next, Lemma 8 shows that even if  $D_i^f$  is not a shifted version of  $D_j^f$ , the statements in Theorem 4 still hold.

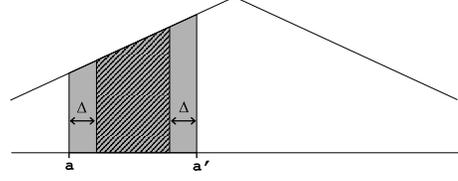


Figure 1: Proof of Lemma 6

**Lemma 5** For any  $\Delta$ , if  $R_1 > 3\Delta + 2R$ , then, for any  $i$ ,

$$\int_{r=-R_1/2}^{R_1/2} (\alpha_i^f(r) - \alpha_i^f(\Delta + r)) \mathbf{d}r \geq \frac{\Delta}{2}$$

Note that the difference between the statement of Theorem 4 and Lemma 5 is that the left-hand side of the equation in Theorem 4 has an absolute value, and hence Lemma 5 makes a stronger statement (under stronger assumptions).

Before we prove Lemma 5, we need the following lemma.

**Lemma 6** Let  $[a, a']$  be any interval of length more than  $2\Delta$ . Then, for any  $i$ ,

$$\begin{aligned} \Delta \cdot \int_a^{a'} \varphi_i^f(r) \mathbf{d}r &\geq \int_{r=a}^{a'} (F_i^f(r + \Delta) - F_i^f(r)) \mathbf{d}r \\ &\geq \Delta \cdot \int_{r=a+\Delta}^{a'-\Delta} \varphi_i^f(r) \mathbf{d}r \end{aligned}$$

**Proof:** For any  $r$ ,

$$F_i^f(r + \Delta) - F_i^f(r) = \int_{t=r}^{r+\Delta} \varphi_i^f(t) \mathbf{d}t$$

We divide the interval  $[a, a']$  into infinitesimal intervals of length  $\bar{\delta}$ . The probability mass of distribution  $D_i$  in an interval  $[t, t + \bar{\delta}]$  is  $\bar{\delta} \cdot \varphi_i^f(t)$ .

Note that in the expression

$$\int_{r=a}^{a'} (F_i^f(r + \Delta) - F_i^f(r)) \mathbf{d}r$$

the probability mass of each interval  $[t, t + \bar{\delta}]$  where  $t$  lies in  $[a + \Delta, a' - \Delta]$  is counted exactly  $\frac{\Delta}{\bar{\delta}}$  times, and the probability mass of  $D_i$  in an interval  $[t, t + \bar{\delta}]$ , where  $t$  lies in the interval  $[a, a + \Delta) \cup (a' - \Delta, a']$  is counted at most  $\frac{\Delta}{\bar{\delta}}$  times – see Figure 1. Since  $\varphi_i^f(t) \geq 0$  for all  $t$ , the lemma follows in the limit when  $\bar{\delta} \rightarrow 0$ .  $\square$

**Proof:**(Of Lemma 5) The shaded area in Figure 2 shows the value of  $\alpha_i^f(r) - \alpha_i^f(r + \Delta)$  for a distribution  $D_i$ .

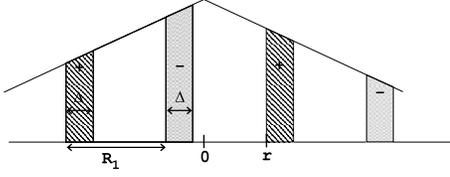


Figure 2: Proof of Lemma 5

We can write:

$$\begin{aligned}
& \int_{r=-R_1/2}^{R_1/2} (\alpha_i^f(r) - \alpha_i^f(r + \Delta)) \mathbf{d}r \\
= & \int_{r=-R_1/2}^{R_1/2} \sum_{\lambda=-\infty}^{\infty} [(F_i^f(r + (2\lambda + 1)R_1) \\
& - F_i^f(r + 2\lambda R_1) - (F_i^f(r + \Delta + (2\lambda + 1)R_1) \\
& - F_i^f(r + \Delta + 2\lambda R_1))] \mathbf{d}r \\
= & \int_{r=-R_1/2}^{R_1/2} \sum_{\lambda=-\infty}^{\infty} [(F_i^f(r + (2\lambda + 1)R_1) \\
& - F_i^f(r + \Delta + (2\lambda + 1)R_1) - (F_i^f(r + 2\lambda R_1) \\
& - F_i^f(r + \Delta + 2\lambda R_1))] \mathbf{d}r \\
= & \int_{r=-R_1/2}^{R_1/2} \sum_{\lambda=-\infty}^{\infty} [(F_i^f(r + 2\lambda R_1 + \Delta) \\
& - F_i^f(r + 2\lambda R_1)) - (F_i^f(r + (2\lambda + 1)R_1 + \Delta) \\
& - F_i^f(r + (2\lambda + 1)R_1))] \mathbf{d}r
\end{aligned}$$

From Lemma 6, the first term is at least

$$\Delta \cdot \sum_{\lambda=-\infty}^{\infty} \int_{r=-R_1/2+\Delta}^{R_1/2-\Delta} \varphi_i^f(r + 2\lambda R_1) \mathbf{d}r$$

This is  $\Delta$  times the total probability mass of  $D_i$  in the intervals  $[2\lambda R_1 - R_1/2 + \Delta, 2\lambda R_1 + R_1/2 - \Delta]$ , for all  $\lambda$ . Since  $R_1 > 2\Delta + 2R$ , this includes the interval  $[-R, R]$ , and as the median of  $D_i$  is at 0 and  $D_i$  has  $\frac{3}{4}$ -radius less than or equal to  $R$ , the value of the first term is at least  $\frac{3\Delta}{4}$ .

From Lemma 6, the second term is at most

$$\Delta \cdot \sum_{\lambda=-\infty}^{\infty} \int_{r=-R_1/2}^{R_1/2} \varphi_i^f(r + (2\lambda + 1)R_1) \mathbf{d}r$$

This is the total probability mass of  $D_i$  in the intervals  $[(2\lambda + 1)R_1 - R_1/2, (2\lambda + 1)R_1 + R_1/2]$ , for all  $\lambda$ . Since  $R_1 > 3\Delta + 2R$ , none of these intervals have any intersection with  $[-R, R]$ . The total probability mass in these intervals is therefore at most  $\frac{1}{4}$ , and therefore the value of the second term is at most  $\frac{\Delta}{4}$ . The lemma follows.  $\square$

Next we show that Theorem 4 holds even if distribution  $D_i^f$  is not a shifted version of distribution  $D_j^f$ . This

is shown by a combination of Lemmas 7 and 8, which are both consequences of the symmetry of the distributions  $D_i^f$  and  $D_j^f$ .

**Lemma 7** Suppose that for any  $i, j$ , and  $f$ ,  $D_i^f, D_j^f$  have property (1) and median 0. Then, for any  $r$ ,

$$\alpha_i^f(r) - \alpha_j^f(r) = \alpha_j^f(-r) - \alpha_i^f(-r)$$

**Proof:** We define

$$\bar{\alpha}_i^f(r) = \sum_{\lambda=-\infty}^{\infty} F_i^f(r + 2\lambda R_1) - F_i^f(r + (2\lambda - 1)R_1)$$

Thus,  $\bar{\alpha}_i^f(r)$  is the probability mass of  $D_i$  in the odd intervals, which is again the probability that  $\psi_r$  maps a random point from  $D_i$  to 1 when the shift chosen is  $r$ . Therefore,  $\bar{\alpha}_i^f(r) = 1 - \alpha_i^f(r)$ . Since  $D_i$  is symmetric with median 0, for any interval  $[a, a']$ ,  $a' > a > 0$ ,  $F_i^f(a') - F_i^f(a) = F_i^f(-a) - F_i^f(-a')$ . Therefore,

$$\begin{aligned}
& \alpha_i^f(-r) \\
= & \sum_{\lambda=-\infty}^{\infty} F_i^f(-r + (2\lambda + 1)R_1) - F_i^f(-r + 2\lambda R_1) \\
= & \sum_{\lambda=-\infty}^{\infty} F_i^f(r - 2\lambda R_1) - F_i^f(r - (2\lambda + 1)R_1) \\
= & \bar{\alpha}_i^f(r)
\end{aligned}$$

The lemma follows because

$$\bar{\alpha}_i^f(r) - \bar{\alpha}_j^f(r) = \alpha_j^f(r) - \alpha_i^f(r)$$

$\square$

**Lemma 8** For any  $i$  and  $j$ , if  $D_i^f$  and  $D_j^f$  have properties (1) and (2), then,

$$\begin{aligned}
& \int_{r=-R_1/2}^{R_1/2} |\alpha_j^f(r + \Delta) - \alpha_i^f(r)| \mathbf{d}r \\
\geq & \int_{r=-R_1/2}^{R_1/2} (\alpha_j^f(r) - \alpha_j^f(r + \Delta)) \mathbf{d}r
\end{aligned}$$

**Proof:** By Lemma 7, for every  $r \in [-R_1/2, R_1/2]$ , there is a unique  $r' = -r$  such that  $\alpha_i^f(r) - \alpha_j^f(r) = \alpha_j^f(r') - \alpha_i^f(r')$ . We claim that for every such pair  $r, r'$ ,

$$\begin{aligned}
& |\alpha_j^f(r + \Delta) - \alpha_i^f(r)| + |\alpha_j^f(r' + \Delta) - \alpha_i^f(r')| \\
\geq & (\alpha_j^f(r) - \alpha_j^f(r + \Delta)) + (\alpha_j^f(r') - \alpha_j^f(r' + \Delta))
\end{aligned}$$

We note that for a fixed pair  $(r, r')$ ,

$$\begin{aligned}
& |\alpha_j^f(r + \Delta) - \alpha_i^f(r)| + |\alpha_j^f(r' + \Delta) - \alpha_i^f(r')| \\
= & |\alpha_j^f(r + \Delta) - \alpha_i^f(r)| + |\alpha_j^f(r' + \Delta) + \alpha_j^f(r) \\
& - \alpha_j^f(r) - \alpha_i^f(r')| \\
\geq & |\alpha_j^f(r + \Delta) - \alpha_i^f(r) + \alpha_j^f(r' + \Delta) + \alpha_j^f(r) \\
& - \alpha_j^f(r) - \alpha_i^f(r')| \\
\geq & |(\alpha_j^f(r + \Delta) - \alpha_j^f(r)) + (\alpha_j^f(r' + \Delta) - \alpha_j^f(r')) \\
& + (\alpha_j^f(r) + \alpha_j^f(r') - \alpha_i^f(r) - \alpha_i^f(r'))|
\end{aligned}$$

The lemma follows by summing over all such pairs  $(r, r')$ .

□

**Proof:** (Of Theorem 4) From Equation 1 and Lemma 5,

$$\begin{aligned}
& \mathbf{E}[|\Pr_{x \sim D_i^f}[\psi_r(x) = 0] - \Pr_{x \sim D_j^f}[\psi_r(x) = 0]|] \\
& \frac{1}{R_1} \int_{-R_1/2}^{R_1/2} |\alpha_j^f(r + \Delta) - \alpha_i^f(r)| \mathbf{d}\mathbf{r} \geq \frac{\Delta}{2R_1}
\end{aligned}$$

The second step follows from Lemma 5. □

## 4.2 Embedding Distributions with Large Separation

In this section, we describe an embedding with the following property. For any  $i, j$ , and  $f$ , if  $D_i^f$  and  $D_j^f$  have properties (1) and (2), and  $|m(D_i^f) - m(D_j^f)| \geq 8R$ , then, the expected gap between the centers of the distributions induced by applying the embeddings on points from  $D_i^f$  and  $D_j^f$  is at least a constant.

The embedding is as follows. Given a random  $\zeta = \{\rho, \{\varepsilon_k\}_{k \in \mathbf{Z}}\}$  where  $\rho$  is a number in  $[0, R_2)$  and  $\{\varepsilon_k\}$  is an infinite sequence of bits, we define  $\phi_\zeta : \mathbf{R} \rightarrow \{0, 1\}$  as follows.

$$\phi_\zeta(x) = \varepsilon_{k(x)}, \text{ where } k(x) = \lfloor \frac{x - \rho}{R_2} \rfloor \quad (2)$$

In other words, if  $x - \rho$  lies in the interval  $[8kR, 8(k + 1)R)$ , then  $\phi_\zeta(x) = \varepsilon_k$ .

The properties of the embedding  $\phi_\zeta$  can be summarized as follows.

**Theorem 9** *For any  $i, j$ , and  $f$ , let  $D_i^f$  and  $D_j^f$  have properties (1) and (2), and let  $|m(D_i^f) - m(D_j^f)| \geq 8R$ . If  $R_2 \geq 8R$ , and if  $\rho$  is generated uniformly at random from the interval  $[0, R_2)$ , and each  $\varepsilon_k$  is generated by an independent toss of a fair coin, then,*

$$\mathbf{E}[|\Pr_{x \sim D_i^f}[\phi_\zeta(x) = 0] - \Pr_{x \sim D_j^f}[\phi_\zeta(x) = 0]|] \geq \frac{1}{8}$$

where the expectation is taken over the distribution of  $\zeta$ .

**Proof:** We say that an interval  $[a, a']$  of length  $8R$  or less is *cut* by the embedding if there exists some  $y \in [a, a']$

such that  $\frac{y-r}{8R}$  is an integer. If  $[a, a']$  is cut at  $y$ , then, with probability  $\frac{1}{2}$  over the choice of  $\{\varepsilon_k\}$ , any point  $x$  in the interval  $[a, y]$  has a different value of  $\phi_\zeta(x)$  than any point in  $(y, a']$ . If an interval is not cut, then all points in the interval have the same value of  $\phi_\zeta$  with probability 1 over the choice of  $\{\varepsilon_k\}$ .

Since the intervals  $[m(D_i^f) - R, m(D_i^f) + R]$  and  $[m(D_j^f) - R, m(D_j^f) + R]$  have length at least  $2R$ ,

$$\Pr[[m(D_i^f) - R, m(D_i^f) + R], [m(D_j^f) - R, m(D_j^f) + R] \text{ are not cut}] \geq 1 - \frac{2R + 2R}{8R} \geq \frac{1}{2}$$

If none of the intervals  $[m(D_i^f) - R, m(D_i^f) + R]$  and  $[m(D_j^f) - R, m(D_j^f) + R]$  are cut,

$$\Pr[\phi_\zeta(m(D_i^f) - R) \neq \phi_\zeta(m(D_j^f) - R)] = \frac{1}{2}$$

Let us assume that the intervals  $[m(D_i^f) - R, m(D_i^f) + R]$  and  $[m(D_j^f) - R, m(D_j^f) + R]$  are not cut and

$$\phi_\zeta(m(D_i^f) - R) \neq \phi_\zeta(m(D_j^f) - R)$$

. From the two equations above, the probability of this event is at least  $\frac{1}{4}$ . Also suppose without loss of generality that  $\phi_\zeta(m(D_i^f) - R) = 0$ . Then, since  $R$  is an upper bound on the  $\frac{3}{4}$ -radius of the distributions  $D_i^f$  and  $D_j^f$ , the probability mass of  $D_i^f$  that maps to 0 is at least  $\frac{3}{4}$ , and the probability mass of  $D_j^f$  that maps to 0 is at most  $\frac{1}{4}$ . Therefore, with probability at least  $\frac{1}{4}$ ,

$$|\Pr_{x \sim D_i^f}[\phi_\zeta(x) = 0] - \Pr_{x \sim D_j^f}[\phi_\zeta(x) = 0]| \geq \frac{1}{2}$$

The theorem follows. □

## 4.3 Combining the Embeddings

In this section, we show how to combine the embeddings of Sections 4.1 and 4.2 to provide a map  $\Phi$  which obeys the guarantees of Theorem 1. Given parameters  $R_1, R_2$ , and  $q$ , we define  $\Phi_f$  for a coordinate  $f$  as follows.

$$\Phi_f(x) = (\phi_{\zeta_1}(x^f), \dots, \phi_{\zeta_q}(x^f), \psi_{r_1}(x^f), \dots, \psi_{r_q}(x^f)) \quad (3)$$

Here,  $\zeta_1, \dots, \zeta_q$  are  $q$  independent random values of  $\zeta = (\rho, \{\varepsilon_k\}_{k \in \mathbf{Z}})$ , where  $\rho$  is drawn uniformly at random from the interval  $[0, R_2)$ , and  $\varepsilon_k$ , for all  $k$ , are generated by independent tosses of an unbiased coin.  $r_1, \dots, r_q$  are  $q$  independent random values of  $r$ , where  $r$  is drawn uniformly at random from the interval  $[0, R_1)$ . Finally, the embedding  $\Phi$  is defined as:

$$\Phi(x) = \Phi_1(x) \oplus \dots \oplus \Phi_n(x) \quad (4)$$

The properties of the embedding  $\Phi$  are summarized in Theorem 1. Next, we prove Theorem 1. We begin with the following lemma, which demonstrates the properties of each  $\Phi_f$ .

**Lemma 10** Let  $R_1 \geq 26R$ ,  $R_2 \geq 8R$ , and  $q = 4\sqrt{n}T \log n \log T$ , and suppose we are given samples from a mixture of product distributions which satisfy conditions (1) and (2). Then, for all  $i$  and  $j$ , the embedding  $\Phi = \bigoplus_f \Phi_f$  defined in Equation 3 satisfies the following conditions. With probability at least  $1 - \frac{1}{n}$  over the randomness in the embedding, for each coordinate  $f$ ,

1. If  $|m(D_i^f) - m(D_j^f)| > 8R$ , then, for some constant  $c_5$ ,

$$\|\mathbf{E}_{x \sim D_i^f}[\Phi_f(x)] - \mathbf{E}_{x \sim D_j^f}[\Phi_f(x)]\| \geq c_5 n^{1/4} T^{1/2} \times (\log n \log T)^{1/2}$$

2. If  $\frac{R}{\sqrt{n}} \leq |m(D_i^f) - m(D_j^f)| \leq 8R$ , then, for some constant  $c_6$ ,

$$\|\mathbf{E}_{x \sim D_i^f}[\Phi_f(x)] - \mathbf{E}_{x \sim D_j^f}[\Phi_f(x)]\| \geq c_6 n^{1/4} T^{1/2} \times (\log n \log T)^{1/2} \frac{|m(D_i^f) - m(D_j^f)|}{R}$$

**Proof:**(Of Lemma 10) The first part of the lemma follows by Theorem 9, along with an application of the Chernoff Bounds, followed by a Union Bound over all  $i, j, f$ . The second part follows similarly by an application of Theorem 4.  $\square$

**Proof:**(Of Theorem 1) We call a coordinate  $f$  *very low* for distributions  $i$  and  $j$  if  $|m(D_i^f) - m(D_j^f)| \leq \frac{R}{\sqrt{n}}$ , *low* if  $\frac{R}{\sqrt{n}} \leq |m(D_i^f) - m(D_j^f)| < 8R$ , and *high* otherwise. Let  $V_{i,j}$ ,  $L_{i,j}$  and  $H_{i,j}$  respectively denote the set of very low, low and high coordinates for distributions  $D_i$  and  $D_j$ . Then,

$$\|\tilde{\mu}_i - \tilde{\mu}_j\|^2 = \sum_{f \in V_{i,j}} \|\tilde{\mu}_i^f - \tilde{\mu}_j^f\|^2 + \sum_{f \in L_{i,j}} \|\tilde{\mu}_i^f - \tilde{\mu}_j^f\|^2 + \sum_{f \in H_{i,j}} \|\tilde{\mu}_i^f - \tilde{\mu}_j^f\|^2$$

From Lemma 10, this sum is at least

$$\sum_{f \in L_{i,j}} c_6 n^{1/2} T \log n \log T \frac{|m(D_i^f) - m(D_j^f)|^2}{R^2} + \sum_{f \in H_{i,j}} c_5 n^{1/2} T \log n \log T$$

which, by the definition of effective distance is at least

$$c_7 n^{1/2} T \log n \log T \left( \frac{d_R^2(m(D_i), m(D_j))}{R^2} - \frac{\sum_{f \in V_{i,j}} (m(D_i^f) - m(D_j^f))^2}{R^2} \right)$$

where  $c_7$  is some constant. Now the contribution from the very low coordinates to the distance between  $m(D_i)$  and  $m(D_j)$  is at most  $\sqrt{\sum_f R^2/n} = R$ . Since

$$d_R(m(D_i), m(D_j)) \geq 2R$$

, this contribution is at most  $\frac{1}{2}$  the total distance. The first part of the theorem therefore follows.

For any sample  $x$  from any  $D_i$  in the mixture, and any  $k, k'$ , coordinates  $(f, k)$  and  $(f', k')$  of  $\Phi(x)$  are function of  $x^f$  and  $x^{f'}$  respectively. As for  $f \neq f'$ ,  $x^f$  and  $x^{f'}$  are independently distributed, the second part of the theorem follows.  $\square$

## 5 Applications: Learning Mixtures

In this section, we show how our embedding in Theorem 1 can be combined with standard algorithm for learning mixture models to yield algorithms that can learn mixtures of heavy-tailed distributions. First, in Section 5.1, we show how to combine our embedding with SVD-based algorithms of [KSV05, AM05]; in Section 5.2, we show how to combine our embedding with the more recent algorithm of [CR08].

### 5.1 Clustering using SVD

In this section, we present Algorithm HT-SVD– a combination of SVD-based algorithms of [AM05, KSV05] with our embedding in Theorem 1. The input to the algorithm is a set  $S$  of samples, and the output is a partitioning of the samples. The algorithm is described in Figure 3.

The properties of Algorithm HT-SVD are summarized by Theorem 2, which we prove for the rest of this section. The two main steps in the proof are as follows: first, we show that after applying our embedding, the transformed distributions have good properties, such as low directional variance and distance-concentration. Next, we show that these properties imply that SVD-based algorithms, such as those of [KSV05, AM05] can learn these mixtures effectively. The following lemma shows that the maximum directional variance of the transformed distributions in the mixture is high; this fact is later used crucially in demonstrating that SVD-based algorithms can effectively cluster the mixture.

**Lemma 11** For any  $i$ , the maximum directional variance of the transformed distribution  $\tilde{D}_i$  is at most  $O(n^{1/2} T \log n \log T)$ .

**Proof:** Let  $v$  be any unit vector in the transformed space. The variance of the transformed distribution  $\tilde{D}_i$  along  $v$

### HT-SVD( $S$ )

1. Let  $R_1 = 26R$ ,  $R_2 = 8R$ , and  $q = 4\sqrt{n}T \log n \log T$ . Compute  $\tilde{S} = \{\Phi(x) | x \in S\}$ . Partition  $\tilde{S}$  into  $\tilde{S}_A$  and  $\tilde{S}_B$  uniformly at random.
2. Construct the  $\frac{s}{2} \times nq$  matrix  $\bar{S}_A$  (respectively  $\bar{S}_B$ ) in which the entry at row  $l$  and column  $l'$  is the  $l'$ -th coordinate of the  $l$ -th sample point in  $\tilde{S}_A$  ( $\tilde{S}_B$  respectively).
3. Let  $\{v_{1,A}, \dots, v_{T,A}\}$  (resp.  $\{v_{1,B}, \dots, v_{T,B}\}$ ) be the top  $T$  singular values of  $\bar{S}_A$  (resp.  $\bar{S}_B$ ). Project each point in  $\tilde{S}_B$  (resp.  $\tilde{S}_A$ ) on the subspace  $\mathcal{K}_A$  (resp.  $\mathcal{K}_B$ ) spanned by  $v_{1,A}, \dots, v_{T,A}$  (resp.  $v_{1,B}, \dots, v_{T,B}$ ).
4. Use a distance-based clustering algorithm as in [AK01] to partition the points in  $\tilde{S}_A$  and  $\tilde{S}_B$  after projection.

Figure 3: Algorithm Using SVDs

can be written as:

$$\begin{aligned}
& \mathbf{E}_{\tilde{x} \sim \tilde{D}_i} [\langle v, \tilde{x} - \mathbf{E}[\tilde{x}] \rangle^2] \\
= & \mathbf{E}_{\tilde{x} \sim \tilde{D}_i} \left[ \sum_{(f,k)} (v^{f,k})^2 \cdot (\tilde{x}^{f,k} - \mathbf{E}[\tilde{x}^{f,k}])^2 \right. \\
& + 2 \sum_{(f,k),(f',k')} v^{f,k} \cdot v^{f',k'} \cdot (\tilde{x}^{f,k} - \mathbf{E}[\tilde{x}^{f,k}]) \\
& \left. \times (\tilde{x}^{f',k'} - \mathbf{E}[\tilde{x}^{f',k'}]) \right] \\
\leq & \mathbf{E}_{\tilde{x} \sim \tilde{D}_i} \left[ \sum_{(f,k)} (v^{f,k})^2 + 2 \sum_{(f,k),(f',k')} v^{f,k} \cdot v^{f',k'} \right. \\
& \left. \times (\tilde{x}^{f,k} - \mathbf{E}[\tilde{x}^{f,k}]) \cdot (\tilde{x}^{f',k'} - \mathbf{E}[\tilde{x}^{f',k'}]) \right] \\
\leq & \mathbf{E}_{\tilde{x} \sim \tilde{D}_i} \left[ \sum_{(f,k)} (v^{f,k})^2 + 2 \sum_f \sum_{k,k'} v^{f,k} v^{f,k'} \right. \\
& \left. \times (\tilde{x}^{f,k} - \mathbf{E}[\tilde{x}^{f,k}]) \cdot (\tilde{x}^{f,k'} - \mathbf{E}[\tilde{x}^{f,k'}]) \right] \\
\leq & \mathbf{E}_{\tilde{x} \sim \tilde{D}_i} \left[ \sum_f \left( \sum_k v^{f,k} \right)^2 \right]
\end{aligned}$$

As  $\tilde{x}^{f,k}$  is distributed independently of  $\tilde{x}^{f',k'}$  when  $f \neq f'$ , in this case,

$$\mathbf{E}_{\tilde{x} \sim \tilde{D}_i} [(\tilde{x}^{f,k} - \mathbf{E}[\tilde{x}^{f,k}]) \cdot (\tilde{x}^{f',k'} - \mathbf{E}[\tilde{x}^{f',k'}])] = 0$$

The lemma follows as  $|\tilde{x}^{f,k} - \mathbf{E}[\tilde{x}^{f,k}]| \leq 1$  for any  $f$  and  $k$ , and there are at most  $O(n^{1/2}T \log n \log T)$  coordinates corresponding to a single  $f$ .  $\square$

Next we show that the transformed distributions also possess some distance-concentration properties.

**Lemma 12** *Let  $\mathcal{H}$  be a  $d$ -dimensional subspace of  $\{0, 1\}^{4n^{3/2}T \log T \log n}$ . Then for any  $i$ ,*

$$\begin{aligned}
\Pr_{\tilde{x} \sim \tilde{D}_i} [\|\mathbf{P}_{\mathcal{H}}(\tilde{x} - \mathbf{E}[\tilde{x}])\| < 4n^{1/4}T^{1/2}(\log n \log T)^{1/2} \\
\times \sqrt{d \log(d/\delta)}] \geq 1 - \delta
\end{aligned}$$

**Proof:** Let  $q = 4n^{1/2}T \log n \log T$ . Let  $v_1, \dots, v_d$  be an orthonormal basis of  $\mathcal{H}$ . As

$$\|\mathbf{P}_{\mathcal{H}}(\tilde{x})\|^2 = \sum_{l=1}^d (\langle v_l, \tilde{x} \rangle)^2$$

we apply the Method of Bounded Differences to bound the value of each  $\langle v_l, \tilde{x} \rangle$ .

$$\langle v_l, \tilde{x} \rangle = \sum_f \sum_k v_l^{f,k} \cdot \tilde{x}^{f,k}$$

As changing each coordinate of the original sample point  $x$  will change at most  $q$  coordinates of  $\tilde{x}$ ,  $\gamma_f$ , the change in  $\langle v_l, \tilde{x} \rangle$  when we change a coordinate  $f$  of the original sample point is at most  $(\sum_k v_l^{f,k})^2$ . Therefore,  $\gamma = \sum_f \gamma_f^2 = \sum_f (\sum_k v_l^{f,k})^2$ . Since  $v_l$  is a unit vector,  $\gamma \leq q$ . Thus, for any  $l$ ,

$$\Pr[|\langle v_l, \tilde{x} \rangle - \langle v_l, \mathbf{E}[\tilde{x}] \rangle| > \sqrt{q \log(d/\delta)}] \leq \frac{\delta}{d}$$

As  $\|\mathbf{P}_{\mathcal{H}}(\tilde{x} - \mathbf{E}[\tilde{x}])\|^2 = \sum_l \langle v_l, \tilde{x} - \mathbf{E}[\tilde{x}] \rangle^2$ , the lemma follows by applying a Union Bound over each vector  $v_l$ .  $\square$

We are now ready to prove Theorem 2. The main tool in our proof is the following lemma, due to [AM05], which shows that if the separation between the transformed centers is large, then, Step 3 of the algorithm will find a subspace in which the transformed centers are far apart.

**Lemma 13** *Let, for each  $i$ ,  $c_{i,A}$  be the empirical centers of  $\tilde{D}_i$  computed from the points in  $\tilde{S}_A$ , and let  $\sigma$  be the maximum directional standard deviation of any  $\tilde{D}_i$ . Then,*

$$\begin{aligned}
\|\mathbf{P}_{\mathcal{K}_B}(c_{i,A} - c_{j,A})\| & \geq \|c_{i,A} - c_{j,A}\| \\
& - \sigma(\omega_i^{-1/2} + \omega_j^{-1/2})
\end{aligned}$$

**Proof:** (Of Theorem 2) Let  $q = 4n^{1/2}T \log n \log T$ . When the distributions in the input mixture obey the separation conditions of Theorem 2, from Theorem 1, for each  $i$  and  $j$ , the distance between the transformed centers  $\tilde{\mu}_i$  and  $\tilde{\mu}_j$  is at least :

$$\Omega(\sqrt{q}) \cdot (w_i^{-1/2} + w_j^{-1/2} + \sqrt{T \log(Tn/\delta)})$$

Since the number of samples is at least  $\Omega(\frac{n^{3/2}}{w_{\min}})$ , the distance between the sample means and actual means of the transformed distributions are at most  $O(1)$ . Therefore, from Theorem 13,

$$\|\mathbf{P}_{\mathcal{K}_B}(c_{i,A} - c_{j,A})\| \geq c_8 \sqrt{qT \log(Tn/\delta)}$$

where  $c_{i,A}$  and  $c_{j,A}$  are the empirical centers of the transformed distributions, and  $c_8$  is some constant. As  $\mathcal{K}_B$  has dimension at most  $T$ , from Lemma 12 and a union bound over all pairs of samples, with probability  $1 - \delta$ , all pairs of samples drawn from a distribution  $D_i$  have distance at most

$$2n^{1/4}T^{1/2}(\log n \log T)^{1/2} \sqrt{2T \log(nT/\delta)}$$

in the subspace  $\mathcal{K}_B$ . On the other hand, for some constant  $a'$ , a sample drawn from  $D_i$  and a sample drawn from  $D_j$  are at least

$$a'n^{1/4}T^{1/2}(\log n \log T)^{1/2} \sqrt{T \log(nT/\delta)}$$

apart in  $\mathcal{K}_B$ . Algorithm HT-SVD therefore works for  $a' > 2\sqrt{2}$ .  $\square$

## 5.2 Clustering Using Correlations

In this section, we present Algorithm HT-CORRELATIONS which is a combination of our embedding with the correlations-based clustering algorithm of [CR08]. Algorithm HT-CORRELATIONS is described in Figure 4. The input to the algorithm is a set  $S$  of  $s$  samples, and the output is a partitioning of the samples.

The properties of Algorithm HT-CORRELATIONS are described in Theorem 3. This section is devoted to proving Theorem 3. The proof proceeds in three steps. First, we deduce from Theorem 1 that if the distributions satisfy the conditions in Theorem 3, then the transformed distributions satisfy the separation and spreading requirements of Theorem 1 in [CR08]. We can then apply Theorem 1 to show that the centers of the transformed distributions are far apart in  $\mathcal{K}_A$  and  $\mathcal{K}_B$ , the subspaces computed in Step 4 of

Algorithm HT-CORRELATIONS. Finally, we use this fact along with Lemmas 11 and 12 to show that distance concentration algorithms work in these output subspaces.

**Proof:**(Of Theorem 3) Let  $q = 4n^{1/2}T \log n \log T$ . From Theorem 1 and Conditions (1) and (2), for each  $i$  and  $j$ , the distance between the transformed centers  $\tilde{\mu}_i$  and  $\tilde{\mu}_j$  is at least

$$\Omega(\sqrt{q})(\sqrt{T \log \Lambda} + \sqrt{T \log(nT/\delta)})$$

We note that the proof of Theorem 1 in [CR08] requires only that for each distribution, the coordinates in  $\mathcal{F}$  are independently distributed from the coordinates in  $\mathcal{G}$ . Since the distribution of any coordinate in  $\mathcal{F}$  is independent of the distribution in  $\mathcal{G}$  (although the coordinates within  $\mathcal{F}$  or  $\mathcal{G}$  are not necessarily independently distributed), we can apply Theorem 1 in [CR08] to conclude that for each  $i$  and  $j$ , there exists some constant  $a$  such that:

$$\begin{aligned} d_{\mathcal{K}_B}(\tilde{\mu}_i, \tilde{\mu}_j) &\geq \Omega(d(\tilde{\mu}_i, \tilde{\mu}_j)) \\ &\geq a(\sqrt{qT \log \Lambda} + \sqrt{qT \log(nT/\delta)}) \end{aligned}$$

As  $\mathcal{K}_B$  has dimension at most  $2T$ , from Lemma 12 and a union bound, with probability  $1 - \delta$ , all pairs of samples drawn from a distribution  $D_i$  have distance at most  $2\sqrt{qT \log(nT/\delta)}$  in the subspace  $\mathcal{K}_B$ . On the other hand, a sample drawn from  $D_i$  and a sample drawn from  $D_j$  are at least  $(a_1 - 2)\sqrt{2qT \log(nT/\delta)}$  apart in  $\mathcal{K}_B$ . Algorithm HT-CORRELATIONS therefore works.  $\square$

## References

- [AK01] S. Arora and R. Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of 33rd ACM Symposium on Theory of Computing*, pages 247–257, 2001.
- [AM05] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 458–469, 2005.
- [CR08] K. Chaudhuri and S. Rao. Learning mixtures of distributions using correlations and independence. In *21st Annual Conference on Learning Theory*, 2008.
- [Das99] S. Dasgupta. Learning mixtures of gaussians. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pages 634–644, 1999.
- [DHKS05] A. Dasgupta, J. Hopcroft, J. Kleinberg, and M. Sandler. On learning mixtures of heavy-tailed distributions. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science*, pages 491–500, 2005.
- [DS00] S. Dasgupta and L. Schulman. A two-round variant of em for gaussian mixtures. In *Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.
- [FOS05] J. Feldman, R. O’Donnell, and R. Servedio. Learning mixtures of product distributions over discrete domains. In *Proceedings of FOCS*, 2005.
- [Ind01] Piotr Indyk. Algorithmic applications of low-distortion geometric embeddings. In *FOCS*, pages 10–33, 2001.

HT-CORRELATIONS( $S$ )

1. Partition the set of coordinates into  $\mathcal{F}$  and  $\mathcal{G}$  uniformly at random.
2. Partition  $S$  uniformly at random into  $S_A$  and  $S_B$ . Let  $R_1 = 26R$ ,  $R_2 = 8R$ , and  $q = 4\sqrt{n}T \log n \log T$ . Compute  $\tilde{S}_A = \{\Phi(x)|x \in S_A\}$  and  $\tilde{S}_B = \{\Phi(x)|x \in S_B\}$ .
3. Construct the  $\frac{nq}{2} \times \frac{nq}{2}$  covariance matrix  $M_A$  (respectively  $M_B$ ), which has a row for each tuple  $(f, k)$ ,  $f \in \mathcal{F}$ ,  $k \in [q]$ , and a column for each tuple  $(g, k)$ ,  $g \in \mathcal{G}$ ,  $k \in [q]$ . The entry at row  $(f, k)$  and column  $(g, k')$  is the covariance between coordinate  $(f, k)$  and  $(g, k')$  of the transformed points over all samples in  $S_A$  ( $S_B$  respectively).
4. Let  $\{v_{1,A}, \dots, v_{T,A}\}$  and  $\{y_{1,A}, \dots, y_{T,A}\}$  ( $\{v_{1,B}, \dots, v_{T,B}\}$  and  $\{y_{1,B}, \dots, y_{T,B}\}$  respectively) be the top  $T$  left and right singular vectors of  $M_A$  (resp.  $M_B$ ). Project each point in  $\tilde{S}_B$  (resp.  $\tilde{S}_A$ ) on the subspace  $\mathcal{K}_A$  (resp.  $\mathcal{K}_B$ ) spanned by  $\{v_{1,A}, \dots, v_{T,A}\} \cup \{y_{1,A}, \dots, y_{T,A}\}$  (resp.  $\{v_{1,B}, \dots, v_{T,B}\} \cup \{y_{1,B}, \dots, y_{T,B}\}$ ).
5. Use a distance-based clustering algorithm [AK01] to partition the points in  $\tilde{S}_A$  and  $\tilde{S}_B$  after projection.

Figure 4: Algorithm Using Correlations

- [KSV05] R. Kannan, H. Salmasian, and S. Vempala. The spectral method for general mixture models. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.
- [VW02] V. Vempala and G. Wang. A spectral algorithm of learning mixtures of distributions. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 113–123, 2002.

---

# Does Unlabeled Data Provably Help?

## Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning

---

Shai Ben-David and Tyler Lu and Dávid Pál  
David R. Cheriton School of Computer Science  
University of Waterloo  
Waterloo, ON, Canada  
{shai,ttl,dpal}@cs.uwaterloo.ca

### Abstract

We study the potential benefits to classification prediction that arise from having access to unlabeled samples. We compare learning in the semi-supervised model to the standard, supervised PAC (distribution free) model, considering both the realizable and the unrealizable (agnostic) settings.

Roughly speaking, our conclusion is that access to unlabeled samples cannot provide sample size guarantees that are better than those obtainable without access to unlabeled data, unless one postulates very strong assumptions about the distribution of the labels.

In particular, we prove that for basic hypothesis classes over the real line, if the distribution of unlabeled data is ‘smooth’, knowledge of that distribution cannot improve the labeled sample complexity by more than a constant factor (e.g., 2). We conjecture that a similar phenomena holds for any hypothesis class and any unlabeled data distribution. We also discuss the utility of semi-supervised learning under the common *cluster assumption* concerning the distribution of labels, and show that even in the most accommodating cases, where data is generated by two uni-modal label-homogeneous distributions, common SSL paradigms may be misleading and may result in poor prediction performance.

## 1 Introduction

While the problem of classification prediction based on labeled training samples has received a lot of research attention and is reasonably well understood, in many practical learning scenarios, labeled data is hard to come by and unlabeled data is more readily available. Consequently, users try to utilize available unlabeled data to assist with the classification learning process. Learning from both labeled and unlabeled data is commonly called semi-supervised learning (SSL). Due to its wide potential applications, this approach is gaining attention in both the application oriented and the theoretical machine learning communities.

However, theoretical analysis of semi-supervised learning has, so far, been scarce and it falls short of providing

unequivocal explanation of merits of using unlabeled examples in learning. We take steps toward rectifying this theory-practice gap by providing formal analysis of some semi-supervised learning settings. The question we focus on is whether unlabeled data can be utilized to provably improve the sample complexity of classification learning.

We investigate what type of assumptions about the data generating distribution (or which circumstances) are sufficient to make the SSL approach yield better bounds on the predictions accuracy than fully supervised learning. The bulk of this paper focuses on showing that without prior knowledge about the distribution of *labels*, SSL cannot guarantee any significant advantages in sample complexity (e.g., no more than a constant factor for learning tasks over the real line).

we carry our analysis in a simplified, utopian, model of semi-supervised learning, in which the learning algorithm has perfect knowledge of the probability distribution of the unlabeled data. We focus on estimating the *labeled sample* complexity of learning. Since our model provides the learner with more information than just a sample of the unlabeled data distribution, lower bounds on the labeled sample complexity of learning in our model imply similar lower bounds for common notions of semi-supervised learning. Upper bounds, or sample size sufficiency results (for the labeled samples) in our model, apply to the common SSL setting only once sufficiently large unlabeled samples are available to the learner. In this paper we mainly discuss lower bounds, and when we address upper bounds we settle for stating that they apply eventually as the unlabeled sample sizes grow.

Our model of semi-supervised learning can be viewed as learning with respect to a fixed distribution, (see Benedek and Itai [5]). However, our emphasis is different. Our goal is to compare how the *knowledge* of the unlabeled distribution helps, as opposed to learning when the only access to the underlying unlabeled data distribution is via the training labeled sample. We call the former setting *semi-supervised* and the latter *supervised* or *fully supervised* learning.

We present explicit formalization of different ways in which the merits of the semi-supervised paradigm can be measured. We then investigate the extent by which SSL can provide provable advantages over fully supervised learning with respect to these measures.

Roughly speaking, we conclude that no special unlabeled data distribution (like, say, one that breaks into clear data clusters) suffices to render SSL an advantage over fully su-

ervised learning. Unlabeled data can make a difference only under strong assumptions (or prior knowledge) about the conditional *labeled* distribution.

One should note however, that in many cases such knowledge can also be utilized by a fully supervised algorithm. The search for justification to the SSL paradigm therefore leaves us with one setting - the cases where there exists prior knowledge about the *relationship* between the labels and the unlabeled data structure (and not just about the labels per se). However, we show in Section 3 that common applications of SSL paradigms for utilizing such relationship (like the popular *cluster assumption* or the related algorithmic bias towards class boundaries that pass through low-density data regions) may lead to poor prediction accuracy, even when the data does comply with the underlying data model (say, the data is generated by a mixture of two Gaussian distributions, one for each label, each generating a homogeneously labeled set of examples).

The potential merits of SSL, in both settings - either with or without making assumptions about the labeled distribution, have been investigated before. Vapnik’s model of transductive learning [15], as well as Kääriäinen’s paper [12] address the setting without restrictions on the way labels are generated while Balcan-Blum’s augmented PAC model for semi-supervised learning [3, 4] offers a framework for formalizing prior knowledge about the relationship between labels and the structure of the unlabeled distribution. We elaborate more about these in the next section on related work. One basic difference between these works and ours is that they try to provide explanations of the success of the SSL paradigm while we focus on investigating its inherent limitations.

This paper does not resolve the issue of the utility of unlabeled data in full generality. Rather, we provide answers for relatively simple classes of concepts over the real line (thresholds and unions of  $d$  intervals). We believe that these answers generalize to other classes in an obvious way. We also pose some conjectures and open questions.

The paper is organized as follows. We start by discussing previous related work in Section 2. In Section 3 and show that a commonly held assumption can result in performance degradation of SSL. We continue on our main path in Section 4 where we formally define our model of semi-supervised learning and introduce notation. Section 5 casts the previous paradigms in our model and formally poses the question of the utility of unlabeled data to sample based label prediction. This question guides the rest of the paper. Section 6 analyzes this question for basic learning tasks over the real line. The section concludes by asking a slightly different question about the possible meaningful formalizations of the SSL and supervised learning comparison. We conclude our paper in section 7 where we also discuss open questions and directions for further research.

## 2 Related Work

Analysis of performance guarantees for semi-supervised learning can be carried out in two main setups. The first focuses on the unlabeled marginal data distribution and does not make any prior assumptions about the conditional label distribution. The second approach focuses on assump-

tions about the conditional labeled distribution, under which the SSL approach has potentially better label prediction performance than learning based on just labeled samples. The investigation of the first setup was pioneered by Vapnik in the late 70s in his model of *transductive learning*, e.g. [15]. There has been growing interest in this model in the recent years due to the popularity of using unlabeled data in practical label prediction tasks. This model assumes that unlabeled examples are drawn IID from an unknown distribution, and then the labels of some randomly picked subset of these examples are revealed to the learner. The goal of the learner is to label the remaining examples minimizing the error. The main difference between this model and SSL is that the error of learner’s hypothesis is judged only with respect to the known initial sample.

However, there are no known bounds in the transductive setting that are strictly better than supervised learning bounds. Vapnik’s bounds [15] are almost identical. El-Yaniv and Pechyony [10] prove bounds that are similar to the usual margin bounds using Rademacher complexity, except that the learner is allowed to decide *a posteriori* the concept class given the unlabeled examples. But they do not show whether it can be advantageous to choose the class in this way. Their earlier paper [9] gave bounds in terms of a notion of *uniform stability* of the learning algorithm, and in the broader setting where examples are not assumed to come IID from an unknown distribution. But again, it’s not clear whether and when the resulting bounds beat the supervised learning bounds.

Kääriäinen [12] proposes a method for semi-supervised learning without prior assumption on the conditional label distributions. The algorithm of Kääriäinen is based on the observation that one can output the function that minimizes the unlabeled data weights in the symmetric differences to all other functions of the version space. This algorithm *can* reduce the error of supervised ERM by a factor of 2. For more details on these algorithms, see Section 5.

Earlier, Benedek and Itai [5] discuss a model of “learning over a fixed distribution”. Such a model can be viewed as SSL learning, since once the unlabeled data distribution is fixed, it can be viewed as being known to the learner. The idea of Benedek and Itai’s algorithm is to construct a minimum  $\epsilon$ -cover of the hypothesis space under the pseudometric induced by the data distribution. The learning algorithm they propose is to apply empirical risk minimization (ERM) on the functions in such a cover. Of course this  $\epsilon$ -cover algorithm requires knowledge of the unlabeled distribution, without which the algorithm reduces to ERM over the original hypothesis class.

The second, certainly more popular, set of semi-supervised approaches focuses on assumptions about the conditional labeled distributions. A recent PAC model of SSL proposed by Balcan and Blum [3, 4] attempts to formally capture such assumptions. They propose a notion of a compatibility function that assigns a higher score to classifiers which “fit nicely” with respect to the unlabeled distribution. The rationale is that by narrowing down the set of classifiers to only compatible ones, the capacity of the set of potential classifiers goes down and the generalization bounds of empirical risk minimization improve. However, since the set of potential classi-

fiers is trimmed down by a compatibility threshold, if the presumed label-structure relationship fails to hold, the learner may be left with only poorly performing classifiers. One serious concern about this approach is that it provides no way of verifying these crucial modeling assumptions. In Section 3 we demonstrate that this approach may damage learning even when the underlying assumptions seem to hold. In Claim 3 we show that without prior knowledge of such relationship that the Balcan and Blum approach has poor worst-case generalization performance.

Common assumptions include the *smoothness assumption* and the related *low density assumption* [7] which suggests that the decision boundary should lie in a low density region. In section 3, we give examples of mixtures of two Gaussians showing that the low density assumption may be misleading even under favourable data generation models, resulting in low density boundary SSL classifiers with larger error than the outcome of straightforward supervised learning that ignores the unlabeled data.

Many other assumptions about the labels/unlabeled data structure relationship have been investigated, most notably co-training [6] and explicit generative data models [8].

However, all these approaches, are based on very strong assumptions about the data generating distributions. Assumptions that are hard to verify, or to justify on the basis of prior knowledge of a realistic learner.

### 3 On SSL and the Cluster Assumption

This paper has several results of the form “as long as one does not make any assumptions about the behavior of the *labels*, SSL cannot help much over algorithms that ignore the unlabeled data.”

However, two arguments can be raised against such claims. First, SSL is not really intended to be used without any prior assumption about the distribution of labels. In fact, SSL can be viewed as applying some prior knowledge (or just belief) that the labels are somehow correlated with the unlabeled structure of the data. Can we say anything (anything negative, naturally ...) under such an assumption?

Second, maybe using unlabeled data can’t *always* help you, but if it can help *sometimes* why not use it (always)? Well, can we show that in some cases the use of unlabeled data can indeed hurt the learner? Of course, nothing of that kind can apply for all potential learners, since a learner can choose to ignore the unlabeled data and then of course not get hurt by “using” it. We are therefore left with asking, “can the use of unlabeled data hurt the performance of *concrete* common SSL paradigms?”

We briefly address these two questions below by demonstrating that for certain *common* SSL strategies (“low density cut” and Balcan-Blum style use of “compatibility threshold”) SSL can sometimes hurt you, even when the (vaguely stated) “cluster assumption” does hold (when the data breaks into clear uni-modal distributions, each labeled homogeneously).

We also show a general lower bound on the sample complexity of SSL under a general model of the cluster assumption.

In Figures 1, 2, and 3 we depict three examples of simple data distributions over the real line. In all of these examples, the data is generated by a mixture of two uni-modal distributions, each of these modes generates examples labeled ho-

mogeneously, each by a different label. However, the minimum density point of the unlabeled mixture data is significantly off the optimal label prediction decision boundary. Figure 1 shows a mixture of two equal-variance symmetric Gaussians, Figure 2 is a mixture of different Gaussians and Figure 3 shows an extreme case of uni-modal density functions for which the error of the minimum density partition has classification error that is twice that of the optimal decision boundary.

Note that in all such examples, not only does the minimum-density bias mislead the learning process, but also, if one follows the paradigm suggested by Balcan and Blum [4], a wrong choice of the compatibility threshold level will doom the learning process to failure (whereas a simple empirical risk minimization that ignores unlabeled data will succeed based on a small number of labeled samples).

In [13] Rigollet present a formal model of the cluster assumption. Given a probability distribution,  $D$  over some Euclidean data domain, define, for any positive real number,  $a$ ,  $L(a) = \{x : p(x) > a\}$ . The *cluster assumption* says that points in each of the connected components of  $L(a)$  [after removal of “lines or thin ribbons”] have the same Bayesian optimum label.

This is a quite strong assumption under which one can apply an SSL approach. However, in spite of this strong cluster assumption, we can prove that the ratio between the sample complexity of SSL and SL is at most  $d$ - the Euclidean dimension of the data.

Namely, on one hand, the results of Section 6, below, provide a lower bound of  $\Omega\left(\frac{k+\ln(1/\delta)}{\epsilon^2}\right)$  on the sample complexity of SSL learning under this cluster assumption, where  $k$  is the number of connected components of  $L(a)$ . On the other hand, a learner that has access to only labeled examples, can apply the basic ERM algorithm to the class of all  $k$ -cell Voronoi partitions of the space. Since the VC-dimension of the class of all  $k$ -cell Voronoi partitions in  $R^d$  is of order  $kd$ , the usual VC-bounds on the sample complexity of such an SL learner is  $O\left(\frac{kd+\ln(1/\delta)}{\epsilon^2}\right)$  examples.

### 4 A No-Prior-Knowledge Model of Semi-Supervised Learning

We work in the common (agnostic) PAC framework, in which a learning problem is modeled by a probability distribution  $P$  over  $X \times \{0, 1\}$  for some domain set,  $X$ . Any function from  $X$  to  $\{0, 1\}$  is called a *hypothesis*. Examples are pairs,  $(x, y) \in X \times \{0, 1\}$ , and a *sample* is a finite sequence  $S = \{(x_i, y_i)\}_{i=1}^m$  of examples.

**Definition 1** (SL and SSL).

- A supervised learning (SL) algorithm is a function,  $L : \bigcup_{m \in \mathbb{N}} (X \times \{0, 1\})^m \rightarrow \{0, 1\}^X$ , that mapping samples to a hypotheses.
- A semi-supervised learning (SSL) algorithm is a function  $L : \bigcup_{m \in \mathbb{N}} (X \times \{0, 1\})^m \times \mathcal{P} \rightarrow \{0, 1\}^X$ , where  $\mathcal{P}$  is a set of probability distributions over  $X$ . Namely, an SSL algorithm takes as input not only a finite labeled sample but also a probability distribution over the domain set (and outputs a hypothesis, as before).

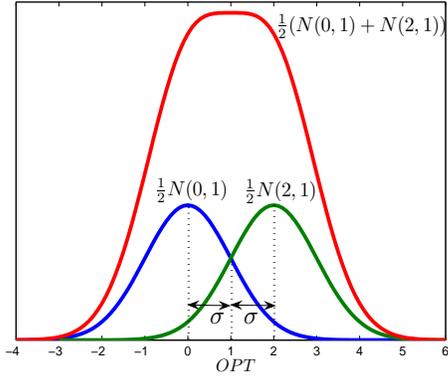


Figure 1: Mixture of two Gaussians  $\mathcal{N}(0, 1)$  (labeled ‘-’) and  $\mathcal{N}(2, 1)$  (labeled ‘+’) shows that the optimum threshold is at  $x = 1$ , the densest point of the unlabeled distribution. The sum of these two Gaussians is unimodal.

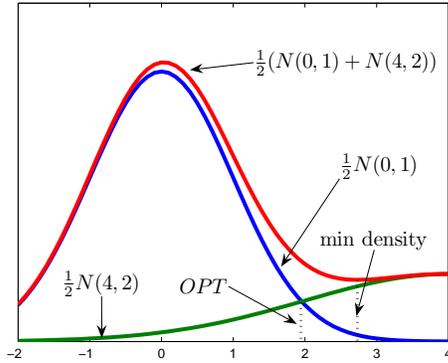


Figure 2: Mixture of two Gaussians  $\mathcal{N}(0, 1)$  (labeled ‘-’) and  $\mathcal{N}(4, 2)$  (labeled ‘+’) with difference variances. The minimum density point of the unlabeled data (the sum of the two distributions) does not coincide with the optimum label-separating threshold where the two Gaussians intersect. The classification error of optimum is  $\approx 0.17$  and that of the minimum density partition is  $\approx 0.21$ .

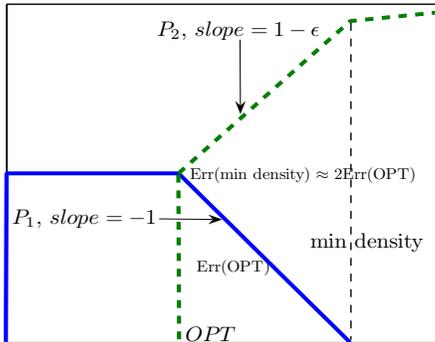


Figure 3: The solid line indicates the distribution  $P_1$  (labeled ‘-’) and the dotted line is  $P_2$  (labeled ‘+’). The  $x$  coordinate of their intersection is the optimum label prediction boundary. The slope of the solid line is slightly steeper than that of the dotted line ( $|-1| > 1 - \epsilon$ ). The minimum density point occurs where the density of  $P_1$  reaches 0. The error of the minimum unlabeled density threshold is twice that of the optimum classifier.

For a distribution  $P$  over  $X \times \{0, 1\}$ , let  $\mathcal{D}(P)$  denote the marginal distribution over  $X$ . That is, formally, for  $X' \subseteq X$  we define  $\mathcal{D}(P)(X') = P(X' \times \{0, 1\})$  (provided that  $X' \times \{0, 1\}$  is  $P$ -measurable). For a learning problem  $P$ , we call  $\mathcal{D}(P)$  the *unlabeled distribution* of  $P$ .

Following the common PAC terminology and notation, the *error* of a hypothesis  $h$ , with respect to  $P$ , is  $\text{Err}^P(h) = \Pr_{(x,y) \sim P}[h(x) \neq y]$ . Similarly, the *empirical error*,  $\text{Err}^S(h)$ , of a hypothesis  $h$  on a sample  $S$  is defined as  $\text{Err}^S(h) = \frac{1}{m} |\{i : i \in \{1, 2, \dots, m\}, h(x_i) \neq y_i\}|$ .

**Definition 2** (The sample complexities (SSL and SL) of a class). For a class  $H$  of hypotheses, the sample complexity of a semi-supervised learning algorithm  $A$  with respect to  $P$ , confidence  $\delta > 0$  and accuracy  $\epsilon > 0$ , is

$$m(A, H, P, \epsilon, \delta) = \min \{m \in \mathbb{N} : \Pr_{S \sim P^m} [\text{Err}^P(A(S, \mathcal{D}(P))) - \inf_{h' \in H} \text{Err}^P(h') > \epsilon] < \delta\}.$$

The *sample complexity* of a supervised learning algorithm  $A$  is defined similarly, except that the second input parameter  $\mathcal{D}(P)$  is omitted.

We consider two settings, realizable and agnostic. In the *agnostic* setting,  $P$  can be arbitrary. The *realizable* setting is defined by assuming that there exists hypothesis  $h \in H$  such that  $\text{Err}^P(h) = 0$ ; consequently  $\inf_{h' \in H} \text{Err}^P(h') = 0$ . In particular, this implies that for any  $x \in X$ , the conditional probabilities,  $P(y = 0 | x)$  and  $P(y = 1 | x)$  are always either 0 or 1; in the agnostic setting the conditionals can be arbitrary.

Without reference to any learning problem, an *unlabeled distribution*  $D$  is simply any distribution over  $X$ . We use  $\text{Ext}(D)$  to denote all possible *extensions* of  $D$ , that is,  $\text{Ext}(D)$  is the family of all possible distributions  $P$  over  $X \times \{0, 1\}$  such that  $\mathcal{D}(P) = D$ . For an unlabeled distribution  $D$  and hypothesis  $h$ ,  $D_h$  denotes the probability distribution in  $\text{Ext}(D)$  such that  $D_h(y = h(x) | x) = 1$ . For a hypothesis  $h$  and an “unlabeled sample”  $S = \{x_i\}_{i=1}^m$ , where  $x_i \in X$ , we denote by  $(S, h(S))$  the sample  $\{(x_i, h(x_i))\}_{i=1}^m$ .

For a subset  $T$  of some domain set, we use  $\mathbf{1}_T$  to denote its characteristic function. In particular, if  $T \subseteq X$  then  $\mathbf{1}_T$  is a hypothesis over  $X$ . For two hypothesis  $g, h$  we use  $g \Delta h$  to denote their “symmetric difference”, that is,  $g \Delta h$  is a hypothesis  $\mathbf{1}\{x \in X : g(x) \neq h(x)\}$ . Finally,  $\text{VC}(H)$  denotes the VC-dimension [14] of hypothesis class  $H$ .

## 5 Previous No Prior Knowledge Paradigms

Previous approaches to SSL algorithms for the no prior knowledge paradigm have used the unlabeled sample to figure out the “geometry” of the hypothesis space with respect to the unlabeled (marginal) distribution. A common approach is to use that knowledge to reduce the hypothesis search space. In doing so, one may improve the generalization upper bounds.

Recall that given an unlabeled distribution  $D$  and a hypothesis class  $H$ , an  $\epsilon$ -cover is a subset  $H' \subseteq H$  such that for any  $h \in H$  there exists  $g \in H'$  such that  $D(g \Delta h) \leq \epsilon$ . Note that if  $H'$  is an  $\epsilon$ -cover for  $H$  with respect to  $D$ , then for every extension  $P \in \text{Ext}(D)$  the  $\inf_{g \in H'} \text{Err}^P(g) \leq \inf_{h \in H} \text{Err}^P(h) + \epsilon$ .

In some cases the construction of a small  $\epsilon$ -cover is a major use of unlabeled data. Benedek and Itai [5] analyze the approach, in the case when the unlabeled distribution is fixed and therefore can be thought of as known to the learner. They show that the smaller an  $\epsilon$ -cover is the better its generalization bound one for the ERM algorithm over this cover.

Balcan and Blum [4] suggest a different way of using the unlabeled data to reduce the hypothesis space. However, we claim that without making any prior assumptions about the relationship between the labeled and unlabeled distributions, their approach boils down to the  $\epsilon$ -cover construction described above.

**Claim 3.** *Let  $H$  be any hypotheses class,  $\epsilon, \delta > 0$ , and  $D$  be any unlabeled distribution. Let  $H' \subseteq H$  be the set of “compatible hypotheses.” Suppose  $A$  is an SSL algorithm that outputs any hypothesis in  $H'$ . If  $H'$  does not contain an  $\epsilon$ -cover of  $H$  with respect to  $D$ , the error of the hypothesis that  $A$  outputs is at least  $\epsilon$  regardless of the size of the labeled sample.*

*Proof.* Since  $H'$  does not contain an  $\epsilon$ -cover of  $H$ , there exist a hypothesis  $h \in H$  such that for all  $g \in H'$ ,  $D(g\Delta h) > \epsilon$ . Thus, for any  $g \in H'$ ,  $\text{Err}^{D_h}(g) > \epsilon$ . Algorithm  $A$  outputs some  $g \in H'$  and the proof follows.  $\square$

Kääriäinen [12] utilizes the unlabeled data in a different way. Given the labeled data his algorithm constructs the version space  $F \subseteq H$  of all sample-consistent hypotheses, and then applies the knowledge of the unlabeled distribution  $D$  to find the “center” of that version space. Namely, a hypothesis  $g \in F$  that minimizes  $\max_{h \in F} D(g\Delta h)$ .

Clearly, all the above paradigms depend on the knowledge of the unlabeled distribution  $D$ . In return, better upper bounds on the sample complexity of the respective algorithms (or equivalently on the errors of the hypotheses produced by such algorithms) can be shown. For example, Benedek and Itai give (for the realizable case) an upper bound on the sample complexity that depends on the size of the  $\epsilon$ -cover—the smaller  $\epsilon$ -cover, the smaller the upper bound.

In the next section we analyze the gains that such knowledge of unlabeled data distribution can make in the no prior knowledge setting. We prove that over the real line for any “smooth” unlabeled distribution  $D$ , ERM over the full hypothesis class  $H$  has worst case sample complexity that is at most by constant factor bigger than the worst case sample complexity of any SSL algorithm. We conjecture that this is a more general phenomenon.

**Conjecture 4.** For any hypothesis class  $H$ , there exists a constant  $c \geq 1$  and a supervised algorithm  $A$ , such that for any distribution  $D$  over the domain and any semi-supervised learning algorithm  $B$ ,

$$\sup_{h \in H} m(A, H, D_h, \epsilon, \delta) \leq c \cdot \sup_{h \in H} m(B, H, D_h, \epsilon, \delta)$$

for any  $\epsilon$  and  $\delta$  small enough, say smaller than  $1/c$ .

**Conjecture 5.** For any hypothesis class  $H$ , there exists a constant  $c \geq 1$  and a supervised algorithm  $A$ , such that for

any distribution  $D$  over the domain and any semi-supervised learning algorithm  $B$ ,

$$\sup_{P \in \text{Ext}(D)} m(A, H, P, \epsilon, \delta) \leq c \cdot \sup_{P \in \text{Ext}(D)} m(B, H, P, \epsilon, \delta)$$

for any  $\epsilon$  and  $\delta$  small enough, say smaller than  $1/c$ .

## 6 Inherent Limitations of Semi-Supervised Learning

This section is devoted to proving the inherent limitations of SSL paradigm in the no prior knowledge model over the real line. In Section 6.2 we prove Conjecture 4 for thresholds on the real line in the realizable setting, under the condition that the unlabeled distribution is absolutely continuous. In Section 6.3 we prove Conjecture 5 for thresholds and union of  $d$  intervals over the real line in the agnostic setting (under the same unlabeled distribution condition).

The former follows from Theorems 8 and 10. The latter follows from Corollary 13 (for thresholds) and from Corollary 16 (for union of  $d$  intervals). To prove the results we rely on a simple “rescaling trick” that we explain in Section 6.1.

We briefly sketch the idea of the proofs. Let us start by defining the hypothesis classes. The class of thresholds is defined as  $H = \{\mathbf{1}(-\infty, t] : t \in \mathbb{R}\}$  and the class of union of  $d$  intervals

$$UI_d = \{\mathbf{1}[a_1, a_2) \cup [a_3, a_4) \cup \dots \cup [a_{2\ell-1}, a_{2\ell}) : \ell \leq d, a_1 \leq a_2 \leq \dots \leq a_{2\ell}\}.$$

The rescaling trick says that the SSL sample complexity of learning  $H$  (resp.  $UI_d$ ) under any two absolutely continuous unlabeled distributions is exactly the same. We can thus focus on the sample complexity of learning under some fixed absolutely continuous distribution; for concreteness and convenience we chose the uniform distribution over  $(0, 1)$ . By proving a sample complexity lower bound on the learning under the uniform distribution over  $(0, 1)$ , we are effectively proving a lower bound on the sample complexity of SSL under any absolutely continuous distribution. Through the use of techniques from the probabilistic method, we obtain lower bounds on the SSL sample complexity that is within a constant factor of the well-known upper bounds on SL sample complexity (e.g. VC upper bounds on the sample complexity of ERM for any unknown distribution).

In Section 6.4 we discuss other possible formulations of the comparison between SL and SSL algorithms.

### 6.1 Rescaling Trick

In this section we show that learning any “natural” hypothesis class on the real line has the same sample complexity for any absolutely continuous unlabeled distribution independent of its shape. Intuitively, if we imagine the real axis made of rubber, then a natural hypothesis class is one that is closed under rescaling (stretching) of the axis. Classes of thresholds and union of  $d$  intervals are examples of such natural classes, since under any rescaling an interval remains an interval. The rescaling will apply also on the unlabeled distribution over the real line and it will allow us to go from any absolutely continuous distribution to the uniform distribution over  $(0, 1)$ .

More formally, a *rescaling* is a continuous increasing function  $f$  from an open interval  $I$  onto an open interval  $J$ . We denote by  $H|_A$  the restriction of a class  $H$  to a subset  $A$ , that is,  $H|_A = \{h|_A : h \in H\}$ . We use  $\circ$  to denote function composition. We say that a hypothesis class  $H$  over  $\mathbb{R}$  is *closed under rescaling* whenever for any rescaling  $f : I \rightarrow J$ , if  $h|_J \in H|_J$ , then  $h|_J \circ f \in H|_I$ . If  $H$  is any class closed under rescaling, then any rescaling  $f$  induces a bijection  $h|_J \mapsto h|_J \circ f$  between  $H|_I$  and  $H|_J$ . (This follows since  $f^{-1}$  is also rescaling.) Clearly, the class of thresholds and the class of unions of  $d$  intervals are closed under rescaling.

We show that the sample complexity is unaffected by the rescalings provided that the hypothesis class is closed under rescalings. We split the results into two lemmas—Lemma 6 and Lemma 7. The first lemma shows that if we have a supervised algorithm with certain sample complexity for the case when the unlabeled distribution is the uniform distribution over  $(0, 1)$ , then the algorithm can be translated into an SSL algorithm with the same sample complexity for the case when the unlabeled distribution is any absolutely continuous distribution. The second lemma shows the translation in the other direction. Namely, that a SSL algorithm with certain sample complexity on some absolutely continuous unlabeled distribution can be translated to a supervised algorithm for the case when unlabeled distribution is uniform over  $(0, 1)$ .

**Lemma 6** (Rescaling trick I). *Let  $H$  be a hypothesis class over  $\mathbb{R}$  closed under rescaling. Let  $U$  be the uniform distribution over  $(0, 1)$ . Let  $\epsilon, \delta > 0$ .*

(a) (Realizable case): *If  $A$  is any supervised or semi-supervised algorithm, then there exists an semi-supervised learning algorithm  $B$  such that for any distribution  $D$  over an open interval  $I$  which is absolutely continuous with respect to Lebesgue measure on  $I$*

$$\sup_{h \in H} m(B, H, D_h, \epsilon, \delta) \leq \sup_{g \in H} m(A, H, U_g, \epsilon, \delta). \quad (1)$$

(b) (Agnostic case): *If  $A$  is any supervised or semi-supervised algorithm, then there exists an semi-supervised learning algorithm  $B$  such that for any distribution  $D$  over an open interval  $I$  which is absolutely continuous with respect to Lebesgue measure on  $I$*

$$\sup_{P \in \text{Ext}(D)} m(B, H, P, \epsilon, \delta) \leq \sup_{Q \in \text{Ext}(U)} m(A, H, Q, \epsilon, \delta). \quad (2)$$

*Proof.* Fix  $H$  and  $A$ . We construct algorithm  $B$  as follows. The algorithm  $B$  has two inputs, a sample  $S = \{(x_i, y_i)\}_{i=1}^m$  and a distribution  $D$ . Based on  $D$  the algorithm computes the cumulative distribution function  $F : I \rightarrow (0, 1)$ ,  $F(t) = D(I \cap (-\infty, t])$ . Then,  $B$  computes from  $S$  transformed sample  $S' = \{(x'_i, y_i)\}_{i=1}^m$  where  $x'_i = F(x_i)$ . On a sample  $S'$  the algorithm  $B$  simulates algorithm  $A$  and computes  $h = A(S')$ . (If  $A$  is semi-supervised we fix its second input to be  $U$ ). Finally,  $B$  outputs  $g = h \circ F$ .

It remains to show that for any  $D$  with continuous cumulative distribution function (1) and (2) holds for any  $\epsilon, \delta > 0$ . We prove (2), the other equality is proved similarly.

Let  $P \in \text{Ext}(D)$ . Slightly abusing notation, we define the “image” distribution  $F(P)$  over  $(0, 1) \times \{0, 1\}$  to be

$$F(P)(M) = P(\{(x, y) : (F(x), y) \in M\})$$

for any (measurable)  $M \subseteq (0, 1) \times \{0, 1\}$ . It is not hard to see that if  $S$  is distributed according to  $P^m$ , then  $S'$  is distributed according to  $(F(P))^m$ . Clearly,  $\mathcal{D}(F(P)) = U$  i.e.  $F(P) \in \text{Ext}(U)$ . Further note that since  $D$  is absolutely continuous,  $F$  is a rescaling. Hence  $\text{Err}^{F(P)}(h) = \text{Err}^P(h \circ F)$  and  $\inf_{h \in H} \text{Err}^P(h) = \inf_{h \in H} \text{Err}^{F(P)}(h)$ . Henceforth, for any  $\epsilon$  and any  $m \in \mathbb{N}$

$$\begin{aligned} & \Pr_{S \sim P^m} [\text{Err}^P(B(S, D)) - \inf_{h \in H} \text{Err}^P(h) > \epsilon] \\ &= \Pr_{S' \sim (F(P))^m} [\text{Err}^P(A(S') \circ F) - \inf_{h \in H} \text{Err}^{F(P)}(h) > \epsilon] \\ &= \Pr_{S' \sim (F(P))^m} [\text{Err}^{F(P)}(A(S')) - \inf_{h \in H} \text{Err}^{F(P)}(h) > \epsilon]. \end{aligned}$$

Therefore, for any  $\epsilon, \delta > 0$ ,

$$\begin{aligned} m(B, H, P, \epsilon, \delta) &= m(A, H, F(P), \epsilon, \delta) \\ &\leq \sup_{Q \in \text{Ext}(P)} m(A, H, Q, \epsilon, \delta). \end{aligned}$$

Taking supremum over  $P \in \text{Ext}(D)$  finishes the proof.  $\square$

**Lemma 7** (Rescaling trick II). *Let  $H$  be a hypothesis class over  $\mathbb{R}$  closed under rescaling. Let  $U$  be the uniform distribution over  $(0, 1)$ . Let  $\epsilon, \delta > 0$ .*

(a) (Realizable case): *If  $B$  is any supervised or semi-supervised algorithm and  $D$  is any distribution over an open interval  $I$ , which is absolutely continuous with respect to the Lebesgue measure on  $I$ , then there exists a supervised learning algorithm  $A$  such that*

$$\sup_{g \in H} m(A, H, U_g, \epsilon, \delta) \leq \sup_{h \in H} m(B, H, D_h, \epsilon, \delta). \quad (3)$$

(b) (Agnostic case): *If  $B$  is any supervised or semi-supervised algorithm and  $D$  is any distribution over an open interval  $I$ , which is absolutely continuous with respect to the Lebesgue measure on  $I$ , then there exists a supervised learning algorithm  $A$  such that*

$$\sup_{Q \in \text{Ext}(U)} m(A, H, Q, \epsilon, \delta) \leq \sup_{P \in \text{Ext}(D)} m(B, H, P, \epsilon, \delta). \quad (4)$$

*Proof.* Fix  $H, B$  and  $D$ . Let  $F : I \rightarrow (0, 1)$  be the cumulative distribution function of  $D$ , that is,  $F(t) = D(I \cap (-\infty, t))$ . Since  $D$  is absolutely continuous,  $F$  is a rescaling and inverse  $F^{-1}$  exists.

Now, we construct algorithm  $A$ . Algorithm  $A$  maps input sample  $S' = \{(x'_i, y_i)\}_{i=1}^m$  to sample  $S = \{(x_i, y_i)\}_{i=1}^m$  where  $x_i = F^{-1}(x'_i)$ . On a sample  $S$  the algorithm  $A$  simulates algorithm  $B$  and computes  $g = B(S, D)$ . (If  $B$  is supervised, then the second input is omitted.) Finally,  $A$  outputs  $h = g \circ F^{-1}$ .

It remains to show that for any  $D$  with continuous cumulative distribution function (3) and (4) holds for any  $\epsilon, \delta > 0$ . We prove (4), the other equality is proved similarly.

Let  $Q \in \text{Ext}(U)$ . Slightly abusing notation, we define the “pre-image” distribution  $F^{-1}(Q)$  over  $I \times \{0, 1\}$  to be

$$F^{-1}(Q)(M) = Q(\{(F(x), y) : (x, y) \in M\})$$

for any (measurable)  $M \subseteq I \times \{0, 1\}$ . It is not hard to see that if  $S'$  is distributed according to  $Q$ , then  $S$  is distributed according to  $(F^{-1}(Q))^m$ . Clearly,  $\mathcal{D}(F^{-1}(U)) = D$  i.e.

$F^{-1}(Q) \in \text{Ext}(D)$ . Since  $F^{-1}$  is a rescaling,  $\text{Err}^{F^{-1}(Q)}(h) = \text{Err}^Q(h \circ F^{-1})$  and  $\inf_{h \in H} \text{Err}^Q(h) = \inf_{h \in H} \text{Err}^{F^{-1}(Q)}(h)$ . Henceforth, for any  $\epsilon > 0$  and any  $m \in \mathbb{N}$

$$\begin{aligned} & \Pr_{S \sim Q^m} [\text{Err}^Q(A(S')) - \inf_{h \in H} \text{Err}^Q(h)] \\ &= \Pr_{S \sim F^{-1}(Q)^m} [\text{Err}^Q(B(S, D) \circ F^{-1}) - \inf_{h \in H} \text{Err}^{F^{-1}(Q)}(h)] \\ &= \Pr_{S \sim F^{-1}(Q)^m} [\text{Err}^{F^{-1}(Q)}(B(S, D)) - \inf_{h \in H} \text{Err}^{F^{-1}(Q)}(h)]. \end{aligned}$$

Therefore, for any  $\epsilon, \delta > 0$ ,

$$\begin{aligned} m(A, H, Q, \epsilon, \delta) &= m(B, H, F^{-1}(Q), \epsilon, \delta) \\ &\leq \sup_{P \in \text{Ext}(D)} m(B, H, P, \epsilon, \delta) \end{aligned}$$

Taking supremum over  $Q \in \text{Ext}(U)$  finishes the proof.  $\square$

## 6.2 Sample Complexity of Learning Thresholds in the Realizable Case

In this section we consider learning the class of thresholds,  $H = \{\mathbf{1}(-\infty, t] : t \in \mathbb{R}\}$ , on the real line in the realizable setting and show that for absolutely continuous unlabeled distributions SSL has at most factor 2 advantage over SL in the sample complexity.

First, in Theorem 8, we show  $\frac{\ln(1/\delta)}{\epsilon}$  upper bound on the sample complexity of supervised learning. This seems to be a folklore result. Second, we consider sample complexity of semi-supervised learning in the case when  $\mathcal{D}(P)$  is absolutely continuous with respect to the Lebesgue measure on  $\mathbb{R}$ . In Theorems 9 and 10 we show that the sample complexity is between  $\frac{\ln(1/\delta)}{2\epsilon} + O(\frac{1}{\epsilon})$  and  $\frac{\ln(1/\delta)}{2.01\epsilon} - O(\frac{1}{\epsilon})$ .<sup>1</sup> Ignoring the lower order terms, we see that the sample complexity of supervised learning is (asymptotically) at most 2-times larger than that of semi-supervised learning.

We will make use the following of two algorithms: supervised algorithm  $L$  and semi-supervised algorithm  $B$  proposed by Kääriäinen [12]. Both algorithms on a sample  $S = ((x_1, y_2), (x_2, y_2), \dots, (x_m, y_m))$  first compute

$$\begin{aligned} \ell &= \max\{x_i : i \in \{1, 2, \dots, m\}, y_i = 1\}, \\ r &= \min\{x_i : i \in \{1, 2, \dots, m\}, y_i = 0\}. \end{aligned}$$

Algorithm  $L$  simply outputs the hypothesis  $\mathbf{1}(-\infty, \ell]$ . Algorithm  $B$  makes use of its second input, distribution  $D$ . Provided that  $\ell < r$ ,  $B$  computes  $t'' = \sup\{t' : D((\ell, t']) \leq D((\ell, r])/2\}$  and outputs hypothesis  $\mathbf{1}(-\infty, t'']$ .

**Theorem 8** (SL upper bound). *Let  $H$  be the class of thresholds and  $L$  be the supervised learning algorithm defined above. For any  $D$ , for any  $\epsilon, \delta > 0$ , and any “target”  $h \in H$ ,*

$$m(A, H, D_h, \epsilon, \delta) \leq \frac{\ln(1/\delta)}{\epsilon}.$$

*Proof.* Let  $h = \mathbf{1}(-\infty, t)$  and let  $s = \sup\{s : D((s, t]) \geq \epsilon\}$ . The event  $\text{Err}^{D_h}(L(S)) \geq \epsilon$  occurs precisely when

<sup>1</sup>The 2.01 in the lower bound can be replaced by arbitrary number strictly greater than 2. This slight imperfection is a consequence of that the true dependence of the sample complexity on  $\epsilon$ , in this case, is of the form  $1/\ln(1-2\epsilon)$  and not  $1/(2\epsilon)$ .

the interval  $(s, t]$  does not contain any sample points. This happens with probability  $(1 - D((s, t]))^m \leq (1 - \epsilon)^m$ . If  $m \geq \frac{\ln(1/\delta)}{\epsilon}$ , then  $(1 - \epsilon)^m \leq \exp(-\epsilon m) \leq \delta$ .  $\square$

**Theorem 9** (SSL upper bound). *Let  $H$  be the class of thresholds and  $B$  be the semi-supervised learning algorithm defined above. For any absolutely continuous distribution  $D$  over an open interval, any  $\epsilon \in (0, \frac{1}{4})$ ,  $\delta \in (0, \frac{1}{2})$ , and any “target”  $h \in H$ ,*

$$m(B, H, D_h, \epsilon, \delta) \leq \frac{\ln(1/\delta)}{2\epsilon} + \frac{\ln 2}{2\epsilon}.$$

*Proof.* By rescaling trick (Lemma 6 part (a)) we can assume that  $D$  is uniform over  $(0, 1)$ . Fix  $\epsilon \in (0, \frac{1}{4})$ ,  $\delta \in (0, \frac{1}{2})$  and  $h \in H$ . We show that, for any  $m \geq 2$ ,

$$\Pr_{S \sim D_h^m} [\text{Err}^{D_h}(B(S, D_h)) \geq \epsilon] \leq 2(1 - 2\epsilon)^m, \quad (5)$$

from which the theorem easily follows, since if  $m \geq \frac{\ln(1/\delta)}{2\epsilon} + \frac{\ln 2}{2\epsilon}$ , then  $m \geq 2$  and  $2(1 - 2\epsilon)^m \leq 2\exp(-2m\epsilon) \leq \delta$ .

In order to prove (5), let  $h = \mathbf{1}(-\infty, t]$  be the “target”. Without loss of generality  $t \in [0, \frac{1}{2}]$ . With a little abuse, we assume that  $\ell \in [0, t]$  and  $r \in [t, 1]$ . For convenience, we define  $a : [0, t] \rightarrow [t, 1]$ ,  $b : [0, t] \rightarrow [t, 1]$  as  $a(\ell) = \max(2t - \ell - 2\epsilon, t)$  and  $b(\ell) = \min(2t - \ell + 2\epsilon, 1)$  respectively. It is easily verified that  $\text{Err}^{D_h}(B(S, D_h)) \leq \epsilon$  if and only if  $r \in [a(\ell), b(\ell)]$ .

We lower bound the probability of success

$$p = \Pr_{S \sim D_h^m} [\text{Err}^{D_h}(B(S, D_h)) \leq \epsilon].$$

There are two cases:

*Case 1:* If  $t > 2\epsilon$ , then we integrate over all possible choices of the rightmost positive example in  $S$  (which determines  $\ell$ ) and leftmost negative example in  $S$  (which determines  $r$ ). There are  $m(m-1)$  choices for the rightmost positive example and leftmost negative example. We have

$$p \geq p_1 = m(m-1) \int_0^t \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} dr d\ell.$$

*Case 2:* If  $t \leq 2\epsilon$ , then we integrate over all possible choices of the rightmost positive example in  $S$  and leftmost negative example in  $S$ . Additionally we also consider samples without positive examples, and integrate over all possible choices of the leftmost (negative) example. We have

$$\begin{aligned} p \geq p_2 &= m(m-1) \int_0^t \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} dr d\ell \\ &\quad + m \int_t^{2\epsilon} (1-r)^{m-1} dr \end{aligned}$$

Both cases split into further subcases.

*Subcase 1a:* If  $t > 2\epsilon$  and  $t + 4\epsilon \leq 1$  and  $t + \epsilon \geq 1/2$ ,

then  $0 \leq 2t + 2\epsilon - 1 \leq t - 2\epsilon \leq t$  and

$$\begin{aligned}
p_1 &= m(m-1) \left[ \int_0^{2t+2\epsilon-1} \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \right. \\
&\quad + \int_{2t+2\epsilon-1}^{t-2\epsilon} \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \\
&\quad \left. + \int_{t-2\epsilon}^t \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \right] \\
&= m(m-1) \left[ \int_0^{2t+2\epsilon-1} \int_{2t-\ell-2\epsilon}^1 (1-r+\ell)^{m-2} \, dr d\ell \right. \\
&\quad + \int_{2t+2\epsilon-1}^{t-2\epsilon} \int_{2t-\ell-2\epsilon}^{2t-\ell+2\epsilon} (1-r+\ell)^{m-2} \, dr d\ell \\
&\quad \left. + \int_{t-2\epsilon}^t \int_t^{2t-\ell+2\epsilon} (1-r+\ell)^{m-2} \, dr d\ell \right] \\
&= 1 - \frac{1}{2}(1-2t-2\epsilon)^m - \frac{1}{2}(-1+2t+6\epsilon)^m - (1-2\epsilon)^m \\
&\geq 1 - 2(1-2\epsilon)^m.
\end{aligned}$$

*Subcase 1b:* If  $t > 2\epsilon$  and  $t + \epsilon \leq 1/2$ , then  $2t + 2\epsilon - 1 \leq 0 \leq t - 2\epsilon \leq t$  and

$$\begin{aligned}
p_1 &= m(m-1) \left[ \int_0^{t-2\epsilon} \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \right. \\
&\quad \left. + \int_{t-2\epsilon}^t \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \right] \\
&= m(m-1) \left[ \int_0^{t-2\epsilon} \int_{2t-\ell-2\epsilon}^{2t-\ell+2\epsilon} (1-r+\ell)^{m-2} \, dr d\ell \right. \\
&\quad \left. + \int_{t-2\epsilon}^t \int_t^{2t-\ell+2\epsilon} (1-r+\ell)^{m-2} \, dr d\ell \right] \\
&= 1 - (1-2\epsilon)^m + \frac{1}{2}(1-2t-2\epsilon)^m - \frac{1}{2}(1-2t+2\epsilon)^m \\
&\geq 1 - \frac{3}{2}(1-2\epsilon)^m.
\end{aligned}$$

*Subcase 1c:* If  $t > 2\epsilon$  and  $t + 4\epsilon \geq 1$ , then  $0 \leq t - 2\epsilon \leq 2t + 2\epsilon - 1 \leq t$ , and

$$\begin{aligned}
p_1 &= m(m-1) \left[ \int_0^{t-2\epsilon} \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \right. \\
&\quad + \int_{t-2\epsilon}^{2t+2\epsilon-1} \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \\
&\quad \left. + \int_{2t+2\epsilon-1}^t \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \right] \\
&= m(m-1) \left[ \int_0^{t-2\epsilon} \int_{2t-\ell-2\epsilon}^1 (1-r+\ell)^{m-2} \, dr d\ell \right. \\
&\quad + \int_{t-2\epsilon}^{2t+2\epsilon-1} \int_t^1 (1-r+\ell)^{m-2} \, dr d\ell \\
&\quad \left. + \int_{2t+2\epsilon-1}^t \int_t^{2t-\ell+2\epsilon} (1-r+\ell)^{m-2} \, dr d\ell \right] \\
&= 1 - (1-2\epsilon)^m - \frac{1}{2}(1-2t+2\epsilon)^m - \frac{1}{2}(2t+2\epsilon-1)^m \\
&\geq 1 - 2(1-2\epsilon)^m.
\end{aligned}$$

*Subcase 2a:* If  $t \leq 2\epsilon$  and  $t + \epsilon \geq 1/2$ , then  $t - 2\epsilon \leq 0 \leq 2t + 2\epsilon - 1 \leq t$  and

$$\begin{aligned}
p_2 &= m(m-1) \left[ \int_0^{2t+2\epsilon-1} \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \right. \\
&\quad \left. + \int_{2t+2\epsilon-1}^t \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \right] \\
&\quad + m \int_t^{2\epsilon} (1-r)^{m-1} \, dr \\
&= m(m-1) \left[ \int_0^{2t+2\epsilon-1} \int_t^1 (1-r+\ell)^{m-2} \, dr d\ell \right. \\
&\quad \left. + \int_{2t+2\epsilon-1}^t \int_t^{2t-\ell+2\epsilon} (1-r+\ell)^{m-2} \, dr d\ell \right] \\
&\quad + (1-t)^m - (1-2\epsilon)^m \\
&= 1 - \frac{3}{2}(1-2\epsilon)^m - \frac{1}{2}(2t+2\epsilon-1)^m \\
&\geq 1 - 2(1-2\epsilon)^m.
\end{aligned}$$

*Subcase 2b:* If  $t \leq 2\epsilon$  and  $t + \epsilon \leq 1/2$ , then  $t - 2\epsilon \leq 0$ ,  $2t + 2\epsilon - 1 \leq 0$  and

$$\begin{aligned}
p_2 &= m(m-1) \int_0^t \int_{a(\ell)}^{b(\ell)} (1-r+\ell)^{m-2} \, dr d\ell \\
&\quad + m \int_t^{2\epsilon} (1-r)^{m-1} \, dr \\
&= m(m-1) \int_0^t \int_t^{2t-\ell+2\epsilon} (1-r+\ell)^{m-2} \, dr d\ell \\
&\quad + (1-t)^m - (1-2\epsilon)^m \\
&= 1 - \frac{3}{2}(1-2\epsilon)^m - \frac{1}{2}(1-2t-2\epsilon)^m \\
&\geq 1 - 2(1-2\epsilon)^m.
\end{aligned}$$

□

**Theorem 10** (SSL lower bound). *For any (randomized) semi-supervised algorithm  $A$ , any  $\epsilon \in (0, 0.001)$ , any  $\delta > 0$ , any absolutely continuous probability distribution  $D$  over an open interval, there exists  $h \in H$ , such that*

$$m(A, H, D_h, \epsilon, \delta) \geq \frac{\ln(1/\delta)}{2.01\epsilon} - \frac{\ln 2}{2.01\epsilon}.$$

*Proof.* By rescaling trick (Lemma 7 part (a)) we can assume that  $D$  is uniform over  $(0, 1)$ . Fix  $A, \epsilon, \delta$ . We show the existence of required  $h$  by a probabilistic argument. We consider picking  $t$  uniformly at random from  $(0, 1)$  and let  $h = \mathbf{1}(-\infty, t]$ . We prove that for any  $m \geq 0$ ,

$$\mathbb{E}_t \Pr_{S \sim D_h^m} [\text{Err}^{D_h}(A(S, D_h)) \geq \epsilon] \geq \frac{1}{2}(1-2\epsilon)^m. \quad (6)$$

The left-hand side can be rewritten as

$$\begin{aligned}
&\mathbb{E}_t \Pr_{S \sim D_h^m} [\text{Err}^{D_h}(A(S, D_h)) \geq \epsilon] \\
&= \mathbb{E}_t \mathbb{E}_{S \sim D_h^m} \mathbf{1}\{(t, S) : \text{Err}^{D_h}(A(S, D)) \geq \epsilon\} \\
&= \mathbb{E}_{S \sim D^m} \mathbb{E}_t \mathbf{1}\{(t, S) : \text{Err}^{D_h}(A((S, h(S)), D)) \geq \epsilon\} \\
&= \mathbb{E}_{S \sim D^m} \Pr_t [\text{Err}^{D_h}(A(h(S), D_h)) \geq \epsilon]
\end{aligned}$$

To lower bound the last expression, fix unlabeled points  $0 \leq x_1 \leq x_2 \leq \dots \leq x_m \leq 1$ . For convenience, let  $x_0 = 0$  and  $x_{m+1} = 1$ . We claim that

$$\Pr_t [\text{Err}^{D_h}(A((S, h(S)), D)) \geq \epsilon] \geq \sum_{i=0}^m \max(x_{i+1} - x_i - 2\epsilon, 0). \quad (7)$$

To prove that we also fix  $i \in \{0, 1, 2, \dots, m\}$  and restrict  $t$  to lie in the interval  $(x_i, x_{i+1}]$ . The labels in  $(S, h(S))$  are hence fixed. Hence the hypothesis  $g = A((S, h(S)), D)$  is fixed. It is not hard to see that regardless of  $g$

$$\int_{x_i}^{x_{i+1}} \mathbf{1}\{t : \text{Err}^{D_h}(g) \geq \epsilon\} dt \geq \max(x_{i+1} - x_i - 2\epsilon, 0),$$

which follows from that the set  $\{t : \text{Err}^{D_h}(g) < \epsilon\}$  is contained in an interval of length at most  $2\epsilon$ . Summing over all  $i$  we obtain (7).

In order to prove (6) we will compute expectation over  $S \sim D^m$  of both sides of (7). Expectation of the left side of (7) equals to the left side of (6). The expectation of the right side of (7) is equal to

$$I_m = m! \underbrace{\int_0^{x_{m+1}} \int_0^{x_m} \int_0^{x_{m-1}} \dots \int_0^{x_2}}_{m \text{ times}} \sum_{i=0}^m \max(x_{i+1} - x_i - 2\epsilon, 0) dx_1 \dots dx_{m-2} dx_{m-1} dx_m,$$

since there are  $m!$  equiprobable choices for the order of the points  $x_1, x_2, \dots, x_m$  among which we choose, without loss of generality, the one with  $x_1 \leq x_2 \leq \dots \leq x_m$ . We look at  $I_m$  as a function of  $x_{m+1}$  and we prove that

$$I_m(x_{m+1}) = (\max(x_{m+1} - 2\epsilon, 0))^{m+1}, \quad (8)$$

for any  $m \geq 0$  and any  $x_{m+1} \in [0, 1]$ . The bound (6) follows from (8), since  $I_m = I_m(1) = (1 - 2\epsilon)^{m+1} \geq \frac{1}{2}(1 - 2\epsilon)^m$  for  $\epsilon \leq 1/4$ . In turn, (8) follows, by induction on  $m$ , from the recurrence

$$I_m(x_{m+1}) = m \int_0^{x_{m+1}} I_{m-1}(x_m) + \max(x_{m+1} - x_m - 2\epsilon, 0) \cdot x_m^{m-1} dx_m,$$

which is valid for all  $m \geq 1$ . In the base case,  $m = 0$ ,  $I_0(x_1) = \max(x_1 - 2\epsilon, 0)$  trivially follows by definition. In the inductive case,  $m \geq 1$ , we consider two cases. First case,  $x_{m+1} < 2\epsilon$ , holds since  $\max(x_{i+1} - x_i - 2\epsilon, 0) = 0$  and hence by definition  $I_m(x_{m+1}) = 0$ . In the second case,  $x_{m+1} \geq 2\epsilon$ , from the recurrence and the induction hypothe-

sis we have

$$\begin{aligned} I_m(x_{m+1}) &= m \int_0^{x_{m+1}} (\max(x_m - 2\epsilon, 0))^m \\ &\quad + \max(x_{m+1} - x_m - 2\epsilon, 0) \cdot x_m^{m-1} dx_m \\ &= m \int_{2\epsilon}^{x_{m+1}} (x_m - 2\epsilon)^m dx_m \\ &\quad + m \int_0^{x_{m+1} - 2\epsilon} (x_{m+1} - x_m - 2\epsilon) x_m^{m-1} dx_m \\ &= \frac{m}{m+1} (x_{m+1} - 2\epsilon)^{m+1} \\ &\quad + \frac{1}{m+1} (x_{m+1} - 2\epsilon)^{m+1} \\ &= (x_{m+1} - 2\epsilon)^{m+1}. \end{aligned}$$

To finish the proof of the theorem, suppose  $m < \frac{\ln(1/\delta)}{2.01\epsilon} - \frac{\ln 2}{2.01\epsilon}$ . Then  $\frac{1}{2}(1 - 2\epsilon)^m > \delta$ , since

$$\begin{aligned} \ln\left(\frac{1}{2}(1 - 2\epsilon)^m\right) &= \\ -\ln 2 + m \ln(1 - 2\epsilon) &> -\ln 2 - m(2.01\epsilon) > \ln \delta, \end{aligned}$$

where we have used that  $\ln(1 - 2\epsilon) > -2.01\epsilon$  for any  $\epsilon \in (0, 0.001)$ . Therefore from (6), for at least one target  $h = \mathbf{1}(-\infty, t]$ , with probability greater than  $\delta$ , algorithm  $A$  fails to output a hypothesis with error less than  $\epsilon$ .  $\square$

*Remark.* The  $\frac{\ln(1/\delta)}{2.01\epsilon} - O(\frac{1}{\epsilon})$  lower bound applies to supervised learning as well. However, we do not know of any supervised algorithm (deterministic or randomized) that has asymptotic sample complexity  $c \frac{\ln(1/\delta)}{\epsilon}$  for any constant  $c < 1$ . For example, the randomized algorithm that outputs with probability  $1/2$  the hypothesis  $\mathbf{1}(-\infty, \ell]$  and with probability  $1/2$  the hypothesis  $\mathbf{1}(-\infty, r]$  still cannot achieve the SSL sample complexity. We conjecture that all supervised algorithms for learning thresholds on real line in the realizable setting have asymptotic sample complexity at least  $\frac{\ln(1/\delta)}{\epsilon}$ .

### 6.3 Sample Complexity in Agnostic Case

In this section, we show that even in the agnostic setting SSL does not have more than constant factor improvement over SL. We prove some lower bounds for some classes over the real line. We introduce the notion of a  $b$ -shatterable distribution, which intuitively, are distributions where there are  $b$  ‘‘clusters’’ that can be shattered by the concept class. The main lower bound of this section are for such distributions (see Theorem 15). We show how this lower bound results in tight sample complexity bounds for two concrete problems. The first is learning thresholds on the real line where we show a bound of  $\Theta(\ln(1/\delta)/\epsilon^2)$ . Then we show sample complexity of  $\Theta\left(\frac{2d + \ln(1/\delta)}{\epsilon^2}\right)$  for the union of  $d$  intervals on the real line.

The sample complexity of the union of  $d$  intervals for a fixed distribution in a noisy setting has also been investigated by Gentile and Helmbold [11]. They show a lower bound of  $\Omega\left(2d \log \frac{1}{\Delta} / (\Delta(1 - 2\eta)^2)\right)$  where  $\Delta$  is the distance to the target that the learning algorithm should guarantee with

high probability, and  $\eta$  is the probability of a wrong label appearing (see classification noise model of [1]). This notation implies that the difference in true error of target and the algorithm's output is  $\epsilon = (1 - 2\eta)\Delta$ . Setting  $\eta = 1/2 - \epsilon/4$  gives  $\Omega(2d/\epsilon^2)$ . We note that we do not make the assumption of a constant level of noise for each unlabeled example. It turns out, however, that in our proofs we do construct worst case distributions that have a constant noise rate that is slightly below  $1/2$ .

We point out two main differences between our results and that of Gentile and Helmbold. The first being that we explicitly construct noisy distributions to obtain  $\epsilon^2$  in the denominator. The second difference is that our technique appears to be quite different from theirs, which uses an information theory approach, whereas we make use of known techniques based on lower bounding how well one can distinguish similar noisy distributions, and then applying an averaging argument. The main tools used in this section come from Anthony and Bartlett [2, Chapter 5].

We first cite a result on how many examples are needed to distinguish two similar, Bernoulli distributions in Lemma 11. Then in Lemma 12 we prove an analogue of this for arbitrary unlabeled distributions. The latter result is used to give us a lower bound in Theorem 15 for  $b$ -shatterable distributions (see Definition 14). Corollary 13 and 16 gives us tight sample complexity bounds for thresholds and union of intervals on  $\mathbb{R}$ .

**Lemma 11** (Anthony and Bartlett [2]). *Suppose that  $P$  is a random variable uniformly distributed on  $\{P_1, P_2\}$  where  $P_1, P_2$  are Bernoulli distributions over  $\{0, 1\}$  with  $P_1(1) = 1/2 - \gamma$  and  $P_2(1) = 1/2 + \gamma$  for  $0 < \gamma < 1/2$ . Suppose that  $\xi_1, \dots, \xi_m$  are IID  $\{0, 1\}$  valued random variables with  $\Pr(\xi_i = 1) = P(1)$  for each  $i$ . Let  $f$  be a function from  $\{0, 1\}^m \rightarrow \{P_1, P_2\}$ . Then*

$$\begin{aligned} \mathbb{E}_P \Pr_{\xi \sim P^m} [f(\xi) \neq P] &> \frac{1}{4} \left( 1 - \sqrt{1 - \exp\left(\frac{-4m\gamma^2}{1 - 4\gamma^2}\right)} \right) \\ &=: F(m, \gamma). \end{aligned}$$

One can view the lemma this way: if one randomly picks two weighted coins with similar biases, then there's a lower bound on the confidence with which one can accurately predict the coin that was picked.

The next result is similar except an unlabeled distribution  $D$  is fixed, and the distributions we want to distinguish will be extensions of  $D$ .

**Lemma 12.** *Fix any  $X, H, D$  over  $X$ , and  $m > 0$ . Suppose there exists  $h, g \in H$  with  $D(h\Delta g) > 0$ . Let  $P_h$  and  $P_g$  be the extension of  $D$  such that  $P_h((x, h(x))|x) = P_g((x, g(x))|x) = 1/2 + \gamma$ . Let  $A_D : (h\Delta g \times \{0, 1\})^m \rightarrow H$  be any function. Then for any  $x_1, \dots, x_m \in h\Delta g$ , there exists  $P \in \{P_h, P_g\}$  such that if  $y_i \sim P_{x_i}$  for all  $i$ ,*

$$\begin{aligned} \Pr_{y_i} [\text{Err}^P(A_D((x_1, y_1), \dots, (x_m, y_m))) - OPT_P \\ > \gamma D(h\Delta g)] > F(m, \gamma). \end{aligned}$$

Where  $P_x$  is the conditional distribution of  $P$  given  $x$ , and  $OPT_P = 1/2 - \gamma$ . Thus if the probability of failure is at

most  $\delta$ , we require

$$m \geq \left( \frac{1}{4\gamma^2} - 1 \right) \ln \frac{1}{8\delta}. \quad (9)$$

*Proof.* Suppose for a contradiction this is not true. Let  $\mathcal{P} = \{P_h, P_g\}$ . Then there exists an  $A_D$  and  $x_1, \dots, x_m$  such that

$$\begin{aligned} \forall P \in \mathcal{P}, \Pr_{y_i} [\text{Err}^P(A_D((x_1, y_1), \dots, (x_m, y_m))) - OPT_P \\ > \gamma D(h\Delta g)] \leq F(m, \gamma). \quad (10) \end{aligned}$$

Then we will show that the lower bound in Lemma 11 can be violated. Now  $h\Delta g$  can be partitioned into  $\Delta_0 = \{x : h(x) = 0\}$  and  $\Delta_1 = \{x : h(x) = 1\}$ . Without loss of generality assume  $\{x_1, \dots, x_l\} \subseteq \Delta_0$  and  $\{x_{l+1}, \dots, x_m\} \subseteq \Delta_1$ . Let  $A = A_D((x_1, y_1), \dots, (x_m, y_m))$ .

From the triangle inequality  $D(A\Delta h) + D(A\Delta g) \geq D(h\Delta g)$ . Thus if  $A$  is closer to  $h$  then  $D(A\Delta g) \geq D(h\Delta g)/2$  and vice versa. Let  $P$  be a random variable uniformly distributed on  $\mathcal{P}$ . We have  $\Pr(y_1 = 1) = \dots = \Pr(y_l = 1) = P_{\Delta_0}(1) = \Pr(y_{l+1} = 0) = \dots = \Pr(y_m = 0) = P_{\Delta_1}(0)$ .

Let  $\xi_1, \dots, \xi_m \sim P_{\Delta_0}$  so that  $\Pr(\xi_i = 1) = 1/2 - \gamma$  when  $P = P_h$  and equal to  $1/2 + \gamma$  when  $P = P_g$ . Let us define the function  $f : \{0, 1\}^m \rightarrow \mathcal{P}$  as follows. It will take as input  $\xi_1, \dots, \xi_m$  then transform this to an input of  $A_D$  as  $I = (x_1, \xi_1), \dots, (x_l, \xi_l), (x_{l+1}, 1 - \xi_{l+1}), \dots, (x_m, 1 - \xi_m)$  so that  $\xi_i$  and  $1 - \xi_j$  is from the same distribution as  $y_i$  and  $y_j$ , respectively, for  $i \leq l, j > l$ . Now define

$$f(\xi_1, \dots, \xi_l) = \begin{cases} P_h & \text{if } D(A_D(I)\Delta h) < D(A_D(I)\Delta g) \\ P_g & \text{otherwise} \end{cases}.$$

We have

$$\begin{aligned} \mathbb{E}_P \Pr_{\xi \sim P_{\Delta_0}^m} [f(\xi) \neq P] \\ &\leq \mathbb{E}_P \Pr_{\xi} [D(A_D(I)\Delta OPT_P) > D(h\Delta g)/2] \\ &\leq \mathbb{E}_P \Pr_{\xi} [\text{Err}^P(A_D(I)) - OPT_P > \gamma D(h\Delta g)] \\ &\leq F(m, \gamma) \end{aligned}$$

where the last inequality follows from (10). This is a contradiction, so the lower bound from Lemma 11 must apply. If the probability of failure  $F(m, \gamma)$  is at most  $\delta$ , solving the inequality for  $m$  gives (9).  $\square$

**Corollary 13.** *The SSL sample complexity of learning thresholds over the uniform distribution over  $(0, 1)$  is  $\Theta(\ln(1/\delta)/\epsilon^2)$ .*

*Proof.* Upper bound comes from any ERM algorithm. Let  $h = \mathbf{1}(-\infty, 0]$  and  $g = \mathbf{1}(-\infty, 1]$  so  $D(h\Delta g) = 1$ . Set  $\gamma = \epsilon$  as in Lemma 12.  $\square$

**Definition 14.** *The triple  $(X, H, D)$  is  $b$ -shatterable if there exists disjoint sets  $C_1, C_2, \dots, C_b$  with  $D(C_i) = 1/b$  for each  $i$ , and for each  $S \subseteq \{1, 2, \dots, b\}$ , there exists  $h \in H$  such that*

$$h \cap \left( \bigcup_{i=1}^b C_i \right) = \bigcup_{i \in S} C_i.$$

**Theorem 15.** *If  $(X, H, D)$  is  $b$ -shatterable and  $H$  contains  $h, g$  with  $D(h\Delta g) = 1$  then a lower bound on the SSL sample complexity for  $0 < \epsilon, \delta < 1/64$  is*

$$\Omega\left(\frac{b + \ln \frac{1}{\delta}}{\epsilon^2}\right).$$

*Proof.* The proof is similar to Theorem 5.2 in Anthony and Bartlett [2]. Let  $G = \{h_1, h_2, \dots, h_{2^b}\}$  be the class of functions that  $b$ -shatters  $D$  with respect to  $C = \{C_1, \dots, C_b\}$ . We construct noisy extensions of  $D$ ,  $\mathcal{P} = \{P_1, P_2, \dots, P_{2^b}\}$  so that for each  $i$ ,  $P_i((x, h_i(x))) = (1 + 2\gamma)/(2b)$ . For any  $h \in H$  let  $\text{snap}(h) = \text{argmin}_{h' \in G} D(h\Delta h')$ . Suppose  $P \in \mathcal{P}$ , let  $h^*$  denote the optimal classifier which is some  $g \in G$  depending on the choice of  $P$ . If  $i \neq j$  and  $N(h_i, h_j)$  is the number of sets in  $C$  where  $h_i$  and  $h_j$  disagree, then  $D(h_i\Delta h_j) \geq N(h_i, h_j)/b$ , and since  $G$  is a  $1/b$ -packing,

$$\begin{aligned} \text{Err}^P(h) &\geq \text{Err}^P(h^*) + \frac{\gamma}{b} N(\text{snap}(h), h^*) \\ &= \frac{1}{2} (\text{Err}^P(\text{snap}(h)) + \text{Err}^P(h^*)). \end{aligned} \quad (11)$$

Modifying the proof of Anthony and Bartlett with the use of Lemma 12 rather than Lemma 11 we get that there exists a  $P \in \mathcal{P}$  such that whenever  $m \leq b/(320\epsilon^2)$ ,

$$\Pr_{S \sim P^m} [\text{Err}^P(\text{snap}(A(D, S))) - \text{Err}^P(h^*) > 2\epsilon] > \delta.$$

Whenever  $A$  fails, we get from (11)

$$\begin{aligned} \text{Err}^P(A(D, S)) - \text{Err}^P(h^*) \\ \geq \frac{1}{2} (\text{Err}^P(\text{snap}(h)) + \text{Err}^P(h^*)) \geq \epsilon. \end{aligned}$$

To get  $\Omega(\ln(1/\delta)/\epsilon^2)$ , apply Lemma 12 with  $h$  and  $g$ .  $\square$

We will now apply the above theorem to give the sample complexity for learning union of intervals on the real line. Recall that by the rescaling trick, we only need to consider the sample complexity with respect to the uniform distribution on  $(0, 1)$ .

**Corollary 16.** *The SSL sample complexity for learning the class of union of at most  $d$  intervals  $UI_d = \{(a_1, a_2) \cup \dots \cup [a_{2l-1}, a_{2l}] : l \leq d, 0 \leq a_1 \leq a_2 \leq \dots \leq a_{2l} \leq 1\}$  over uniform distribution on  $(0, 1)$  is*

$$\Theta\left(\frac{2d + \ln \frac{1}{\delta}}{\epsilon^2}\right).$$

*Proof.* We have  $\text{VC}(UI_d) = 2d$ , thus the upper bound follows immediately. Construct  $2d$ -shatterable sets by letting  $C_i = [(i-1)/2d, i/2d]$  for  $i = 1, \dots, 2d$ . For any  $S \subseteq \{1, \dots, 2d\}$  define  $h_S = \bigcup_{i \in S} C_i$ . Now if  $|S| \leq d$  then clearly  $h_S \in UI_d$ , if  $|S| > d$  then  $h_{\bar{S}} \in UI_d$  since  $|\bar{S}| < d$ . But then  $[0, 1] \setminus h_{\bar{S}}$  can be covered by at most  $d$  intervals, so  $h_S \in UI_d$ . Thus the set  $\{h_S : S \subseteq \{1, \dots, 2d\}\}$   $2d$ -shatters  $D$  on  $[0, 1]$ . Also let  $h = [0, 0] = \emptyset$  and  $g = [0, 1]$ . Now apply Theorem 15 for the bound.  $\square$

## 6.4 No Optimal Semi-Supervised Algorithm

One could imagine a different formulation of the comparison between SL and SSL paradigms. For example, one might ask naively whether, for given class  $H$ , there is a semi-supervised algorithm  $A$ , such that for any supervised algorithm  $B$ , and any  $\epsilon, \delta$ , on any probability distribution  $P$  the sample complexity of  $A$  is no higher than the sample complexity of  $B$ . The answer to the question is easily seen to be negative, because for any  $P$  there exists a supervised learning algorithm  $B_P$  that ignores the labeled examples and simply outputs hypothesis  $h \in H$  with minimum error  $\text{Err}^P(h)$  (or even Bayes optimal classifier for  $P$ ). On  $P$  the sample complexity of  $B_P$  is zero, unfortunately, on  $P'$ , sufficiently different from  $P$ , the sample complexity of  $B_P$  is infinite.

One might disregard algorithms such as  $B_P$  and ask the same question as above, except that one quantifies over only the subset of algorithms that on *any* distribution over  $X \times \{0, 1\}$  have sample complexity that is polynomial in  $1/\epsilon$  and  $\ln(1/\delta)$ . Such algorithms are often called PAC (Probably Approximately Correct). The following theorem demonstrates that such restriction does not help and the answer to the question is still negative.

**Theorem 17.** *Let  $H = \{\mathbf{1}(-\infty, t] : t \in \mathbb{R}\}$  be the class of thresholds over the real line. For any absolutely continuous distribution  $D$  (with respect to Lebesgue measure on  $\mathbb{R}$ ), any semi-supervised algorithm  $A$ , any  $\epsilon > 0$  and  $\delta \in (0, \frac{1}{2})$ , there exists a distribution  $P \in \text{Ext}(D)$  and a supervised PAC learning algorithm  $B$  such that*

$$m(A, H, P, \epsilon, \delta) > m(B, H, P, \epsilon, \delta).$$

*Proof.* Fix any  $A, D$  and  $m$ . Let  $L$  be the algorithm that chooses the left most empirical error minimizer, that is, on a sample  $S$ ,  $L$  outputs  $\mathbf{1}(-\infty, \ell]$ , where

$$\ell = \inf \left\{ t \in \mathbb{R} : \text{Err}^S(\mathbf{1}(-\infty, t]) = \min_{h' \in H} \text{Err}^S(h') \right\}.$$

For any  $h \in H$  we also define algorithm  $L_h$ , which outputs  $h$  if  $\text{Err}^S(h) = 0$ , and otherwise  $L_h$  outputs  $L(S)$ . First, note that  $L \equiv L_{\emptyset}$ . Second, for any  $h$ ,  $L_h$  outputs a hypothesis that minimizes empirical error, and since  $\text{VC}(H) = 1$ , it is a PAC algorithm. Third, clearly the sample complexity of  $L_h$  on  $D_h$  is zero (regardless of  $\epsilon$  and  $\delta$ ).

Theorem 10 shows that there exists  $h \in H$  such that the sample complexity of  $A$  on  $D_h$  is positive, in fact, it is increasing as  $\epsilon$  and  $\delta$  approach zero. Thus there exists supervised algorithm  $B = L_h$  with lower sample complexity than  $A$ .  $\square$

## 7 Conclusion

We provide a formal analysis of the sample complexity of semi-supervised learning compared to that of learning from labeled data only. We focus on bounds that do not depend on assumptions concerning the relationship between the labels and unlabeled data distribution.

Our main conclusion is that in such a setting semi-supervised learning has limited advantage. Formally, we show that for basic concept classes over the real line this advantage is never more than a constant factor of the sample size. We believe that this phenomena applies much more widely.

We also briefly address the error bounds under common assumptions on the relationship between unlabeled data and the labels. We demonstrate that even when such assumptions apply common SSL paradigms may be inferior to standard empirical risk minimization. We conclude that prior beliefs like the cluster assumption should be formulated more precisely to reflect the known practical merits of SSL. This discussion highlights a dire deficiency in current approach to semi-supervised learning; common assumptions about these labels-unlabeled structure relationships do not offer any method for reliably checking if they hold (in any given learning problem).

The paper calls attention to, and formalizes, some natural fundamental questions about the theory-practice gap concerning semi-supervised learning. The major open question we raise is whether any semi-supervised learning algorithm can achieve sample size guarantees that are unattainable without access to unlabeled data. This is formalized in Conjectures 5 and 4.

**Acknowledgements.** We like to thank Nati (Nathan) Srebro and Vitaly Feldman for useful discussions.

## References

- [1] Dana Angluin and Philip D. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1987.
- [2] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, January 1999.
- [3] Maria-Florina Balcan and Avrim Blum. A PAC-style model for learning from labeled and unlabeled data. In *Proceedings of 18th Annual Conference on Learning Theory 2005*, pages 111–126. Springer, 2005.
- [4] Maria-Florina Balcan and Avrim Blum. An augmented PAC model for semi-supervised learning. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, chapter 21, pages 61–89. MIT Press, September 2006.
- [5] Gyora M. Benedek and Alon Itai. Learnability with respect to fixed distributions. *Theoretical Computer Science*, 86(2):377–389, 1991.
- [6] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.
- [7] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. MIT Press, September 2006.
- [8] Fabio Cozman and Ira Cohen. Risks of semi-supervised learning: How unlabeled data can degrade performance of generative classifiers. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, chapter 4, pages 57–72. MIT Press, September 2006.
- [9] Ran El-Yaniv and Dmitry Pechyony. Stable transductive learning. In *COLT*, pages 35–49, 2006.
- [10] Ran El-Yaniv and Dmitry Pechyony. Transductive rademacher complexity and its applications. In *COLT*, pages 157–171, 2007.
- [11] Claudio Gentile and David P. Helmbold. Improved lower bounds for learning from noisy examples: and information-theoretic approach. In *Proceedings of COLT 1998*, pages 104–115. ACM, 1998.
- [12] Matti Kääriäinen. Generalization error bounds using unlabeled data. In *Proceedings of COLT 2005*, pages 127–142. Springer, 2005.
- [13] P. Rigollet. Generalization error bounds in semi-supervised classification under the cluster assumption. *The Journal of Machine Learning Research*, 8:1369–1392, 2007.
- [14] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.
- [15] Vladimir N. Vapnik. Transductive inference and semi-supervised learning. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, chapter 24, pages 453–472. MIT Press, September 2006.

---

# The True Sample Complexity of Active Learning

---

**Maria-Florina Balcan**

Computer Science Department  
Carnegie Mellon University  
ninamf@cs.cmu.edu

**Steve Hanneke**

Machine Learning Department  
Carnegie Mellon University  
shanneke@cs.cmu.edu

**Jennifer Wortman**

Computer and Information Science  
University of Pennsylvania  
wortmanj@seas.upenn.edu

## Abstract

We describe and explore a new perspective on the sample complexity of active learning. In many situations where it was generally believed that active learning does not help, we show that active learning does help in the limit, often with exponential improvements in sample complexity. This contrasts with the traditional analysis of active learning problems such as non-homogeneous linear separators or depth-limited decision trees, in which  $\Omega(1/\epsilon)$  lower bounds are common. Such lower bounds should be interpreted carefully; indeed, we prove that it is always possible to learn an  $\epsilon$ -good classifier with a number of samples asymptotically smaller than this. These new insights arise from a subtle variation on the traditional definition of sample complexity, not previously recognized in the active learning literature.

## 1 Introduction

Machine learning research has often focused on the problem of learning a classifier from labeled examples sampled independent from the particular learning algorithm that is used. However, for many contemporary practical problems such as classifying web pages or detecting spam, there is often an abundance of *unlabeled* data available, from which a relatively small subset is selected to be labeled and used for learning. In such scenarios, the question arises of how to select that subset of examples to be labeled.

One possibility, which has recently been generating substantial interest, is *active learning*. In active learning, the learning algorithm itself is allowed to select the subset of unlabeled examples to be labeled. It does this sequentially (i.e., interactively), using the requested label information from previously selected examples to inform its decision of which example to select next. The hope is that by only requesting the labels of informative examples, the algorithm can learn a good classifier using significantly fewer labels than would be required if the labeled set were sampled at random.

A number of active learning analyses have recently been proposed in a PAC-style setting, both for the realizable and for the agnostic cases, resulting in a sequence of important positive and negative results [6, 7, 8, 2, 10, 4, 9, 13, 12].

In particular, the most concrete noteworthy positive result for when active learning helps is that of learning homogeneous (i.e., through the origin) linear separators, when the data is linearly separable and distributed uniformly over the unit sphere, and this example has been extensively analyzed [8, 2, 10, 4, 9]. However, few other positive results are known, and there are simple (almost trivial) examples, such as learning intervals or non-homogeneous linear separators under the uniform distribution, where previous analyses of sample complexities have indicated that perhaps active learning does not help at all [8].

In this work, we approach the analysis of active learning algorithms from a different angle. Specifically, we point out that traditional analyses have studied the number of label requests required before an algorithm can both produce an  $\epsilon$ -good classifier *and* prove that the classifier's error is no more than  $\epsilon$ . These studies have turned up simple examples where this number is no smaller than the number of random labeled examples required for passive learning. This is the case for learning certain nonhomogeneous linear separators and intervals on the real line, and generally seems to be a common problem for many learning scenarios. As such, it has led some to conclude that active learning *does not help* for most learning problems. One of the goals of our present analysis is to dispel this misconception. Specifically, we study the number of labels an algorithm needs to request before it can produce an  $\epsilon$ -good classifier, even if there is no accessible confidence bound available to verify the quality of the classifier. With this type of analysis, we prove that active learning can essentially always achieve asymptotically superior sample complexity compared to passive learning when the VC dimension is finite. Furthermore, we find that for most natural learning problems, including the negative examples given in the previous literature, active learning can achieve exponential<sup>1</sup> improvements over passive learning with respect to dependence on  $\epsilon$ . This situation is characterized in Figure 1.1.

### 1.1 A Simple Example: Unions of Intervals

To get some intuition about when these types of sample complexity are different, consider the following example. Suppose that  $C$  is the class of all intervals over  $[0, 1]$  and  $D$  is

---

<sup>1</sup>We slightly abuse the term “exponential” throughout the paper. In particular, we refer to any  $\text{polylog}(1/\epsilon)$  as being an exponential improvement over  $1/\epsilon$ .

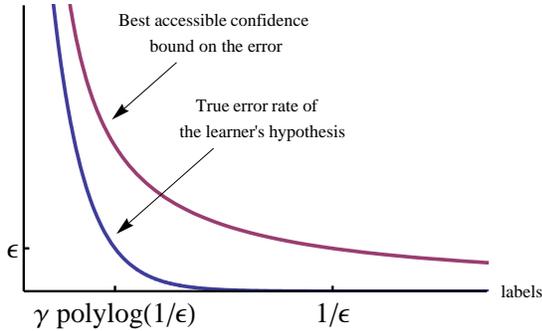


Figure 1.1: Active learning can often achieve exponential improvements, though in many cases the amount of improvement cannot be detected from information available to the learning algorithm. Here  $\gamma$  may be a target-dependent constant.

a uniform distribution over  $[0, 1]$ . If the target function is the empty interval, then for any sufficiently small  $\epsilon$ , in order to *verify* with high confidence that this (or any) interval has error  $\leq \epsilon$ , we need to request labels in at least a constant fraction of the  $\Omega(1/\epsilon)$  intervals  $[0, \epsilon], [\epsilon, 2\epsilon], \dots$ , requiring  $\Omega(1/\epsilon)$  total label requests.

However, no matter what the target function is, we can *find* an  $\epsilon$ -good classifier with only a logarithmic sample complexity via the following extremely simple 2-phase learning algorithm. We start with a large ( $\Omega(1/\epsilon)$ ) set of unlabeled examples. In the first phase, on each round we choose a point  $x$  uniformly at random from the unlabeled sample and query its label. We repeat this until we observe the first  $+1$  label, at which point we enter the second phase. In the second phase, we alternate between running one binary search on the examples between  $0$  and that  $x$  and a second on the examples between that  $x$  and  $1$  to approximate the end-points of the interval. At the end, we output a smallest interval consistent with the observed positive labels.

If the target  $h^*$  labels every point as  $-1$  (the so-called *all-negative* function), the algorithm described above would output a hypothesis with  $0$  error even after  $0$  label requests. On the other hand, if the target is an interval  $[a, b] \subseteq [0, 1]$ , where  $b - a = w > 0$ , then after roughly  $O(1/w)$  queries (a constant number that depends only on the target), a positive example will be found. Since only  $O(\log(1/\epsilon))$  queries are required to run the binary search to reach error rate  $\epsilon$ , the sample complexity is at worst logarithmic in  $1/\epsilon$ . Thus, we see a sharp distinction between the sample complexity required to *find* a good classifier (logarithmic) and the sample complexity needed to both find a good classifier *and verify* that it is good.

This example is particularly simple, since there is effectively only *one* “hard” target function (the all-negative target). However, most of the spaces we study are significantly more complex than this, and there are generally many targets for which it is difficult to achieve good verifiable complexity.

**Our Results:** We show that in many situations where it was previously believed that active learning cannot help, active learning does help in the limit. Our main specific contri-

butions are as follows:

- We distinguish between two different variations on the definition of sample complexity. The traditional definition, which we refer to as *verifiable sample complexity*, focuses on the number of label requests needed to obtain a confidence bound indicating an algorithm has achieved at most  $\epsilon$  error. The newer definition, which we refer to simply as *sample complexity*, focuses on the number of label requests before an algorithm actually achieves at most  $\epsilon$  error. We point out that the latter is often significantly smaller than the former, in contrast to passive learning where they are often equivalent up to constants for most nontrivial learning problems.
- We prove that *any* distribution and finite VC dimension concept class has active learning sample complexity asymptotically smaller than the sample complexity of passive learning for nontrivial targets. A simple corollary of this is that finite VC dimension implies  $o(1/\epsilon)$  active learning sample complexity.
- We show it is possible to actively learn with an *exponential rate* a variety of concept classes and distributions, many of which are known to require a linear rate in the traditional analysis of active learning: for example, intervals on  $[0, 1]$  and non-homogeneous linear separators under the uniform distribution.
- We show that even in this new perspective, there do exist lower bounds; it is possible to exhibit somewhat contrived distributions where exponential rates are not achievable even for some simple concept spaces (see Theorem 12). The learning problems for which these lower bounds hold are much more intricate than the lower bounds from the traditional analysis, and intuitively seem to represent the core of what makes a hard active learning problem.

## 2 Background and Notation

Let  $\mathcal{X}$  be an instance space and  $\mathcal{Y} = \{-1, 1\}$  be the set of possible labels. Let  $C$  be the hypothesis class, a set of measurable functions mapping from  $\mathcal{X}$  to  $\mathcal{Y}$ , and assume that  $C$  has VC dimension  $d$ . We consider here the realizable setting in which it is assumed that the instances are labeled by a target function  $h^*$  in the class  $C$ . The *error rate* of a hypothesis  $h$  with respect to a distribution  $D$  over  $\mathcal{X}$  is defined as  $\text{er}(h) = \mathbb{P}_D(h(x) \neq h^*(x))$ .

We assume the existence of an infinite sequence  $x_1, x_2, \dots$  of examples sampled i.i.d. according to  $D$ . The learning algorithm may access any finite initial segment  $x_1, x_2, \dots, x_m$ . Essentially, this means we allow the algorithm access to an arbitrarily large, but finite, sequence of random unlabeled examples. In active learning, the algorithm can select any example  $x_i$  and request the label  $h^*(x_i)$  that the target assigns to that example, observing the labels of all previous requests before selecting the next example to query. The goal is to find a hypothesis  $h$  with small error with respect to  $D$ , while simultaneously minimizing the number of label requests that the learning algorithm makes.

## 2.1 Two Definitions of Sample Complexity

The following definitions present a subtle but significant distinction we refer to throughout the paper. Several of the results that follow highlight situations where these two definitions of sample complexity can have dramatically different dependence on  $\epsilon$ .

**Definition 1** A function  $S(\epsilon, \delta, h^*)$  is a verifiable sample complexity for a pair  $(C, D)$  if there exists an active learning algorithm  $A(t, \delta)$  that outputs both a classifier  $h_t$  and a value  $\hat{\epsilon}_t \in \mathbb{R}$  after making at most  $t$  label requests, such that for any target function  $h^* \in C$ ,  $\epsilon \in (0, 1/2)$ ,  $\delta \in (0, 1/4)$ , for any  $t \geq S(\epsilon, \delta, h^*)$ ,

$$\mathbb{P}_D(\text{er}(h_t) \leq \hat{\epsilon}_t \leq \epsilon) \geq 1 - \delta.$$

**Definition 2** A function  $S(\epsilon, \delta, h^*)$  is a sample complexity for a pair  $(C, D)$  if there exists an active learning algorithm  $A(t, \delta)$  that outputs a classifier  $h_t$  after making at most  $t$  label requests, such that for any target function  $h^* \in C$ ,  $\epsilon \in (0, 1/2)$ ,  $\delta \in (0, 1/4)$ , for any  $t \geq S(\epsilon, \delta, h^*)$ ,

$$\mathbb{P}_D(\text{er}(h_t) \leq \epsilon) \geq 1 - \delta.$$

Note that both types of sample complexity can be target-dependent and distribution-dependent. The only distinction is whether or not there is an accessible guarantee on the error of the chosen hypothesis that is also at most  $\epsilon$ . This confidence bound can only depend on quantities accessible to the learning algorithm, such as the  $t$  requested labels. Thus, any verifiable sample complexity function is also a sample complexity function, but we study a variety of cases where the reverse is not true. In situations where there are sample complexity functions significantly smaller than any achievable verifiable sample complexities, we sometimes refer to the smaller quantity as the *true sample complexity* to distinguish it from the verifiable sample complexity.

A common alternative formulation of verifiable sample complexity is to let  $A$  take  $\epsilon$  as an argument and allow it to choose online how many label requests it needs in order to guarantee error at most  $\epsilon$  [8]. This alternative definition is essentially equivalent (either definition can be reduced to the other without significant loss), as the algorithm must be able to produce a confidence bound of size at most  $\epsilon$  on the error of its hypothesis in order to decide when to stop requesting labels anyway.<sup>2</sup>

## 2.2 The Verifiable Sample Complexity

To date, there has been a significant amount of work studying the verifiable sample complexity (though typically under the aforementioned alternative formulation). It is clear from standard results in passive learning that verifiable sample complexities of  $O((d/\epsilon) \log(1/\epsilon) + (1/\epsilon) \log(1/\delta))$  are

<sup>2</sup>There is some question as to what the “right” formal model of active learning is in general. For instance, we could instead let  $A$  generate an infinite sequence of  $h_t$  hypotheses (or  $(h_t, \hat{\epsilon}_t)$  in the verifiable case), where  $h_t$  can depend only on the first  $t$  label requests made by the algorithm along with some initial segment of unlabeled examples (as in [5]), representing the case where we are not sure a-priori of when we will stop the algorithm. However, for our present purposes, such alternative models are equivalent in sample complexity up to constants.

easy to obtain for any learning problem, by requesting the labels of random examples. As such, there has been much interest in determining when it is possible to achieve verifiable sample complexity *smaller* than this, and in particular, when the verifiable sample complexity is a polylogarithmic function of  $1/\epsilon$  (representing exponential improvements over passive learning).

One of the earliest active learning algorithms in this model is the selective sampling algorithm of Cohn, Atlas, and Ladner [6], henceforth referred to as CAL. This algorithm keeps track of two spaces—the current *version space*  $C_i$ , defined as the set of hypotheses in  $C$  consistent with all labels revealed so far, and the current *region of uncertainty*  $R_i = \{x \in \mathcal{X} : \exists h_1, h_2 \in C_i \text{ s.t. } h_1(x) \neq h_2(x)\}$ . In each round  $i$ , the algorithm picks a random unlabeled example from  $R_i$  and requests its label, eliminating all hypotheses in  $C_i$  inconsistent with the received label to make the next version space  $C_{i+1}$ . The algorithm then defines  $R_{i+1}$  as the region of uncertainty for the new version space  $C_{i+1}$  and continues. Its final hypothesis can then be taken arbitrarily from  $C_t$ , the final version space, and we use the diameter of  $C_t$  for the  $\hat{\epsilon}_t$  error bound.

While there are a small number of cases in which this algorithm and others have been shown to achieve exponential improvements in the verifiable sample complexity for all targets (most notably, the case of homogeneous linear separators under the uniform distribution), there exist extremely simple concept classes for which  $\Omega(1/\epsilon)$  labels are needed for some targets. For example, consider the class of intervals in  $[0, 1]$  under the uniform distribution. In order to distinguish the all-negative target from the set of hypotheses that are positive on a region of weight  $\epsilon$  and make a high probability guarantee,  $\Omega(1/\epsilon)$  labeled examples are needed [8].

Recently, there have been a few quantities proposed to measure the verifiable sample complexity of active learning on any given concept class and distribution. Dasgupta’s *splitting index* [8], which is dependent on the concept class, data distribution, target function, and a parameter  $\tau$ , quantifies how easy it is to make progress toward reducing the diameter of the version space by choosing an example to query. Another quantity to which we will frequently refer is Hanneke’s *disagreement coefficient* [12], defined as follows.

**Definition 3** For any  $h \in C$  and  $r > 0$ , let  $B(h, r)$  be a ball of radius  $r$  around  $h$  in  $C$ . That is,

$$B(h, r) = \{h' \in C : \mathbb{P}_D(h(x) \neq h'(x)) \leq r\}.$$

For any hypothesis class  $C$ , define the region of disagreement as

$$\text{DIS}(C) = \{x \in \mathcal{X} : \exists h_1, h_2 \in C : h_1(x) \neq h_2(x)\}.$$

Additionally, let  $\bar{C}$  denote any countable dense subset of  $C$ .<sup>3</sup> For our purposes, the disagreement coefficient of a hypothesis  $h$ , denoted  $\theta_h$ , is defined as

$$\theta_h = \sup_{r>0} \frac{\mathbb{P}(\text{DIS}(\bar{B}(h, r)))}{r}.$$

<sup>3</sup>That is,  $\bar{C}$  is countable and  $\forall h \in C, \forall \epsilon > 0, \exists h' \in \bar{C} : \mathbb{P}(h(X) \neq h'(X)) \leq \epsilon$ . Such a subset exists, for example, in any  $C$  with finite VC dimension. We introduce this countable dense subset to avoid certain degenerate behaviors, such as when  $\text{DIS}(B(h, 0)) = \mathcal{X}$ .

The disagreement coefficient for a concept space  $C$  is defined as  $\theta = \sup_{h \in C} \theta_h$ .

The disagreement coefficient is often a useful quantity for analyzing the verifiable sample complexity of active learning algorithms. For example, it has been shown that the algorithm of Cohn, Atlas, and Ladner described above achieves a verifiable sample complexity at most  $\theta_{h^*} d \cdot \text{polylog}(1/(\epsilon\delta))$  when run with concept class  $\bar{C}$  for target function  $h^* \in C$  [12]. We will see that both the disagreement coefficient and splitting index are also useful quantities for analyzing true sample complexities, though their use in that case is less direct.

### 2.3 The True Sample Complexity

This paper focuses on situations where true sample complexities are significantly smaller than verifiable sample complexities. In particular, we show that many common pairs  $(C, D)$  have sample complexity that is polylogarithmic in both  $1/\epsilon$  and  $1/\delta$  and linear only in some finite target-dependent constant  $\gamma_{h^*}$ . This contrasts sharply with the infamous  $1/\epsilon$  lower bounds mentioned above, which have been identified for verifiable sample complexity. The implication is that, for any fixed target  $h^*$ , such lower bounds vanish as  $\epsilon$  approaches 0. This also contrasts with passive learning, where  $1/\epsilon$  lower bounds are typically unavoidable [1].

**Definition 4** We say that  $(C, D)$  is actively learnable at an exponential rate if there exists an active learning algorithm achieving sample complexity

$$S(\epsilon, \delta, h^*) = \gamma_{h^*} \cdot \text{polylog}(1/(\epsilon\delta))$$

for some finite  $\gamma_{h^*} = \gamma(h^*, D)$  independent of  $\epsilon$  and  $\delta$ .

## 3 Strict Improvements of Active Over Passive

In this section, we describe conditions under which active learning can achieve a sample complexity asymptotically superior to passive learning. The results are surprisingly general, indicating that whenever the VC dimension is finite, essentially any passive learning algorithm is asymptotically dominated by an active learning algorithm on all targets.

**Definition 5** A function  $S(\epsilon, \delta, h^*)$  is a passive learning sample complexity for a pair  $(C, D)$  if there exists an algorithm  $A(((x_1, h^*(x_1)), (x_2, h^*(x_2)), \dots, (x_t, h^*(x_t))), \delta)$  that outputs a classifier  $h_t$ , such that for any target function  $h^* \in C$ ,  $\epsilon \in (0, 1/2)$ ,  $\delta \in (0, 1/4)$ , for any  $t \geq S(\epsilon, \delta, h^*)$ ,

$$\mathbb{P}_D(\text{er}(h_t) \leq \epsilon) \geq 1 - \delta.$$

Thus, a passive learning sample complexity corresponds to a restriction of an active learning sample complexity to algorithms that specifically request the first  $t$  labels in the sequence and ignore the rest. In particular, it is known that for any finite VC dimension class, there is always an  $O(1/\epsilon)$  passive learning sample complexity [14]. Furthermore, this is often tight (though not always), in the sense that for any passive algorithm, there exist targets for which the corresponding passive learning sample complexity is  $\Omega(1/\epsilon)$  [1]. The following theorem states that for any passive learning sample complexity, there exists an achievable active learning sample complexity with a strictly slower asymptotic rate of growth. Its proof is included in Appendix D.

**Theorem 6** Suppose  $C$  has finite VC dimension, and let  $D$  be any distribution on  $\mathcal{X}$ . For any passive learning sample complexity  $S_p(\epsilon, \delta, h)$  for  $(C, D)$ , there exists an active learning algorithm achieving a sample complexity  $S_a(\epsilon, \delta, h)$  such that, for all targets  $h \in C$  for which  $S_p(\epsilon, \delta, h) = \omega(1)$ ,<sup>4</sup>

$$S_a(\epsilon, \delta, h) = o(S_p(\epsilon/4, \delta, h)).$$

In particular, this implies the following simple corollary.

**Corollary 7** For any  $C$  with finite VC dimension, and any distribution  $D$  over  $\mathcal{X}$ , there is an active learning algorithm that achieves a sample complexity  $S(\epsilon, \delta, h)$  such that

$$S(\epsilon, \delta, h) = o(1/\epsilon)$$

for all targets  $h \in C$ .

**Proof:** Let  $d$  be the VC dimension of  $C$ . The passive learning algorithm of Haussler, Littlestone & Warmuth [14] is known to achieve a sample complexity no more than  $(kd/\epsilon) \log(1/\delta)$ , for some universal constant  $k < 200$  [14]. Applying Theorem 6 now implies the result. ■

Note the interesting contrast, not only to passive learning, but also to the known results on the verifiable sample complexity of active learning. This theorem definitively states that the  $\Omega(1/\epsilon)$  lower bounds common in the literature on verifiable sample complexity can never arise in the analysis of the true sample complexity of finite VC dimension classes.

## 4 Composing Hypothesis Classes

Recall the simple example of learning the class of intervals over  $[0, 1]$  under the uniform distribution. It is well known that the verifiable sample complexity of the “all-negative” classifier in this class is  $\Omega(1/\epsilon)$ . However, consider the more limited class  $C_1 \subset C$  containing only the intervals  $h$  with  $w(h) = \mathbb{P}(h(X) = +1) > 0$ . Using the simple algorithm described in Section 1.1, this restricted class can be learned with a (verifiable) sample complexity of only  $O(1/w(h) + \log(1/\epsilon))$ . Furthermore, the remaining set of classifiers  $C_2 = C \setminus C_1$  (which consists of only the all-negative classifier) has sample complexity 0. Thus,  $C = C_1 \cup C_2$ , and both  $(C_1, D)$  and  $(C_2, D)$  are learnable at an exponential rate.

It turns out that it is often convenient to view concept classes in terms of such well-constructed, possibly infinite sequences of subsets. Generally, given a distribution  $D$  and a function class  $C$ , suppose we can construct a sequence of subclasses,  $C_1, C_2, \dots$ , where  $C = \cup_{i=1}^{\infty} C_i$ , such that it is possible to actively learn any subclass  $C_i$  with only

<sup>4</sup>Recall that we say a non-negative function  $\phi(\epsilon) = o(1/\epsilon)$  iff  $\lim_{\epsilon \rightarrow 0} \phi(\epsilon)/(1/\epsilon) = 0$ . Similarly,  $\phi(\epsilon) = \omega(1)$  iff  $\lim_{\epsilon \rightarrow 0} 1/\phi(\epsilon) = 0$ . Here and below, the  $o(\cdot)$ ,  $\omega(\cdot)$ ,  $\Omega(\cdot)$  and  $O(\cdot)$  notation should be interpreted as  $\epsilon \rightarrow 0$  (from the + direction), treating all other parameters (e.g.,  $\delta$  and  $h^*$ ) as fixed constants. Note that any algorithm achieving a sample complexity  $S_p(\epsilon, \delta, h) \neq \omega(1)$  is guaranteed, with probability  $\geq 1 - \delta$ , to achieve error zero using a finite number of samples, and therefore we cannot hope to achieve a slower asymptotic growth in sample complexity.

$S_i(\epsilon, \delta, h)$  sample complexity. Thus, if we know that the target  $h^*$  is in  $C_i$ , it is straightforward to guarantee  $S_i(\epsilon, \delta, h^*)$  sample complexity. However, it turns out it is also possible to learn with sample complexity  $O(S_i(\epsilon/2, \delta/2, h^*))$  even without this information. This can be accomplished by using an aggregation algorithm.

We describe a simple algorithm for aggregation below in which multiple algorithms are run on different subclasses  $C_i$  in parallel and we select among their outputs by comparisons. Within each subclass  $C_i$  we run an active learning algorithm  $A_i$ , such as Dasgupta’s splitting algorithm [8] or CAL, with some sample complexity  $S_i(\epsilon, \delta, h)$ .

---

**Algorithm 1** The Aggregation Procedure. Here it is assumed that  $C = \cup_{i=1}^{\infty} C_i$ , and that for each  $i$ ,  $A_i$  is an algorithm achieving sample complexity at most  $S_i(\epsilon, \delta, h)$  for the pair  $(C_i, D)$ . The procedure takes  $t$  and  $\delta$  as parameters.

---

```

Let  $k$  be the largest integer s.t.  $k^2 \lceil 72 \ln(4k/\delta) \rceil \leq t/2$ 
for  $i = 1, \dots, k$  do
  Let  $h_i$  be the output of running  $A_i(\lfloor t/(4i^2) \rfloor, \delta/2)$  on
  the sequence  $\{x_{2n-1}\}_{n=1}^{\infty}$ 
end for
for  $i, j \in \{1, 2, \dots, k\}$  do
  if  $\mathbb{P}(h_i(X) \neq h_j(X)) > 0$  then
    Let  $R_{ij}$  be the first  $\lceil 72 \ln(4k/\delta) \rceil$  elements in the se-
    quence  $\{x_{2n}\}_{n=1}^{\infty}$  for which  $h_i(x) \neq h_j(x)$ 
    Request the labels of all examples in  $R_{ij}$ 
    Let  $m_{ij}$  be the number of elements in  $R_{ij}$  on which
     $h_i$  makes a mistake
  else
    Let  $m_{ij} = 0$ 
  end if
end for
Return  $\hat{h}_t = h_i$  where  $i = \operatorname{argmin}_{i \in \{1, 2, \dots, k\}} \max_{j \in \{1, 2, \dots, k\}} m_{ij}$ 

```

---

Using this algorithm, we can show the following sample complexity bound. The proof appears in Appendix A.

**Theorem 8** For any distribution  $D$ , let  $C_1, C_2, \dots$  be a sequence of classes such that for each  $i$ , the pair  $(C_i, D)$  has sample complexity at most  $S_i(\epsilon, \delta, h)$  for all  $h \in C_i$ . Let  $C = \cup_{i=1}^{\infty} C_i$ . Then  $(C, D)$  has a sample complexity at most

$$\min_{i: h \in C_i} \max \left\{ 4i^2 \lceil S_i(\epsilon/2, \delta/2, h) \rceil, 2i^2 \left\lceil 72 \ln \frac{4i}{\delta} \right\rceil \right\},$$

for any  $h \in C$ . In particular, Algorithm 1 achieves this, when used with the  $A_i$  algorithms that each achieve the  $S_i(\epsilon, \delta, h)$  sample complexity.

A particularly interesting implication of Theorem 8 is that, if we can decompose  $C$  into a sequence of classes  $C_i$  such that each  $(C_i, D)$  is learnable at an exponential rate, then this procedure achieves exponential rates. Since it is more abstract and it allows us to use known active learning algorithms as a black box, we often use this compositional view throughout the remainder of the paper. In particular, since the verifiable sample complexity of active learning is presently much better understood in the existing literature, it will often be useful to use this result in combination with

an algorithm with a known bound on its *verifiable* sample complexity. As the following theorem states, at least for the case of exponential rates, this approach of constructing algorithms with good true sample complexity by reduction to algorithms with known verifiable complexity on subspaces loses nothing in generality. The proof is included in Appendix B.

**Theorem 9** For any  $(C, D)$  learnable at an exponential rate, there exists a sequence  $C_1, C_2, \dots$  with  $C = \cup_{i=1}^{\infty} C_i$ , and a sequence of active learning algorithms  $A_1, A_2, \dots$  such that the algorithm  $A_i$  achieves verifiable sample complexity at most  $\gamma_i \operatorname{polylog}_i(1/(\epsilon\delta))$  for the pair  $(C_i, D)$ . Thus, the aggregation algorithm (Algorithm 1) achieves exponential rates when used with these algorithms.

Note that decomposing a given  $C$  into a sequence of  $C_i$  subsets that have good verifiable sample complexities is not always a simple task. One might be tempted to think a simple decomposition based on increasing values of verifiable sample complexity with respect to  $(C, D)$  would be sufficient. However, this is not always the case, and generally we need to use information more detailed than verifiable complexity with respect to  $(C, D)$  to construct a good decomposition. We have included in Appendix C a simple heuristic approach that can be quite effective, and in particular yields good sample complexities for every  $(C, D)$  described in Section 5.

## 5 Exponential Rates

The results in Section 3 tell us that the sample complexity of active learning can be made strictly superior to any passive learning sample complexity when the VC dimension is finite. We now ask how much better that sample complexity can be. In particular, we describe a number of concept classes and distributions that are learnable at an *exponential* rate, many of which are known to require  $\Omega(1/\epsilon)$  *verifiable* sample complexity.

### 5.1 Exponential rates for simple classes

We begin with a few simple observations, to point out situations in which exponential rates are trivially achievable; in fact, in each of the cases mentioned in this subsection, the sample complexity is actually  $O(1)$ .

Clearly if  $|\mathcal{X}| < \infty$  or  $|C| < \infty$ , we can always achieve exponential rates. In the former case, we may simply request the label of every  $x$  in the support of  $D$ , and thereby perfectly identify the target. The corresponding  $\gamma = |\mathcal{X}|$ . In the latter case, for every pair  $h_1, h_2 \in C$  such that  $\mathbb{P}(h_1(X) \neq h_2(X)) > 0$ , we may request the label of any  $x_i$  such that  $h_1(x_i) \neq h_2(x_i)$ , and there will be only one (up to measure zero differences)  $h \in C$  that gets all of these examples correct: namely, the target function. So in this case, we learn with an exponential rate with  $\gamma = |C|^2$ .

Less obvious is the fact that this argument extends to any *countably infinite* hypothesis class  $C$ . In particular, in this case we can list the classifiers in  $C$ :  $h_1, h_2, \dots$ . Then we define the sequence  $C_i = \{h_i\}$ , and simply use Algorithm 1. By Theorem 8, this gives an algorithm with sample complexity  $S(\epsilon, \delta, h_i) = 2i^2 \lceil 72 \ln(4i/\delta) \rceil = O(1)$ .

## 5.2 Geometric Concepts, Uniform Distribution

Many interesting geometric concepts in  $\mathbb{R}^n$  are learnable at an exponential rate if the underlying distribution is uniform on some subset of  $\mathbb{R}^n$ . Here we provide some examples; interestingly, every example in this subsection has some targets for which the *verifiable* sample complexity is  $\Omega(1/\epsilon)$ . As we see in Section 5.3, all of the results in this section can be extended to many other types of distributions as well.

**Unions of  $k$  intervals under arbitrary distributions:** Let  $\mathcal{X}$  be the interval  $[0, 1]$  and let  $C^{(k)}$  denote the class of unions of at most  $k$  intervals. In other words,  $C^{(k)}$  contains functions described by a sequence  $\langle a_0, a_1, \dots, a_\ell \rangle$ , where  $a_0 = 0$ ,  $a_\ell = 1$ ,  $\ell \leq 2k + 1$ , and  $a_0, \dots, a_\ell$  is the (nondecreasing) sequence of transition points between negative and positive segments (so  $x$  is labeled  $+1$  iff  $x \in [a_i, a_{i+1})$  for some *odd*  $i$ ). For any distribution, this class is learnable at an exponential rate, by the following decomposition argument. First, let

$$C_1 = \{h \in C^{(k)} : \mathbb{P}(h(X) = +1) = 0\}.$$

That is,  $C_1$  contains the all-negative function, or any function that is equivalent given the distribution  $D$ . For  $i = 2, 3, \dots, k + 1$ , inductively define

$$C_i = \{h \in C^{(k)} : \exists h' \in C^{(i-1)} \text{ s.t. } \mathbb{P}(h(X) \neq h'(X)) = 0\} \setminus \cup_{j < i} C_j.$$

In other words,  $C_i$  contains all of the functions that can be represented as unions of  $i - 1$  intervals but cannot be represented as unions of fewer intervals. Clearly  $C_1$  has verifiable sample complexity 0. For  $i > 1$ , within each subclass  $C_i$ , the disagreement coefficient is bounded by something proportional to  $k + 1/w(h)$ , where

$$w(h) = \min\{\mathbb{P}([a_j, a_{j+1})) : 0 \leq j < \ell, \mathbb{P}([a_j, a_{j+1})) > 0\}$$

is the weight of the smallest positive or negative interval and  $\langle a_0, a_1, \dots, a_\ell \rangle$  is the sequence of transition points corresponding to this  $h$ . Thus, running CAL with  $\bar{C}_i$  achieves polylogarithmic (verifiable) sample complexity for any  $h \in C_i$ . Since  $C^{(k)} = \cup_{i=1}^{k+1} C_i$ , by Theorem 8,  $C^{(k)}$  is learnable at an exponential rate.

**Ordinary Binary Classification Trees:** Let  $\mathcal{X}$  be the cube  $[0, 1]^n$ ,  $D$  be the uniform distribution on  $\mathcal{X}$ , and  $C$  be the class of binary decision trees using a finite number of axis-parallel splits (see e.g., Devroye et al. [11], Chapter 20). In this case, (similarly to the previous example) we let  $C_i$  be the set of decision trees in  $C$  distance zero from a tree with  $i$  leaf nodes, not contained in any  $C_j$  for  $j < i$ . For any  $i$ , the disagreement coefficient for any  $h \in C_i$  (with respect to  $(C_i, D)$ ) is a finite constant, and we can choose  $\bar{C}_i$  to have finite VC dimension, so each  $(C_i, D)$  is learnable at an exponential rate (by running CAL with  $\bar{C}_i$ ), and thus by Theorem 8,  $(C, D)$  is learnable at an exponential rate.

### 5.2.1 Linear Separators

**Theorem 10** *Let  $C$  be the hypothesis class of linear separators in  $n$  dimensions, and let  $D$  be the uniform distribution over the surface of the unit sphere. The pair  $(C, D)$  is learnable at an exponential rate.*

**Proof:** (Sketch) There are multiple ways to achieve this. We describe here a simple proof that uses a decomposition as follows. Let  $\lambda(h)$  be the probability mass of the minority class under hypothesis  $h$ .  $C_1$  contains only the separators  $h$  with  $\lambda(h) = 0$ , and  $C_2 = C \setminus C_1$ . As before, we can use a black box active learning algorithm such as CAL to learn within each class  $C_i$ . To prove that we indeed get the desired exponential rate of active learning, we show that the disagreement coefficient of any separator  $h$  with respect to  $(C, D)$  is at most  $\propto \sqrt{n}/\lambda(h)$ . Hanneke's results concerning the CAL algorithm [12] then imply that  $C_2$  is learnable at an exponential rate. Since  $C_1$  trivially has sample complexity 1, combined with Theorem 8, this would imply the result.

We describe the key steps involved in computing the disagreement coefficient. First we can show that for any two linear separators  $h(x) = \text{sign}(w \cdot x + b)$  and  $h'(x) = \text{sign}(w' \cdot x + b')$ , we can lower bound the distance between them as

$$\mathbb{P}(h(X) \neq h'(X)) \geq \max\left\{|\lambda - \lambda'|, \frac{2\alpha}{\pi} \min\{\lambda, \lambda'\}\right\},$$

where  $\alpha = \arccos(w \cdot w')$  is the angle between  $w$  and  $w'$ ,  $\lambda$  is the probability mass of the minority class under  $h$ , and  $\lambda'$  is the probability mass of the minority class under  $h'$ . Assume for now that  $h$  and  $h'$  are close enough together to have the same minority class; it's not necessary, but simplifies things.

We are now ready to compute the disagreement coefficient. Assume  $r < \lambda/\sqrt{n}$ . From the previous claim we have

$$B(h, r) \subseteq \left\{h' : \max\left\{|\lambda - \lambda'|, \frac{2\alpha}{\pi} \min\{\lambda, \lambda'\}\right\} \leq r\right\}$$

where  $B(h, r)$  is the ball of radius  $r$  around  $h$  in the hypothesis space. The region of disagreement of the set on the left is contained within

$$\text{DIS}(\{h' : w' = w \wedge |\lambda' - \lambda| \leq r\}) \cup \text{DIS}\left(\left\{h' : \frac{2\alpha}{\pi}(\lambda - r) \leq r \wedge |\lambda - \lambda'| = r\right\}\right).$$

By some trigonometry, we can show this region is contained within

$$\text{DIS}(\{h' : w' = w \wedge |\lambda' - \lambda| \leq r\}) \cup \left\{x : |w \cdot x + b_1| \leq c \frac{r}{\lambda}\right\} \cup \left\{x : |w \cdot x + b_2| \leq c \frac{r}{\lambda}\right\}$$

for some constants  $b_1, b_2, c$ . Using previous results [2, 12], it is possible to show that the measure of this region is at most  $2r + c'(\sqrt{n}/\lambda)r = c''(\sqrt{n}/\lambda)r$ . This finally implies that for any target function, the disagreement coefficient is at most  $c''(\sqrt{n}/\lambda)$ , where  $\lambda$  is the probability of the minority class of the target function. ■

## 5.3 Composition results

We can also extend the results from the previous subsection to other types of distributions and concept classes in a variety of ways. Here we include a few results to this end.

**Close distributions:** If  $(C, D)$  is learnable at an exponential rate, then for any distribution  $D'$  such that for all measurable

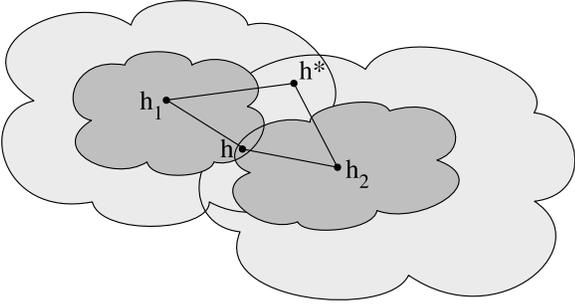


Figure 5.1: Illustration of the proof of Theorem 11. The dark gray regions represent  $B_{D_1}(h_1, 2r)$  and  $B_{D_2}(h_2, 2r)$ . The function  $h$  that gets returned is in the intersection of these. The light gray regions represent  $B_{D_1}(h_1, \epsilon/3)$  and  $B_{D_2}(h_2, \epsilon/3)$ . The target function  $h^*$  is in the intersection of these. We therefore must have  $r \leq \epsilon/3$ , and by the triangle inequality  $\text{er}(h) \leq \epsilon$ .

$A \subseteq \mathcal{X}$ ,  $\lambda \mathbb{P}_D(A) \leq \mathbb{P}_{D'}(A) \leq (1/\lambda) \mathbb{P}_D(A)$  for some  $\lambda \in (0, 1]$ ,  $(C, D')$  is also learnable at an exponential rate. In particular, we can simply use the algorithm for  $(C, D)$ , filter the examples from  $D'$  so that they appear like examples from  $D$ , and then any  $t$  large enough to find an  $\epsilon\lambda$ -good classifier with respect to  $D$  is large enough to find an  $\epsilon$ -good classifier with respect to  $D'$ .

**A composition theorem for mixtures of distributions:** Suppose there exist algorithms  $\mathcal{A}_1$  and  $\mathcal{A}_2$  for learning a class  $C$  at an exponential rate under distributions  $D_1$  and  $D_2$  respectively. It turns out we can also learn under any mixture of  $D_1$  and  $D_2$  at an exponential rate, by using  $\mathcal{A}_1$  and  $\mathcal{A}_2$  as black boxes. In particular, the following theorem relates the sample complexity under a mixture to the sample complexities under the mixing components.

**Theorem 11** *Let  $C$  be an arbitrary hypothesis class. Assume that the pairs  $(C, D_1)$  and  $(C, D_2)$  have sample complexities  $S_1(\epsilon, \delta, h^*)$  and  $S_2(\epsilon, \delta, h^*)$  respectively, where  $D_1$  and  $D_2$  have density functions  $\mathbb{P}_{D_1}$  and  $\mathbb{P}_{D_2}$  respectively. Then for any  $\alpha \in [0, 1]$ , the pair  $(C, \alpha D_1 + (1 - \alpha) D_2)$  has sample complexity at most  $2 \lceil \max\{S_1(\epsilon/3, \delta/2, h^*), S_2(\epsilon/3, \delta/2, h^*)\} \rceil$ .*

**Proof:** If  $\alpha = 0$  or 1 then the theorem statement holds trivially. Assume instead that  $\alpha \in (0, 1)$ . We describe an algorithm in terms of  $\alpha$ ,  $D_1$ , and  $D_2$ , which achieves this sample complexity bound.

Suppose algorithms  $\mathcal{A}_1$  and  $\mathcal{A}_2$  achieve the stated sample complexities under  $D_1$  and  $D_2$  respectively. At a high level, the algorithm we define works by “filtering” the distribution over input so that it appears to come from two streams, one distributed according to  $D_1$ , and one distributed according to  $D_2$ , and feeding these filtered streams to  $\mathcal{A}_1$  and  $\mathcal{A}_2$  respectively. To do so, we define a random sequence  $u_1, u_2, \dots$  of independent uniform random variables in  $[0, 1]$ . We then run  $\mathcal{A}_1$  on the sequence of examples  $x_i$  from the unlabeled data sequence satisfying

$$u_i < \frac{\alpha \mathbb{P}_{D_1}(x_i)}{\alpha \mathbb{P}_{D_1}(x_i) + (1 - \alpha) \mathbb{P}_{D_2}(x_i)},$$

and run  $\mathcal{A}_2$  on the remaining examples, allowing each to make an equal number of label requests.

Let  $h_1$  and  $h_2$  be the classifiers output by  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . Because of the filtering, the examples that  $\mathcal{A}_1$  sees are distributed according to  $D_1$ , so after  $t/2$  queries, the current error of  $h_1$  with respect to  $D_1$  is, with probability  $1 - \delta/2$ , at most  $\inf\{\epsilon' : S_1(\epsilon', \delta/2, h^*) \leq t/2\}$ . A similar argument applies to the error of  $h_2$  with respect to  $D_2$ .

Finally, let

$$r = \inf\{r : B_{D_1}(h_1, r) \cap B_{D_2}(h_2, r) \neq \emptyset\}.$$

Define the output of the algorithm to be any  $h \in B_{D_1}(h_1, 2r) \cap B_{D_2}(h_2, 2r)$ . If a total of  $t \geq 2 \lceil \max\{S_1(\epsilon/3, \delta/2, h^*), S_2(\epsilon/3, \delta/2, h^*)\} \rceil$  queries have been made ( $t/2$  by  $\mathcal{A}_1$  and  $t/2$  by  $\mathcal{A}_2$ ), then by a union bound, with probability at least  $1 - \delta$ ,  $h^*$  is in the intersection of the  $\epsilon/3$ -balls, and so  $h$  is in the intersection of the  $2\epsilon/3$ -balls. By the triangle inequality,  $h$  is within  $\epsilon$  of  $h^*$  under both distributions, and thus also under the mixture. (See Figure 5.1 for an illustration of these ideas.) ■

## 5.4 Lower Bounds

Given the previous discussion, one might suspect that *any* pair  $(C, D)$  is learnable at an exponential rate, under some mild condition such as finite VC dimension. However, we show in the following that this is *not* the case, even for some simple geometric concept classes when the distribution is especially nasty.

**Theorem 12** *There exists a pair  $(C, D)$ , with the VC dimension of  $C$  equal 1, that is not learnable at an exponential rate (in the sense of Definition 4).*

**Proof:** (Sketch) Let  $T$  be a fixed infinite tree in which each node at depth  $i$  has  $c_i$  children;  $c_i$  is defined shortly. We consider learning the hypothesis class  $C$  where each  $h \in C$  corresponds to a path down the tree starting at the root; every node along this path is labeled 1 while the remaining nodes are labeled  $-1$ . Clearly for each  $h \in C$  there is precisely one node on each level of the tree labeled 1 by  $h$  (i.e. one node at each depth  $d$ ).  $C$  has VC dimension 1 since knowing the identity of the node labeled 1 on level  $i$  is enough to determine the labels of all nodes on levels  $0, \dots, i$  perfectly. This learning problem is depicted in Figure 5.2.

Now we define  $D$ , a “bad” distribution for  $C$ . Let  $\ell_i$  be the total probability of all nodes on level  $i$  according to  $D$ . Assume all nodes on level  $i$  have the same probability according to  $D$ , and call this  $p_i$ . By definition, we have  $p_i = \ell_i / \prod_{j=0}^{i-1} c_j$ .

We show that it is possible to define the parameters above in such a way that for any  $\epsilon_0 > 0$ , there exists some  $\epsilon < \epsilon_0$  such that for some level  $j$ ,  $p_j = \epsilon$  and  $c_{j-1} \geq (1/p_j)^{1/2} = (1/\epsilon)^{1/2}$ . This implies that  $\Omega(1/\epsilon^{1/2})$  labels are needed to learn with error less than  $\epsilon$ , for the following reason. We know that there is exactly one node on level  $j$  that has label 1, and that any successful algorithm must identify this node (or have a lucky guess at which one it is) since it has probability  $\epsilon$ . By the usual probabilistic method trick (picking the target at random by choosing the positive node at each level  $i + 1$  uniformly from the children of the positive at level  $i$ ), we

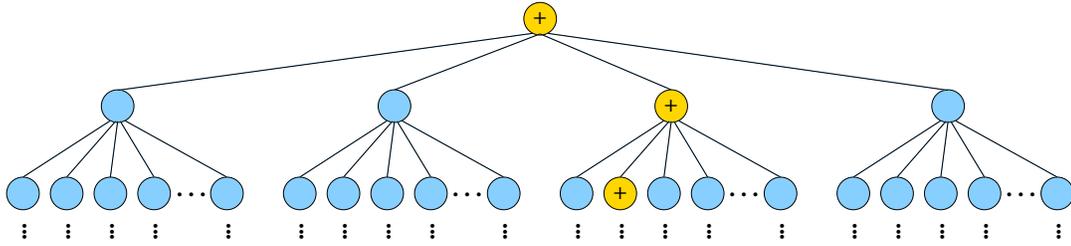


Figure 5.2: A learning problem where exponential rates are not achievable. The instance space is an infinite-depth tree. The target labels nodes along a single infinite path as +1, and labels all other nodes -1. When the number of children and probability mass of each node at each subsequent level are set in a certain way, sample complexities of  $o(1/\sqrt{\epsilon})$  are not achievable.

can argue that in order to label that node positive with at least some constant probability, we need to query at least a constant fraction of the node’s siblings, so we need to query on the order of  $c_{j-1}$  nodes on level  $j$ .

Thus it is enough to show that we can define the values above such that for all  $i$ ,  $c_{i-1} \geq (1/p_i)^{1/2}$ , and such that  $p_i$  gets arbitrarily small as  $i$  gets big.

To start, notice that if we recursively define the values of  $c_i$  as  $c_i = \prod_{j=0}^{i-1} c_j / \ell_{i+1}$  then

$$c_{i-1}^2 = c_{i-1} \left( \frac{\prod_{j=0}^{i-2} c_j}{\ell_i} \right) = \frac{\prod_{j=0}^{i-1} c_j}{\ell_i} = \frac{1}{p_i}$$

and  $c_{i-1} \geq (1/p_i)^{1/2}$  as desired.

To enforce that  $p_i$  gets arbitrarily small as  $i$  gets big, we simply need to set  $\ell_i$  appropriately. In particular, we need  $\lim_{i \rightarrow \infty} \ell_i / \prod_{j=0}^{i-1} c_j = 0$ . Since the denominator is increasing in  $i$ , it suffices to show  $\lim_{i \rightarrow \infty} \ell_i = 0$ . Defining the values of  $\ell_i$  to be any positive probability distribution over  $i$  that goes to 0 in the limit completes the proof. ■

For essentially any function  $\phi = o(1/\epsilon)$ , the tree example in the proof can be modified to construct a pair  $(C, D)$  with the VC dimension of  $C$  equal to 1 such that no algorithm achieves  $o(\phi(\epsilon))$  sample complexity for all targets: simply choose  $c_i = \lfloor \phi(p_{i+1}) \rfloor$ , where  $\{p_j\}$  is any sequence strictly decreasing to 0 s.t.  $p_{i+1} \phi(p_{i+1}) \prod_{j < i} c_j \leq \ell_{i+1}$  and  $\phi(p_{i+1}) \geq 1$ , where as before  $\{\ell_j\}$  is any sequence of positive values summing to 1; we can (arbitrarily) assign any left-over probability mass to the root node;  $\phi = o(1/\epsilon)$  guarantees that such a  $\{p_j\}$  sequence exists for any  $\phi = \omega(1)$ . Thus, the  $o(1/\epsilon)$  guarantee of Corollary 7 is in some sense the tightest guarantee we can make at that level of generality, without using a more detailed description of the structure of the problem beyond the finite VC dimension assumption.

This type of example can be realized by certain nasty distributions, even for a variety of simple hypothesis classes: for example, linear separators in  $\mathbb{R}^2$  or axis-aligned rectangles in  $\mathbb{R}^2$ . We remark that this example can also be modified to show that we cannot expect intersections of classifiers to preserve exponential rates. That is, the proof can be extended to show that there exist classes  $C_1$  and  $C_2$ , such that both  $(C_1, D)$  and  $(C_2, D)$  are learnable at an exponential rate, but  $(C, D)$  is not, where  $C = \{h_1 \cap h_2 : h_1 \in C_1, h_2 \in C_2\}$ .

## 6 Discussion and Open Questions

The implication of our analysis is that in many interesting cases where it was previously believed that active learning could not help, it turns out that active learning *does help asymptotically*. We have formalized this idea and illustrated it with a number of examples and general theorems throughout the paper. This realization dramatically shifts our understanding of the usefulness of active learning: while previously it was thought that active learning could *not* provably help in any but a few contrived and unrealistic learning problems, in this alternative perspective we now see that active learning essentially *always* helps, and does so significantly in all *but* a few contrived and unrealistic problems.

The use of decompositions of  $C$  in our analysis also generates another interpretation of these results. Specifically, Dasgupta [8] posed the question of whether it would be useful to develop active learning techniques for looking at unlabeled data and “placing bets” on certain hypotheses. One might interpret this work as an answer to this question; that is, some of the decompositions used in this paper can be interpreted as reflecting a preference partial-ordering of the hypotheses, similar to ideas explored in the passive learning literature [16, 15, 3]. However, the construction of a good decomposition in active learning seems more subtle and quite different from previous work in the context of supervised or semi-supervised learning.

It is interesting to examine the role of target- and distribution-dependent constants in this analysis. As defined, both the verifiable and true sample complexities may depend heavily on the particular target function and distribution. Thus, in both cases, we have interpreted these quantities as fixed when studying the asymptotic growth of these sample complexities as  $\epsilon$  approaches 0. It has been known for some time that, with only a few unusual exceptions, any target- and distribution-independent bound on the verifiable sample complexity could typically be no better than the sample complexity of passive learning; in particular, this observation lead Dasgupta to formulate his splitting index bounds as both target- and distribution-dependent [8]. This fact also applies to bounds on the true sample complexity as well. Indeed, the entire distinction between verifiable and true sample complexities collapses if we remove the dependence on these unobservable quantities.

There are many interesting open problems within this framework. Perhaps two of the most interesting are formulating general necessary and sufficient conditions for

learnability at an exponential rate, and determining whether Theorem 6 can be extended to the agnostic case.

**Acknowledgments:** We thank Eyal Even-Dar, Michael Kearns, and Yishay Mansour for numerous useful discussions and for helping us to initiate this line of thought. We are also grateful to Larry Wasserman and Eric Xing for their helpful feedback.

Maria-Florina is supported in part by an IBM Graduate Fellowship and by a Google Research Grant. Steve is funded by the NSF grant IIS-0713379 awarded to Eric Xing.

## References

- [1] A. Antos and G. Lugosi. Strong minimax lower bounds for learning. *Machine Learning*, 30:31–56, 1998.
- [2] M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.
- [3] M.-F. Balcan and A. Blum. A PAC-style model for learning from labeled and unlabeled data. Book chapter in “Semi-Supervised Learning”, O. Chapelle and B. Schölkopf and A. Zien, eds., MIT press, 2006.
- [4] M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Proceedings of the 20th Annual Conference on Learning Theory*, 2007.
- [5] R. Castro and R. Nowak. Minimax bounds for active learning. In *Proceedings of the 20th Annual Conference on Learning Theory*, 2007.
- [6] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.
- [7] S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems*, 2004.
- [8] S. Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems*, 2005.
- [9] S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*, 2007.
- [10] S. Dasgupta, A. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.
- [11] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.
- [12] S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.
- [13] S. Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the 20th Conference on Learning Theory*, 2007.
- [14] D. Haussler, N. Littlestone, and M. Warmuth. Predicting  $\{0, 1\}$ -functions on randomly drawn points. *Information and Computation*, 115:248–292, 1994.
- [15] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.
- [16] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., 1998.

## Appendix

### A Proof of Theorem 8

First note that the total number of label requests used by the aggregation procedure in Algorithm 1 is at most  $t$ . Initially running the algorithms  $A_1, \dots, A_k$  requires  $\sum_{i=1}^k \lceil t/(4i^2) \rceil \leq t/2$  labels, and the second phase of the algorithm requires  $k^2 \lceil 72 \ln(4k/\delta) \rceil$  labels, which by definition of  $k$  is also less than  $t/2$ . Thus this procedure is a valid learning algorithm.

Now suppose that  $h^* \in C_i$ , and assume that

$$t \geq \max \left\{ 4i^2 \lceil S_i(\epsilon/2, \delta/2, h^*) \rceil, 2i^2 \lceil 72 \ln(4i/\delta) \rceil \right\}.$$

We must show that for any such value of  $t$ ,  $er(\hat{h}_t) \leq \epsilon$  with probability at least  $1 - \delta$ .

First notice that since  $t \geq 2i^2 \lceil 72 \ln(4i/\delta) \rceil$ ,  $k \geq i$ . Furthermore, since  $t/(4i^2) \geq \lceil S_i(\epsilon/2, \delta/2, h^*) \rceil$ , with probability at least  $1 - \delta/2$ , running  $\mathcal{A}_i(\lceil t/(4i^2) \rceil, \delta/2)$  returns a function  $h_i$  with  $er(h_i) \leq \epsilon/2$ .

Let  $j^* = \operatorname{argmin}_j er(h_j)$ . By Hoeffding’s inequality, with probability at least  $1 - \delta/4$ , for all  $\ell$ ,

$$m_{j^* \ell} \leq \frac{7}{12} \lceil 72 \ln(4k/\delta) \rceil,$$

and thus

$$\min_j \max_{\ell} m_{j\ell} \leq \frac{7}{12} \lceil 72 \ln(4k/\delta) \rceil.$$

Furthermore, by Hoeffding’s inequality and a union bound, with probability at least  $1 - \delta/4$ , for any  $\ell$  such that

$$m_{\ell j^*} \leq \frac{7}{12} \lceil 72 \ln(4k/\delta) \rceil$$

we have that

$$er(h_{\ell} | h_{\ell}(x) \neq h_{j^*}(x)) \leq \frac{2}{3}$$

and thus  $er(h_{\ell}) \leq 2er(h_{j^*})$ . By a union bound over these three events, we find that, as desired, with probability at least  $1 - \delta$ ,

$$er(\hat{h}_t) \leq 2er(h_{j^*}) \leq 2er(h_i) \leq \epsilon. \quad \blacksquare$$

### B Proof of Theorem 9

Assume that  $(C, D)$  is learnable at an exponential rate. That means there exists an algorithm  $A$  such that for any target  $h^*$  in  $C$ , there exist constants  $\gamma_{h^*} = \gamma(h^*, D)$  and  $k_{h^*}$  such that for any  $\epsilon$  and  $\delta$ , with probability at least  $1 - \delta$ , for any  $t \geq \gamma_{h^*} (\log(1/(\epsilon\delta)))^{k_{h^*}}$ , after  $t$  label requests,  $A(t, \delta)$  outputs an  $\epsilon$ -good classifier.

We define  $C_i = \{h \in C : \gamma_h \leq i, k_h \leq i\}$ . For every  $i$ , we define an algorithm  $A_i$  that achieves the required polylog verifiable sample complexity as follows. We first run  $A$  to obtain function  $h_A$ . We then let  $A_i$  always output the closest classifier in  $C_i$  to  $h_A$ . If  $t \geq i(\log(2/(\epsilon\delta)))^i$ , then after  $t$  label requests, with probability at least  $1 - \delta$ ,  $A(t, \delta)$  outputs an  $\epsilon/2$ -good classifier, so by the triangle inequality, with probability at least  $1 - \delta$ ,  $A_i(t, \delta)$  outputs an  $\epsilon$ -good classifier. Furthermore,  $A_i$  can output  $\hat{\epsilon}_t = (2/\delta) \exp\{-t/i^{1/i}\}$ , which is no more than  $\epsilon$ . Combining this with Theorem 8 we get the desired result.  $\blacksquare$

## C Heuristic Approaches to Decomposition

As mentioned, decomposing purely based on verifiable complexity with respect to  $(C, D)$  typically cannot yield a good decomposition even for very simple problems, such as unions of intervals. The reason is that the set of classifiers with high verifiable sample complexity may itself have high verifiable complexity.

Although we do not yet have a general method that can provably always find a good decomposition when one exists (other than the trivial method in the proof of Theorem 9), we often find that a heuristic recursive technique can be quite effective. That is, we can define  $C_1 = C$ . Then for  $i > 1$ , we recursively define  $C_i$  as the set of all  $h \in C_{i-1}$  such that  $\theta_h = \infty$  with respect to  $(C_{i-1}, D)$ . Suppose that for some  $N$ ,  $C_{N+1} = \emptyset$ . Then for the decomposition  $C_1, C_2, \dots, C_N$ , every  $h \in C$  has  $\theta_h < \infty$  with respect to at least one of the sets in which it is contained. Thus, the verifiable sample complexity of  $h$  with respect to that set is  $O(\text{polylog}(1/\epsilon\delta))$ , and the aggregation algorithm can be used to achieve polylog sample complexity.

We could alternatively perform a similar decomposition using a suitable definition of splitting index [8], or more generally using

$$\limsup_{\epsilon \rightarrow 0} \frac{S_{C_{i-1}}(\epsilon, \delta, h)}{\left(\log\left(\frac{1}{\epsilon\delta}\right)\right)^k}$$

for some fixed constant  $k > 0$ .

While this procedure does not always generate a good decomposition, certainly if  $N < \infty$  exists, then this creates a decomposition for which the aggregation algorithm, combined with an appropriate sequence of algorithms  $\{A_i\}$ , can achieve exponential rates. In particular, this is the case for all of the  $(C, D)$  described in Section 5. In fact, even if  $N = \infty$ , as long as every  $h \in C$  does end up in *some* set  $C_i$  for finite  $i$ , this decomposition would still provide exponential rates.

## D Proof of Theorem 6

We now finally prove Theorem 6. This section is mostly self-contained, though we do make use of Theorem 8 from Section 4 in the final step of the proof.

For any  $V \subseteq C$  and  $h \in C$ , define

$$\bar{B}_V(h, r) = \{h' \in \bar{V} : \mathbb{P}_D(h(x) \neq h'(x)) \leq r\},$$

where  $\bar{V}$  is, as before, a countable dense subset of  $V$ . Define the *boundary* of  $h$  with respect to  $D$  and  $V$ , denoted  $\partial_V h$ , as

$$\partial_V h = \lim_{r \rightarrow 0} \text{DIS}(\bar{B}_V(h, r)).$$

The proof will proceed according to the following outline. We begin in Lemma 13 by describing special conditions under which a CAL-like algorithm has the property that the more unlabeled examples it processes, the smaller the fraction of them it requests the labels of. Since CAL always identifies the target's true label on any example it processes, we end up with a set of labeled examples growing strictly faster than the number of label requests used to obtain it; we can use this as a training set in any passive learning algorithm. However, the special conditions under which this happens are rather limiting, so we require an additional step, in Lemma 14; there, we exploit a subtle relation between

overlapping boundary regions and shatterable sets to show that we can decompose any finite VC dimension class into a countable number of subsets satisfying these special conditions. This, combined with the aggregation algorithm, extends Lemma 13 to the general conditions of Theorem 6.

**Lemma 13** *Suppose  $(C, D)$  is such that  $C$  has finite VC dimension  $d$ , and  $\forall h \in C, \mathbb{P}(\partial_C h) = 0$ . Then for any passive learning sample complexity  $S_p(\epsilon, \delta, h)$  for  $(C, D)$ , there exists an active learning algorithm achieving a sample complexity  $S_a(\epsilon, \delta, h)$  such that, for any target function  $h^* \in C$  where  $S_p(\epsilon, \delta, h^*) = \omega(1)$ ,*

$$S_a(\epsilon, \delta/2, h^*) = o(S_p(\epsilon/2, \delta, h^*)).$$

**Proof:** We perform the learning in two phases. The first is a passive phase: we simply request the labels of  $x_1, x_2, \dots, x_{\lfloor t/3 \rfloor}$ , and let

$$V = \{h \in \bar{C} : \forall i \leq \lfloor t/3 \rfloor, h(x_i) = h^*(x_i)\}.$$

In other words,  $V$  is the set of all hypotheses that correctly label the first  $\lfloor t/3 \rfloor$  examples. By standard consistency results [11], with probability at least  $1 - \delta/8$ , there is a universal constant  $c > 0$  such that

$$\sup_{h_1, h_2 \in V} \mathbb{P}_D(h_1(x) \neq h_2(x)) \leq c \left( \frac{d \ln t + \ln \frac{1}{\delta}}{t} \right).$$

In particular, on this event, we have

$$\mathbb{P}(\text{DIS}(V)) \leq \mathbb{P}\left(\text{DIS}\left(\bar{B}\left(h^*, c \frac{d \ln t + \ln \frac{1}{\delta}}{t}\right)\right)\right).$$

Let us denote this latter quantity by  $\Delta_t$ . Note that  $\Delta_t$  goes to 0 as  $t$  grows.

If ever we have  $\mathbb{P}(\text{DIS}(V)) = 0$  for some finite  $t$ , then clearly we can return any  $h \in V$ , so this case is easy.

Otherwise, let  $n_t = \lfloor t/(36\mathbb{P}(\text{DIS}(V)) \ln(8/\delta)) \rfloor$ , and suppose  $t \geq 3$ . By a Chernoff bound, with probability at least  $1 - \delta/8$ , in the sequence of examples  $x_{\lfloor t/3 \rfloor + 1}, x_{\lfloor t/3 \rfloor + 2}, \dots, x_{\lfloor t/3 \rfloor + n_t}$ , at most  $t/3$  of the examples are in  $\text{DIS}(V)$ . If this is not the case, we fail and output an arbitrary  $h$ ; otherwise, we request the labels of every one of these  $n_t$  examples that are in  $\text{DIS}(V)$ . Now construct a sequence  $\mathcal{L} = \{(x'_1, y'_1), (x'_2, y'_2), \dots, (x'_{n_t}, y'_{n_t})\}$  of labeled examples such that  $x'_i = x_{\lfloor t/3 \rfloor + i}$ , and  $y'_i$  is either the label agreed upon by all the elements of  $V$ , or it is the  $h^*(x_{\lfloor t/3 \rfloor + i})$  label value we explicitly requested. Note that because  $\inf_{h \in V} \text{er}(h) = 0$  with probability 1, we also have that with probability 1 every  $y'_i = h^*(x'_i)$ . We may therefore use these  $n_t$  examples as iid training examples for the passive learning algorithm.

Specifically, let us split up the sequence  $\mathcal{L}$  into  $k = 4$  sequences  $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_k$ , where

$$\begin{aligned} \mathcal{L}_i = & \{(x'_{(i-1)\lfloor n_t/k \rfloor + 1}, y'_{(i-1)\lfloor n_t/k \rfloor + 1}), \\ & (x'_{(i-1)\lfloor n_t/k \rfloor + 2}, y'_{(i-1)\lfloor n_t/k \rfloor + 2}), \\ & \dots, (x'_{i\lfloor n_t/k \rfloor}, y'_{i\lfloor n_t/k \rfloor})\}. \end{aligned}$$

Suppose  $A$  is the passive learning algorithm that guarantees  $S_p(\epsilon, \delta, h)$  passive sample complexities. Then for  $i \in \{1, 2, \dots, k-1\}$ , let  $h_i$  be the classifier returned by  $A(\mathcal{L}_i, \delta)$ .

Additionally, let  $h_k$  be any classifier in  $V$  consistent with the labels in  $\mathcal{L}_k$ .

Finally, for each  $i, j \in \{1, 2, \dots, k\}$ , request the labels of the first  $\lfloor t/(3k^2) \rfloor$  examples in the sequence  $\{x_{\lfloor t/3 \rfloor + n_t + 1}, x_{\lfloor t/3 \rfloor + n_t + 2}, \dots\}$  that satisfy  $h_i(x) \neq h_j(x)$  and let  $R_{ij}$  denote these  $\lfloor t/(3k^2) \rfloor$  labeled examples ( $R_{ij} = \emptyset$  if  $\mathbb{P}_D(h_i(x) \neq h_j(x)) = 0$ ). Let  $m_{ij}$  denote the number of mistakes  $h_i$  makes on the set  $R_{ij}$ . Finally, let  $\hat{h}_t = h_i$  where

$$i = \underset{i}{\operatorname{argmin}} \max_j m_{ij}.$$

This will be the classifier we return.

It is known (see, e.g., [11]) that if  $\lfloor n_t/k \rfloor \geq c'((d/\epsilon) \log(1/\epsilon) + (1/\epsilon) \log(1/\delta))$  for some finite universal constant  $c'$ , then with probability at least  $1 - \delta/8$  over the draw of  $\mathcal{L}_k$ ,  $er(h_k) \leq \epsilon$ . Define

$$\bar{S}_p(\epsilon, \delta, h^*) = \min \left\{ S_p(\epsilon, \delta, h^*), c' \frac{d \log(1/\epsilon) + \log(1/\delta)}{\epsilon} \right\}.$$

We have chosen  $k$  large enough so that, if  $\lfloor n_t/k \rfloor \geq \bar{S}_p(\epsilon, \delta, h^*)$ , then with probability at least  $1 - \delta/8$  over the draw of  $\mathcal{L}$ ,  $\min_i er(h_i) \leq \epsilon$ . Furthermore, by a Hoeffding bound argument (similar to the proof of Theorem 8), for any  $t \geq t_0 = 3k^2 \lceil 72 \ln(16k/\delta) \rceil$ , we have that with probability at least  $1 - \delta/8$ ,  $er(\hat{h}_t) \leq 2 \min_i er(h_i)$ . Define

$$S_a(2\epsilon, \delta/2, h^*) = 1 + \inf \left\{ s \geq t_0 : s \geq 144k \ln \frac{8}{\delta} \bar{S}_p(\epsilon, \delta, h^*) \Delta_s \right\}.$$

Note that if  $t \geq S_a(2\epsilon, \delta/2, h^*)$ , then (with probability  $\geq 1 - \delta/8$ )

$$\bar{S}_p(\epsilon, \delta, h^*) \leq \frac{t}{144k \ln \frac{8}{\delta} \Delta_t} \leq \lfloor n_t/k \rfloor.$$

So, by a union bound over the possible failure events listed above ( $\delta/8$  for  $\mathbb{P}(\text{DIS}(V)) > \Delta_t$ ,  $\delta/8$  for more than  $t/3$  examples of  $\mathcal{L}$  in  $\text{DIS}(V)$ ,  $\delta/8$  for  $\min_i er(h_i) > \epsilon$ , and  $\delta/8$  for  $er(\hat{h}_t) > 2 \min_i er(h_i)$ ), if  $t \geq S_a(2\epsilon, \delta/2, h^*)$ , then with probability at least  $1 - \delta/2$ ,  $er(\hat{h}_t) \leq 2\epsilon$ . So  $S_a(\epsilon, \delta, h^*)$  is a valid sample complexity function, achieved by the described algorithm. Furthermore,

$$S_a(\epsilon, \delta/2, h^*) \leq 1 + \max \left\{ t_0, 144k \ln \frac{8}{\delta} \bar{S}_p(\epsilon/2, \delta, h^*) \Delta_{S_a(\epsilon, \delta/2, h^*) - 2} \right\}.$$

$S_p(\epsilon, \delta, h^*) = \omega(1)$  implies  $S_a(\epsilon, \delta/2, h^*) = \omega(1)$ , so we know that  $\Delta_{S_a(\epsilon, \delta/2, h^*) - 2} = o(1)$ . Thus,  $S_a(\epsilon, \delta/2, h^*) = o(\bar{S}_p(\epsilon/2, \delta, h^*))$ , and thus we have  $S_a(\epsilon, \delta/2, h^*) = o(S_p(\epsilon/2, \delta, h^*))$ . ■

As an interesting aside, it is also true (by essentially the same argument) that under the conditions of Lemma 13, the *verifiable* sample complexity of active learning is strictly smaller than the *verifiable* sample complexity of passive learning in this same sense. In particular, this implies a verifiable sample complexity that is  $o(1/\epsilon)$  under these conditions. For instance, with some effort one can show that these conditions are satisfied when the VC dimension of  $C$

is 1, or when the support of  $D$  is at most countably infinite. However, for more complex learning problems, this condition will typically not be satisfied, and as such we require some additional work in order to use this lemma toward a proof of the general result in Theorem 6. Toward this end, we again turn to the idea of a decomposition of  $C$ , this time decomposing it into subsets satisfying the condition in Lemma 13.

**Lemma 14** *For any  $(C, D)$  where  $C$  has finite VC dimension  $d$ , there exists a countably infinite sequence  $C_1, C_2, \dots$  such that  $C = \cup_{i=1}^{\infty} C_i$  and  $\forall i, \forall h \in C_i, \mathbb{P}(\partial_{C_i} h) = 0$ .*

**Proof:** The case of  $d = 0$  is clear, so assume  $d > 0$ . A decomposition procedure is given in Algorithm 2. We will show that, if we let  $\mathbb{H} = \text{Decompose}(C)$ , then the maximum recursion depth is at most  $d$  (counting the initial call as depth 0). Note that if this is true, then the lemma is proved, since it implies that  $\mathbb{H}$  can be uniquely indexed by a  $d$ -tuple of integers, of which there are at most countably many.

---

**Algorithm 2**  $\text{Decompose}(\mathcal{H})$

---

Let  $\mathcal{H}_\infty = \{h \in \mathcal{H} : \mathbb{P}(\partial_{\mathcal{H}} h) = 0\}$

**if**  $\mathcal{H}_\infty = \mathcal{H}$  **then**

Return  $\{\mathcal{H}\}$

**else**

For  $i \in \{1, 2, \dots\}$ , let  $\mathcal{H}_i =$

$$\{h \in \mathcal{H} : \mathbb{P}(\partial_{\mathcal{H}} h) \in ((1+2^{-(d+3)})^{-i}, (1+2^{-(d+3)})^{1-i})\}$$

Return  $\bigcup_{i \in \{1, 2, \dots\}} \text{Decompose}(\mathcal{H}_i) \cup \{\mathcal{H}_\infty\}$

**end if**

---

For the sake of contradiction, suppose that the maximum recursion depth of  $\text{Decompose}(C)$  is more than  $d$  (or is infinite). Thus, based on the first  $d+1$  recursive calls in one of those deepest paths in the recursion tree, there is a sequence of sets

$$C = \mathcal{H}^{(0)} \supseteq \mathcal{H}^{(1)} \supseteq \mathcal{H}^{(2)} \supseteq \dots \mathcal{H}^{(d+1)} \neq \emptyset$$

and a corresponding sequence of finite positive integers  $i_1, i_2, \dots, i_{d+1}$  such that for each  $j \in \{1, 2, \dots, d+1\}$ , every  $h \in \mathcal{H}^{(j)}$  has

$$\mathbb{P}(\partial_{\mathcal{H}^{(j-1)}} h) \in \left( (1+2^{-(d+3)})^{-i_j}, (1+2^{-(d+3)})^{1-i_j} \right).$$

Take any  $h_{d+1} \in \mathcal{H}^{(d+1)}$ . There must exist some  $r > 0$  such that  $\forall j \in \{1, 2, \dots, d+1\}$ ,

$$\mathbb{P}(\text{DIS}(\bar{B}_{\mathcal{H}^{(j-1)}}(h_{d+1}, r))) \in \left( (1+2^{-(d+3)})^{-i_j}, (1+2^{-(d+2)})(1+2^{-(d+3)})^{-i_j} \right).$$

In particular, any set of  $\leq 2^{d+1}$  classifiers  $T \subset \bar{B}_{\mathcal{H}^{(j)}}(h_{d+1}, r/2)$  must have  $\mathbb{P}(\cap_{h \in T} \partial_{\mathcal{H}^{(j-1)}} h) > 0$ .

We now construct a shattered set of points of size  $d+1$ . Consider constructing a binary tree with  $2^{d+1}$  leaves as follows. The root node contains  $h_{d+1}$  (call this level 0). Let  $h_d \in \bar{B}_{\mathcal{H}^{(d)}}(h_{d+1}, r/4)$  be some classifier with  $\mathbb{P}(h_d(X) \neq h_{d+1}(X)) > 0$ . Let the left child of the root be  $h_{d+1}$  and the right child be  $h_d$  (call this level 1). Define  $A_1 = \{x :$

$h_d(x) \neq h_{d+1}(x)$ , and let  $\Delta_1 = 2^{-(d+2)}\mathbb{P}(A_1)$ . Now for each  $j \in \{d-1, d-2, \dots, 0\}$  in decreasing order, we define the  $d-j+1$  level of the tree as follows. Let  $T_{j+1}$  denote the nodes at the  $d-j$  level in the tree, and let  $A'_{d-j+1} = \bigcap_{h \in T_{j+1}} \partial_{\mathcal{H}^{(j)}} h$ . We iterate over the elements of  $T_{j+1}$  in left-to-right order, and for each one  $h$ , we find  $h' \in B_{\mathcal{H}^{(j)}}(h, \Delta_{d-j})$  with

$$\mathbb{P}_D(h(x) \neq h'(x) \wedge x \in A'_{d-j+1}) > 0.$$

We then define the left child of  $h$  to be  $h$  and the right child to be  $h'$ , and we update

$$A'_{d-j+1} \leftarrow A'_{d-j+1} \cap \{x : h(x) \neq h'(x)\}.$$

After iterating through all the elements of  $T_{j+1}$  in this manner, define  $A_{d-j+1}$  to be the final value of  $A'_{d-j+1}$  and  $\Delta_{d-j+1} = 2^{-(d+2)}\mathbb{P}(A_{d-j+1})$ . The key is that, because every  $h$  in the tree is within  $r/2$  of  $h_{d+1}$ , the set  $A'_{d-j+1}$  always has nonzero measure, and is contained in  $\partial_{\mathcal{H}^{(j)}} h$  for any  $h \in T_{j+1}$ , so there always exists an  $h'$  arbitrarily close to  $h$  with  $\mathbb{P}_D(h(x) \neq h'(x) \wedge x \in A'_{d-j+1}) > 0$ .

Note that for  $i \in \{1, 2, \dots, d+1\}$ , every node in the left subtree of any  $h$  at level  $i-1$  is strictly within distance  $2\Delta_i$  of  $h$ , and every node in the right subtree of any  $h$  at level  $i-1$  is strictly within distance  $2\Delta_i$  of the right child of  $h$ . Since  $2\Delta_i 2^{d+1} = \mathbb{P}(A_i)$ , there must be some set  $A_i^* \subseteq A_i$  with  $\mathbb{P}(A_i^*) > 0$  such that for every  $h$  at level  $i-1$ , every node in its left subtree agrees with  $h$  on every  $x \in A_i^*$  and every node in its right subtree disagrees with  $h$  on every  $x \in A_i^*$ . Therefore, taking any  $\{x_1, x_2, \dots, x_d, x_{d+1}\}$  such that each  $x_i \in A_i^*$  creates a shatterable set (shattered by the set of leaf nodes in the tree). This contradicts VC dimension  $d$ , so we must have that the maximum recursion depth is at most  $d$ . ■

**Proof:**[Theorem 6] Theorem 6 now follows by a simple combination of Lemmas 13 and 14, along with Theorem 8. That is, the passive learning algorithm achieving passive learning sample complexity  $S_p(\epsilon, \delta, h)$  on  $(C, D)$  also achieves  $S_p(\epsilon, \delta, h)$  on any  $(C_i, D)$ , where  $C_1, C_2, \dots$  is the decomposition from Lemma 14. So Lemma 13 guarantees the existence of active learning algorithms  $A_1, A_2, \dots$  such that  $A_i$  achieves a sample complexity  $S_i(\epsilon, \delta/2, h) = o(S_p(\epsilon/2, \delta, h))$  on  $(C_i, D)$  for all  $h \in C_i$  s.t.  $S_p(\epsilon, \delta, h) = \omega(1)$ . Finally, Theorem 8 tells us that this implies the existence of an active learning algorithm based on these  $A_i$  combined with Algorithm 1, achieving sample complexity  $o(S_p(\epsilon/4, \delta, h))$  on  $(C, D)$ . ■

Note there is nothing special about 4 in Theorem 6. Using a similar argument, it can be made arbitrarily close to 1.

---

# Extracting Certainty from Uncertainty: Regret Bounded by Variation in Costs

---

**Elad Hazan**  
IBM Almaden  
650 Harry Rd, San Jose, CA 95120  
hazan@us.ibm.com

**Satyen Kale**  
Microsoft Research  
1 Microsoft Way, Redmond, WA 98052  
sakale@microsoft.com

## Abstract

Prediction from expert advice is a fundamental problem in machine learning. A major pillar of the field is the existence of learning algorithms whose average loss approaches that of the best expert in hindsight (in other words, whose average regret approaches zero). Traditionally the regret of online algorithms was bounded in terms of the number of prediction rounds.

Cesa-Bianchi, Mansour and Stoltz [4] posed the question whether it is possible to bound the regret of an online algorithm by the *variation* of the observed costs. In this paper we resolve this question, and prove such bounds in the fully adversarial setting, in two important online learning scenarios: prediction from expert advice, and online linear optimization.

## 1 Introduction

A cornerstone of modern machine learning are algorithms for prediction from expert advice. The seminal work of Littlestone and Warmuth [12], Vovk [13] and Freund and Schapire [6] gave algorithms which, under fully adversarial cost sequences, attain average cost approaching that of the best expert in hindsight.

To be more precise, consider a prediction setting in which an online learner has access to  $n$  experts. Iteratively, the learner may choose the advice of any expert deterministically or randomly. After choosing a course of action, an adversary reveals the cost of following the advice of the different experts, from which the expected cost of the online learner is derived. The classic results mentioned above give algorithms which sequentially produce randomized decisions, such that the difference between the (expected) cost of the algorithm and the best expert in hindsight grows like  $O(\sqrt{T \log n})$ , where  $T$  is the number of prediction iterations. This extra additive cost is known as the regret of the online learning algorithm.

However, *a priori* it is not clear why online learning algorithms should have high regret (growing with the number of iterations) in an unchanging environment. As an extreme example, consider a setting in which there are only two experts. Suppose that the first expert always incurs cost 1, whereas

the second expert always incurs cost  $\frac{1}{2}$ . One would expect to “figure out” this pattern quickly, and focus on the second expert, thus incurring a total cost that is at most  $\frac{T}{2}$  plus at most a constant extra cost (irrespective of the number of rounds  $T$ ), thus having only constant regret. However, any straightforward application of previously known analyses of expert learning algorithms only gives a regret bound of  $\Theta(\sqrt{T})$  in this simple case (or very simple variations of it).

More generally, the natural bound on the regret of a “good” learning algorithm should depend on *variation* in the sequence of costs, rather than purely on the number of iterations. If the cost sequence has low variation, we expect our algorithm to be able to perform better.

This intuition has a direct analog in the stochastic setting: here, the sequence of experts’ costs are independently sampled from a distribution. In this situation, a natural bound on the rate of convergence to the optimal expert is controlled by the variance of the distribution (low variance should imply faster convergence). This was formalized by Cesa-Bianchi, Mansour and Stoltz [4], who assert that “*proving such a rate in the fully adversarial setting would be a fundamental result*”.

In this paper we prove the first such regret bounds on online learning algorithms in two important scenarios: prediction from expert advice, and the more general framework of online linear optimization. Our algorithms have regret bounded by the variation of the cost sequence, in a manner that is made precise in the following sections. Thus, our bounds are tighter than *all* previous bounds, and hence yield better bounds on the applications of previous bounds (see, for example, the applications in [4]).

### 1.1 Online linear optimization

Online linear optimization [10] is a general framework for online learning which has received much attention recently. In this framework the decision set is an arbitrary bounded, closed, convex set in Euclidean space  $K \subseteq \mathbb{R}^n$  rather than a fixed set of experts, and the costs are determined by adversarially constructed vectors,  $f_1, f_2, \dots \in \mathbb{R}^n$ , such that the cost of point  $x \in K$  is given by  $f_t \cdot x$ . The online learner iteratively chooses a point in the convex set  $x_t \in K$ , and then the cost vector  $f_t$  is revealed and the cost  $f_t \cdot x_t$  is incurred. The performance of online learning algorithms is measured by the regret, which is defined as the difference in the total cost of the sequence of points chosen by the algorithm, viz.

$\sum_{t=1}^T f_t \cdot x_t$ , and the total cost of the least cost fixed point in hindsight, viz.  $\min_{x \in K} \sum_{t=1}^T f_t \cdot x$ .

Several decision problems fit very naturally in this framework. For example, in the online shortest path problem the online learner has to repeatedly choose a path in a given graph from a source node to a destination node. Her cost is the total length of the path according to weights which are chosen by an adversary. This problem can be cast as an online linear optimization problem, where the decision space is the set of all distributions over paths in the graph connecting the source to the destination. Even though this set sits in exponential dimensional Euclidean space, by thinking of a distribution over paths as a flow in the graph, it is possible to efficiently represent the decision space as a polytope in  $\mathbb{R}^{|E|}$  ( $E$  denotes the set of edges in the given graph), described by  $O(|E|)$  constraints, and translate the cost functions to this new domain as well.

The general online linear optimization framework allows for efficient and natural algorithms based on the gradient descent update rule coupled with Euclidean projections [8, 14]. Specifically, we consider Zinkevich’s Lazy Projection algorithm [14]. This algorithm runs online gradient descent on an auxiliary sequence of points and chooses the projections of these auxiliary points on the convex set in every iteration. This algorithm was shown to have regret  $O(\sqrt{T})$ .

The crucial geometric intuition which allows us to prove regret bounds based on the variation of the cost sequence can be summarized by the following intuitive fact: the distance between successive projections for the Lazy Projection algorithm is directly related to the variation of the cost sequence.

We now describe our bounds. Define the variation of the sequence of cost functions to be  $\text{VAR}_T = \sum_{t=1}^T \|f_t - \mu_T^*\|^2$ , where  $\mu_T^* = \frac{1}{T} \sum_{t=1}^T f_t$  is the mean of the sequence. Our analysis of the Lazy Projection algorithm yields the following regret bound:

$$\text{Regret} \leq O(\sqrt{\text{VAR}_T}).$$

## 1.2 Prediction from expert advice

Prediction from expert advice can be cast as a special case of online linear optimization: the decision space is the simplex of all distributions on  $n$  experts. The expectation operator provides a linear cost function on the simplex via the costs of the experts. Hence, our result for online linear optimization already implies variation bounds for regret in the case of prediction from expert advice.

However, this bound is suboptimal, as it depends on the variation of all experts rather than, say, the maximum variation of a single expert. This issue is familiar to learning theorists: “Euclidean algorithms” such as gradient descent attain performance which relates to the Euclidean norm of the cost functions (or variations in our case). While this Euclidean flavor is optimal in certain cases (i.e. when the underlying convex set is the hyper-cube), for certain convex bodies such as the simplex, better performance can be achieved. The multiplicative update algorithms such as EG [11] and FPL\* [10] attain regret which is proportional to  $O(R\sqrt{T \log n})$  where  $R$  is a bound on the  $\ell_\infty$  norm of the cost functions.

By analogy with the online linear optimization case, for a sequence of cost vectors  $f_1, f_2, \dots, f_T \in \mathbb{R}^n$ , where  $f_t(i)$

is the cost of expert  $i$  in the  $t^{\text{th}}$  round, we would expect to be able to bound the regret of online linear optimization over the simplex by something like  $O(\sqrt{\text{VAR}_T^\infty \log n})$ , where

$$\text{VAR}_T^\infty = \max_{i \in n} \left\{ \sum_{t=1}^T |f_t(i) - \mu_T^*(i)|^2 \right\}$$

is the maximum variation in costs amongst the different experts (as before,  $\mu_T^*(i) = \frac{1}{T} \sum_{t=1}^T f_t(i)$  is the mean cost of the  $i^{\text{th}}$  expert). In fact, our bound is even a bit stronger,

$$\text{Regret}(T) = O\left(\sqrt{\text{VAR}_T^{\max} \log n}\right).$$

Here  $\text{VAR}_T^{\max} \leq \text{VAR}_T^\infty$ , and is defined to be

$$\text{VAR}_T^{\max} = \max_{t \leq T} \{\text{VAR}_t(\ell_t)\},$$

where  $\text{VAR}_t(i)$  is the variation in costs of expert  $i$  up to the  $t^{\text{th}}$  round, and  $\ell_t$  is the best expert till the  $t^{\text{th}}$  round.

Whereas for the general online linear optimization we consider the well-known Lazy Projection algorithms and our results are novel by tighter analysis, for the case of prediction from expert advice we need to consider a new algorithm. We can prove that existing variants of the multiplicative weights algorithms do not attain the performance above, and instead consider a different variant of update rule, in which the distribution at time  $t$ , denoted  $x_t$  is defined to be

$$x_t(i) \propto \exp\left(-\eta \sum_{\tau=1}^{t-1} f_\tau(i) - 4\eta^2 \sum_{\tau=1}^{t-1} (f_\tau(i) - \mu_\tau(i))^2\right),$$

where  $\eta$  is a learning rate parameter and  $\mu_t = \frac{1}{t} \sum_{\tau=1}^{t-1} f_\tau$  is the (approximate) mean cost vector at iteration  $t$ . That is, *the distribution over experts explicitly takes into account the variation in their costs*. As far as we know this is a new variant of the multiplicative update algorithms family, and it is necessary to include this feature to prove variation bounds on the regret.

## 1.3 Discussion of the results

Cesa-Bianchi, Mansour and Stoltz [4] discussed desiderata for *fundamental* regret bounds for the expert prediction problem: invariance under translation and rescaling of costs vectors. Invariance under translation implies that the bounds depend only on the effective ranges of the cost vectors in each round, rather than the absolute ranges (by effective range, we mean the maximum difference between the costs in any given round). This is because if any round, the cost vectors are all changed by the same amount, the difference between the expected cost of the algorithm in that round and the cost of any given expert remains the same as without the translation. Our regret bounds enjoy this translation invariance property: this is a direct consequence of the variation bound. This implies, for instance, that it doesn’t matter what sign the costs are, and in fact our bounds are robust enough to handle mixed signs in costs.

Rescaling invariance implies that the bound continues to hold even if all the cost vectors are scaled by the same factor. Again, our regret bounds enjoy rescaling invariance since the regret and the square-root variation scale by the same factors.

We make crucial use of these invariance properties in our analysis; the invariance allows us to normalize the cost vectors in ways that make them easier to reason about.

## 1.4 The stationary stochastic setting vs. an adversary

A point made by [4] is that the variation bounds on the regret essentially match the performance of a natural algorithm in the stochastic setting in which the payoffs are generated by a stationary stochastic process. Let us give a rough sketch of why this is true. Consider a setting of online linear optimization over the unit ball. Suppose that the cost functions are generated by a stationary stochastic process, such that in each iteration the cost function is independently sampled from a fixed distribution with some mean vector  $\mu$ . For a long enough sequence of cost functions drawn from this distribution, the best point in hindsight is essentially the least cost point with respect to the cost vector  $\mu$ .

Let  $\bar{\mu}$  be the observed mean of samples. The natural algorithm uses  $\bar{\mu}$  as proxy for the actual mean and chooses its point with  $\bar{\mu}$  as a cost vector, and this can be shown to be optimal. It is a standard fact that the variance of  $\bar{\mu}$  decreases inversely with the number of samples. Thus, if  $\sigma^2$  is the variance of the distribution, then the variance of  $\bar{\mu}$  after  $t$  iterations is  $\frac{\sigma^2}{t}$ . The expected regret on iteration  $t$  is proportional to the standard deviation  $\frac{\sigma}{\sqrt{t}}$ , and thus the total regret of the optimal predictor is on the order of  $\sum_{t=1}^T \frac{\sigma}{\sqrt{t}} = O(\sqrt{\sigma^2 T}) = O(\sqrt{\text{VAR}_T})$ .

Hence, the optimal achievable regret in this simple setting is proportional to square root of the total variation. In the sequel we prove that the same regret (up to constant factors) can be achieved in the fully adversarial setting, i.e. in a setting in which the cost functions are chosen completely adversarially. In the stationary stochastic setting, the average cost converges to the average optimum cost at a speed that depends on the variance of the distribution: lower variance implies faster convergence. Hence, by proving the variation bounds on the regret, we give strong indication that *online linear optimization in the adversarial setting is as efficient as in the stationary stochastic setting*.

## 1.5 A brief history of prediction

It is incredible that as early as the late fifties, Hannan [7] devised an efficient algorithm for online decision making. Hannan's algorithm proceeds by adding a perturbation to the costs of actions seen so far, and choosing the action with least cost (taking into account the perturbations). He proves that the regret of an online player using his algorithm grows like  $O(\sqrt{T})$  where  $T$  is the number of prediction iterations.

Since then much water has flown under the bridge and many experts have predicted: this includes the aforementioned influential multiplicative update family of algorithms [12, 13, 6], Cover's universal portfolio prediction problem [5] and the extensions of Follow-The-Perturbed-Leader [10] to online optimization and complex decision problems such as online shortest paths. The machine learning community has extended these fundamental results into a beautiful theory of general prediction using Bregman divergences and generalized projections (in order to do justice to the numerous contributors we refer to the credits in the comprehensive book of [3]). This work refined upon the basic regret bound of  $O(\sqrt{T})$ . This refinement, however, deals with the constants multiplying the  $\sqrt{T}$  term.

Freund and Schapire [6] showed that a Multiplicative Weights algorithm based on the Weighted Majority algorithm attains regret bounds of  $O\left(\sqrt{R \sum_{t=1}^T f_t(i^*) \log n}\right)$ , where

it is assumed that all costs are in the range  $[0, R]$ , and  $i^*$  is the best expert in hindsight. In the case when the costs lie in the range  $[-R, R]$ , Allenberg-Neeman and Neeman [1] showed that there is an expert  $i$  such that the regret can be bounded by

$O\left(\sqrt{R \sum_{t=1}^T |f_t(i)| \log n}\right)$ . Most recently Cesa-Bianchi,

Mansour and Stoltz [4] gave the first second-order regret bounds: they proved a bound of  $O\left(\sqrt{A_T^{\max} \log n}\right)$  where  $A_T^{\max} = \max_{t \leq T} \left\{ \sum_{\tau=1}^t f_\tau(\ell_t)^2 \right\}$  is the maximum, over all the time periods  $t$ , of the sum of *squares* of losses up to time  $t$  of the best expert at time  $t$ . They suggest, and indeed as we argue in the previous section it makes intuitive sense, that the it should be possible to get a bound that scales as  $\sqrt{\text{VAR}_T^{\max}}$ .

In this paper we prove their conjecture to be correct, in effect providing the optimal regret bounds up to constant factors.

## 2 Notation and background

The following definitions and derivations may be familiar to experts in learning theory, who may wish to proceed directly to the next section.

In the online linear optimization problem, the decision space is a closed, bounded, convex set  $K \in \mathbb{R}^n$ , and we are sequentially given a series of linear cost functions  $f_t : K \rightarrow \mathbb{R}$  for  $t = 1, 2, \dots$ . With some abuse of notation, we also write the functions as  $f_t(x) = f_t \cdot x$  for some vector  $f_t \in \mathbb{R}^n$ .

The algorithm iteratively produces a point  $x_t \in K$  in every round  $t$ , without knowledge of  $f_t$  (but using the past sequence of cost functions), and incurs the cost  $f_t(x_t)$ . The regret at time  $T$  is defined to be

$$\text{Regret}(f_1, f_2, \dots, f_T) := \sum_{t=1}^T f_t(x_t) - \min_{x \in K} \sum_{t=1}^T f_t(x).$$

Usually, we will drop the cost vectors from the regret notation when they are clear from context. For convenience, we define  $f_0 = 0$ , and let  $F_t = \sum_{\tau=0}^{t-1} f_\tau$ .

We proceed to describe a widely used algorithmic technique in online learning, on the basis of which we will derive our algorithms.

Since our goal is to get regret bounded by the variation in the cost sequence, intuitively, a Follow-The-Leader (FTL) type algorithm, which always chooses the best point so far to use in the next round, should perform well if the variation is low. The FTL algorithm by itself doesn't usually guarantee low regret, mainly because it is inherently unstable: it may swing wildly from one point to another from one iteration to the next at very little provocation (for example, consider the case of expert prediction with 2 experts for the following sequence of cost vectors:  $(1/2, 0)$ ,  $(0, 1)$ ,  $(1, 0)$ ,  $(0, 1), \dots$ ). To make it stable, we add a strictly convex regularization function  $R(x)$  before computing the leader. The generic algorithm which results is shown below, and is called Follow The Regularized Leader (FTRL):

---

**Algorithm 1** FTRL

---

- 1: Let  $K$  be a convex set
  - 2: Input: parameter  $\eta > 0$ , regularization function  $R(x)$ .
  - 3: **for**  $t = 1$  to  $T$  **do**
  - 4:   Use  $x_t \triangleq \min_{x \in K} \left( F_t \cdot x + \frac{1}{\eta} R(x) \right)$ ,
  - 5:   Receive  $f_t$
  - 6: **end for**
- 

A crucial observation regarding the FTRL algorithm which we use in the analysis is its equivalence to the following algorithm, which we call Follow the Lazy Projected Leader (FLPL). This algorithm maintains an auxiliary sequence of points which are updated using a gradient descent type algorithm, which are then projected into the convex set using the Bregman divergence  $B^R$  defined by  $R$ :

$$B^R(x, y) = R(x) - R(y) - \nabla R(y) \cdot (x - y).$$

The algorithm as it is given has an implicit update, whose implementation we ignore for now (in this paper we are only concerned with the Euclidean and Relative Entropy divergences, in which case the updates are efficient).

---

**Algorithm 2** FLPL

---

- 1: Let  $K$  be a convex set
- 2: Input: parameter  $\eta > 0$ , regularizer function  $R(x)$ .
- 3: **for**  $t = 1$  to  $T$  **do**
- 4:   If  $t = 1$ , choose  $y_1$  such that  $\nabla R(y_1) = 0$ .
- 5:   If  $t > 1$ , choose  $y_t$  such that  $\nabla R(y_t) = \nabla R(y_{t-1}) - \eta f_{t-1}$ .
- 6:   Project according to  $B^R$ :

$$x_t = \arg \min_{x \in K} B^R(x, y_t)$$

- 7: **end for**
- 

In fact, the two algorithms above are identical. This is perhaps not surprising, given what is known about the so called “mirror-descent” algorithm (e.g. [3]). Nevertheless this fact is crucial for our later derivations, and we did not find this precise statement elsewhere, hence we include a short proof.

**Lemma 1** *The two algorithms above produce identical predictions, i.e.*

$$\arg \min_{x \in K} \left( F_t \cdot x + \frac{1}{\eta} R(x) \right) = \arg \min_{x \in K} B^R(x, y_t).$$

**Proof:** First, let us observe that the unconstrained optimum  $x^* = \arg \min_{x \in \mathbb{R}^n} \left( F_t \cdot x + \frac{1}{\eta} R(x) \right)$  satisfies

$$F_t + \frac{1}{\eta} \nabla R(x^*) = 0$$

By induction, the above equation is also satisfied for  $y_t$ . Since  $R(x)$  is assumed to be strictly convex, there is only one solution for the above equation and thus  $y_t = x^*$ . Hence,

$$\begin{aligned} B_R(x, y_t) &= R(x) - R(y_t) - \nabla R(y_t) \cdot (x - y_t) \\ &= R(x) - R(y_t) + \eta F_t \cdot (x - y_t). \end{aligned}$$

Since  $R(y_t)$  and  $F_t \cdot y_t$  are constants (i.e. independent of  $x$ ),  $B_R(x, y_t)$  is minimized at the point  $x$  that minimizes  $R(x) + \eta F_t \cdot x$ , which implies that

$$\arg \min_{x \in K} B_R(x, y_t) = \arg \min_{x \in K} \left( F_t \cdot x + \frac{1}{\eta} R(x) \right). \quad \blacksquare$$

One important property which follows from the first characterization of  $x_t$  is the following standard bound on the regret, due to Kalai and Vempala [10], called the Follow-The-Leader/Be-The-Leader (FTL-BTL) inequality:

**Lemma 2** *The regret of the FTRL (or equivalently, the FLPL) algorithm is bounded as:*

$$\text{Regret} \leq \sum_{t=1}^T f_t \cdot (x_t - x_{t+1}) + \frac{1}{\eta} [R(x_T) - R(x_0)].$$

### 3 Algorithms and main results

In this section we describe the algorithms for which we prove variation bounds, and state formally their performance guarantees.

#### 3.1 Online linear optimization

We start by describing our result for online linear optimization. Following the notation defined in the previous section, we assume that  $K \subseteq \mathbb{B}_n$ , where  $\mathbb{B}_n$  is the unit ball in  $\mathbb{R}^n$ , and that  $0 \in K$ . This is without loss of generality, and can be assumed by a suitable scaling and translation of  $K$ . Scaling  $K$  down by its diameter  $D$  makes the diameter 1 and scales the regret down by  $D$  as well, and changing the coordinate system so that  $K$  contains the origin doesn't change the regret bound. Here, we are making use of the translation invariance of our regret bounds.

We also assume that for all  $t$ ,  $\|f_t\| \leq 1$ . If we have some other bound  $R$  on  $\|f_t\|$ , then we scale down the  $f_t$ 's by  $R$  to get new cost vectors  $f'_t$  such that  $\|f'_t\| \leq 1$ . We can then run the algorithm pretending as if  $f'_t$  is the sequence of cost vectors.

Define the variation of sequence of cost vectors  $f_1, \dots, f_T$  to be

$$\text{VAR}_T(f_1, f_2, \dots, f_T) = \sum_{t=1}^T \|f_t - \mu\|^2,$$

where  $\mu = \frac{1}{T} \sum_{t=1}^T f_t$  is the vector that minimizes the above expression. Usually, we will drop the cost vectors from the notation for the variation, refer to it simply as  $\text{VAR}_T$ , when the cost vectors are clear from context. To see that scaling has no effect on the regret bound, note

$$\text{VAR}_T(f'_1, \dots, f'_T) = \frac{1}{R^2} \text{VAR}_T(f_1, \dots, f_T),$$

and

$$\text{Regret}(f'_1, \dots, f'_T) = \frac{1}{R} \text{Regret}(f_1, \dots, f_T).$$

Thus, if  $\text{Regret}(f'_1, \dots, f'_T) = O(\sqrt{\text{VAR}_T(f'_1, \dots, f'_T)})$ , then  $\text{Regret}(f_1, \dots, f_T) = O(\sqrt{\text{VAR}_T(f_1, \dots, f_T)})$ . This is exactly the rescaling invariance discussed earlier.

For ease of notation, we define  $f_0 = 0$ , and for any  $t > 0$ , let  $F_t = \sum_{\tau=0}^{t-1} f_\tau$  and  $\mu_t = \frac{1}{t}F_t = \frac{1}{t}\sum_{\tau=0}^{t-1} f_\tau$ . We instantiate the FTRL/FLPL algorithm with the regularization function  $R(x) = \frac{1}{2}\|x\|^2$ . This regularization was considered many times before, and the only change hereby is to choose a different “learning rate”  $\eta$ , which will enable us to prove the novel regret bounds. Since  $\nabla R(x) = x$  for this regularization, the algorithm that results is:

---

**Algorithm 3** Lazy Projection

---

- 1: Let  $K$  be a convex set
  - 2: Input: parameter  $\eta > 0$ .
  - 3: **for**  $t = 1$  to  $T$  **do**
  - 4:   If  $t = 1$ , choose  $y_1 = 0$ .
  - 5:   If  $t > 1$ , let  $y_t = y_{t-1} - \eta f_{t-1}$ .
  - 6:   Use  $x_t = \arg \min_{x \in K} \|x - y_t\|$ .
  - 7: **end for**
- 

Our main theorem with respect to online linear optimization is:

**Theorem 3** *Let  $f_t$ , for  $t = 1, 2, \dots, T$ , be a sequence of cost vectors to the experts so that  $\|f_t\| \leq 1$ . Setting  $\eta = \min\{2/\sqrt{\text{VAR}_T}, 1/6\}$ , the regret of the Lazy Projection algorithm is bounded by*

$$\text{Regret} \leq 16\sqrt{\text{VAR}_T}.$$

Of course, an upper bound on the total variation  $\text{VAR}_T$  may not be known in advance. Even so, standard  $\eta$ -halving tricks (start with  $\eta = 1/6$ , and halve  $\eta$  as soon as the variation quadruples, and restart the algorithm) immediately give an  $O(\sqrt{\text{VAR}_T})$  regret bound. The formal application of this standard trick is omitted from this extended abstract.

### 3.2 Prediction from expert advice

In the expert learning problem, we assume that we have access to  $n$  experts. In each round  $t$ , we choose a distribution  $x_t$  over the experts and choose an expert from it. Then, we obtain a cost vector  $f_t$  which specifies a cost  $f_t(i)$  for every expert, and we incur the cost of the chosen expert. Our goal is to bound the total expected cost of the algorithm (i.e.  $\sum_{t=1}^T f_t \cdot x_t$ ) relative to the total cost of the expert with minimum total cost in hindsight (i.e.  $\min_i \sum_{t=1}^T f_t(i)$ ).

For simplicity, we assume that all costs  $f_t(i) \in [0, 1]$ . This can be assumed without loss of generality from the rescaling and translation invariance of our final regret bounds. In general, all we need is bound  $R$  on the maximum value of  $f_t(i) - f_t(j)$  over all rounds and all pairs of experts  $i, j$ . In each round, the algorithm can be run by scaling the costs of all experts down by  $R$ , and then subtracting out the minimum cost in each round. As in the case of online linear optimization, this scaling and translation doesn’t affect the square-root variation bound on the regret.

As mentioned before, this setting is a special case of the online linear optimization where the domain  $K$  is the simplex (denoted  $\Delta$ ) of distributions over the experts. To design an algorithm for this special case, we need a different regularization function,  $\text{ne}(x) = \sum_i x_i \ln x_i - x_i$ . The Bregman

divergence which arises from this is the un-normalized relative entropy (c.f Herbster and Warmuth [9]), defined on  $\mathbb{R}_+^n$ , called as follows:

$$D_{\text{ne}}(x, y) := \sum_i y_i \cdot \ln \frac{y_i}{x_i} + y_i - x_i.$$

Note that when  $x, y \in \Delta$ ,  $D_{\text{ne}}(x, y)$  is the relative entropy between  $x$  and  $y$ , and  $\text{ne}(x)$  is the negative entropy of  $x$ . The Bregman projection on the simplex with the un-normalized relative entropy divergence is implemented simply by scaling all the coordinates so that they sum to 1.

A significant twist on the usual multiplicative weights algorithm is that we modify the cost functions to explicitly take into account the variation: we actually run the FTRL/FLPL algorithm on the sequence of cost vectors  $\tilde{f}_1, \tilde{f}_2, \dots$  where

$$\tilde{f}_t(i) = [f_t(i) + 4\eta(f_t(i) - \mu_t(i))^2],$$

where  $\mu_t = \frac{1}{t}\sum_{\tau=0}^{t-1} f_\tau$ . As before, we use the convention that  $f_0 = 0$ .

For ease of notation, for a vector  $x$ , we define the vector  $x^2$  as  $x^2(i) = x(i)^2$ . Thus, we can write  $\tilde{f}_t$  compactly as  $\tilde{f}_t = f_t + 4\eta(f_t - \mu_t)^2$ . The algorithm which results is given below:

---

**Algorithm 4** Variation MW

---

- 1: Input: parameter  $\eta > 0$ .
  - 2: **for**  $t = 1$  to  $T$  **do**
  - 3:   If  $t = 1$ , choose  $y_1 = \vec{1}$ , the all 1’s vector.
  - 4:   If  $t > 1$ , let  $y_t(i) = y_{t-1}(i) \exp(-\eta \tilde{f}_{t-1}(i))$ , where  $\tilde{f}_{t-1}(i) = f_{t-1}(i) + 4\eta(f_{t-1}(i) - \mu_{t-1}(i))^2$ .
  - 5:   Use  $x_t = y_t/Z_t$ , where  $Z_t = \sum_i y_t(i)$ .
  - 6: **end for**
- 

Define

$$\text{VAR}_T^{\text{max}} = \max_{t \leq T} \{\text{VAR}_t(\ell_t)\},$$

where  $\ell_t$  is the best expert till the  $t^{\text{th}}$  round, and  $\text{VAR}_t(i) = \sum_{\tau=1}^t (f_\tau(i) - \mu_\tau^*(i))^2$  where  $\mu_\tau^*(i) = \frac{1}{\tau}\sum_{\sigma=1}^{\tau} f_\sigma(i)$  is the mean cost of the  $i^{\text{th}}$  expert till the  $t^{\text{th}}$  round. Our main result concerning prediction from expert advice is

**Theorem 4** *Let  $f_t$ , for  $t = 1, 2, \dots, T$ , be a sequence of cost vectors to the experts so that  $f_t(i) \in [0, 1]$ . Setting  $\eta = \min\left\{\sqrt{\log(n)/4\text{VAR}_T^{\text{max}}}, 1/10\right\}$ , the regret of the Variation MW algorithm is bounded by*

$$\text{Regret} \leq 8\sqrt{\text{VAR}_T^{\text{max}} \log(n)} + 10 \log(n).$$

Again,  $\eta$ -halving tricks can be used to obtain this result in the case when  $\text{VAR}_T^{\text{max}}$  is not known ahead of time. The additive  $\log(n)$  term is inherent in all expert learning algorithms and also appears in all previously known regret bounds.

## 4 Analysis of the Lazy Projection algorithm

In this section we prove Theorem 3. The proof uses the dual characterization of the FTRL type algorithms introduced previously: on one hand we follow the standard methodology of the Follow-The-Leader type algorithms, bounding the regret by distance between consecutive predictions. On the other hand we use the fact that these predictions are projections of aggregate cost functions, and analyze the distance between successive projections. In fact, this latter analysis is the main crux of the proof - we refine previous approaches by giving a tighter bound on this distance which is based on simple geometrical intuition.

### Proof: (Theorem 3)

In order to aid understanding, we present the proof as a series of lemmas. We defer the proofs of the lemmas to after the present proof. We start by invoking the FTL-BTL inequality (Lemma 2) to obtain the following bound:

#### Lemma 5

$$\text{Regret} \leq \sum_{t=1}^T (f_t - \mu_t) \cdot (x_t - x_{t+1}) + \frac{1}{\eta}.$$

We proceed to relate the distance between successive projections to the variation in the cost vectors. This lemma is the main crux of the proof, and is based on the geometric intuition depicted in Figure 1. The idea in the proof is that if the sequence of cost vectors has low variation, then the cumulative cost vector  $F_t$  is far away from the convex body, and in such a case, the distance between successive projections can be bounded in terms of the length of the component of  $f_t$  orthogonal to  $F_t$ , which can in turn be bounded in terms of  $\|f_t - \mu_t\|$ , since  $\mu_t = \frac{1}{t}F_t$ .

**Lemma 6** For all  $t$ , we have:

$$\|x_t - x_{t+1}\| \leq \frac{3\eta}{2} \|f_t - \mu_t\| + \frac{2}{t}.$$

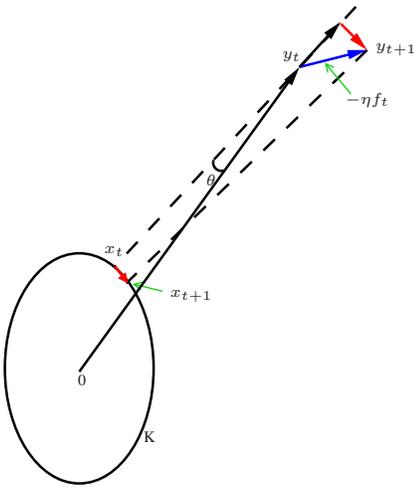


Figure 1: The distance between successive projections, viz.  $\|x_t - x_{t+1}\|$ , is bounded by the length of the component of  $-\eta f_t$  orthogonal to the  $y_t - x_t$ .

For ease of notation, we define a parameter of the cost vectors which will be further used in the analysis:

$$\rho(T) := \sum_{t=1}^T \frac{1}{t} \|f_t - \mu_t\|.$$

This parameter measures the variation of the cost vectors. Using the Cauchy-Schwartz inequality and Lemma 6 we get

$$\begin{aligned} (f_t - \mu_t) \cdot (x_t - x_{t+1}) &\leq \|f_t - \mu_t\| \cdot \left[ \frac{3\eta}{2} \|f_t - \mu_t\| + \frac{2}{t} \right] \\ &\leq \frac{3\eta}{2} \|f_t - \mu_t\|^2 + \frac{2\|f_t - \mu_t\|}{t}. \end{aligned}$$

Plugging this into the regret bound of Lemma 5 gives us the following bound:

$$\text{Regret} \leq \frac{3\eta}{2} \sum_{t=1}^T \|f_t - \mu_t\|^2 + 2\rho(T) + \frac{1}{\eta}. \quad (1)$$

To proceed from here, we use the following Lemma (which, curiously enough, is proved using the analysis of an online learning algorithm that has nothing to do with the present setting!):

**Lemma 7** For any vector  $\mu$ , we have:

$$\sum_{t=1}^T \|f_t - \mu\|^2 \leq \sum_{t=1}^T \|f_t - \mu_t\|^2 + 4\rho(T).$$

Plugging into equation (1) we get that, for any vector  $\mu$  (and in particular, for  $\mu = \mu_T^* := \frac{1}{T} \sum_{t=1}^T f_t$ ),

$$\text{Regret} \leq \frac{3\eta}{2} \sum_{t=1}^T \|f_t - \mu\|^2 + (2 + 6\eta)\rho(T) + \frac{1}{\eta}.$$

We can bound on  $\rho(T)$  as follows:

**Lemma 8** For any vector  $\mu$ , we have:

$$\rho(T) \leq 3 \sqrt{\sum_{t=1}^T \|f_t - \mu\|^2}.$$

Finally, by setting  $\eta = \min\{2/\sqrt{\text{VAR}_T}, 1/6\}$ , the proof is complete.  $\blacksquare$

We now give the omitted proofs of Lemmas used in the above proof.

### Proof: (Lemma 5)

By definition of  $x_t$ , we know that

$$F_t \cdot x_t + \frac{1}{2\eta} \|x_t\|^2 \leq F_t \cdot x_{t+1} + \frac{1}{2\eta} \|x_{t+1}\|^2.$$

Recall that  $\mu_t = F_t/t$ . Hence,

$$\begin{aligned} \sum_{t=1}^T \mu_t \cdot (x_t - x_{t+1}) &= \sum_{t=1}^T \frac{F_t}{t} \cdot (x_t - x_{t+1}) \\ &\leq \sum_{t=1}^T \frac{1}{t} \cdot \frac{1}{2\eta} (\|x_{t+1}\|^2 - \|x_t\|^2) \\ &\leq \frac{1}{2\eta} \sum_{t=2}^T \|x_t\|^2 \cdot \left( \frac{1}{t-1} - \frac{1}{t} \right) + \frac{\|x_{T+1}\|^2}{2\eta T} \\ &\leq \frac{1}{2\eta}. \end{aligned}$$

Here, we use the fact that  $\|x_t\| \leq 1$ . The stated bound then follows from Lemma 2.  $\blacksquare$

**Proof: (Lemma 6)**

We split up the analysis in two cases:

1.  $\|F_t\| \leq 2/\eta$ : Assume that  $\|F_t\| > 0$ . Since  $x_t$  and  $x_{t+1}$  are the projections of  $y_t$  and  $y_{t+1}$  respectively on  $K$ , by the Projection Lemma 9 we have

$$\begin{aligned} \|x_t - x_{t+1}\| &\leq \|y_t - y_{t+1}\| \\ &= \eta \|f_t\| \\ &\leq \eta \|f_t - \mu_t\| + \eta \|\mu_t\| \\ &\leq \eta \|f_t - \mu_t\| + \frac{2\|\mu_t\|}{\|F_t\|} \\ &= \eta \|f_t - \mu_t\| + \frac{2}{t}. \end{aligned}$$

If  $\|F_t\| = 0$ , then the Projection Lemma 9 implies that  $\|x_t - x_{t+1}\| \leq \eta \|f_t\| = \eta \|f_t - \mu_t\|$ , so the stated bound still holds.

2.  $\|F_t\| \geq 2/\eta$ : we first show the following bound:

$$\|x_t - x_{t+1}\| \leq \eta \|f_t - \mu_t\| + \|f_t\|/\|F_t\|. \quad (2)$$

Consider two unit vectors:  $u$  in the direction  $y_t - x_t$ , and  $v$  in the direction  $y_t$ . We claim that the sine of the angle  $\theta$  between these vectors is at most  $1/\eta\|F_t\|$ . To see this, consider the triangle formed by the points  $0, x_t, y_t$ . We are interested in the angle  $\theta$  at vertex  $y_t$  (see Figure 1). Let  $\vartheta$  be the angle at  $x_t$ . By the law of sines, we have

$$\sin(\theta) = \frac{\|x_t\| \sin(\vartheta)}{\|y_t\|} \leq \frac{1}{\|y_t\|} = \frac{1}{\eta\|F_t\|},$$

where the inequality follows because  $\|x_t\| \leq 1$  and  $\sin(\vartheta) \leq 1$ .

Now, we consider the components of  $f_t$  along  $u$  and  $v$ : define  $f_t^u = (f_t \cdot u)u$  and  $f_t^v = (f_t \cdot v)v$ . Consider the point  $y_t - \eta f_t^u$ . Since it lies on the line joining  $y_t$  to  $x_t$ , its projection on  $K$  is also  $x_t$ . Here, we use the fact that  $y_t - \eta f_t^u$  is outside  $K$ : this is because

$$\|y_t - \eta f_t^u\| \geq \|y_t\| - \eta \|f_t^u\| \geq \eta \|F_t\| - \eta \geq 1.$$

By the Projection Lemma 9, we have

$$\|x_{t+1} - x_t\| \leq \|y_{t+1} - (y_t - \eta f_t^u)\| = \eta \|f_t - f_t^u\|. \quad (3)$$

Let  $x$  be the projection of  $f_t^v$  on the subspace spanned by  $u$  (i.e.  $x = (f_t^v \cdot u)u$ ). Then, since  $f_t^u$  is the projection of  $f_t$  in the subspace spanned by  $u$ , it is the closest point to  $f_t$  in the subspace, and since  $x$  is also in the subspace, we have

$$\begin{aligned} \|f_t - f_t^u\| &\leq \|f_t - x\| \\ &\leq \|f_t - f_t^v\| + \|f_t^v - x\| \\ &= \|f_t - f_t^v\| + \|f_t^v\| \sin(\theta) \\ &\leq \|f_t - f_t^v\| + \|f_t\|/\eta \|F_t\| \\ &\leq \|f_t - \mu_t\| + \|f_t\|/\eta \|F_t\|. \end{aligned}$$

The last inequality follows because  $f_t^v$  is the closest point to  $f_t$  in the subspace spanned by  $v$ , and  $\mu_t$  is a point in this subspace. Plugging this bound into (3), we get (2).

Now, we have the following bound on  $\|f_t\|/\|F_t\|$ :

$$\frac{\|f_t\|}{\|F_t\|} \leq \frac{\|f_t - \mu_t\| + \|\mu_t\|}{\|F_t\|} \leq \frac{\eta}{2} \|f_t - \mu_t\| + \frac{1}{t}. \quad (4)$$

Plugging (4) into (2), we get the required bound.  $\blacksquare$

**Proof: (Lemma 7)**

We may assume that  $\|\mu\| \leq 1$ , since the right hand side is minimized at  $\mu = \frac{1}{T} \sum_{t=1}^T f_t$ . The statement of the lemma is essentially bounding the regret of the FTL algorithm played on the sequence of cost functions  $c_t(x) = \|x - f_t\|^2$ , for  $t = 0, 1, 2, \dots, T$ , with the convex domain the unit ball  $\mathbb{B}_n$ . This is because the leader in round  $t$  is

$$\arg \min_{x \in \mathbb{B}_n} \left\{ \sum_{\tau=0}^{t-1} \|x - f_\tau\|^2 \right\} = \frac{1}{t} \sum_{\tau=0}^{t-1} f_\tau = \mu_t.$$

We assume here that the first point played by the algorithm is 0. Then by the FTL-BTL inequality (Lemma 2), the regret of the FTL algorithm can be bounded as (here, the regularization function  $R(x)$  is null):

$$\begin{aligned} \text{Regret} &\leq c_0(0) - c_0(\mu_1) + \sum_{t=1}^T c_t(\mu_t) - c_t(\mu_{t+1}) \\ &\leq \sum_{t=1}^T \nabla c_t(\mu_t) \cdot (\mu_t - \mu_{t+1}) \quad (\because c_t \text{ is convex}) \\ &\leq \sum_{t=1}^T \|\nabla c_t(\mu_t)\| \|\mu_t - \mu_{t+1}\| \\ &\leq \sum_{t=1}^T \|2(f_t - \mu_t)\| \cdot \|\mu_t - \mu_{t+1}\|. \end{aligned}$$

Now, we have

$$\begin{aligned} \|\mu_t - \mu_{t+1}\| &= \left\| \mu_t - \frac{t\mu_t + f_t}{t+1} \right\| \\ &\leq \frac{1}{t+1} (\|\mu_t\| + \|f_t\|) \\ &\leq \frac{2}{t}. \end{aligned}$$

Thus, the regret is bounded by  $4\rho(T)$ . ■

**Proof: (Lemma 8)** We may assume without loss of generality that  $\mu = 0$ : using the vectors  $f_t - \mu$  instead of  $f_t$  doesn't change the value of  $\rho(T)$ . We have

$$\begin{aligned}
\rho(T) &= \sum_{t=1}^T \frac{1}{t} \|f_t - F_t/t\| \\
&\leq \sum_{t=1}^T \frac{1}{t} \left[ \|f_t\| + \frac{1}{t} \|F_t\| \right] \\
&\leq \sum_{t=1}^T \left[ \frac{1}{t} \|f_t\| + \frac{1}{t^2} \sum_{\tau=1}^{t-1} \|f_\tau\| \right] \\
&\leq \sum_{t=1}^T \frac{2}{t} \|f_t\| \quad \left( \because \sum_{\tau=t+1}^T \frac{1}{t^2} \leq \frac{1}{t} \right) \\
&\leq \sqrt{\left[ \sum_{t=1}^T \|f_t\|^2 \right] \left[ \sum_{t=1}^T \frac{4}{t^2} \right]} \quad (\text{Cauchy-Schwarz}) \\
&\leq 3 \sqrt{\sum_{t=1}^T \|f_t\|^2},
\end{aligned}$$

as required. ■

The projection lemma which follows is a well-known fact from convex optimization theory. We include the proof for completeness.

**Lemma 9 (Projection lemma)** Let  $K$  be a convex set, and let  $x$  and  $y$  be any two points. Let  $x'$  and  $y'$  be their respective projections on  $K$ . Then

$$\|x' - y'\| \leq \|x - y\|.$$

**Proof:** Assume that  $x' \neq y'$ , otherwise the inequality is trivial. By the properties of projections on convex sets, we have

$$(x - x') \cdot (y' - x') \leq 0 \text{ and } (y - y') \cdot (x' - y') \leq 0. \quad (5)$$

Consider the line  $\ell$  passing through  $x'$  and  $y'$ , and consider the projections  $x''$  and  $y''$  of  $x$  and  $y$  respectively on this line. The inequalities (5) imply that along  $\ell$ , the order of the points is  $(x'', x', y', y'')$ . Thus, we have

$$\|x' - y'\| \leq \|x'' - y''\| \leq \|x - y\|,$$

where the last inequality follows because the projection of any line segment on any line is no longer than the segment itself. ■

## 5 Analysis of the Variation MW algorithm

The analysis of the Variation MW is straightforward, though complicated somewhat due to heavy algebraic manipulations. We outline the main ideas in the analysis now. Our starting point is Lemma 10, a well-known bound which relates the regret of the Multiplicative Weights algorithm with the expected *squared* losses of the experts (the expectation being taken under the distributions generated by the algorithm).

Next, we make crucial use of the fact that the Multiplicative Weights algorithm puts exponentially higher weight on experts with lower cost than those with higher costs. Since we explicitly factor in the variation in the costs of each expert before computing their exponential weights, eventually the algorithm starts to concentrate all the weight on experts with lower cost and lower variation. This yields the desired regret bound.

We now describe a regret bound on the performance of the Multiplicative Weights algorithm. This bound is well-known (see, for e.g. [4, 2]), we include the short proof for completeness.

**Lemma 10** Suppose in round  $t$  of the expert prediction problem, expert  $i$  incurs cost  $g_t(i)$ , where  $|g_t(i)| \leq M$ . Consider the Multiplicative Weights algorithm, that in round  $t$  chooses expert  $i$  with probability  $x_t(i) \propto \exp(-\eta \sum_{\tau=1}^{t-1} g_\tau(i))$ . Then, if  $\eta \leq 1/M$ ,

$$\text{Regret} \leq \eta \sum_{t=1}^T g_t^2 \cdot x_t + \frac{\log n}{\eta}.$$

**Proof:** Let  $w_t(i) = \exp(-\eta \sum_{\tau=1}^{t-1} g_\tau(i))$ , and let  $Z_t = \sum_i w_t(i)$ . Then the distribution on the experts at time  $t$  is exactly  $w_t/Z_t$ . We think of  $Z_t$  as a potential function, and track how it changes over time. Initially,  $Z_1 = n$ . We have

$$\begin{aligned}
Z_{t+1} &= \sum_i w_t(i) \exp(-\eta g_t(i)) \\
&\leq \sum_i w_t(i) (1 - \eta g_t(i) + \eta^2 g_t(i)^2) \quad (6) \\
&= Z_t (1 - \eta (g_t \cdot x_t) + \eta^2 (g_t^2 \cdot x_t)) \\
&\leq Z_t \exp(-\eta (g_t \cdot x_t) + \eta^2 (g_t^2 \cdot x_t)).
\end{aligned}$$

In (6), we used the fact that for  $|x| \leq 1$ , we have  $\exp(x) \leq 1 + x + x^2$ . Thus, by induction, we have

$$Z_{T+1} \leq n \exp \left( -\eta \sum_{t=1}^T (g_t \cdot x_t) + \eta^2 \sum_{t=1}^T (g_t^2 \cdot x_t) \right).$$

Also, for any expert  $i$  we have the bound

$$Z_{T+1} \geq w_{T+1}(i) = \exp \left( -\eta \sum_{\tau=1}^T g_\tau(i) \right).$$

Putting these two inequalities together, taking logarithms and simplifying, we get the desired bound on the regret. ■

For our analysis, we use a slightly different notion of variation of the experts' costs: for any round  $t$  and any expert  $i$ , define

$$Q_t(i) = \sum_{\tau=1}^{t-1} (f_\tau(i) - \mu_\tau(i))^2.$$

Recall that the usual definition of variation of an experts cost up to the  $t^{\text{th}}$  round is simply

$$\text{VAR}_t(i) = \sum_{\tau=1}^t (f_\tau(i) - \mu_t^*(i))^2,$$

where  $\mu_t^*(i) = \frac{1}{t} \sum_{\tau=1}^t f_\tau(i)$ . But it is easily seen from (the 1 dimensional version of) Lemmas 7 and 8 that

$$Q_t(i) \leq \text{VAR}_t(i) + 12\sqrt{\text{VAR}_t(i)}. \quad (7)$$

and thus  $Q_t(i)$  can serve as a proxy for the true variation (up to constant factors).

Recall that  $\ell_t$  is the best expert till time  $t$ , and  $\text{VAR}_T^{\max} = \max_{t \leq T} \{\text{VAR}_t(\ell_t)\}$ . Define  $Q_T^{\max} = \max_{t \leq T} Q_t(\ell_t)$ . Then, we have that

$$Q_T^{\max} \leq 4\text{VAR}_T^{\max},$$

assuming that  $\text{VAR}_T^{\max} \geq 16$ . Then, the following Lemma combined with inequality (7) implies Theorem 4.

**Lemma 11** *Let  $f_t$ , for  $t = 1, 2, \dots, T$ , be a sequence of cost vectors to the experts so that  $f_t(i) \in [0, 1]$ . Let  $\ell_t$  be the best expert at time  $t$ , and let  $Q$  be an upper bound on  $Q_T^{\max} = \max_{t \leq T} \{Q_t(\ell_t)\}$ . Then setting  $\eta = \min\{\sqrt{\log(n)}/4Q, 1/10\}$ , the regret of the Variation MW algorithm is bounded by*

$$\text{Regret} \leq 4\sqrt{Q \log(n)} + 10 \log(n).$$

**Proof:** Define  $g_t = \tilde{f}_t - \alpha_t \vec{1}$ , where  $\alpha_t = \mu_t(\ell_t) + \frac{4\eta}{t} Q_t(\ell_t)$ , and  $\vec{1}$  is the all 1's vector. Note that for any  $i$ ,

$$\exp\left(-\eta \sum_{\tau=1}^{t-1} g_\tau(i)\right) = \frac{1}{Z} \exp\left(-\eta \sum_{\tau=1}^{t-1} \tilde{f}_\tau(i)\right),$$

where  $Z$  is a scaling constant independent of  $i$ . Hence, scaling either the weights  $\exp(-\eta \sum_{\tau=1}^{t-1} g_\tau(i))$  or the weights  $\exp(-\eta \sum_{\tau=1}^{t-1} \tilde{f}_\tau(i))$  to sum up to 1 yields the same distribution, viz.  $x_t$ .

Since we assumed that the  $f_t(i) \in [0, 1]$ , we conclude that  $g_t(i) \in [-2, 2]$  (since  $4\eta \leq 1$ ). Applying Lemma 10 to the sequence of cost vectors  $g_t$ , we get the following regret bound, where  $\ell_T$  is the final best expert:

$$\sum_{t=1}^T \tilde{f}_t \cdot x_t - \sum_{t=1}^T \tilde{f}_t(\ell_T) \leq \eta \sum_{t=1}^T g_t^2 \cdot x_t + \frac{\log n}{\eta}.$$

Here, we used the fact that the  $\sum_{t=1}^T \alpha_t \vec{1} \cdot x_t = \sum_{t=1}^T \alpha_t$ . Simplifying using the definition of  $\tilde{f}_t$ , we get

$$\begin{aligned} & \sum_{t=1}^T f_t \cdot x_t - \sum_{t=1}^T f_t(\ell_T) \\ & \leq \eta \sum_{t=1}^T g_t^2 \cdot x_t + \frac{\log n}{\eta} \\ & \quad - 4\eta \sum_{t=1}^T (f_t - \mu_t)^2 \cdot x_t + 4\eta \sum_{t=1}^T (f_t(\ell_T) - \mu_t(\ell_T))^2 \\ & \leq \eta \sum_{t=1}^T [g_t^2 - 4(f_t - \mu_t)^2] \cdot x_t + 4\eta(Q+1) + \frac{\log n}{\eta}, \end{aligned} \quad (8)$$

since  $\sum_{t=1}^T (f_t(\ell_T) - \mu_t(\ell_T))^2 \leq Q_T(\ell_T) + 1 \leq Q + 1$ .

The following lemma bounds the first term in (8). The proof is a straightforward calculation, and so we defer its proof to after the present proof.

**Lemma 12** *If  $\eta \leq 1/10$ , then for any  $i$ , we have*

$$g_t^2(i) - 4(f_t(i) - \mu_t(i))^2 \leq 2(\mu_t(i) - \alpha_t)^2.$$

Plugging this bound into (8), we get that

$$\text{Regret} \leq 2\eta \sum_{t=1}^T (\mu_t - \alpha_t \vec{1})^2 \cdot x_t + \frac{\log n}{\eta} + 4\eta(Q+1). \quad (9)$$

We now proceed to bound  $\sum_{t=1}^T (\mu_t - \alpha_t \vec{1})^2 \cdot x_t$ . We bound each term in the summation separately. For any  $t \leq \frac{\log n}{\eta}$ , we simply bound  $|\mu_t(i) - \alpha_t| \leq 2$  and hence we have  $(\mu_t - \alpha_t \vec{1})^2 \cdot x_t \leq 2$ .

Now let  $t > \frac{\log n}{\eta}$ . For convenience of notation, we drop the subscript  $t$  from  $x_t(i)$  and refer to them as  $x(i)$ .

$$\begin{aligned} & (\mu_t - \alpha_t \vec{1})^2 \cdot x \\ & = \sum_{i: \mu_t(i) \leq \alpha_t} (\mu_t(i) - \alpha_t)^2 x(i) + \sum_{i: \mu_t(i) > \alpha_t} (\mu_t(i) - \alpha_t)^2 x(i) \\ & \leq \sum_{i: \mu_t(i) \leq \alpha_t} \left[ \frac{4\eta}{t} Q_t(\ell_t) \right]^2 x(i) + \sum_{i: \mu_t(i) > \alpha_t} (\mu_t(i) - \alpha_t)^2 x(i) \end{aligned} \quad (10)$$

$$\leq \left[ \frac{4\eta}{t} Q_t(\ell_t) \right]^2 + \sum_{i: \mu_t(i) > \alpha_t} (\mu_t(i) - \alpha_t)^2 x(i) \quad (11)$$

Here, (10) follows because when  $\mu_t(i) \leq \alpha_t = \mu_t(\ell_t) + \frac{4\eta}{t} Q_t(\ell_t)$ , we have  $|\mu_t - \alpha_t| \leq \frac{4\eta}{t} Q_t(\ell_t)$  since  $\mu_t(i) \geq \mu_t(\ell_t)$ .

We now bound each term of (11) separately. The proof of the following lemma is a straightforward calculation and we defer it to after the present proof.

**Lemma 13** *The first term of (11), summed over all  $t$ , can be bounded as:*

$$\sum_{t=1}^T \left[ \frac{4\eta}{t} Q_t(\ell_t) \right]^2 \leq 32\eta^2 Q.$$

The hard part is to bound the second term of (11). We now proceed to do so. The intuition in the following analysis is that the Variation MW algorithm tends to concentrate exponentially high weight on the experts that have low cost.

Let  $I$  be the index set of all  $i$  such that  $\mu_t(i) > \alpha_t$ . Note that  $\ell_t \notin I$ . Now, we have  $x(i) \propto \exp(-\eta t \mu_t(i) - 4\eta^2 Q_t(i))$ , and thus  $x(\ell_t) \propto \exp(-\eta t \alpha_t)$ . Thus,  $x(i)$  can be written as:

$$\begin{aligned} x(i) & = \frac{\exp(-\eta t \mu_t(i) - 4\eta^2 Q_t(i))}{\exp(-\eta t \alpha_t) + \sum_{j \neq \ell_t} \exp(-\eta t \mu_t(j) - 4\eta^2 Q_t(j))} \\ & = \frac{\lambda(i) \exp(-\eta t (\mu_t(i) - \alpha_t))}{1 + \sum_{j \neq \ell_t} \lambda(j) \exp(-\eta t (\mu_t(j) - \alpha_t))}, \end{aligned}$$

where  $\lambda(i) = \exp(-4\eta^2 Q_t(i))$ . Note that all  $\lambda(i) \in (0, 1]$ . Define, for all  $i$ ,  $d(i) = (\mu_t(i) - \alpha_t)$ . Note that for  $i \in I$ ,  $d(i) \in [0, 1]$ . Thus, we have

$$\sum_{i \in I} d(i)^2 x(i) = \sum_{i \in I} \frac{\lambda(i) d(i)^2 \exp(-\eta t d(i))}{1 + \sum_{j \neq \ell_t} \lambda(j) \exp(-\eta t d(j))}.$$

To upper bound  $\sum_{i \in I} d(i)^2 x(i)$ , we can neglect the factors in the denominator which depend on  $i \notin I \cup \{\ell_t\}$ ; this only increases the value. Let  $d^I$  and  $\lambda^I$  be the vectors  $d$  and  $\lambda$  restricted to the index set  $I$ . Define the function  $h : (0, 1]^{|I|} \times [0, 1]^{|I|} \rightarrow \mathbb{R}$  as

$$h(\lambda^I, d^I) = \sum_{i \in I} \frac{\lambda(i) d(i)^2 \exp(-\eta t d(i))}{1 + \sum_{j \in I} \lambda(j) \exp(-\eta t d(j))}.$$

This maximum value of this function on its domain gives an upper bound on the expression above.

**Lemma 14** For  $t > \frac{\log n}{\eta}$ , and for any  $(\lambda^I, d^I) \in (0, 1]^{|I|} \times [0, 1]^{|I|}$ , we have

$$h(\lambda^I, d^I) \leq \frac{2 \log^2 n}{\eta^2 t^2}.$$

Putting Lemmas 13 and 14 together, we have that

$$\begin{aligned} \sum_{t=1}^T (\mu_t - \alpha_t \bar{1})^2 \cdot x_t &\leq \sum_{t \leq \frac{\log n}{\eta}} 2 + \sum_{t=1}^T \left[ \frac{4\eta}{t} Q_t(\ell_t) \right]^2 \\ &\quad + \sum_{t > \frac{\log n}{\eta}} \frac{2 \log^2 n}{\eta^2 t^2} \\ &\leq 32\eta^2 Q + \frac{4 \log n}{\eta}. \end{aligned}$$

Plugging this bound into (9), we get

$$\text{Regret} \leq \frac{\log n}{\eta} + 64\eta^3 Q + 8 \log(n) + 4\eta(Q + 1).$$

Now, if we set  $\eta = \{\sqrt{\log n / 4Q}, 1/10\}$ , we get that the regret is bounded by

$$\text{Regret} \leq 4\sqrt{Q} \cdot \log n + 10 \log(n). \quad \blacksquare$$

Again, it may not be possible to get an upper bound on  $Q_T^{\max}$  a priori, but we can use the same  $\eta$ -halving idea (start with  $\eta = 1/10$ , and halve  $\eta$  as soon as this maximum quadruples, and restart the algorithm) and get regret that bounded by

$$\text{Regret} \leq O\left(\sqrt{Q_T^{\max} \log(n)} + \log(Q_T^{\max} \log(n))\right).$$

The details of this bound are standard and are hence omitted from this extended abstract.

We now give the omitted proofs of Lemmas 12, 13, and 14.

**Proof: (Lemma 12)**

We have:

$$\begin{aligned} g_t(i)^2 &= (f_t(i) - \alpha_t + 4\eta(f_t(i) - \mu_t(i))^2)^2 \\ &= (f_t(i) - \alpha_t)^2 + 8\eta(f_t(i) - \alpha_t)(f_t(i) - \mu_t(i))^2 \\ &\quad + 16\eta^2(f_t(i) - \mu_t(i))^4 \\ &\leq (f_t(i) - \alpha_t)^2 + (16\eta + 16\eta^2)(f_t(i) - \mu_t(i))^2 \end{aligned} \quad (12)$$

$$\leq 2(\mu_t(i) - \alpha_t)^2 + (2 + 16\eta + 16\eta^2)(f_t(i) - \mu_t(i))^2 \quad (13)$$

$$\leq 2(\mu_t(i) - \alpha_t)^2 + 4(f_t(i) - \mu_t(i))^2. \quad (14)$$

Here, inequality (12) follows because  $|f_t(i) - \mu_t(j)| \leq 1$  for any  $i, j$ , and  $|f_t(i) - \alpha_t| \leq 2$ , inequality (13) follows from the fact that  $(a+b)^2 \leq 2a^2 + 2b^2$  for any real numbers  $a, b$ , and inequality (14) follows since  $16\eta + 16\eta^2 \leq 2$  if  $\eta \leq 1/10$ . The lemma follows.  $\blacksquare$

**Proof: (Lemma 13)**

Note that for  $t \leq Q$ ,  $Q_t(\ell_t) = \sum_{\tau=1}^{t-1} (f_\tau(i) - \mu_\tau(i))^2 \leq t$ , and for  $t > Q$ ,  $Q_t(\ell_t) \leq Q$ . Thus we have

$$\sum_{t=1}^T \left[ \frac{4\eta}{t} Q_t(\ell_t) \right]^2 \leq 16\eta^2 \cdot \left[ \sum_{t \leq Q} 1^2 + \sum_{t > Q} \frac{Q^2}{t^2} \right] \leq 32\eta^2 Q. \quad \blacksquare$$

**Proof: Lemma 14)**

Let  $S = \{i : d(i) \leq \frac{\log n}{\eta}\}$ , and let  $S' = I \setminus S$ . We upper bound  $h(\lambda^I, d^I)$  as follows:

$$\begin{aligned} h(\lambda^I, d^I) &\leq \sum_{i \in S} \frac{\lambda(i) d(i)^2 \exp(-\eta t d(i))}{\sum_{j \in S} \lambda(j) \exp(-\eta t d(j))} \\ &\quad + \sum_{i \in S'} \lambda(i) d(i)^2 \exp(-\eta t d(i)) \\ &\leq \max_{i \in S'} \left\{ \frac{\lambda(i) d(i)^2 \exp(-\eta t d(i))}{\lambda(i) \exp(-\eta t d(i))} \right\} \end{aligned} \quad (15)$$

$$+ \sum_{i \in S'} \frac{\log^2 n}{\eta^2 t^2} \exp(-\log n) \quad (16)$$

$$\leq \frac{2 \log^2 n}{\eta^2 t^2}.$$

In (15) we use the inequality  $\frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} \leq \max_{i \leq n} \frac{a_i}{b_i}$  for positive reals  $a_i$  and  $b_i$ . In (16), we used the following facts (a)  $\lambda(i) \leq 1$ , and (b) the function  $x^2 \exp(-\eta t x)$  has a negative derivative (and is thus decreasing) when  $x > \frac{2}{\eta t}$ , and thus its maximum over the range  $[\frac{\log n}{\eta t}, 1]$  is obtained at  $\frac{\log n}{\eta t}$ .  $\blacksquare$

## 6 Conclusions and Future Work

In this paper, we investigated the possibility of bounding the regret of online learning algorithms by terms which depend on the variation of the cost sequence, rather than the number of prediction rounds. We analyzed two algorithms, Lazy Projection and Variation MW, and showed that these algorithms obtain variation-bounded regret. Such bounds are significant not only because they show that it is possible to suffer much less regret than previously believed when the cost sequence is particularly benign, but also because they match the regret bounds of natural regret minimizing algorithms in the stochastic setting of independent cost functions from a fixed distribution.

We believe that this work opens up many new directions for future research, all related to bounding the regret in terms of the variation of the cost sequence in the various different scenarios in which regret minimizing algorithms have been devised: bandit settings, strictly convex cost functions, on-line convex optimization and so on. We conjecture in all such scenarios, it is possible to get variation-bounded regret.

Specifically, we conjecture that any dependence on  $T$ , the number of prediction rounds, in the regret bound can be replaced by the same dependence on the variation of the cost sequence. In other scenarios, the variation needs to be defined carefully in settings in which it is not natural or obvious, such as in the case of online convex optimization.

## Acknowledgements

We thank Martin Zinkevich for initial discussions on the possibility of variation bounds on the regret.

## References

- [1] Chamy Allenberg-Neeman and Benny Neeman. Full information game with gains and losses. In *15'th International Conference on Algorithmic Learning Theory*, 2004.
- [2] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2003.
- [3] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [4] Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Mach. Learn.*, 66(2-3):321–352, 2007.
- [5] T. Cover. Universal portfolios. *Math. Finance*, 1:1–19, 1991.
- [6] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.
- [7] James Hannan. Approximation to bayes risk in repeated play. In *M. Dresher, A. W. Tucker, and P. Wolfe, editors, Contributions to the Theory of Games, volume III*, pages 97–139, 1957.
- [8] D. P. Helmbold, J. Kivinen, and M. K. Warmuth. Relative loss bounds for single neurons. *IEEE Transactions on Neural Networks*, 10(6):1291–1304, November 1999.
- [9] Mark Herbster and Manfred K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.
- [10] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.
- [11] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Inf. Comput.*, 132(1):1–63, 1997.
- [12] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.
- [13] V. Vovk. A game of prediction with expert advice. *J. Comput. Syst. Sci.*, 56(2):153–173, 1998.
- [14] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.



---

# Online Learning of Approximate Maximum $p$ -Norm Margin Classifiers with Bias

---

Kosuke Ishibashi, Kohei Hatano and Masayuki Takeda

Department of Informatics, Kyushu University  
744 Motoooka, Nishi-ku, Fukuoka 819-0395, Japan  
{k-ishi, hatano, takeda}@i.kyushu-u.ac.jp

## Abstract

We propose a new online learning algorithm which provably approximates maximum margin classifiers with bias, where the margin is defined in terms of  $p$ -norm distance. Although learning of linear classifiers with bias can be reduced to learning of those without bias, the known reduction might lose the margin and slow down the convergence of online learning algorithms. Our algorithm, unlike previous online learning algorithms, implicitly uses a new reduction which preserves the margin and avoids such possible deficiencies. Our preliminary experiments show that our algorithm runs much faster than previous algorithms especially when the underlying linear classifier has large bias.

## 1 Introduction

Large margin classification methods are quite popular among Machine Learning and related research areas. Various generalization bounds (e.g., [32, 34, 10]) guarantee that linear classifiers with large margin over training data have small generalization error with high probability. The Support Vector Machine (SVM) [5] is one of the most powerful among such methods. The central idea of SVM is to find the maximum 2-norm margin hyperplane over linearly separable data. Further, by using kernels and soft margin formulations, it can learn large margin hyperplane over linearly inseparable data as well. The problem of finding the maximum 2-norm margin hyperplane over data is formulated as a quadratic programming problem. So the task of SVM can be solved in polynomial time by using standard optimization methods.

On the other hand, solving quadratic programming problems is time-consuming, especially for huge data which is now common in many applications. This motivates many researches for making SVM more scalable. One of major approaches is to decompose the original quadratic programming problem into smaller problems which are to solve [28, 29, 16, 8, 17]. Another popular approach is to apply online learning algorithms. Online learning algorithms such as Perceptron [31, 27, 26] and its variants [1, 11, 22, 13] work in iterations, where at each iteration, they process only one instance and update their hypotheses successively. Online learning algorithms use less memory, and are easy to

implement. Many online learning algorithms that find large margin classifiers have been proposed, including, e.g., Kernel Adatron [12], Voted Perceptron [11], Max Margin Perceptron [21], ROMMA [22], ALMA [13], NORMA [19], LASVM [4], MICRA [35], and Pegasos [33].

However, most of these online learning algorithms do not fully exploit the linear separability of data. More precisely, they are designed to learn homogeneous hyperplanes, i.e., hyperplanes that lie on the origin, and they cannot learn linear classifiers with bias directly. So, in order to learn linear classifiers with bias, typical online learning algorithms map instances from the original space  $\mathbb{R}^n$  to an augmented space  $\mathbb{R}^{n+1}$  with an extra dimension by using the mapping  $\phi : \mathbf{x} \mapsto \tilde{\mathbf{x}} = (\mathbf{x}, -R)$ , where  $R$  is the maximum 2-norm of instances [10]. Then, a hyperplane with bias  $(\mathbf{w}, b)$  in the original space corresponds to the hyperplane without bias  $\tilde{\mathbf{w}} = (\mathbf{w}, -b/R)$  in the augmented space since  $\mathbf{w} \cdot \mathbf{x} + b = \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}$ . So, by using this mapping, learning linear classifiers with bias can be reduced to learning those without bias. But, this mapping weakens the guarantee of margin. Suppose that for a sequence of labeled examples  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$  ( $\mathbf{x}_t \in \mathbb{R}^n$  and  $y_t \in \{-1, +1\}$  for  $t = 1, \dots, T$ ), there is a hyperplane with bias  $(\mathbf{u}, b)$  that has margin

$$\gamma = \min_{t=1, \dots, T} \frac{y_t(\mathbf{u} \cdot \mathbf{x}_t + b)}{\|\mathbf{u}\|_2 R},$$

where instances are normalized by  $R$  so as to limit the maximum 2-norm of instances to be one. Then, the corresponding hyperplane  $\tilde{\mathbf{u}} = (\mathbf{u}, -b/R)$  over the augmented space, in which the maximum norm of instances is bounded by  $\tilde{R}$ , has margin

$$\tilde{\gamma} = \frac{y(\tilde{\mathbf{u}} \cdot \tilde{\mathbf{x}})}{\|\tilde{\mathbf{u}}\|_2 \tilde{R}} \geq \frac{y(\mathbf{u} \cdot \mathbf{x} + b)}{2\|\mathbf{u}\|_2 R} = \frac{1}{2}\gamma,$$

since  $\|\tilde{\mathbf{u}}\|_2^2 = \|\mathbf{u}\|^2 + b^2/R^2 \leq 2\|\mathbf{u}\|^2$ , and  $\|\tilde{\mathbf{x}}\|_2^2 \leq 2R$ . Even though the loss of margin is at most by a constant factor, it might cause significant difference in prediction performance over practical applications.

In this paper, we propose a new online learning algorithm that approximately maximizes the margin. Our algorithm, PUMMA (P-norm Utilizing Maximum Margin Algorithm), is an extension of ROMMA [22] in two ways. First, PUMMA can optimize the bias directly by using an implicit reduction from learning of linear classifiers with bias to learning those without bias, instead of using the mapping  $\phi$ .

Second, PUMMA can provably approximate the maximum  $p$ -norm margin classifier for  $p \geq 2$ . A benefit of maximizing  $p$ -norm margin is that we can find sparse linear classifiers quickly. Technically speaking, PUMMA is a variant of  $p$ -norm algorithm [15, 14]. It is known that, if we set  $p = \infty$  or  $p = O(\ln n)$ , the  $p$ -norm algorithm behaves like online multiplicative update algorithms such as Winnow [23], which can converge exponentially faster than Perceptron, when the underlying linear classifier is sparse. For example, if the target concept is a  $k$ -disjunction over  $n$  boolean variables, Winnow can find a consistent hypothesis in  $O(k \ln n)$  mistakes, while Perceptron needs  $\Omega(kn)$  mistakes [20].

We show that PUMMA, given a parameter  $\delta$  ( $0 < \delta \leq 1$ ) and  $p \geq 2$ , finds a linear classifier which has  $p$ -norm margin at least  $(1 - \delta)\gamma$  in  $O(\frac{(p-1)R^2}{\delta^2\gamma^2})$  updates, when there exists a hyperplane with  $p$ -norm margin  $\gamma$  that separates the given sequence of data. The worst-case iteration bound of PUMMA is as the same as those of typical Perceptron-like algorithms when  $p=2$  and that of ALMA [13] for  $p > 2$ , PUMMA is potentially faster than these previous algorithms especially when the underlying linear classifier has large bias.

For linearly inseparable data, PUMMA can use kernels and the 2-norm soft margin formulation [9] for  $p = 2$ , as well as previous Perceptron-like online learning algorithms. Further, we extend PUMMA to deal with 2-norm soft margin formulation for  $p > 2$ . Note that in standard implementations of the SVM [16, 8, 17], the 1-norm soft margin formulation (see, e.g., [10]) is preferred since it often requires less computation time. However, in general, both soft margin formulations are incomparable in terms of generalization ability, which depends on data and choices of kernels. For online-based implementations of the SVM with 1-norm soft margin see LASVM [4] and Pegasos [33].

There are other related works. For  $p = 2$ , previous algorithms such as Kernel Adatron [12], NPA [18], SMO algorithm [29], Max Margin Perceptron [21], and LASVM [4] can find bias directly as well. However, the first three algorithms are not suitable for the online setting since they need to store past examples to compute the bias. Max Margin Perceptron finds the same solution of our algorithm, but its upperbound of updates is  $\ln(R/\gamma)$  times worse than that of PUMMA. For LASVM, there is no theoretical analysis of its convergence rate. For  $p = \infty$ , ROME algorithm [24] is also similar to our present work. It is an online learning algorithm that finds an accurate linear classifier quickly when the margin of the underlying classifier is defined as  $\infty$ -norm distance. On the other hand, ROME requires prior knowledge of the margin and bias. For a more general convex optimization technique which includes ROMMA as a special case, see [3].

In our preliminary experiments, PUMMA converges faster than previous online algorithms over artificial dataset, especially when the underlying linear classifier has large bias. In particular, for  $p = O(\ln n)$ , PUMMA is from 2 to 10 times faster than ALMA. Over real datasets, PUMMA often outperforms previous online algorithms.

## 2 Preliminaries

### 2.1 Norm

For any vector  $\mathbf{x} \in \mathbb{R}^n$  and  $p > 0$ ,  $p$ -norm  $\|\mathbf{x}\|_p$  of  $\mathbf{x}$  is given as  $(\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$ . In particular,  $\|\mathbf{x}\|_\infty$  is given as  $\|\mathbf{x}\|_\infty = \max_i |x_i|$ . It can be shown that, for any fixed  $\mathbf{x} \in \mathbb{R}^n$ , the  $p$ -norm  $\|\mathbf{x}\|_p$  is decreasing with respect to  $p$ , i.e.,  $\|\mathbf{x}\|_{p'} \leq \|\mathbf{x}\|_p$  for any  $0 < p \leq p'$ . For  $p > 1$ ,  $q$ -norm is *dual* to  $p$ -norm if  $1/q = 1 - 1/p$ . For  $p \geq 1$  and  $q$  such that  $1/p + 1/q = 1$ , it is known that

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_p \leq \|\mathbf{x}\|_1 \leq n^{1/p} \|\mathbf{x}\|_\infty.$$

### 2.2 Online learning

We consider the standard setting of online learning of linear classifiers, in which learning proceeds in trials. At each trial  $t$ , the learner receives an instance  $\mathbf{x}_t \in \mathbb{R}^n$ , and it predicts a label  $\hat{y}_t \in \{-1, +1\}$ . Then the learner receives the true label  $y_t \in \{-1, +1\}$  and then it possibly updates its current hypothesis depending on the received label. In this paper, we assume that labels are determined by a linear classifier  $f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$  for some weight vector  $\mathbf{w} \in \mathbb{R}^n$  and bias  $b \in \mathbb{R}$ , where  $\text{sign}(a) = +1$  if  $a \geq 0$ , otherwise  $\text{sign}(a) = -1$ . In particular, if  $y_t \neq \hat{y}_t$ , we say that the learner makes a *mistake*. A typical goal of online learning is to minimize the number of mistakes as small as possible. Most of known online algorithms are *mistake-driven*, that is, they update their hypotheses when they make a mistake.

The  $p$ -norm distance between a hyperplane and a point is computed as follows:

**Lemma 1 ([25])** Let  $V = \{\mathbf{v} \in \mathbb{R}^n \mid \mathbf{w} \cdot \mathbf{v} + b = 0\}$ . Then, for any  $\mathbf{x} \in \mathbb{R}^n$ ,

$$\min_{\mathbf{v} \in V} \|\mathbf{x} - \mathbf{v}\|_p = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|_q},$$

where  $q = 1/(1 - 1/p)$ <sup>1</sup>.

Based on Lemma 1, the  $p$ -norm (geometric) *margin* of a hyperplane  $(\mathbf{w}, b)$  over an example  $(\mathbf{x}, y)$  is defined as

$$\frac{y(\mathbf{w} \cdot \mathbf{x} + b)}{\|\mathbf{w}\|_q}.$$

For any sequence of examples  $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T))$  ( $T \geq 1$ ), the *margin* of a hyperplane  $(\mathbf{w}, b)$  over  $S$  is defined as the minimum margin of examples in  $S$ . The algorithms we consider update their hypotheses if not only they make a mistake, but also their hypotheses have insufficient margin. In this paper, the learner's goal is to minimize the number of updates in order to obtain a linear classifier with approximately maximum  $p$ -norm margin over the given sequence of examples.

### 2.3 Convex duality

We review the basic results on convex analysis. Let  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  be a strictly convex differentiable function. The *Legendre dual* of  $F$ , denoted as  $F^*$ , is defined by

$$F^*(\boldsymbol{\theta}) = \sup_{\mathbf{w} \in \mathbb{R}^n} (\boldsymbol{\theta} \cdot \mathbf{w} - F(\mathbf{w})).$$

<sup>1</sup>More generally, this lemma holds for an arbitrary norm and its dual norm.

It can be verified that  $F^*$  is also strictly convex and differentiable. Then the following lemma holds:

- Lemma 2 ([30, 7])** 1.  $F^{**} = F$ .  
 2.  $F(\mathbf{w}) + F^*(\boldsymbol{\theta}) = \boldsymbol{\theta} \cdot \mathbf{w}$  if and only if  $\boldsymbol{\theta} = \nabla F(\mathbf{w})$ .  
 3.  $\nabla F^* = (\nabla F)^{-1}$ .

In particular, we use  $F(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_q^2$  throughout this paper. Let  $\mathbf{f} = \nabla F$ , that is,

$$\mathbf{f}(\mathbf{w})_i = \frac{\text{sign}(w_i)|w_i|^{q-1}}{\|\mathbf{w}\|_q^{q-2}}$$

By Lemma 2 and some calculations, we obtain the following property.

- Lemma 3 ([14])** 1. The inverse  $\mathbf{f}^{-1}$  of  $\mathbf{f}$  is given as

$$\mathbf{f}^{-1}(\mathbf{w})_i = \frac{\text{sign}(w_i)|w_i|^{p-1}}{\|\mathbf{w}\|_p^{p-2}},$$

where  $1/p + 1/q = 1$ .

2.  $\|\mathbf{f}(\mathbf{w})\|_p = \|\mathbf{w}\|_q$ .  
 3.  $\mathbf{w} \cdot \mathbf{f}(\mathbf{w}) = \|\mathbf{f}(\mathbf{w})\|_p^2 = \|\mathbf{w}\|_q^2$ .

Finally, we will use the following bound later.

**Proposition 1 ([15, 14])** Let  $G(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|_p^2$  with  $p \geq 2$  and let  $\mathbf{g} = \nabla G$ . Then it holds for any  $\mathbf{x}$  and  $\mathbf{a}$  that

$$G(\boldsymbol{\theta} + \mathbf{a}) \leq G(\boldsymbol{\theta}) + \mathbf{g}(\boldsymbol{\theta}) \cdot \mathbf{a} + \frac{(p-1)}{2}\|\mathbf{a}\|_p^2.$$

### 3 PUMMA

We consider the learning of maximum  $p$ -norm margin classifiers in the online learning setting. By Lemma 1, the problem of finding the maximum  $p$ -norm margin hyperplane over a sequence of labeled examples  $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m))$  is formulated as follows:

$$\min_{\mathbf{w}, b} \frac{1}{2}\|\mathbf{w}\|_q^2, \quad (1)$$

subject to :

$$y_t(\mathbf{w} \cdot \mathbf{x}_t + b) \geq 1 \quad (1 \leq t \leq T),$$

where  $q$  is such that  $1/p + 1/q = 1$ . Since the problem (1) is a convex optimization problem with linear inequality constraints, it can be solved by optimization methods such as interior-point methods [6]. However, in the context of online learning, it is time-consuming to solve the problem (1) at each trial. Further, it is necessary to store all the past given examples.

For  $p = 2$ , Li and Long proposed an elegant solution of the problem (1) in the online learning setting [22]. Their algorithm, ROMMA, is an online learning algorithm that finds approximate 2-norm maximum margin hyperplanes without bias. At each trial  $t$ , given an instance  $\mathbf{x}_t$ , ROMMA predicts  $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$  such that

$$\mathbf{w}_t = \arg \min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2, \quad (2)$$

subject to

$$y_{t-1}\mathbf{w} \cdot \mathbf{x}_{t-1} \geq 1 \text{ and } \mathbf{w} \cdot \mathbf{w}_{t-1} \geq \|\mathbf{w}_{t-1}\|_2^2.$$

It can be shown that the constraints of the problem (2) is *relaxed*, that is, the constraints of the problem (2) is weaker than those of the problem (1) when  $p = 2$  and  $b_t$  is fixed with 0. In fact, the second constraint in (2) corresponds to the hyperspace that contains the polyhedron which representing the constraints  $y_j(\mathbf{w} \cdot \mathbf{x}_j) \geq 1$  ( $j = 1, \dots, t-2$ ).

Our algorithm, PUMMA, generalizes ROMMA in two folds: (i) PUMMA can maximize any  $p$ -norm margin with  $p \geq 2$ . (ii) PUMMA can directly learn non-homogeneous hyperplanes. PUMMA takes  $\delta$  ( $0 \leq \delta < 1$ ) and  $p$  ( $p \geq 2$ ) as parameters. For initialization, it requires initial weight vector  $\mathbf{w}_0 = \mathbf{0} \in \mathbb{R}^n$  and positive and negative instances  $\mathbf{x}_1^{pos}$  and  $\mathbf{x}_1^{neg}$ , respectively. These two examples are easily obtained by keep predicting  $-1$  until the first positive example appears and predicting  $+1$  until the first negative example comes. If either a positive or negative example cannot be obtained, then the number of updates is at most 1.

Then, given a sequence  $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{t-1}, y_{t-1}))$  of examples and an instance  $\mathbf{x}_t$ , PUMMA predicts  $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t + b_t)$ , where  $\mathbf{w}_t$  and  $b_t$  is given as follows:

$$(\mathbf{w}_t, b_t) = \arg \min_{\mathbf{w}, b} \frac{1}{2}\|\mathbf{w}\|_q^2, \quad (3)$$

subject to :

$$\mathbf{w} \cdot \mathbf{x}_t^{pos} + b \geq 1, \quad \mathbf{w} \cdot \mathbf{x}_t^{neg} + b \leq -1$$

$$\mathbf{w} \cdot \mathbf{f}(\mathbf{w}_{t-1}) \geq \|\mathbf{w}_{t-1}\|_q^2,$$

where  $q = 1/(1 - 1/p)$ ,  $\mathbf{x}_t^{pos}$  and  $\mathbf{x}_t^{neg}$  are the last positive and negative examples which incur updates, respectively. If  $y_t(\mathbf{w}_t \cdot \mathbf{x}_t + b_t) < 1 - \delta$ , PUMMA $_p(\delta)$  updates  $(\mathbf{x}_{t+1}^{pos}, \mathbf{x}_{t+1}^{neg}) = (\mathbf{x}_t, \mathbf{x}_t^{neg})$ , if  $y_t = +1$ , and  $(\mathbf{x}_{t+1}^{pos}, \mathbf{x}_{t+1}^{neg}) = (\mathbf{x}_t^{pos}, \mathbf{x}_t)$ , otherwise.

#### 3.1 Solution of the optimization problem (3)

Now we show the solution of the optimization problem (3). In this subsection, for simplicity, we denote  $\mathbf{v} = \mathbf{w}_{t-1}$ ,  $\boldsymbol{\theta} = \mathbf{f}(\mathbf{w}_{t-1})$ ,  $\mathbf{x}^{pos} = \mathbf{x}_t^{pos}$  and  $\mathbf{x}^{neg} = \mathbf{x}_t^{neg}$ . Let  $L$  be the Lagrangian, that is,

$$\begin{aligned} L(\mathbf{w}, b, \alpha, \beta) &= \frac{1}{2}\|\mathbf{w}\|_q^2 \\ &+ \sum_{\ell \in \{pos, neg\}} \alpha^\ell \{1 - y^\ell(\mathbf{w} \cdot \mathbf{x}^\ell + b)\} \\ &+ \beta(\|\mathbf{v}\|_q^2 - \boldsymbol{\theta} \cdot \mathbf{w}), \end{aligned} \quad (4)$$

where  $y^{pos} = +1$  and  $y^{neg} = -1$ . Then the partial derivative of  $L$  w.r.t.  $w_i$  and  $b$  is given respectively as

$$\frac{\partial L}{\partial w_i} = \mathbf{f}(\mathbf{w})_i - \sum_{\ell \in \{pos, neg\}} y^\ell \alpha^\ell x_i^\ell - \beta \theta_i, \text{ and} \quad (5)$$

$$\frac{\partial L}{\partial b} = \alpha^{pos} - \alpha^{neg}. \quad (6)$$

Since the solution  $(\mathbf{w}^*, b^*)$  must enforce the partial derivatives (5) and (6) to be zero, the vector  $\mathbf{w}^*$  is specified as

$$\mathbf{w}^* = \mathbf{f}^{-1}(\alpha \mathbf{z} + \beta \boldsymbol{\theta}),$$

where  $\alpha = \alpha^{pos} = \alpha^{neg}$ ,  $\mathbf{z} = \mathbf{x}^{pos} - \mathbf{x}^{neg}$  and

$$\mathbf{f}^{-1}(\boldsymbol{\theta})_i = \frac{\text{sign}(\theta_i)|\theta_i|^{p-1}}{\|\boldsymbol{\theta}\|_p^{p-2}}.$$

**PUMMA**  $_p(\delta)$ **begin**

1. (Initialization) Get examples  $(\mathbf{x}_1^{pos}, +1)$  and  $(\mathbf{x}_1^{neg}, -1)$ . Let  $\mathbf{w}_0 = (0, \dots, 0) \in \mathbb{R}^n$ .

2. For  $t = 1$  to  $T$ ,

(a) Receive an instance  $\mathbf{x}_t$ .

(b) Let

$$(\mathbf{w}_t, b_t) = \arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|_q^2,$$

subject to :

$$(\mathbf{w} \cdot \mathbf{x}_t^{pos} + b) \geq 1$$

$$(\mathbf{w} \cdot \mathbf{x}_t^{neg} + b) \leq -1$$

$$\mathbf{w} \cdot \mathbf{f}(\mathbf{w}_{t-1}) \geq \|\mathbf{w}_{t-1}\|_q^2.$$

(c) Predict  $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t + b_t)$ .

(d) Receive the label  $y_t$ . If  $y_t(\mathbf{w}_t \cdot \mathbf{x}_t + b_t) < 1 - \delta$ , update

$$(\mathbf{x}_{t+1}^{pos}, \mathbf{x}_{t+1}^{neg}) = \begin{cases} (\mathbf{x}_t, \mathbf{x}_t^{neg}) & , (y_t = +1) \\ (\mathbf{x}_t^{pos}, \mathbf{x}_t) & , (y_t = -1). \end{cases}$$

Otherwise, let

$$(\mathbf{x}_{t+1}^{pos}, \mathbf{x}_{t+1}^{neg}) = (\mathbf{x}_t^{pos}, \mathbf{x}_t^{neg}).$$

**end.**

Figure 1: The description of PUMMA .

Further, by KKT conditions, the parameters  $\alpha$  and  $\beta$  satisfy that

$$\alpha(1 - \mathbf{w}^* \cdot \mathbf{x}^{pos} - b^*) = 0, \quad (7)$$

$$\alpha(1 + \mathbf{w}^* \cdot \mathbf{x}^{neg} + b^*) = 0, \quad (8)$$

$$1 - \mathbf{w}^* \cdot \mathbf{x}^{pos} - b^* \leq 0, \quad (9)$$

$$1 + \mathbf{w}^* \cdot \mathbf{x}^{neg} + b^* \leq 0, \quad (10)$$

$$\alpha \geq 0, \quad (11)$$

$$\beta(\|\mathbf{v}\|_q^2 - \mathbf{w}^* \cdot \boldsymbol{\theta}) = 0, \quad (12)$$

$$\|\mathbf{v}\|_q^2 - \mathbf{w}^* \cdot \boldsymbol{\theta} \leq 0, \quad (13)$$

$$\text{and } \beta \geq 0. \quad (14)$$

We show that  $\alpha > 0$  by contradiction. Assuming that  $\alpha = 0$ , we have  $\mathbf{w}^* = \mathbf{f}(\beta\boldsymbol{\theta}) = \beta\mathbf{v}$ . Then the conditions (12), (13) and (14) implies  $\beta = 1$  and thus  $\mathbf{w}^* = \mathbf{v}$ . However, the conditions (9) or (10) cannot be satisfied for  $\mathbf{w}^* = \mathbf{v}$ , which is a contradiction.

Now we consider two cases. (i) Suppose that  $\beta = 0$ . Then, since  $\alpha > 0$  and the conditions (7) and (8) hold, the vector  $\mathbf{w}^*$  is given as

$$\mathbf{w}^* = \alpha \mathbf{f}^{-1}(\mathbf{z}), \quad (15)$$

where  $\alpha = 2/\|\mathbf{z}\|_p^2$ . (ii) Otherwise, i.e., if  $\beta > 0$ , by the conditions (7), (8), and (12),

$$\mathbf{w}^* = \mathbf{f}^{-1}(\alpha\mathbf{z} + \beta\mathbf{v}), \quad (16)$$

where  $\alpha$  and  $\beta$  where  $\alpha$  and  $\beta$  satisfies the following equations

$$\begin{cases} \mathbf{f}^{-1}(\alpha\mathbf{z} + \beta\boldsymbol{\theta}) \cdot \mathbf{z} = 2, \\ \mathbf{f}^{-1}(\alpha\mathbf{z} + \beta\boldsymbol{\theta}) \cdot \boldsymbol{\theta} = \|\mathbf{v}\|_q^2. \end{cases} \quad (17)$$

That is, the optimal solution  $\mathbf{w}^*$  satisfies the constraints of the problem(3) with equality. In this case, the solution can be obtained by maximizing its Lagrange dual  $L^*$  which is defined as

$$L^*(\alpha, \beta) = \min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha, \beta).$$

Further, with some calculations,  $L^*$  is computed as

$$L^*(\alpha, \beta) = -\frac{1}{2} \|\alpha\mathbf{z} + \beta\boldsymbol{\theta}\|_p^2 + 2\alpha + \beta \|\boldsymbol{\theta}\|_p^2. \quad (18)$$

Then, Note that the partial derivatives of  $L^*$  are

$$\frac{\partial L^*}{\partial \alpha} = -\mathbf{f}^{-1}(\alpha\mathbf{z} + \beta\boldsymbol{\theta}) \cdot \mathbf{z} + 2$$

$$\frac{\partial L^*}{\partial \beta} = -\mathbf{f}^{-1}(\alpha\mathbf{z} + \beta\boldsymbol{\theta}) \cdot \boldsymbol{\theta} + \|\boldsymbol{\theta}\|_p^2.$$

Since  $L^*$  is concave, the equations (17) is satisfied if and only if  $L^*$  is maximized. So, given an initial assignment  $(\alpha_0, \beta_0)$ , we can approximate  $(\alpha, \beta)$  by repeating the Newton update

$$\begin{pmatrix} \alpha_{k+1} \\ \beta_{k+1} \end{pmatrix} = \begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} - \nabla^2 L^*(\alpha, \beta)^{-1} \nabla L^*(\alpha_k, \beta_k)$$

for sufficiently many steps, where

$$\frac{\partial^2 L^*}{\partial \alpha^2} = \sum_i \mathbf{f}^{-1'}(\alpha\mathbf{z} + \beta\boldsymbol{\theta})_i z_i^2,$$

$$\frac{\partial^2 L^*}{\partial \beta \partial \alpha} = \sum_i \mathbf{f}^{-1'}(\alpha\mathbf{z} + \beta\boldsymbol{\theta})_i z_i \theta_i,$$

$$\frac{\partial^2 L^*}{\partial \alpha \partial \beta} = \sum_i \mathbf{f}^{-1'}(\alpha\mathbf{z} + \beta\boldsymbol{\theta})_i z_i \theta_i,$$

$$\frac{\partial^2 L^*}{\partial \beta^2} = \sum_i \mathbf{f}^{-1'}(\alpha\mathbf{z} + \beta\boldsymbol{\theta})_i \theta_i^2,$$

and

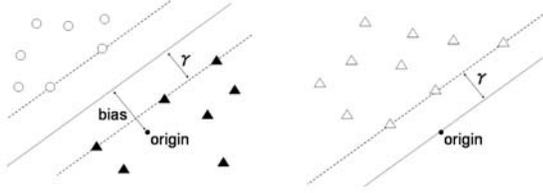
$$\begin{aligned} \mathbf{f}^{-1'}(\boldsymbol{\theta})_i &= \frac{\partial \mathbf{f}^{-1}(\boldsymbol{\theta})}{\partial \theta_i} \\ &= -(p-2) \frac{|\theta_i|^{2(p-1)}}{\|\boldsymbol{\theta}\|_p^{2p-2}} + (p-1) \frac{|\theta_i|^{p-2}}{\|\boldsymbol{\theta}\|_p^{p-2}}. \end{aligned}$$

In our implementation, we set initial values as  $\alpha_0 = 0$  and  $\beta_0 = 1$ .

In particular, for  $p = 2$ , it holds that  $\mathbf{f}(\mathbf{x}) = \mathbf{f}^{-1} = \mathbf{x}$ . So, we have the following analytical solution for equations (17):

$$\begin{aligned} \alpha &= \frac{\|\mathbf{v}\|^2(2 - \mathbf{v} \cdot \mathbf{z})}{\|\mathbf{v}\|^2 \|\mathbf{z}\|^2 - (\mathbf{v} \cdot \mathbf{z})^2} \text{ and} \\ \beta &= \frac{\|\mathbf{v}\|^2 \|\mathbf{z}\|^2 - 2(\mathbf{v} \cdot \mathbf{z})}{\|\mathbf{v}\|^2 \|\mathbf{z}\|^2 - (\mathbf{v} \cdot \mathbf{z})^2}. \end{aligned} \quad (19)$$

Figure 2: Illustration of the implicit reduction which preserves the margin. Each pair of positive and negative examples in the original space (left) corresponds to a positive example in the new space (right).



As a summary, in order to obtain the solution  $\mathbf{w}^*$ , we first assume the case (i) and check whether the condition  $\mathbf{w}^* \cdot \boldsymbol{\theta} > \|\mathbf{v}\|_q^2$  holds or not. If it does, the solution is given as (15). Otherwise, the case (ii) holds and the solution is (19) for  $p = 2$ , or we apply Newton method for  $p > 2$ .

In either case (i) or (ii), the bias  $b^*$  is given as

$$b^* = -\frac{\mathbf{w}^* \cdot \mathbf{x}^{pos} + \mathbf{w}^* \cdot \mathbf{x}^{neg}}{2}. \quad (20)$$

### 3.2 Implicit reduction to learning classifiers without bias

We show an interpretation of PUMMA from the viewpoint of reduction. Let us fix  $p = 2$ . Then, it is easily verified that the update of PUMMA is identical to that of ROMMA for the instance  $\mathbf{z} = (\mathbf{x}_t^{pos} - \mathbf{x}_t^{neg})/2$  whose label is positive. This observation implies a reduction from learning linear classifiers with bias to learning of those without bias. Let  $\mathcal{X} = \mathcal{X}^{pos} \cup \mathcal{X}^{neg}$  be a subset of  $\mathbb{R}^n$ , where  $\mathcal{X}^{pos}$  and  $\mathcal{X}^{neg}$  are positive and negative set of instances and  $\mathcal{X}^{pos} \cap \mathcal{X}^{neg} = \emptyset$ . Assume that there exists  $(\mathbf{u}, b)$  such that  $\mathbf{u} \cdot \mathbf{x}^{pos} + b \geq 1$  for each  $\mathbf{x}^{pos} \in \mathcal{X}^{pos}$ , and  $\mathbf{u} \cdot \mathbf{x}^{neg} + b \leq -1$  for each  $\mathbf{x}^{neg} \in \mathcal{X}^{neg}$ . Then we consider the set

$$\mathcal{Z} = \left\{ \frac{\mathbf{x}^{pos} - \mathbf{x}^{neg}}{2} \mid \mathbf{x}^{pos} \in \mathcal{X}^{pos}, \mathbf{x}^{neg} \in \mathcal{X}^{neg} \right\}.$$

That is, from a set of positive and negative instances, we define the set of positive instances. Then, the following property holds for  $\mathcal{Z}$ .

**Theorem 2** Fix any  $p$  satisfying  $2 \leq p < \infty$ . Let  $(\mathbf{u}, b)$  be the maximum  $p$ -norm hyperplane over  $\mathcal{X}$ . Then,  $\mathbf{u}$  is the maximum  $p$ -norm hyperplane over  $\mathcal{Z}$  as well. Also, the opposite holds for some  $b$ .

**Proof:** Let  $\mathbf{u}'$  be the maximum  $p$ -norm hyperplane over  $\mathcal{Z}$ . Note that  $\mathbf{u} \cdot \mathbf{z} \geq 1$  for each  $\mathbf{z} \in \mathcal{Z}$  (See Figure 2). So, we have  $\|\mathbf{u}\|_q^2 \geq \|\mathbf{u}'\|_q^2$  for  $q$  s.t.  $1/p + 1/q = 1$ . Now let  $b' = \mathbf{u}' \cdot (\tilde{\mathbf{x}}^{pos} + \tilde{\mathbf{x}}^{neg})/2$ , where  $\tilde{\mathbf{x}}^{pos}$  and  $\tilde{\mathbf{x}}^{neg}$  satisfies  $\mathbf{u}' \cdot (\tilde{\mathbf{x}}^{pos} - \tilde{\mathbf{x}}^{neg}) = 2$ , for any  $\mathbf{x}^{pos} \in \mathcal{X}^{pos}$ . Note that such a pair  $(\tilde{\mathbf{x}}^{pos}, \tilde{\mathbf{x}}^{neg})$  always exists since  $\mathbf{u}'$  is the maximum

$p$ -norm margin hyperplane. Then, we have

$$\begin{aligned} \mathbf{u}' \cdot \mathbf{x}^{pos} + b' &= \mathbf{u}' \cdot \tilde{\mathbf{x}}^{pos} + b' + \mathbf{u}' \cdot (\mathbf{x}^{pos} - \tilde{\mathbf{x}}^{pos}) \\ &= \frac{\mathbf{u}' \cdot (\tilde{\mathbf{x}}^{pos} - \tilde{\mathbf{x}}^{neg})}{2} + \mathbf{u}' \cdot (\mathbf{x}^{pos} - \tilde{\mathbf{x}}^{pos}) \\ &= 1 + \mathbf{u}' \cdot (\mathbf{x}^{pos} - \tilde{\mathbf{x}}^{neg} - \tilde{\mathbf{x}}^{pos} + \tilde{\mathbf{x}}^{neg}) \\ &\geq 1 + 2 - 2 = 1. \end{aligned}$$

Similarly, it holds for any  $\mathbf{x}^{neg} \in \mathcal{X}^{neg}$  that  $\mathbf{u}' \cdot \mathbf{x}^{neg} + b' \leq -1$ . So, we get  $\|\mathbf{u}'\|_q^2 \geq \|\mathbf{u}\|_q^2$ . Finally, since the function  $\|\cdot\|_q^2$  ( $1 < q \leq 2$ ) is strictly convex, the minimum is unique. Therefore we obtain  $\mathbf{u} = \mathbf{u}'$ .  $\blacksquare$

This theorem ensures that finding the maximum margin hyperplane with bias can be reduced to finding those without bias over pairs of positive and negative instances. Observe that this reduction does not reduce the margin.

PUMMA can be viewed as a ‘‘wrapper’’ algorithm of ROMMA equipped with this reduction. Given positive and negative instances  $\mathbf{x}^{pos}$  and  $\mathbf{x}^{neg}$ , PUMMA constructs a positive instance  $\mathbf{z} = (\mathbf{x}^{pos} - \mathbf{x}^{neg})/2$  and train ROMMA with  $\mathbf{z}$  for a trial. Then PUMMA receives a weight vector  $\mathbf{w}$  and set bias  $b$  as  $b = -(\mathbf{w} \cdot (\mathbf{x}^{pos} + \mathbf{x}^{neg}))/2$ . If PUMMA makes a mistake (or does not have enough margin) over a new instance, it updates  $\mathbf{z}$  and train ROMMA again.

It is possible to use any online learning algorithm that finds maximum margin linear classifier without bias as sub-routines if it satisfies the following requirement: such an algorithm must output a weight vector whose support vector is  $\mathbf{z}$ . However, most of known online algorithms maximizing the margin does not satisfy this requirement and ROMMA seems to be the only one satisfying the requirement so far.

### 3.3 Convergence proof

We prove an upperbound of updates made by PUMMA. First of all, by the KKT conditions for equations (7) and (8), the following property holds:

**Lemma 4** For  $t \geq 1$ , it holds that

$$\mathbf{w}_t \cdot \mathbf{x}_t^{pos} + b_t = 1 \quad \text{and} \quad \mathbf{w}_t \cdot \mathbf{x}_t^{neg} + b_t = -1.$$

Then we prove that the optimal solution of the offline optimization problem (1) is a feasible solution of the PUMMA’s optimization problem (3).

**Lemma 5** Let  $(\mathbf{u}, b) \in \mathbb{R}^n \times \mathbb{R}$  be a hyperplane such that  $y_j(\mathbf{u} \cdot \mathbf{x}_j + b) \geq 1$  for  $j = 1, \dots, t$ . Then, it holds that  $\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_t) \geq \|\mathbf{w}_t\|_q^2$  and  $\|\mathbf{u}\|_q \geq \|\mathbf{w}_t\|_q$ .

**Proof:** For convenience of the proof, we denote  $\boldsymbol{\theta}_t = \mathbf{f}(\mathbf{w}_t)$ . Without loss of generality, we can assume that an update is made at each trial  $t \geq 1$ . The proof for the first inequality is done by induction on  $t$ . For  $t = 1$ , the vector is written as  $\mathbf{w}_1 = \mathbf{f}^{-1}(\boldsymbol{\theta}_1)$ , where  $\boldsymbol{\theta}_1 = \alpha(\mathbf{x}_1^{pos} - \mathbf{x}_1^{neg})$  for some  $\alpha \geq 0$ . By the definition of  $\mathbf{u}$  and  $b$ , it holds that  $\mathbf{u} \cdot \mathbf{x}_1^{pos} + b \geq 1$  and  $\mathbf{u} \cdot \mathbf{x}_1^{neg} + b \leq -1$ , respectively. So, we obtain

$$\begin{aligned} \mathbf{u} \cdot \boldsymbol{\theta}_1 &= \alpha(\mathbf{u} \cdot \mathbf{x}_1^{pos} - \mathbf{u} \cdot \mathbf{x}_1^{neg}) \\ &\geq \alpha(1 - b + 1 + b) = 2\alpha. \end{aligned}$$

On the other hand, by Lemma 4, we have

$$\|\mathbf{w}_1\|_q^2 = \mathbf{w}_1 \cdot \boldsymbol{\theta}_1 = \alpha \mathbf{w}_1 \cdot (\mathbf{x}_1^{pos} - \mathbf{x}_1^{neg}) = 2\alpha,$$

which shows  $\mathbf{u} \cdot \boldsymbol{\theta}_1 \geq \|\mathbf{w}_1\|_q^2$ .

Suppose that for  $t < t'$ , the statement is true. Then, there are two cases: (i)  $\mathbf{w}_{t'} \cdot \boldsymbol{\theta}_{t'-1} = \|\mathbf{w}_{t'-1}\|_q^2$ , and  $\mathbf{w}_{t'} = \mathbf{f}^{-1}(\boldsymbol{\theta}_{t'})$ , where  $\boldsymbol{\theta}_{t'} = \alpha(\mathbf{x}_{t'}^{pos} - \mathbf{x}_{t'}^{neg}) + \beta \boldsymbol{\theta}_{t'-1}$  for some  $\alpha$  and  $\beta$ , or (ii)  $\mathbf{w}_{t'} \cdot \boldsymbol{\theta}_{t'-1} > \|\mathbf{w}_{t'-1}\|_q^2$ , and  $\mathbf{w}_{t'} = \mathbf{f}^{-1}(\boldsymbol{\theta}_{t'})$ , where  $\boldsymbol{\theta}_{t'} = \alpha(\mathbf{x}_t^{pos} - \mathbf{x}_t^{neg})$ . For the case (ii), the proof follows the same argument for  $t = 1$ , so we only consider the case (i). By the inductive assumption, we have

$$\begin{aligned} \mathbf{u} \cdot \boldsymbol{\theta}_{t'} &= \alpha(\mathbf{u} \cdot \mathbf{x}_{t'}^{pos} - \mathbf{u} \cdot \mathbf{x}_{t'}^{neg}) + \beta \mathbf{u} \cdot \boldsymbol{\theta}_{t'-1} \\ &\geq 2\alpha + \beta \|\mathbf{w}_{t'-1}\|_q^2 \end{aligned}$$

By Lemma 4,

$$\begin{aligned} \|\mathbf{w}_{t'}\|_q^2 &= \mathbf{w}_{t'} \cdot \boldsymbol{\theta}_{t'} \\ &= \mathbf{w}_{t'} \cdot \alpha(\mathbf{x}_{t'}^{pos} - \mathbf{x}_{t'}^{neg}) + \beta \mathbf{w}_{t'} \cdot \boldsymbol{\theta}_{t'-1} \\ &= 2\alpha + \beta \|\mathbf{w}_{t'-1}\|_q^2. \end{aligned}$$

So, we get  $\mathbf{u} \cdot \boldsymbol{\theta}_{t'} \geq \|\mathbf{w}_{t'}\|_q^2$  and thus we prove the first inequality. The second inequality holds immediately since both  $(\mathbf{u}, b)$  and  $(\mathbf{w}_t, b_t)$  satisfy the same constraints in (3) and  $(\mathbf{w}_t, b_t)$  minimizes the norm by definition. ■

Next, prove the following lemma:

**Lemma 6** For each trial  $t \geq 1$  in which an update is incurred,

$$\|\mathbf{w}_{t+1}\|_q^2 - \|\mathbf{w}_t\|_q^2 \geq \frac{\delta^2}{2(p-1)R^2},$$

where  $R = \max_{j=1, \dots, t} \|\mathbf{x}_j\|_p$ .

**Proof:** By the weak duality theorem (see, e.g., [6]), the optimum of the problem (3) is bounded below by the Lagrangian dual  $L^*(\alpha, \beta)$  in (18) for any  $\alpha \geq 0$  and  $\beta \geq 0$ . Therefore, using the notations in the derivation of update,

$$\frac{1}{2} \|\mathbf{w}^*\|_q^2 - \frac{1}{2} \|\mathbf{v}\|_q^2 \geq L^*(\alpha, \beta) - \frac{1}{2} \|\mathbf{v}\|_q^2.$$

So, by using Proposition 1 and letting  $\beta = 1$ , we have

$$\begin{aligned} &L^*(\alpha, 1) - \frac{1}{2} \|\mathbf{v}\|_q^2 \\ &= -\mathbf{G}(\boldsymbol{\theta} + \alpha \mathbf{z}) + \mathbf{G}(\boldsymbol{\theta}) + 2\alpha \\ &\geq -\mathbf{g}(\boldsymbol{\theta}) \cdot \alpha \mathbf{z} - \frac{(p-1)}{2} \alpha^2 \|\mathbf{z}\|_p^2 + 2\alpha \\ &= -\alpha \mathbf{v} \cdot \mathbf{z} - \frac{(p-1)}{2} \alpha^2 \|\mathbf{z}\|_p^2 + 2\alpha. \end{aligned}$$

The right hand side of the inequality above is maximized if

$$\alpha = \frac{2 - \mathbf{v} \cdot \mathbf{z}}{(p-1)\|\mathbf{z}\|_p^2}. \quad (21)$$

Note that  $\alpha$  is positive since  $\mathbf{v} \cdot \mathbf{z} \leq 2 - \delta$ . Substiting (21),

$$L^*(\alpha, 1) - \frac{1}{2} \|\mathbf{v}\|_q^2 \geq \frac{(2 - \mathbf{v} \cdot \mathbf{z})^2}{2(p-1)\|\mathbf{z}\|_p^2} \geq \frac{\delta^2}{2(p-1)R^2}. \quad \blacksquare$$

Now we are ready to prove our main result.

**Theorem 3** Suppose that for a sequence  $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{w}_T, y_T))$ , there exists a hyperplane  $(\mathbf{u}, b) \in \mathbb{R}^n \times \mathbb{R}$  such that  $y_t(\mathbf{u} \cdot \mathbf{x}_t + b) \geq 1$  for  $t = 1, \dots, T$  and the hyperplane  $(\mathbf{u}, b)$  has  $p$ -norm margin  $\gamma$  over  $S$ . Further, let  $R = \max_{t=1, \dots, T} \|\mathbf{x}_t\|_p$ . (i) Then the number of updates made by PUMMA $_p(\delta)$  is at most

$$O\left(\frac{(p-1)R^2\|\mathbf{u}\|_q^2}{\delta^2}\right).$$

(ii) PUMMA $_p(\delta)$  outputs a hypothesis with  $p$ -norm margin at least  $(1 - \delta)\gamma$  after at most the updates above.

**Proof:** As in Lemma 5, without loss of generality, we assume that PUMMA updates for  $t = 1, \dots, M$  ( $M \leq T$ ). By Lemma 5, we have  $\|\mathbf{w}_t\|_q \leq \|\mathbf{u}\|_q$  for  $t \geq 1$ . Further, by Lemma 6, it holds that after  $M$  updates

$$\|\mathbf{u}\|_q^2 \geq \|\mathbf{w}_T\|_q^2 \geq \frac{\delta^2 M}{2(p-1)R^2},$$

which implies  $M \leq \frac{2\|\mathbf{u}\|_q^2 R^2}{\delta^2}$ . Further, after at most  $\frac{2\|\mathbf{u}\|_q^2 R^2}{\delta^2}$  updates, we have  $y_t(\mathbf{w}_t + b_t) \geq 1 - \delta$  for  $t \geq T$ . Then the achieved margin is at least

$$\frac{1 - \delta}{\|\mathbf{w}\|_q} \geq \frac{1 - \delta}{\|\mathbf{u}\|_q} = (1 - \delta)\gamma. \quad \blacksquare$$

Since it holds that  $\|\mathbf{x}\|_p \leq \|\mathbf{x}\|_1$  for  $p \geq 1$  and  $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_p \leq n^{1/p} \|\mathbf{x}\|_\infty$ , we obtain the following corollary (A similar result was shown in [13]).

**Corollary 4** Assume that for a sequence  $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{w}_T, y_T))$ , there exists a hyperplane  $(\mathbf{u}, b) \in \mathbb{R}^n \times \mathbb{R}$  such that  $y_t(\mathbf{u} \cdot \mathbf{x}_t + b) \geq 1$  for  $t = 1, \dots, T$  and the hyperplane  $(\mathbf{u}, b)$  has  $\infty$ -norm margin  $\gamma$  over  $S$ . Further, let  $R = \max_{t=1, \dots, T} \|\mathbf{x}_t\|_\infty$ . Then, by setting  $p = c \ln n$  ( $c > 0$ ), (i) the number of updates made by PUMMA $_p(\delta)$  is at most

$$O\left(\frac{R^2\|\mathbf{u}\|_1^2 \ln n}{\delta^2}\right).$$

(ii) PUMMA $_p(\delta)$  outputs a hypothesis with  $\infty$ -norm margin at least  $\frac{1-\delta}{e^{1/c}}\gamma$  after at most the updates above.

## 4 Kernel and Soft Margin Extensions

### 4.1 Kernel Extension

As well as SVM, ROMMA and other Perceptron-like online algorithms, PUMMA can use kernel functions for  $p = 2$ . Note that, at trial  $t$ , the weight vector  $\mathbf{w}_t$  is written as

$$\mathbf{w}_t = \sum_{j=1}^{t-1} \left( \prod_{n=j+1}^{t-1} \alpha_n \right) \beta_j \mathbf{z}_j,$$

thus an inner product  $\mathbf{w}_t \cdot \mathbf{x}_t$  is given as a weighted sum of inner products  $\mathbf{x}_j \cdot \mathbf{x}_{j'}$  between instances since  $\mathbf{z}_j = \mathbf{x}_j^{pos} - \mathbf{x}_j^{neg}$ . Therefore, we can apply kernel methods by replacing each inner product  $\mathbf{x}_j \cdot \mathbf{x}_{j'}$  with  $K(\mathbf{x}_j, \mathbf{x}_{j'})$  for some kernel  $K$ . More practically, we can compute the inner products between  $\mathbf{w}_t$  and a mapped instance using the recurrence  $\mathbf{w}_t = \alpha_t(\mathbf{x}_t^{pos} - \mathbf{x}_t^{neg}) + \beta_t \mathbf{w}_{t-1}$ .

## 4.2 2-norm Soft Margin Extension

In order to apply PUMMA to linearly inseparable data, as in [21, 22], we employ the 2-norm soft margin minimization [9, 10], which is formulated as follows: Given a sequence  $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T))$  and letting  $\mathcal{S}$  be the set of examples in  $S$ ,

$$\begin{aligned} & \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^q + \frac{C}{2} \sum_{(\mathbf{x}, y) \in \mathcal{S}} \xi_{\mathbf{x}}^2, \quad (22) \\ & \text{subject to} \\ & y(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 - \xi_{\mathbf{x}} \quad ((\mathbf{x}, y) \in \mathcal{S}), \end{aligned}$$

where the constant  $C > 0$  is given as a parameter. Here, we implicitly assume that labels are consistent, i.e., if  $\mathbf{x}_t = \mathbf{x}_{t'}$  then  $y_t = y_{t'}$ . So we drop  $y$  from the subscript of  $\xi$ .

For  $p = 2$ , it is well known that this formulation is equivalent to the 2-norm minimization problem over linearly separable examples in an augmented space:

$$\begin{aligned} & \min_{\tilde{\mathbf{w}}, b, \xi} \frac{1}{2} \|\tilde{\mathbf{w}}\|^2, \\ & \text{subject to:} \\ & y(\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} + b) \geq 1 \quad (\mathbf{x} \in \mathcal{S}), \end{aligned}$$

where  $\tilde{\mathbf{w}} = (\mathbf{w}, \sqrt{C}\xi)$ ,  $\tilde{\mathbf{x}} = (\mathbf{x}, \frac{y}{\sqrt{C}}\mathbf{e}_x)$  for each  $(\mathbf{x}, y) \in \mathcal{S}$ , and each  $\mathbf{e}_x$  is a unit vector in  $\mathbb{R}^{|\mathcal{S}|}$  whose element corresponding to  $\mathbf{x}$  is 1 and other elements are set to 0. To use a kernel function  $K$  with this soft margin formulation, we just modify  $K$  as follows:

$$\tilde{K}(\mathbf{x}_j, \mathbf{x}_i) = K(\mathbf{x}_i, \mathbf{x}_j) + \frac{\Delta_{ij}}{C}, \quad (23)$$

where  $\Delta_{ij} = 1$  if  $i = j$ , otherwise  $\Delta_{ij} = 0$ .

For  $p > 2$ , we modify PUMMA so that, given  $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{t-1}, y_{t-1}))$  and an instance  $\mathbf{x}_t$ , it predicts  $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t + b_t)$ , where  $(\mathbf{w}_t, b_t, \xi_t)$  is specified as follows:

$$(\mathbf{w}_t, b_t, \xi_t) = \arg \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^q + \frac{C}{2} \sum_{(\mathbf{x}, y) \in \mathcal{M}_t} \xi_{\mathbf{x}}^2, \quad (24)$$

subject to:

$$\begin{aligned} & \mathbf{w} \cdot \mathbf{x}_t^{pos} + b \geq 1 - \xi_t^{pos}, \quad (25) \\ & \mathbf{w} \cdot \mathbf{x}_t^{neg} + b \leq -1 + \xi_t^{neg}, \\ & \mathbf{w} \cdot \mathbf{f}(\mathbf{w}_{t-1}) + C \sum_{(\mathbf{x}, y) \in \mathcal{M}_{t-1}} \xi_{\mathbf{x}} \xi_{t-1, \mathbf{x}} \geq \\ & \|\mathbf{w}_{t-1}\|_q^2 + C \sum_{(\mathbf{x}, y) \in \mathcal{M}_{t-1}} \xi_{t-1, \mathbf{x}}^2, \end{aligned}$$

where  $\mathcal{M}_t$  denotes the set of examples in  $S$  which have incurred updates of PUMMA in  $t-1$  trials,  $\xi_t^{pos} = \xi_{\mathbf{x}_t^{pos}}$  and  $\xi_t^{neg} = \xi_{\mathbf{x}_t^{neg}}$ . Then the modified PUMMA update  $\mathbf{x}_{t+1}^{pos}$  or  $\mathbf{x}_t^{neg}$  if  $y_{t+1}(\mathbf{w}_{t+1} \cdot \mathbf{x}_{t+1} + b_{t+1}) < 1 - \delta - \xi_{\mathbf{x}_{t+1}}$ , where  $\xi_{\mathbf{x}_{t+1}} = \xi_{\mathbf{x}_t}$  if  $\mathbf{x}_{t+1} = \mathbf{x}_t$  such that  $(\mathbf{x}_t, y_t) \in \mathcal{M}_t$ . Otherwise,  $\xi_{\mathbf{x}_{t+1}} = 0$ .

**Solution** The Lagrangian function is given as

$$\begin{aligned} & L(\mathbf{w}, b, \xi, \alpha, \beta) \\ & = \frac{1}{2} \|\mathbf{w}\|_q^2 + \frac{C}{2} \sum_{(\mathbf{x}, y) \in \mathcal{M}_t} \xi_{\mathbf{x}}^2 \\ & + \sum_{\ell \in \{\text{pos}, \text{neg}\}} \alpha^\ell (1 - \xi_t^\ell - y_t^\ell \mathbf{w} \cdot \mathbf{x}_t^\ell) \\ & + \beta \left( \|\mathbf{w}_{t-1}\|_q^2 - \mathbf{w} \cdot \mathbf{f}(\mathbf{w}_{t-1}) \right. \\ & \left. + C \sum_{\mathbf{x} \in \mathcal{M}_{t-1}} \xi_{t-1, \mathbf{x}}^2 - C \sum_{\mathbf{x} \in \mathcal{M}_{t-1}} \xi_{\mathbf{x}} \xi_{t-1, \mathbf{x}} \right). \end{aligned}$$

To simplify descriptions, without loss of generality, we assume that  $\mathbf{x}_t$  is a positive instance. Note that every solution is as same as when  $\mathbf{x}_t$  is a negative one. As done in the separable case, by using KKT conditions, we consider the following two cases:

(i) Suppose that  $\beta = 0$ ,  $\alpha > 0$ . Then the optimal solution  $(\mathbf{w}^*, b^*, \xi^*)$  is given as

$$\begin{aligned} & \mathbf{w}^* = \alpha \mathbf{f}^{-1}(\mathbf{z}), \\ & \alpha = \frac{2}{\frac{2}{C} + \|\mathbf{z}\|_p^2}, \\ & \xi_t^{pos*} = \xi_t^{neg*} = \frac{\alpha}{C}, \\ & \xi_{\mathbf{x}}^* = 0 \quad (\mathbf{x} \in \mathcal{M}_t \setminus \{\mathbf{x}_t^{pos}, \mathbf{x}_t^{neg}\}), \end{aligned}$$

where  $\mathbf{z} = \mathbf{x}_t^{pos} - \mathbf{x}_t^{neg}$ . (ii) Otherwise,  $\beta \neq 0$ ,  $\alpha > 0$ . Let  $\boldsymbol{\theta} = \mathbf{f}(\mathbf{w}_{t-1})$ . Then we have

$$\begin{aligned} & \mathbf{w}^* = \mathbf{f}^{-1}(\alpha \mathbf{z} + \beta \boldsymbol{\theta}), \\ & \xi_t^{pos*} = \begin{cases} \frac{\alpha}{C}, & \text{if } (\mathbf{x}_t, y_t) \notin \mathcal{M}_t, \\ \frac{\alpha}{C} + \beta \xi_{t-1}^{neg}, & \text{if } (\mathbf{x}_t, y_t) \in \mathcal{M}_t, \end{cases} \\ & \xi_t^{neg*} = \frac{\alpha}{C} + \beta \xi_{t-1}^{neg}, \\ & \xi_{\mathbf{x}} = \beta \xi_{t-1, \mathbf{x}} \quad (\mathbf{x} \in \mathcal{M}_t \setminus \{\mathbf{x}_t^{pos}, \mathbf{x}_t^{neg}\}), \end{aligned}$$

where  $\alpha$  and  $\beta$  are the maximizers of the Lagrange dual

$$\begin{aligned} & L^*(\alpha, \beta) = \min_{\mathbf{w}, b, \xi} L(\mathbf{w}, b, \xi, \alpha, \beta) \\ & = \frac{1}{2} \|\alpha \mathbf{z} + \beta \boldsymbol{\theta}\|_p^2 \\ & - 2\alpha + \frac{\alpha^2}{C} + \alpha \beta \xi_{t-1}^{neg} \\ & - \beta \|\boldsymbol{\theta}\|_p^2 - C(\beta - \frac{\beta^2}{2}) \sum_{\mathbf{x} \in \mathcal{M}_{t-1}} \xi_{\mathbf{x}, j}^2. \end{aligned}$$

Again, we can approximate  $(\alpha, \beta)$  by repeating the Newton update

$$\begin{pmatrix} \alpha_{k+1} \\ \beta_{k+1} \end{pmatrix} = \begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} - \nabla^2 L^*(\alpha, \beta)^{-1} \nabla L^*(\alpha_k, \beta_k)$$

for sufficiently many steps, where

$$\begin{aligned}\frac{\partial^2 L^*}{\partial^2 \alpha} &= \sum_i \mathbf{f}^{-1'}(\alpha \mathbf{z} + \beta \boldsymbol{\theta})_i z_i^2 + \frac{2}{C} \\ \frac{\partial^2 L^*}{\partial \beta \partial \alpha} &= \frac{\partial^2 L^*}{\partial \alpha \partial \beta} \\ &= \sum_i \mathbf{f}^{-1'}(\alpha \mathbf{z} + \beta \boldsymbol{\theta})_i z_i \theta_i + \xi_{t-1}^{neg} \\ \frac{\partial^2 L^*}{\partial^2 \beta} &= \sum_i \mathbf{f}^{-1'}(\alpha \mathbf{z} + \beta \boldsymbol{\theta})_i \theta_i^2 + C \sum_{\mathbf{x} \in M_{t-1}} \xi_{\mathbf{x},j}^2.\end{aligned}$$

As in the case without soft margin, in order to acquire the solution  $\mathbf{w}^*$  and  $\boldsymbol{\xi}^*$ , we first assume the case (i) and check whether the third constraint of the problem (24) holds with strict inequality or not. If it does, then the case (i) is true. Otherwise, the case (ii) holds. Finally, the bias  $b^*$  is given as

$$b^* = -\frac{\mathbf{w}^* \cdot \mathbf{x}^{pos} + \mathbf{w}^* \cdot \mathbf{x}^{neg} + (\xi_t^{pos} - \xi_t^{neg})}{2}.$$

By the same argument as Section 3, we obtain the following:

**Theorem 5** For a sequence  $S = ((x_1, y_1), \dots, (w_T, y_T))$ , let  $(\mathbf{u}, b, \boldsymbol{\xi}) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^{|S|}$  be the optimal solution of the problem (22). Further, let  $R = \max_{t=1, \dots, T} \|\mathbf{x}_t\|_p$ . (i) Then the number of updates made by  $\text{PUMMA}_p(\delta)$  is at most

$$O\left(\frac{\{(p-1)R^2 + \frac{2}{C}\} \left(\|\mathbf{u}\|_q^2 + \sum_{(x,y) \in S} \xi_{\mathbf{x}}^2\right)}{\delta^2}\right).$$

(ii)  $\text{PUMMA}_p(\delta)$  outputs a hypothesis with  $p$ -norm margin whose objective value for the problem (22) is at most  $\frac{1}{(1-\delta)^2}$  times the optimum after at most the updates above.

## 5 Experiments

### 5.1 Experiments over artificial datasets

We examine PUMMA, ALMA and ROMMA over artificial datasets generated by sparse linear classifiers. Each artificial dataset consists of  $n$ -dimensional  $\{-1, +1\}$ -valued vectors with  $n = 100$ . Each vector is labeled with a  $r$ -of- $k$  threshold function  $f$ , which is represented as  $f(\mathbf{x}) = \text{sign}(x_{i_1} + \dots + x_{i_k} + k - 2r + 1)$  for some  $i_1, \dots, i_k$  s.t.  $1 \leq i_1 \leq i_2 \leq \dots \leq i_k \leq n$ , and it outputs  $+1$  if at least  $r$  of  $k$  relevant features have value  $+1$ , and outputs  $-1$ , otherwise.

For  $k = 16$  and  $r \in \{1, 4, 8\}$  (equivalently, the bias  $b \in \{15, 9, 1\}$ , respectively), we generate random 1000 examples labeled by the  $r$ -of- $k$  threshold function, so that positive and negative examples are equally likely. For ALMA and ROMMA, we add an extra dimension with value  $-R$  to each vector to learn linear classifiers with bias, where  $R = \max \|\mathbf{x}\|_p$ . Note that one can choose different values other than  $-R$ , say, 1. However, as remarked in [10], such a choice for the value in the extra dimension increases the number of iterations by  $O(R^2)$  times when the underlying hyperplane has large bias. So our choice seems to be fair.

We set parameters so that each algorithm is guaranteed to achieve at least 0.9 times the maximum  $p$ -norm margin.

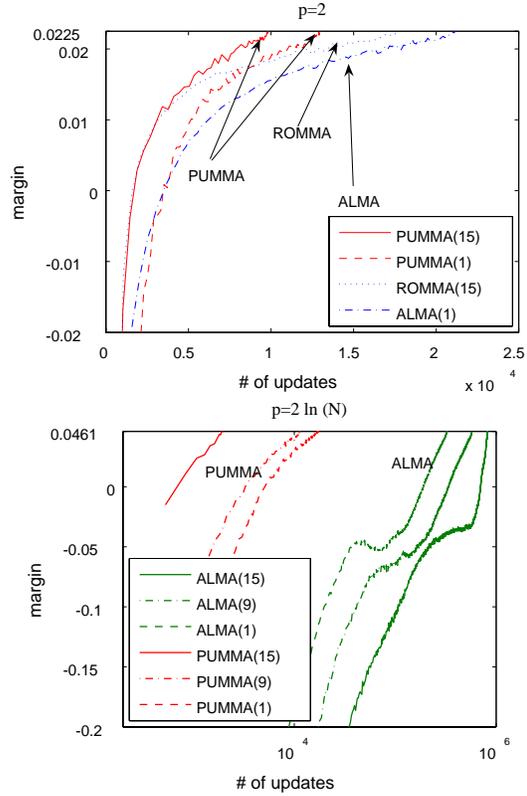


Figure 3: Number of updates and margin over artificial data set in the case  $p = 2$  (upper) and  $p = 2 \ln(n)$  (lower). We set x-axes log scale since the numbers of updates of ALMA are quite larger than PUMMA's. And we hide the result of the case  $p = 2$  and  $b = 9$  since we make the figure easy to view. The parenthetical digits denote the value of bias.

That is, we set  $\alpha = 0.1$  (note the parameter  $\alpha$  is defined differently in [13]) for ALMA and  $\delta = 0.1$  for ROMMA and PUMMA. We examine  $p \in \{2, 2 \ln n\}$ .

We train each algorithm until its hypothesis converges by running it in epochs, where, in one epoch, we make each algorithm go through the whole training data once. At end of each epoch, for each algorithm, we record number of updates, margin incurred during the training and real computation time. Note that we measure the margin of each hypothesis over the original space. We execute these operations 10 times, changing the randomly generated data, and we average the results over 10 executions. The experiments are conducted on a 3.8 GHz Intel Xeon processor with 8 GB RAM running Linux. We use MATLAB for the experiments.

The results are represented in Figure 3 and 4. We observe that PUMMA converges faster. PUMMA's computation time is quite shorter than that of ALMA, although it uses Newton method in each update. Note that we omit the result of ALMA in the case  $p = 2$  since the result is worse than the others. For  $p = 2$ , we don't use Newton method in the execution of PUMMA because we have the analytical solution of the optimal value of  $\alpha$  and  $\beta$  by solving the optimization problem directly.

Table 1: Computation time (sec.) and obtained margin (denoted as  $\gamma'$ ) on some UCI datasets.

dataset	SVM <sup>light</sup>		PUMMA		ROMMA		MICRA	
	sec.	$10^2\gamma'$	sec.	$10^2\gamma'$	sec.	$10^2\gamma'$	sec.	$10^2\gamma'$
ionosphere	0.06	10.55	0.54	10.49	3.12	10.50	0.48	10.04
house-votes	0.03	17.42	0.26	17.31	0.62	17.36	0.09	16.51
adult-1k	0.47	4.95	5.40	4.50	15.83	4.91	2.34	4.03
adult-2k	2.13	3.40	25.38	3.37	82.70	3.38	5.61	2.81
adult-4k	9.33	2.40	159.54	2.38	496.52	2.38	55.91	2.00
adult-8k	232.42	1.69	807.46	1.67	2167.40	1.67	189.13	1.46
adult-16k	1271.06	1.20	3365.47	1.18	12503.62	1.18	2050.84	1.13
adult-full	5893.20	0.83	44480.59	0.82	71296.34	0.82	12394.86	0.79

## 5.2 Experiments over some UCI datasets

We compare PUMMA with some other learning algorithms over the real datasets. The algorithms are SVM<sup>light</sup> [16], MICRA [35], and ROMMA [22]. We used the following datasets of UCI Machine Learning Repository [2]. (i) The ionosphere dataset consists of 351 instances which have 34 continuous attributes. (ii) The house-vote dataset consists of 435 instances which have 16 discrete attributes  $\{y, n, ?\}$ . We change these attributes to  $\{1, -1, 0\}$ . (iii) The adult dataset consists of 32561 instances which have 14 attributes. Among the attributes, 6 of them are discrete and the others are continuous. We change this 14 attributes to 123 binary attributes as Platt did in [29]. The name of dataset 'adult- $m$ k' in Table 1 denotes a subset of the adult dataset which contains  $1000 \times m$  instances. Note that all the datasets have binary class and we change the range of labels with  $\{1, -1\}$ .

To optimize the 2-norm soft margin for this linearly inseparable dataset, we use the following modified inner product

$$IP(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j + \frac{\Delta_{ij}}{C}.$$

We added a dimension which denotes the bias as in Section 1 when we run MICRA and ROMMA which can't deal with bias directly.

We modify SVM<sup>light</sup> so as not to optimize 1-norm soft margin, and we change the inner product so that it optimizes 2-norm soft margin. We set  $\delta = 0.01$  for PUMMA and ROMMA to achieve 99% of the maximum margin. The parameters of MICRA are changed for each dataset as in [35]. But, parameters might not be completely the same as them because some datasets are different from those they used. Finally we set 2-norm soft margin parameter  $C = 1$  for all algorithms. In order to converge faster, we use the following heuristics for each online algorithm.

**Active Set** We try to improve the order of given examples to feed for each online algorithm. First, we give all the examples to each online learning algorithm once. Then, we make a new dataset called "active set", containing the examples which causes updates. After that, we give each example in the active set to the algorithm. If the example doesn't cause any updates, we remove the example from the active set, and we repeat this procedure until the active set becomes empty. Finally, we give all the examples again and check if the al-

gorithm makes any updates. If some updates occur, we construct an active set again and repeat the whole procedure.

We run each algorithm and we measure its real computation time as well as its obtained margin. The experiments on real datasets are conducted on a 3.0 GHz Intel Xeon processor with 16 GB RAM running Linux. We implemented each algorithm in C.

Table 1 shows the real computation time and obtained margin. As can be seen, PUMMA converges quite faster than ROMMA. On the other hand, PUMMA converges slower than MICRA. However, the parameters of MICRA are quite sensitive to datasets and nontrivial to tune appropriately. The results on all the real data set show that SVM<sup>light</sup> is the fastest, whereas MICRA is reported to be faster than SVM<sup>light</sup> over some datasets and with tuned parameters [35]. Note that this might be due to our selection of active sets which is different from theirs.

## 5.3 Experiments over MNIST dataset

Next, we compare these algorithms over MNIST dataset. Since the dataset is not linearly separable, we use polynomial kernel and 2-norm soft margin as follows.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{s}\right)^d + \frac{\Delta_{i,j}}{C}.$$

Since computing kernels is time-consuming, we use some extra heuristics in addition to our active set selection.

**Kernel Cache** Since we have to compute kernel values of the same examples repeatedly, we memorize them in a cache matrix. In the cache matrix, each row memorizes the kernel values of a support vector and all the examples, where a support vector is an instance which causes an update. The length of each row equals to the number of training instances. The number of rows depends on the memory size. When the new kernel value of a support vector and an instance is required, we search the cached value in the cache matrix. If we fails, we calculate the value and store it in the cache matrix. To do this, we search the row of the corresponding support vector. We store the value if we succeed, or we make the new row otherwise. If the matrix is full, we replace the least referenced row by the new row.

**Inner Product Cache** In our experiments, we keep giving examples to each online learning algorithms until they

Table 2: Computation time (sec.) and obtained margin (denoted as  $\gamma'$ ) on MNIST datasets.

class	SVM <sup>light</sup>		SVM <sup>light</sup> w/o bias		PUMMA		ROMMA	
	sec.	$\gamma'$	sec.	$\gamma'$	sec.	$\gamma'$	sec.	$\gamma'$
0	256.51	1.339	164.83	1.155	373.48	1.330	218.97	1.150
1	152.10	0.712	119.62	0.712	291.54	0.706	231.82	0.708
2	413.43	0.810	309.77	0.765	1674.08	0.804	870.58	0.761
3	566.84	0.763	384.17	0.722	2654.19	0.757	2296.34	0.719
4	333.04	0.650	267.65	0.629	905.16	0.645	505.11	0.626
5	428.36	0.672	301.99	0.664	1480.44	0.667	1007.91	0.661
6	246.47	0.941	184.39	0.880	534.80	0.934	308.18	0.876
7	322.90	0.621	304.54	0.611	860.89	0.616	584.36	0.608
8	694.17	0.810	437.48	0.727	5648.12	0.804	5074.51	0.723
9	599.78	0.558	399.08	0.541	5290.33	0.554	6057.92	0.538
avg.	401.36	0.788	287.35	0.741	1971.30	0.782	1715.57	0.737

make no update on all the examples. Assume that at trial  $t = t_1, t_2 (t_1 < t_2)$ , the weight vector  $\mathbf{w}_t$  is updated by the same example  $x_{t_1}$ . The weight vector  $\mathbf{w}_{t_2}$  is written as

$$\begin{aligned} \mathbf{w}_{t_2} &= \sum_{j=1}^{t_2-1} \left( \prod_{k=j+1}^{t_2-1} \alpha_k \right) \beta_j \mathbf{z}_j \\ &= \left( \prod_{k=t_1}^{t_2-1} \alpha_k \right) \mathbf{w}_{t_1} + \sum_{j=t_1}^{t_2-1} \left( \prod_{n=j+1}^{t_2-1} \alpha_n \right) \beta_j \mathbf{z}_j. \end{aligned}$$

So, if we memorize the inner product  $\mathbf{w}_{t_1} \cdot \mathbf{x}_{t_1}$ , we can calculate  $\mathbf{w}_{t_2} \cdot \mathbf{x}_{t_1}$  easier. This technique is efficient when we use kernel.

**Halving  $\delta$**  It is reported that by decreasing  $\delta$  in a dynamical way, ROMMA converges faster [22]. Similar to their approach, we shrink the parameter  $\delta$  by halving repeatedly. More precisely, we set  $\delta = 1$  at first, and halve  $\delta$  when the algorithm makes no update for all the examples. We repeat this procedure until  $\delta$  is as small as we require. Note that if  $\delta$  is smaller than the required value  $\delta_{\text{target}}$ , we set  $\delta = \delta_{\text{target}}$ . When we use kernels, this halving heuristics can reduce support vectors in the early stage of learning, which contributes faster convergence.

MNIST dataset contains 60,000 matrix and labels. Each  $(28 \times 28)$  matrix represents the image of the hand written digit. The value of each element is in  $\{0, \dots, 255\}$ , which denotes the density. Each label takes the value  $\{0, \dots, 9\}$ . MNIST dataset has 10 classes. Since each algorithm can deal with only binary class, we change each label so that one class is positive and the others are negative. Then we get 10 binary labeled datasets.

We run three learning algorithms, SVM<sup>light</sup>, ROMMA and PUMMA on these datasets until they converge. We omit the evaluation of MICRA since it needs careful tuning of parameters to converge fast. We record the real computation time and margin. Note that we use our heuristics for ROMMA and PUMMA. And we set some kernel parameters,  $s = 1100^2$ ,  $d = 5$  and  $C = 1/30$  as in [22]. We set  $\delta_{\text{target}} = 0.01$  and use 1 GB kernel cache. We also run SVM<sup>light</sup> with the same size of cache memory, but its caching

strategy is different from ours. The experiments on MNIST dataset are conducted on the same machine as the experiments on UCI dataset.

The results are shown in Table 2. PUMMA gains higher margin than ROMMA over almost all of the datasets. On the other hand, PUMMA requires more computation time. This seems to be due to the fact that ROMMA solves the different optimization problem, i.e., maximization of margin without bias. We observe the same tendency between SVM<sup>light</sup> with and without bias. Further, computation times of PUMMA are worse than SVM<sup>light</sup>. But, PUMMA and ROMMA might be improved if we employ a different strategy for active set selection.

## 6 Conclusion and Future work

In this paper, we propose PUMMA which obtains the maximum  $p$ -norm margin classifier with bias approximately. Our algorithm often runs faster than previous online learning algorithms when the underlying linear classifier has large bias, by taking advantage of finding bias directly.

Although the worst case upperbound on iterations of our algorithm is the same as those of previous algorithms, our experiments over artificial datasets suggest that our iteration bound might be better. For example, when the target function is a  $r$ -of- $k$  threshold function, iteration bound of PUMMA is  $O(k^2 \ln n)$  with  $p = O(\ln n)$ . However, in our experiments, PUMMA seems to converge in  $O(rk \ln n)$  iterations, which is the best upperbound obtained by Winnow when  $k$  and  $r$  are known a priori. Unfortunately, we have not yet succeeded in proving better iteration bounds. It is still open if there exists an online learning algorithms that learns  $r$ -of- $k$  threshold functions in  $O(rk \ln n)$  updates without knowing  $k$  and  $r$  [23].

So far PUMMA or ALMA approximates  $\infty$ -norm margin indirectly by setting  $p = O(\ln n)$ . Developing an adaptive online algorithm that directly maximizes  $\infty$ -norm margin is also an open problem. One of the future work is to extend our algorithm to handle 1-norm soft margin which is commonly used in SVM. Further, we would like to apply PUMMA to learning sparse classifiers in practical applica-

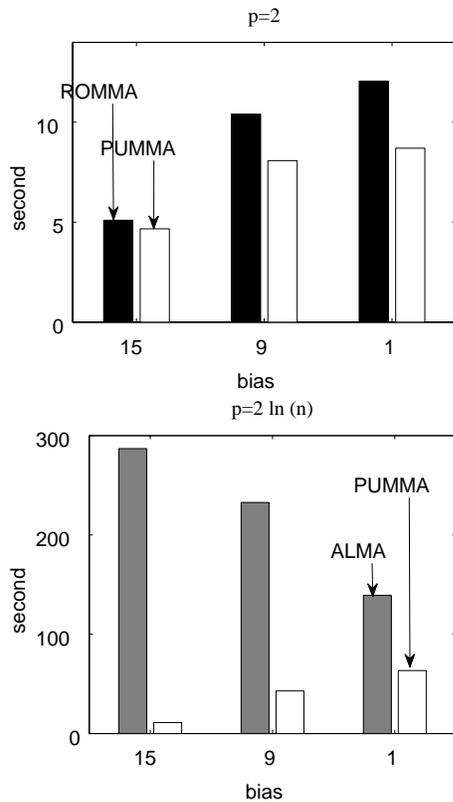


Figure 4: Computation time over artificial data set in the case  $p = 2$  (upper) and  $p = 2 \ln(n)$  (lower).

tions.

## Acknowledgments

We thank anonymous referees for helpful comments.

## References

- [1] J. K. Anlauf and M. Biehl. The adatron; an adaptive perceptron algorithm. *Europhysics Letters*, 10:687–692, 1989.
- [2] A. Asuncion and D. J. Newman. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, <http://mllearn.ics.uci.edu/MLRepository.html>, 2007.
- [3] H. H. Bauschke and P. L. Combettes. A weak-to-strong convergence principle for Fejér-monotone methods in hilbert spaces. *Mathematics of Operations Research*, 26(2):248–264, 2001.
- [4] A. Bordes, S. Ertekin, J. Weston, and Léon Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6:1579–1619, 2005.
- [5] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.
- [6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [7] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [8] C. C. Chang and C. J. Lin. Libsvm: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [9] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.
- [10] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machine*. Cambridge University Press, 2000.
- [11] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–299, 1999.
- [12] T. Friess, N. Cristianini, and C. Campbell. The kernel adatron algorithm: a fast and simple learning procedure for support vector machine. In *Proceedings of the 15th International Conference on Machine Learning*, 1998.
- [13] C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2001.
- [14] C. Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3):265–299, 2003.
- [15] A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. In *Proceedings of the tenth annual conference of Computational learning theory*, pages 171–183, 1997.
- [16] T. Joachims. Making large-scale support vector machine learning practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in kernel methods - Support vector learning*, pages 169–184. MIT Press, 1999.
- [17] T. Joachims. Training linear svms in linear time. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.
- [18] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE Transactions on Neural Networks*, 11(1):124–136, 2000.
- [19] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2004.
- [20] J. Kivinen, M. K. Warmuth, and P. Auer. The perceptron algorithm versus winnow: linear versus logarithmic mistake bounds when few input variables are relevant. *Artificial Intelligence*, 97(1-2):325–343, 1997.
- [21] A. Kowalczyk. Maximum margin perceptron. In B. Schölkopf A. Smola, P. Bartlett and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 75–114. MIT Press, 2000.
- [22] Y. Li and P. M. Long. The relaxed online maximum margin algorithm. *Machine Learning*, 46(1-3):361–387, 2002.
- [23] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.
- [24] P. M. Long and X. Wu. Mistake bounds for maximum entropy discrimination. In *Advances in Neural Inform-*

- mation Processing Systems 17*, pages 833–840, 2004.
- [25] O. L. Mangasarian. Arbitrary-norm separating plane. *Operations Research Letters*, 24:15–23, 1999.
  - [26] M. L. Minsky and S. A. Papert. *Perceptrons*. MIT Press, 1969.
  - [27] A. B. Novikoff. On convergence proofs on perceptrons. In *Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622. Polytechnic Institute of Brooklyn, 1962.
  - [28] E. Osuna, R. Freund, and F. Girosi. Improved training algorithm for support vector machines. In *Proceedings of IEEE NNSP'97*, 1997.
  - [29] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Scholköpf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 185–208. MIT Press, 1999.
  - [30] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.
  - [31] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1959.
  - [32] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.
  - [33] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.
  - [34] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.
  - [35] P. Tsampouka and J. Shawe-Taylor. Approximate maximum margin algorithms with rules controlled by the number of mistakes. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

---

# Minimizing Wide Range Regret with Time Selection Functions

---

Subhash Khot and Ashok Kumar Ponnuswami

New York University and Georgia Tech

khot@cs.nyu.edu and pashok@cc.gatech.edu

## Abstract

We consider the problem of minimizing regret with respect to a given set  $\mathcal{S}$  of pairs of time selection functions and modifications rules. We give an online algorithm that has  $O(\sqrt{T \log |\mathcal{S}|})$  regret with respect to  $\mathcal{S}$  when the algorithm is run for  $T$  time steps and there are  $N$  actions allowed. This improves the upper bound of  $O(\sqrt{TN \log(|\mathcal{I}||\mathcal{F}|)})$  given by Blum and Mansour [BM07a] for the case when  $\mathcal{S} = \mathcal{I} \times \mathcal{F}$  for a set  $\mathcal{I}$  of time selection functions and a set  $\mathcal{F}$  of modification rules. We do so by giving a simple reduction that uses an online algorithm for external regret as a black box.

## 1 Introduction

We consider the following online optimization problem. At the beginning of each day (a time step), we have to choose one of the  $N$  allowed actions. Instead of picking one action deterministically, we may come up with a distribution over the actions. At the end of the day, an adversary, with the knowledge of the distribution we picked, fixes a loss for each action. We give a concrete example from Cesa-Bianchi et al. [CBFH<sup>+</sup>97]. Suppose we want to predict the probability that it rains on a day based on the predictions of  $N$  weather forecasting websites. But we don't know which of these "experts" give good forecasts. We come up with some weights on the websites using an online algorithm and use the weighted prediction as our guess for the probability of raining. At the end of the day, based on whether or not it rained, everyone incurs a loss depending on how inaccurate their prediction was. Usually it is assumed that the loss for each action is picked from a fixed interval, like  $[0, 1]$ . For example, we could charge a person who predicts  $p$  as the probability of rain  $1 - p$  if it rains and  $p$  if it does not. After  $T$  days, we compare the loss incurred by the online algorithm we used to the loss incurred if we had followed a simple strategy (like just picking the same action each day). Our goal is to minimize our *regret* for not following one of the simple strategies. One may also compare the algorithm's performance to the performance if the distribution over actions at each time step were modified using a certain set of rules. We consider the problem of designing algorithms with low regret with respect to a given set of strategies or modification rules.

The most basic regret studied is *external regret*, which is the difference between the loss incurred by the algorithm and the loss incurred by the best action in hindsight. Another kind of regret commonly studied is called *internal regret*. This was introduced by Foster and Vohra [FV98]. Here, we consider the set of modification rules where for each pair  $(a, b)$  of actions we have a rule of the kind: Every time the algorithm suggests picking  $a$ , pick  $b$  instead. The internal regret of the algorithm is the regret of not having applied one of these modification rules. Each rule here can be considered as a function  $f_{a,b}$  that maps every action to itself, except action  $a$  which gets mapped to  $b$ . If we consider the set of modification rules corresponding to all functions mapping the set of actions into itself, we get the notion of *swap regret*. Finally, we can allow any subset of these mappings as the set of allowed modification rules which gives the notion of *wide range regret*. This was defined by Lehrer [Leh03]. Lehrer also associates *time selection* function with each rule that indicates whether a rule is "active" at a given time or not. A related model is that of "sleeping experts" or "specialists" defined in Freund et al. [FSSW97]. Here, at the beginning of time  $t$ , each specialist can decide whether or not the current situation is her area of speciality and make a prediction only if it does. In addition, Blum and Mansour [BM07a] consider the case where the experts can be "partially awake". One way to interpret the activeness function is that it measures degree of confidence that the corresponding rule will perform well at a given time. In this case, we weigh the loss incurred by the algorithm and the modified action with the time selection function to calculate the regret.

The first algorithm with external regret sublinear in  $T$  was developed by Hannan [Han57]. An algorithm whose external regret has only logarithmic dependence on  $N$  was given by Littlestone and Warmuth [LW94] and Cesa-Bianchi et al. [CBFH<sup>+</sup>97].

**Lemma 1 ([CBFH<sup>+</sup>97])** *There exists an online algorithm with external regret at most  $O(\sqrt{T \log N})$  when the losses are picked from  $[-1, +1]$ . The running time is polynomial in  $T$  and  $N$ .*

The number of time steps  $T$  for which it will be run need not be provided as an input to the above algorithm. Stoltz and Lugosi [SL05] give a general method to convert any "weighted average predictor" algorithm for external regret to a low internal or swap regret algorithm. At a high level, they pretend there is an expert for each modification rule

who always suggests using that rule. At each time step, the expert is charged the loss that would be incurred if his modification rule were actually used. The weighted average predictor would give a distribution over the experts. The distribution over the actual actions is found by computing the fixed point of the expected modification rule picked from the distribution over the experts. This gives algorithms with  $O(\sqrt{T \log N})$  internal regret and  $O(\sqrt{TN \log N})$  swap regret. Our approach for wide range regret with time selection functions is based on the same idea. A drawback of a swap regret algorithm constructed this way is that it needs to maintain  $N^N$  weights. Blum and Mansour [BM07a] give an algorithm that has  $O(\sqrt{TN \log N})$  swap regret and runs in time polynomial in  $N$  too. They also give an algorithm that has  $O(\sqrt{TN \log(KM)})$  regret with respect to  $K$  modification rules and  $M$  time selection functions. Here, for each modification rule and time selection function, the regret of not having modified the algorithm's action by the rule with the losses weighed by the time selection function is considered. In this case, we can think of there being  $M$  people who are interested in following an algorithm's predictions. They have varying degrees of importance associated with each day (given by their corresponding time selection function) and want to minimize regret with respect to all the modification rules. The algorithm's goal is to minimize the maximum regret of a person. This is a bit different from the model considered in Lehrer [Leh03]. But with some effort, one can check that the result of Blum and Mansour [BM07a] can be generalized to the model of Lehrer [Leh03]. We refer the reader to [BM07b] for other bounds on the regret minimization and the relation of various kinds of regret to equilibriums in games.

The paper is organized as follows. In the next section, we define the model we work with formally and state our main result. We state the ideas we use from related results in Section 3. We prove our main result of an improved upper bound for wide range regret in Section 4. We conclude with a "first-order" upper bound in Section 5.

## 2 Our Model and Result

Let the set of actions be  $[N] = \{1, 2, \dots, N\}$ . Consider the following  $T$  round game between an online algorithm  $H$  and an adversary. At the beginning of time  $t = 1, 2, \dots, T$ , the algorithm picks a probability vector<sup>12</sup>  $\mathbf{p}^t = (p_1^t, p_2^t, \dots, p_N^t)$ . The adversary then picks the loss vector  $\mathbf{l}^t = (l_1^t, l_2^t, \dots, l_N^t)$  for time  $t$ . The entries of  $\mathbf{l}^t$  are picked from a fixed interval. In this paper, we assume the losses are either picked from  $[0, 1]$  or from  $[-1, +1]$ .

Define the regret of  $H$  with respect to action  $a \in [N]$  to be

$$R_{H,a} = \sum_{t=1}^T \left( \sum_{b \in [N]} p_b^t l_b^t - l_a^t \right) = \sum_{t=1}^T \sum_{b \in [N]} p_b^t (l_b^t - l_a^t).$$

This can be interpreted as the difference between the expected loss of  $H$  and the loss of action  $a$ . Define the *external*

<sup>1</sup>A probability vector is a vector in which the entries are non-negative and sum to 1.

<sup>2</sup>All vectors we consider are column vectors. We will use  $^\top$  to denote the transpose.

regret of  $H$  to be

$$R_{H,ext} = \max_{a \in [N]} R_{H,a}.$$

We now define the model with time selection functions from Blum and Mansour [BM07a]. A time selection function is a function  $I : \mathbb{N} \rightarrow [0, 1]$ . Let  $\mathcal{I}$  be the set of time selection functions. At the beginning of time  $t$ , the adversary sets the values of  $I(t)$  for each  $I \in \mathcal{I}$ . The algorithm then picks  $\mathbf{p}^t$  after which the adversary now picks  $\mathbf{l}^t$  as before. Given a modification rule  $f : [N] \rightarrow [N]$ , define  $\mathbf{M}_f$  to be the matrix with a 1 in column  $f(i)$  of row  $i$  for all  $i$  and zeros everywhere else. Define the regret of  $H$  with respect to time selection function  $I$  and a modification rule  $f$  to be

$$\begin{aligned} R_{H,I,f} &= \sum_t I(t) \sum_{a \in [N]} p_a^t (l_a^t - l_{f(a)}^t) \\ &= \sum_t I(t) (\mathbf{p}^t \cdot \mathbf{l}^t - \mathbf{p}^{\top} \mathbf{M}_f \mathbf{l}^t). \end{aligned}$$

Informally, we first weigh all the losses at time  $t$  by  $I(t)$ , the significance attached to time  $t$ . Then we look at the difference between the expected loss of  $H$  and the expected loss if the output of  $H$  were modified every time by applying  $f$ . That is, we measure the regret of not having played action  $f(a)$  every time we played  $a$ . Given a set  $\mathcal{S}$  of pairs  $(I, f)$ , where  $I$  is a time selection function and  $f$  is a modification rule, the *wide range regret* of  $H$  with respect to  $\mathcal{S}$  is defined as

$$R_{H,\mathcal{S}} = \max_{(I,f) \in \mathcal{S}} R_{H,I,f}.$$

Let  $\mathbb{1} : \mathbb{N} \rightarrow [0, 1]$  be the function that always outputs 1, i.e.,  $\mathbb{1}(t) = 1$ . For simplicity of notation, we will use  $f$  to also denote the pair  $(\mathbb{1}, f)$  when we are not concerned with time selection functions, in which case we assume that the adversary always sets  $\mathbb{1}(t)$  to 1. It is easy to check that external regret is the same  $R_{H,\mathcal{F}_{ext}}$  where  $\mathcal{F}_{ext} = \{f_a\}_{a \in [N]}$  and  $\forall b \in [N] : f_a(b) = a$ . The *internal regret* of  $H$  is defined to be  $R_{H,\mathcal{F}_{int}}$ , where  $\mathcal{F}_{int} = \{f_{a,b}\}_{a,b \in [N]}$  and  $f_{a,b}(a) = b$  while  $f_{a,b}(c) = c$  for  $c \neq a$ . The *swap regret* of  $H$  is defined to be  $R_{H,\mathcal{F}_{swap}}$ , where  $\mathcal{F}_{swap}$  is the set of all functions  $f : [N] \rightarrow [N]$ .

We prove the following theorem for minimizing wide range regret.

**Theorem 2** *There exists an online algorithm  $H$  that for any given set  $\mathcal{S}$  satisfies*

- $R_{H,\mathcal{S}} = O(\sqrt{T \log |\mathcal{S}|})$  when the losses are picked from the  $[0, 1]$ .
- The running time of  $H$  is polynomial in  $T$ ,  $N$  and  $|\mathcal{S}|$ .

Note that this matches (upto a constant) the results for external, internal and swap regret if we are not concerned with time selection functions. A drawback of our approach is that if the size of the set  $\mathcal{S}$  is large, the running time is high. For example, for swap regret with time selection functions, we may need time polynomial in  $T$  and  $N^N$ . But for this case, the result of Blum and Mansour already gives a more efficient algorithm with the same regret (upto a constant).

### 3 Previous Results

We use ideas from Stoltz and Lugosi [SL05] and Blum and Mansour [BM07a].

We first describe the approach of Stoltz and Lugosi [SL05] for internal regret. The idea is to simulate a low external regret algorithm for  $N(N-1)$  imaginary experts. Start with any “weighted average predictor”  $H_{ext}$  with low external regret. There are  $N(N-1)$  imaginary experts, one for each modification rule  $f_{a,b}$ . The expert corresponding to  $f_{a,b}$  always suggests playing  $b$  instead of  $a$ . We will specify how the probability weights over the actual actions are calculated from the output of  $H_{ext}$  and how the losses are generated for the imaginary experts of  $H_{ext}$ .

At time  $t$ , suppose  $H_{ext}$  outputs probability  $q_{a,b}^t$  for the expert corresponding to  $f_{a,b}$ . Then compute the probability vector  $\mathbf{p}^t = (p_1^t, p_2^t, \dots, p_N^t)$  on the actual actions as a fixed point of

$$\mathbf{p}^t = \sum_{a,b \in [N]} q_{a,b}^t \mathbf{p}_{a \rightarrow b}^t,$$

where  $\mathbf{p}_{a \rightarrow b}^t$  denotes the probability vector obtained from  $\mathbf{p}^t$  by changing the weight of action  $a$  to zero at putting it on action  $b$ . This can also be expressed as

$$\mathbf{p}^{t\top} = \sum_{a,b} q_{a,b}^t \mathbf{p}^{t\top} \mathbf{M}_{f_{a,b}} = \mathbf{p}^{t\top} \sum_{a,b} q_{a,b}^t \mathbf{M}_{f_{a,b}}.$$

Let the adversary return back  $\mathbf{l}^t$  as the loss vector at time  $t$ . The loss incurred at time  $t$  by each of the imaginary experts for  $f_{a,b}$  is calculated as

$$l_{f_{a,b}}^t = \mathbf{l}^t \cdot \mathbf{p}_{i \rightarrow j}^t = \mathbf{p}^{t\top} \mathbf{M}_{f_{a,b}} \mathbf{l}^t.$$

This quantity can be thought of as the loss incurred if we followed the expert’s suggestion of playing  $b$  instead of  $a$ . Stoltz and Lugosi [SL05] showed that this achieves low internal regret. For an arbitrary set of modification rules  $\mathcal{F}$ , we have an expert for each modification rule  $f \in \mathcal{F}$  and the probability and loss vectors are now calculated as

$$\mathbf{p}^t = \mathbf{p}^{t\top} \sum_{f \in \mathcal{F}} q_f^t \mathbf{M}_f$$

and

$$l_f^t = \mathbf{p}^{t\top} \mathbf{M}_f \mathbf{l}^t.$$

We now discuss the ideas we use from Blum and Mansour [BM07a]. We start with the case where  $\mathcal{S} = \mathcal{I} \times \mathcal{F}_{ext}$  for some  $\mathcal{I}$ . In this case, there is an expert for each  $(I, f_a) \in \mathcal{S}$ . There is a weight  $w_{I,a}^t$  associated with this expert at the end of time  $t$  where

$$w_{I,a}^t = \beta^{-\tilde{R}_{I,a}^t}$$

and

$$\tilde{R}_{I,a}^t = \sum_{t'=1}^t I(t') (\beta l_H^{t'} - l_a^{t'})$$

for some parameter  $\beta \in (0, 1)$ . Above,  $l_H^t$  is the actual loss incurred at time  $t$ . The quantity  $\tilde{R}_{I,a}^t$  is called a “less-strict” external regret. The probability  $p_a^t$  associated with the action  $a$  at time  $t$  is then proportional to  $\sum_{I \in \mathcal{I}} I(t) w_{I,a}^t$ . By optimizing for the parameter  $\beta$ , Blum and Mansour [BM07a]

show that this achieves a low external regret with respect to all time selection functions.

To generalize this idea for wide range regret, where  $\mathcal{S} = \mathcal{I} \times \mathcal{F}$ , they introduce an expert for each  $a \in [N]$ ,  $I \in \mathcal{I}$  and  $f \in \mathcal{F}$ . There is a weight  $w_{a,I,f}^t$  for each such expert. Note that this does not simplify to the reduction in the previous paragraph for the case when  $\mathcal{F} = \mathcal{F}_{ext}$ . Instead, in the next section we obtain a reduction where there are experts only for each  $(I, f) \in \mathcal{S}$ . Intuitively, this is where we remove the polynomial dependence of wide range regret on  $N$  and obtain a slightly simpler reduction.

### 4 A Reduction from Wide Range Regret to External Regret

We will prove Theorem 2 in this section. We first give an algorithm that when given a low external regret algorithm as a black box uses it to guarantee low wide range regret.

**Theorem 3** *Given an algorithm  $H_{ext}$  with external regret  $R(T, N)$  when the losses are from  $[-1, +1]$ , one can construct an algorithm  $H$  that when given losses from  $[0, 1]$  satisfies:*

- $R_{H,\mathcal{S}} = R(T, |\mathcal{S}|)$
- The running time of  $H$  is polynomial in the running time of  $H_{ext}$ ,  $T$ ,  $N$ , and  $|\mathcal{S}|$ .

**Idea:**  $H$  will basically simulate an instance of  $H_{ext}$  with the elements of  $\mathcal{S}$  being the actions. Figure 1 shows the inputs and outputs of  $H$  and  $H_{ext}$  at time  $t$ . At time  $t$ ,  $H_{ext}$  produces some  $q_{I,f}^t$  for each  $(I, f) \in \mathcal{S}$ , where the  $q_{I,f}^t$  form a probability distribution over  $\mathcal{S}$ .  $H$  will then use this to come up with a probability vector  $\mathbf{p}^t = (p_1^t, p_2^t, \dots, p_N^t)$  on the actual actions.  $H$  will basically pick a random  $(I, f)$  with probability proportional to  $I(t)q_{I,f}^t$ . After this, it picks a vector  $\mathbf{p}^t$  over the actual actions such that  $\mathbf{p}^t$  is a fixed point of such a random  $f$ , i.e., modifying  $\mathbf{p}^t$  by  $f$  in expectation just yields  $\mathbf{p}^t$ . Intuitively, the loss passed to the black box  $H_{ext}$  for  $(I, f)$  is such that  $q_{I,f}^t$  measures the regret with respect to time selection function  $I$  of not having modified the output of  $H$  using function  $f$ . Multiplying this by  $I(t)$  takes care of the relevance of  $(I, f)$  at time  $t$ . Basically, the algorithm makes sure that if the regret with respect to  $(I, f)$  was large so far, then that regret doesn’t increase at the current step.

**Proof:** We first specify how  $H$  computes  $\mathbf{p}^t$  and  $\mathbf{l}^t$  at time  $t$ . To compute  $\mathbf{p}^t$ , get  $\mathbf{q}^t$  from  $H_{ext}$ . If  $\sum_{(I,f) \in \mathcal{S}} I(t)q_{I,f}^t = 0$ , then output any probability vector  $\mathbf{p}$ . Otherwise define  $\mathbf{p}^t$  to be any vector satisfying

$$\mathbf{p}^{t\top} = \mathbf{p}^{t\top} \left( \frac{\sum_{(I,f) \in \mathcal{S}} I(t)q_{I,f}^t \mathbf{M}_f}{\sum_{(I,f) \in \mathcal{S}} I(t)q_{I,f}^t} \right). \quad (1)$$

This is well defined since  $\sum_{(I,f) \in \mathcal{S}} I(t)q_{I,f}^t \neq 0$ . Such a vector  $\mathbf{p}^t$  exists since every row of

$$\frac{\sum_{(I,f) \in \mathcal{S}} I(t)q_{I,f}^t \mathbf{M}_f}{\sum_{(I,f) \in \mathcal{S}} I(t)q_{I,f}^t} \quad (2)$$

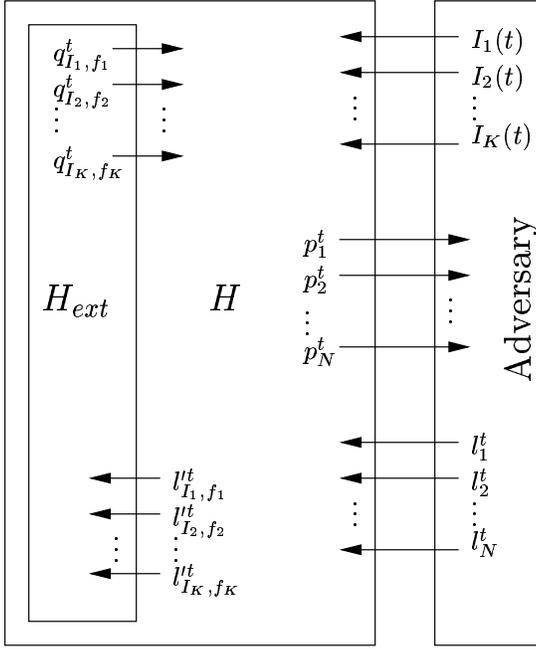


Figure 1: The reduction from wide range to external regret.

is a probability vector because  $\mathbf{M}_f$  has exactly one 1 in each row. That is, (2) defines the transition matrix of a Markov chain. When  $H$  gets back loss vector  $\mathbf{l}^t$ , it computes

$$l_{I,f}^t = I(t) \sum_{a \in N} p_a^t (l_{f(a)}^t - l_a^t) = I(t) \mathbf{p}^{t\top} (\mathbf{M}_f - \mathbf{I}) \mathbf{l}^t$$

where  $\mathbf{I}$  is the identity matrix. This yields

$$\sum_t l_{I,f}^t = \sum_t I(t) \mathbf{p}^{t\top} (\mathbf{M}_f - \mathbf{I}) \mathbf{l}^t = -R_{H,I,f}. \quad (3)$$

That is,  $l_{I,f}^t$  is exactly the decrease at time  $t$  of the regret with respect to  $(I, f)$ . It is easy to check that  $l_{I,f}^t \in [-1, +1]$ .

From the low external regret guarantee of  $H_{ext}$ , for all  $(I, f) \in \mathcal{S}$ :

$$\sum_t \sum_{(J,g) \in \mathcal{S}} q_{J,g}^t l_{J,g}^t \leq \sum_t l_{I,f}^t + R(T, |\mathcal{S}|). \quad (4)$$

We will next show that

$$\sum_{(J,g) \in \mathcal{S}} q_{J,g}^t l_{J,g}^t = 0. \quad (5)$$

Together with (3) and (4), this will show that for all  $(I, f) \in \mathcal{S}$ ,

$$0 \leq -R_{H,I,f} + R(T, |\mathcal{S}|),$$

or  $R_{H,I,f} \leq R(T, |\mathcal{S}|)$  which proves the theorem.

We now proceed to prove (5).

$$\begin{aligned} \sum_{(J,g) \in \mathcal{S}} q_{J,g}^t l_{J,g}^t &= \sum_{(J,g)} q_{J,g}^t J(t) \mathbf{p}^{t\top} (\mathbf{M}_g - \mathbf{I}) \mathbf{l}^t \\ &= \sum_{(J,g)} q_{J,g}^t J(t) \mathbf{p}^{t\top} \mathbf{M}_g \mathbf{l}^t - \sum_{(J,g)} q_{J,g}^t J(t) \mathbf{p}^{t\top} \mathbf{l}^t \\ &= \mathbf{p}^{t\top} \left( \sum_{(J,g)} J(t) q_{J,g}^t \mathbf{M}_g \right) \mathbf{l}^t - \left( \sum_{(J,g)} q_{J,g}^t J(t) \right) (\mathbf{p}^{t\top} \mathbf{l}^t). \end{aligned}$$

**(Case 1:)** Suppose  $\sum_{(J,g)} J(t) q_{J,g}^t \neq 0$ . In this case we can use (1) to get

$$\begin{aligned} \sum_{(J,g) \in \mathcal{S}} q_{J,g}^t l_{J,g}^t &= \left( \sum_{(J,g)} q_{J,g}^t J(t) \right) (\mathbf{p}^{t\top} \mathbf{l}^t) \\ &\quad - \left( \sum_{(J,g)} q_{J,g}^t J(t) \right) (\mathbf{p}^{t\top} \mathbf{l}^t) \\ &= 0. \end{aligned}$$

**(Case 2:)** Assume  $\sum_{(J,g)} J(t) q_{J,g}^t = 0$ . Then  $J(t) q_{J,g}^t = 0$  for all pairs  $(J, g)$  since  $J(t)$  and  $q_{J,g}^t$  are all non-negative, which implies

$$\sum_{(J,g) \in \mathcal{S}} q_{J,g}^t l_{J,g}^t = 0. \quad \blacksquare$$

It can be seen easily that Theorem 3 and Lemma 1 imply Theorem 2.

## 5 A First-Order Bound for Wide Range Regret

If we are only concerned with regret bounds as a function of  $T$  and  $N$  (called “zero-order” bounds in Cesa-Bianchi et al. [CBMS05]), Theorem 2 matches (up to a constant) the known upper bounds for external, internal and swap regret. One can also try to obtain “first-order” bounds, bounds that depend on the sum of payoffs of actions instead of the time. For example, Blum and Mansour [BM07a] show a  $O(\sqrt{L_{min} \log(NM)} + \log(NM))$  upper bound for minimizing external regret with respect to a set  $\mathcal{I}$  of  $M$  time selection functions, where  $L_{min} = \max_I \min_a L_{I,a}$  and  $L_{I,a} = \sum_t I(t) l_a^t$ . For the case when there is at least one “real” expert that does well most of the time, such a bound will be much tighter than a zero-order bound. One can hope to use external regret algorithms with good first-order bounds like the following to come up with good first-order bounds for wide range regret.

**Lemma 4** (Cesa-Bianchi et al. [CBFH<sup>+</sup>97]) *There exists an algorithm with running time polynomial in  $T$  and  $N$  and external regret  $O(\sqrt{L_{min} \log N} + \log N)$  when the losses are picked from  $[0, 1]$ .*

We need an algorithm that can handle losses from the interval  $[-1, +1]$  in Theorem 3. One way to use the algorithm from Lemma 4 is to map the losses  $l_{I,f}^t$  to the interval  $[0, 1]$  by a linear transformation. But this also changes the loss of best action and makes the first order bound obtained very weak. Another alternative is to tinker with the quantity that  $l_{I,f}^t$  signifies. If we are concerned only with modification rules (and not time selection functions), we can redefine  $l_f^t$  as

$$l_f^t = \sum_{a \in N} p_a^t l_{f(a)}^t.$$

But for technical reasons, this can’t be done if we are also working with time selection functions. Note that the only term in (4) that depends on  $I$  and  $f$  is  $l_{I,f}^t$ , and hence it must

capture all the terms that depend on *either*  $I$  or  $f$  in the definition of  $R_{H,I,f}$ . So we give a method based on the approach of Blum and Mansour [BM07a]. The main idea is to define a *reduced regret* for each pair  $(I, f)$ .

**Theorem 5** *There exists an online algorithm that for any  $\mathcal{S}$  satisfies:*

- The wide range regret with respect to  $\mathcal{S}$  is at most  $O(\sqrt{L_{min} \log |\mathcal{S}|} + \log |\mathcal{S}|)$ , where

$$L_{min} = \max_I \min_{(I,f) \in \mathcal{S}} \sum_t I(t) \mathbf{p}^{t\top} \mathbf{M}_f \mathbf{l}^t.$$

- The running time is polynomial in  $T$ ,  $N$ , and  $|\mathcal{S}|$ .

**Proof:** Define the loss of  $H$  with respect to  $I$  till time  $t$  as

$$L_{H,I}^t = \sum_{t'=1}^t I(t') \mathbf{p}^{t'} \cdot \mathbf{l}^{t'},$$

and the loss of  $H$  with respect to  $(I, f)$  till time  $t$  as

$$L_{H,I,f}^t = \sum_{t'=1}^t I(t') \mathbf{p}^{t'\top} \mathbf{M}_f \mathbf{l}^{t'}.$$

We assume that at any time  $t$ , not all  $I(t)$  are zero. This is without loss of generality since in this case, the losses defined above don't change at time  $t$ . For some  $\beta \in (0, 1)$  to be fixed later, we basically run an exponentially weighted predictor with a weight for each pair  $(I, f)$ . The weight of  $(I, f)$  at the end of time  $t$  is  $w_{I,f}^t = \beta^{-\tilde{R}_{H,I,f}^t}$ , where

$$\tilde{R}_{H,I,f}^t = \beta L_{H,I}^t - L_{H,I,f}^t.$$

That is,  $\tilde{R}_{H,I,f}^t$  is a regret of  $H$  with respect to  $(I, f)$  where the incurred loss is reduced by a factor  $\beta$ . We define  $q_{I,f}^t = w_{I,f}^{t-1} / W^{t-1}$ , where  $W^t = \sum_{(I,f) \in \mathcal{S}} w_{I,f}^t$  is the sum of the weights.

At time  $t$ , the algorithm does the following. It computes  $q_{I,f}^t$  as above. The probability vector  $\mathbf{p}^t$  over the actual actions is picked as in (2). This is well defined since  $w_{I,f}^t$  (and hence  $q_{I,f}^t$ ) are all non-zero and at least one of the  $I(t)$  is also non-zero (by assumption). Then the algorithm updates all the losses and weights when it gets back  $\mathbf{l}^t$  from the adversary. We first show that the sum of the weights can not increase at any time.

**Claim 6**

$$\forall t: \sum_{(I,f) \in \mathcal{S}} w_{I,f}^t \leq \sum_{(I,f) \in \mathcal{S}} w_{I,f}^{t-1}$$

**Proof:** We will use the fact that for any  $\beta \in (0, 1)$  and  $x \in [0, 1]$ ,  $\beta^x \leq 1 - (1 - \beta)x$  and  $\beta^{-x} \leq 1 + (1 - \beta)x/\beta$ . This

gives

$$\begin{aligned} \sum_{(I,f) \in \mathcal{S}} w_{I,f}^t &= \sum_{(I,f)} w_{I,f}^{t-1} \beta^{I(t) (\mathbf{p}^{t\top} \mathbf{M}_f \mathbf{l}^t - \beta \mathbf{p}^t \cdot \mathbf{l}^t)} \\ &\leq \sum_{(I,f)} \left[ w_{I,f}^{t-1} \left( 1 - (1 - \beta) I(t) \mathbf{p}^{t\top} \mathbf{M}_f \mathbf{l}^t \right) \right. \\ &\quad \left. \times \left( 1 + (1 - \beta) I(t) \mathbf{p}^t \cdot \mathbf{l}^t \right) \right] \\ &\leq \sum_{(I,f)} w_{I,f}^{t-1} - \left[ (1 - \beta) W^{t-1} \sum_{(I,f)} q_{I,f}^t I(t) \mathbf{p}^{t\top} \mathbf{M}_f \mathbf{l}^t \right] \\ &\quad + \left[ (1 - \beta) W^{t-1} \sum_{(I,f)} q_{I,f}^t I(t) \mathbf{p}^t \cdot \mathbf{l}^t \right] \\ &= \sum_{(I,f)} w_{I,f}^{t-1} - \left[ (1 - \beta) W^{t-1} \mathbf{p}^{t\top} \left( \sum_{(I,f)} q_{I,f}^t I(t) \mathbf{M}_f \right) \mathbf{l}^t \right] \\ &\quad + \left[ (1 - \beta) W^{t-1} \left( \sum_{(I,f)} q_{I,f}^t I(t) \right) (\mathbf{p}^t \cdot \mathbf{l}^t) \right] \\ &= \sum_{(I,f)} w_{I,f}^{t-1}. \end{aligned}$$

Above, the second inequality follows from the definition of  $q_{I,f}^t$  and the last equality follows from (2). ■

We now get back to the proof of the theorem. The claim implies that for all  $(I, f) \in \mathcal{S}$ ,

$$\beta^{-(\beta L_{H,I}^T - L_{H,I,f}^T)} = \beta^{-\tilde{R}_{H,I,f}^T} = w_{I,f}^T \leq \sum_{(J,g) \in \mathcal{S}} w_{J,g}^0 = |\mathcal{S}|$$

which gives

$$(\beta L_{H,I}^T - L_{H,I,f}^T) \log(1/\beta) \leq \log |\mathcal{S}|$$

or

$$L_{H,I} \leq \frac{L_{H,I,f} + \frac{\log |\mathcal{S}|}{\log(1/\beta)}}{\beta}.$$

Since for a given  $I$ , the statement is true for all  $f$  such that  $(I, f) \in \mathcal{S}$ , we can rewrite it as:

$$L_{H,I} \leq \frac{L_{H,I,\min} + \frac{\log |\mathcal{S}|}{\log(1/\beta)}}{\beta}$$

where

$$L_{H,I,\min} = \min_{f:(I,f) \in \mathcal{S}} L_{H,I,f}.$$

Setting  $\beta$  so that

$$\beta^{-1} = 1 + \min \left\{ \sqrt{\frac{\log |\mathcal{S}|}{L_{min}}}, \frac{1}{2} \right\}$$

gives the theorem. ■

## Acknowledgments

The authors would like to thank Yishay Mansour for comments on an early draft of the paper.

## References

- [BM07a] Avrim Blum and Yishay Mansour. From external to internal regret. *J. Mach. Learn. Res.*, 8:1307–1324, 2007.
- [BM07b] Avrim Blum and Yishay Mansour. Learning, regret minimization and equilibria. In Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani, editors, *Algorithmic Game Theory*, chapter 4. Cambridge University Press, 2007.
- [CBFH<sup>+</sup>97] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997.
- [CBMS05] Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. In *COLT*, pages 217–232, 2005.
- [FSSW97] Yoav Freund, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. Using and combining predictors that specialize. In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 334–343, 1997.
- [FV98] Dean Foster and Rakesh V. Vohra. Asymptotic calibration. *Biometrika*, 85:379–390, 1998.
- [Han57] J. Hannan. Approximation to bayes risk in repeated plays. In M.Dresher, A. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pages 97–139. Princeton University Press, 1957.
- [Leh03] Ehud Lehrer. A wide range no-regret theorem. *Games and Economic Behavior*, 42(1):101–115, 2003.
- [LW94] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.
- [SL05] Gilles Stoltz and Gábor Lugosi. Internal regret in on-line portfolio selection. *Mach. Learn.*, 59(1-2):125–159, 2005.

---

# An Efficient Reduction of Ranking to Classification

---

**Nir Ailon**

Google Research  
76 Ninth Ave, 4th Floor  
New York, NY 10011  
nailon@google.com

**Mehryar Mohri**

Courant Institute and Google Research  
251 Mercer Street  
New York, NY 10012  
mohri@cims.nyu.edu

## Abstract

This paper describes an efficient reduction of the learning problem of ranking to binary classification. The reduction is randomized and guarantees a pairwise misranking regret bounded by that of the binary classifier, improving on a recent result of Balcan et al. (2007) which ensures only twice that upper-bound. Moreover, our reduction applies to a broader class of ranking loss functions, admits a simple proof, and the expected time complexity of our algorithm in terms of number of calls to a classifier or preference function is also improved from  $\Omega(n^2)$  to  $O(n \log n)$ . In addition, when the top  $k$  ranked elements only are required ( $k \ll n$ ), as in many applications in information extraction or search engine design, the time complexity of our algorithm can be further reduced to  $O(k \log k + n)$ . Our reduction and algorithm are thus practical for realistic applications where the number of points to rank exceeds several thousands. Much of our results also extend beyond the bipartite case previously studied. To further complement them, we also derive lower bounds for any deterministic reduction of ranking to binary classification, proving that randomization is necessary to achieve our reduction guarantees.

## 1 Introduction

The learning problem of ranking arises in many modern applications, including the design of search engines, information extraction, and movie recommendation systems. In these applications, the ordering of the documents or movies returned is a critical aspect of the system.

The problem has been formulated within two distinct settings. In the *score-based setting*, the learning algorithm receives a labeled sample of pairwise preferences and returns a *scoring function*  $f: U \rightarrow \mathbb{R}$  which induces a linear ordering of the points in the set  $U$ . Test points are simply ranked according to the values of  $f$  for those points. Several ranking algorithms, including RankBoost (Freund et al., 2003; Rudin et al., 2005), SVM-type ranking (Joachims, 2002), and other algorithms such as PRank (Crammer & Singer, 2001; Agarwal & Niyogi, 2005), were designed for this setting. Gener-

alization bounds have been given in this setting for the pairwise misranking error (Freund et al., 2003; Agarwal et al., 2005), including margin-based bounds (Rudin et al., 2005). Stability-based generalization bounds have also been given in this setting for wide classes of ranking algorithms both in the case of bipartite ranking (Agarwal & Niyogi, 2005) and the general case (Cortes et al. 2007b; 2007a).

A somewhat different two-stage scenario was considered in other publications starting with (Cohen et al., 1999), and later (Balcan et al., 2007), which we will refer to as the *preference-based setting*. In the first stage of that setting, a preference function  $h: U \times U \mapsto [0, 1]$  is learned, where values of  $h(u, v)$  closer to one indicate that  $u$  is ranked above  $v$  and values closer to zero the opposite.  $h$  is typically assumed to be the output of a classification algorithm trained on a sample of labeled pairs, and can be for example a convex combination of simpler preference functions as in (Cohen et al., 1999). A crucial difference with the score-based setting is that, in general, the preference function  $h$  may not induce a linear ordering. The relation it induces may be non-transitive, thus we may have for example  $h(u, v) = h(v, w) = h(w, u) = 1$  for three distinct points  $u, v$ , and  $w$ . To rank a test subset  $V \subseteq U$ , in the second stage, the algorithm orders the points in  $V$  by making use of the preference function  $h$  learned in the first stage. The subset ranking set-up examined by Cossock and Zhang (2006), though distinct, also bears some resemblance with this setting.

This paper deals with the preference-based ranking setting just described. The advantage of this setting is that the learning algorithm is not required to return a linear ordering of all points in  $U$ , which may be impossible to achieve faultlessly in accordance with a general possibly non-transitive pairwise preference labeling. This is more likely to be achievable exactly or with a better approximation when the algorithm is requested instead, to supply a linear ordering, only for limited subsets  $V \subseteq U$ .

When the preference function is obtained as the output of a binary classification algorithm, the preference-based setting can be viewed as a reduction of ranking to classification. The second stage specifies how the ranking is obtained using the preference function.

Cohen et al. (1999) showed that in the second stage of the preference-based setting, the general problem of finding a linear ordering with as few pairwise misrankings as possible with respect to the preference function  $h$  is NP-complete. The authors presented a greedy algorithm based on the tour-

nement degree, that is, for a given element  $u$ , the difference between the number of elements it is preferred to versus the number of those preferred to  $u$ . The bound proven by the authors, formulated in terms of the pairwise disagreement loss  $l$  with respect to the preference function  $h$ , can be written as  $l(\sigma_{greedy}, h) \leq 1/2 + l(\sigma_{optimal}, h)/2$ , where  $l(\sigma_{greedy}, h)$  is the loss achieved by the permutation  $\sigma_{greedy}$  returned by their algorithm and  $l(\sigma_{optimal}, h)$  the one achieved by the optimal permutation  $\sigma_{optimal}$  with respect to the preference function  $h$ . This bound was given for the general case of ranking, but, in the particular case of bipartite ranking, a random ordering can achieve a pairwise disagreement loss of  $1/2$  and thus the bound is not informative. Note that the algorithm can be viewed as a derandomization technique.

More recently, Balcan et al. (2007) studied the bipartite ranking problem. In this particular case, the loss of an output ranking is measured by counting pairs of ranked elements, one of which is positive and the other negative (based on some ground truth). They showed that sorting the elements of  $V$  according to the same tournament degree used by Cohen et al. (1999) guarantees a regret of at most  $2r$  using a binary classifier with regret  $r$ . (The regret is defined as a calibration of the loss function that aligns a theoretical optimum with 0.) However, due to the quadratic nature of the definition of the tournament degree, their algorithm requires  $\Omega(n^2)$  calls to the preference function  $h$ , where  $n = |V|$  is the number of objects to rank.

We describe an efficient randomized algorithm for the second stage of preference-based setting and thus for reducing the learning problem of ranking to binary classification. We improve on the recent result of Balcan et al. (2007), by guaranteeing a pairwise misranking regret of at most  $r$  using a binary classifier with regret  $r$ , thereby improving the bound by a factor of 2. Our reduction applies, with different constants, to a broader class of ranking loss functions, admits a simple proof, and the expected running time complexity of our algorithm in terms of number of calls to a classifier or preference function is improved from  $\Omega(n^2)$  to  $O(n \log n)$ . Furthermore, when the top  $k$  ranked elements only are required ( $k \ll n$ ), as in many applications in information extraction or search engines, the time complexity of our algorithm can be further reduced to  $O(k \log k + n)$ . Our reduction and algorithm are thus practical for realistic applications where the number of points to rank exceeds several thousands. The price paid for this improvement is in resorting to nondeterminism. Indeed, our algorithms are randomized, but this turns out to be necessary. We give a simple proof of a lower bound of  $2r$  for *any* deterministic reduction of ranking to binary classification with classification regret  $r$ , thereby generalizing to all deterministic reductions a lower bound result of Balcan et al. (2007).

To appreciate our improvement of the reduction bound from a factor of 2 to 1, consider the case of a binary classifier with an error rate of just 25%, which is quite reasonable in many applications. Assume that the Bayes error is close to zero for the classification problem and similarly that for the ranking problem that the regret and loss approximately coincide. Then, the bound of Balcan et al. (2007) guarantees for the ranking algorithm a pairwise misranking error of at most 50%. But, since a random ranking can achieve 50% pairwise

misranking error, the bound turns out not to be informative in that case. Instead, with a factor of 1, the bound ensures a pairwise misranking of at most 25%.

Much of our results also extend beyond the bipartite case previously studied by Balcan et al. (2007) to the general case of ranking. A by-product of our proofs is a bound on the pairwise disagreement loss with respect to the preference function  $h$  that we will compare to the result given by Cohen et al. (1999).

The algorithm used by Balcan et al. (2007) to produce a ranking based on the preference function is known as sort-by-degree and has been recently used in the context of minimizing the feedback arcset in tournaments (Coppersmith et al., 2006). Here, we use a different algorithm, QuickSort, which has also been recently used for minimizing the feedback arcset in tournaments (Ailon et al. 2005; 2007). The techniques presented build upon earlier work by Ailon et al. (2005; 2007) on combinatorial optimization problems over rankings and clustering.

The remainder of the paper is structured as follows. In Section 2, we introduce the definitions and notation used in future sections and introduce a general family of loss functions for ranking. Section 3 describes a simple and efficient algorithm for reducing ranking to binary classification, proves several bounds guaranteeing the quality of the ranking produced by the algorithm, and analyzes the running-time complexity of our algorithm. In Section 4, we derive a lower bound for any deterministic reduction of ranking to binary classification. In Section 5, we discuss the relationship of the algorithm and its proof with previous related work in combinatorial optimization, and discuss key assumptions related to the notion of regret in this context.

## 2 Preliminaries

This section introduces several preliminary definitions necessary for the presentation of our results. In what follows,  $U$  will denote a universe of elements, e.g., the collection of all possible query-result pairs returned by a web search task, and  $V \subseteq U$  will denote a small subset thereof, e.g., a preliminary list of relevant results for a given query. For simplicity of notation we will assume that  $U$  is a set of integers, so that we are always able to choose a minimal canonical element in a finite subset, as we do in (9) below. This arbitrary ordering should not be confused with the ranking problem we are considering.

### 2.1 General Definitions and Notation

We first briefly discuss the learning setting and assumptions made here and compare them with those of Balcan et al. (2007) and Cohen et al. (1999).

In what follows,  $V \subseteq U$  represents a finite subset extracted from some arbitrary universe  $U$ , which is the set we wish to rank at each round. The notation  $S(V)$  denotes the set of *rankings* on  $V$ , that is the set of injections from  $V$  to  $[n] = \{1, \dots, n\}$ , where  $n = |V|$ . If  $\sigma \in S(V)$  is such a ranking, then  $\sigma(u)$  is the rank of an element  $u \in V$ , where lower ranks are interpreted as preferable ones. More precisely, we say that  $u$  is preferred over  $v$  with respect to  $\sigma$  if  $\sigma(u) < \sigma(v)$ . For convenience, and abusing notation, we

also write  $\sigma(u, v) = 1$  if  $\sigma(u) < \sigma(v)$  and  $\sigma(u, v) = 0$  otherwise. We let  $\binom{V}{k}$  denote the collection of all subsets of size exactly  $k$  of  $V$ . To distinguish between functions taking ordered vs. unordered arguments in what follows, we will use the notation  $F_{u_1 u_2 \dots u_k}$  to denote  $k$  unordered arguments for a function  $F$  defined on  $\binom{V}{k}$  and  $F(u_1, u_2, \dots, u_k)$  to denote  $k$  ordered arguments for a function  $F$  defined on  $\underbrace{V \times \dots \times V}_k$ .

## 2.2 Ground truth

As in standard learning scenarios, at each round, there is an underlying unknown ground truth which we wish the output of the learning algorithm to agree with as much as possible. The ground truth is a ranking that we denote by  $\sigma^* \in S(V)$ , equipped with a function  $\omega$  assigning different *importance* weight to pairs of positions. The combination  $(\sigma^*, \omega)$  is extremely expressive, as we shall see below in Section 2.5. It can encode in particular the standard average pairwise mis-ranking or AUC loss assumed by Balcan et al. (2007) in a bipartite setting, but also more sophisticated ones capturing misrankings among the top  $k$ , and other losses that are close but distinct from those considered by Cléménçon and Vayatis (2007).

## 2.3 Preference function

As with both (Cohen et al., 1999) and (Balcan et al., 2007), we assume that a preference function  $h: U \times U \rightarrow [0, 1]$  is learned in a first learning stage. The convention is that the higher  $h(u, v)$  is, the more our belief that  $u$  should be preferred to  $v$ . The function  $h$  satisfies *pairwise consistency*:  $h(u, v) + h(v, u) = 1$ , but need not even be transitive on 3-tuples (cycles may be induced). The second stage uses  $h$  to output a proper ranking  $\sigma$ , as we shall further discuss below. The running time complexity of the second stage is measured with respect to the number of calls to  $h$ .

## 2.4 Output of Learning Algorithm

The final output of the second stage of the algorithm,  $\sigma$ , is a proper ranking of  $V$ . Its cost is measured differently in (Balcan et al., 2007) and (Cohen et al., 1999). In the former, it is measured against the unknown ground truth and compared to the cost of  $h$  against the ground truth. The rationale is that the information encoded in  $h$  contains all pairwise preference information using the state-of-the-art binary classification. In (Cohen et al., 1999),  $\sigma$  is measured against the given preference function  $h$ , and compared to the theoretically best one can obtain. Thus, there  $h$  plays the role of a known ground truth.

## 2.5 Loss Functions

We are now ready to define the loss functions used to measure the quality of an output ranking  $\sigma$  either with respect to  $\sigma^*$ , as in (Balcan et al., 2007), or with respect to  $h$ , as in (Cohen et al., 1999).

The following general loss function  $L_\omega$  measures the quality of a ranking  $\sigma$  with respect to a desired one  $\sigma^*$  using a weight function  $\omega$  (described below):

$$L_\omega(\sigma, \sigma^*) = \binom{n}{2}^{-1} \sum_{u \neq v} \sigma(u, v) \sigma^*(v, u) \omega(\sigma^*(u), \sigma^*(v)).$$

The sum is over all pairs  $u, v$  in the domain  $V$  of the rankings  $\sigma, \sigma^*$ . It counts the number of inverted pairs  $u, v \in V$  weighed by  $\omega$ , which assigns importance coefficients to pairs, based on their positions in the ground truth  $\sigma^*$ . The function  $\omega$  must satisfy the following three natural axioms, which will be necessary in our analysis:

(P1) Symmetry:  $\omega(i, j) = \omega(j, i)$  for all  $i, j$ ;

(P2) Monotonicity:  $\omega(i, j) \leq \omega(i, k)$  if either  $i < j < k$  or  $i > j > k$ ;

(P3) Triangle inequality:  $\omega(i, j) \leq \omega(i, k) + \omega(k, j)$ .

This definition is very general and encompasses many useful, well studied distance functions. Setting  $\omega(i, j) = 1$  for all  $i \neq j$  yields the unweighted pairwise misranking measure or the so-called Kemeny distance function.

For a fixed integer  $k$ , the following function

$$\omega(i, j) = \begin{cases} 1 & \text{if } ((i \leq k) \vee (j \leq k)) \wedge (i \neq j) \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

can be used to emphasize ranking at the top  $k$  elements. Mis-ranking of pairs with one element ranked among the top  $k$  is penalized by this function. This can be of interest in applications such as information extraction or search engines where the ranking of the top documents matters more. For this emphasis function, all elements ranked below  $k$  are in a tie. In fact, it is possible to encode any tie relation using  $\omega$ .

**Bipartite Ranking.** In a bipartite ranking scenario,  $V$  is partitioned into a positive and negative set  $V^+$  and  $V^-$  of sizes  $m^+$  and  $m^-$  respectively, where  $m^+ + m^- = |V| = n$ . For this scenario (Balcan et al., 2007; Hanley & McNeil, 1982; Lehmann, 1975), we are often interested in the AUC score of  $\sigma \in S(V)$  defined as follows:

$$1 - \text{AUC}(V^+, V^-, \sigma) = \frac{1}{m^- m^+} \sum_{u, v \in V} \mathbf{1}_{(u, v) \in V^+ \times V^-} \sigma(v, u).$$

This expression measures the probability given a random *crucial* pair of elements, one of which is positive and the other negative, that the pair is misordered in  $\sigma$ . It is immediate to verify that this is equal to  $L_\omega(\sigma, \sigma^*)$ , where  $\sigma^*$  is any ranking placing  $V^+$  ahead of  $V^-$ , and

$$\omega(i, j) = \frac{\binom{n}{2}}{m^- m^+} \begin{cases} 1 & (i \leq m^+) \wedge (j > m^+) \\ 1 & (j \leq m^+) \wedge (i > m^+) \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

**Simplified notation.** To avoid carrying  $\sigma^*$  and  $\omega$ , we will define for convenience

$$\tau^*(u, v) = \sigma^*(u, v) \omega(\sigma^*(u), \sigma^*(v))$$

and

$$L(\sigma, \tau^*) := L_\omega(\sigma, \sigma^*) = \binom{n}{2}^{-1} \sum_{u \neq v} \sigma(u, v) \tau^*(v, u).$$

We will formally call  $\tau^*$  a *generalized ranking*, and it will take the role of the ground truth. If  $\omega$  is obtained as in (2) for some integers  $m^+, m^-$  satisfying  $m^+ + m^- = n$  then we will say that the corresponding  $\tau^*$  is *bipartite*.

It is immediate to verify from the properties of the weight function  $\omega$  that for all  $u, v, w \in V$ ,

$$\tau^*(u, v) \leq \tau^*(u, w) + \tau^*(w, v) . \quad (3)$$

If  $\tau^*$  is bipartite, then additionally,

$$\begin{aligned} \tau^*(u, v) + \tau^*(v, w) + \tau^*(w, u) = \\ \tau^*(v, u) + \tau^*(w, v) + \tau^*(u, w) . \end{aligned} \quad (4)$$

## 2.6 Preference Loss Function

We need to extend the definition to measure the loss of a preference function  $h$  with respect to  $\sigma^*$ . In contrast with the loss function just defined, we need to define a *preference loss* measuring a generalized ranking's disagreements with respect to a preference function  $h$  when measured against  $\tau^*$ . We can readily extend the loss definitions defined above as follows:

$$L(h, \tau^*) = L_\omega(h, \sigma^*) = \sum_{u \neq v} h(u, v) \tau^*(v, u) .$$

As explained above,  $L(h, \tau^*)$  is the ideal loss the learning algorithm will aim to achieve with the output ranking hypothesis  $\sigma$ .

## 2.7 Input Distribution

The set  $V$  we wish to rank together with the ground truth  $\tau^*$  are drawn as pair from a distribution we denote by  $D$ . In other words,  $\tau^*$  may be a random function of  $V$ . For our analysis of the loss though, it is convenient to think of  $V$  and  $\tau^*$  as fixed, because our bounds will be conditioned on fixed  $V, \tau^*$  and will easily generalize to the stochastic setting. Finally, we say that  $D$  is bipartite if  $\tau^*$  is bipartite with probability 1.

## 2.8 Regret Functions

The notion of regret is commonly used to measure the difference between the loss incurred by a learning algorithm and that of some *best* alternative. This section introduces the definitions of regret that we will be using to quantify the quality of a ranking algorithm in this context. We will define a notion of *weak* and *strong* regret for both ranking and classification losses as follows.

To define a strong ranking regret, we subtract from the loss function the minimal loss that could have been obtained from a global ranking  $\tilde{\sigma}$  of  $U$ . More precisely, we define:

$$\begin{aligned} \mathcal{R}_{rank}(A, D) = E_{V, \tau^*, s} [L(A_s(V), \tau^*)] \\ - \min_{\tilde{\sigma} \in S(U)} E_{V, \tau^*} [L(\tilde{\sigma}|_V, \tau^*)] , \end{aligned}$$

where  $\tilde{\sigma}|_V \in S(V)$  is defined by restricting the ranking  $\tilde{\sigma} \in S(U)$  to  $V$  in a natural way, and  $A$  is a possibly randomized algorithm using a stream of random bits  $s$  (and a pre-learned preference function  $h$ ) to output a ranking  $A_s(V)$  in  $S(V)$ .

As for the strong preference loss, it is natural to subtract the minimal loss over all, possibly cyclic, preference functions on  $U$ .

More precisely, we define:

$$\mathcal{R}_{class}(h, D) = E_{V, \tau^*} [L(h|_V, \tau^*)] - \min_{\tilde{h}} E_{V, \tau^*} [L(\tilde{h}|_V, \tau^*)] ,$$

where the minimum is over  $\tilde{h}$ , a preference function over  $U$ , and  $\cdot|_V$  is a restriction operator on preference functions defined in the natural way.

The weak ranking and classification regret functions  $\mathcal{R}'_{rank}$  and  $\mathcal{R}'_{class}$  are defined as follows:

$$\begin{aligned} \mathcal{R}'_{rank}(A, D) = E_{V, \tau^*, s} [L(A_s(V), \tau^*)] \\ - E_V \min_{\tilde{\sigma} \in S(V)} E_{\tau^*|V} [L(\tilde{\sigma}, \tau^*)] \end{aligned} \quad (5)$$

$$\begin{aligned} \mathcal{R}'_{class}(h, D) = E_{V, \tau^*} [L(h|_V, \tau^*)] \\ - E_V \min_{\tilde{h}} E_{\tau^*|V} [L(\tilde{h}, \tau^*)] , \end{aligned} \quad (6)$$

where  $\tau^*|V$  is the random variable  $\tau^*$  conditioned on fixed  $V$ . The difference between  $\mathcal{R}$  and  $\mathcal{R}'$  for both ranking and classification is that in their definition the min operator and the  $E_V$  operator are permuted.

The following inequalities follow from the concavity of min and Jensen's inequality:

$$\begin{aligned} \mathcal{R}'_{rank}(A, D) \geq \mathcal{R}_{rank}(A, D) \quad \text{and} \\ \mathcal{R}'_{class}(A, D) \geq \mathcal{R}_{class}(A, D) . \end{aligned} \quad (7)$$

For a fixed  $V$  and any  $u, v \in V$ , let

$$e(u, v) = E_{\tau^*|V} [\tau^*(u, v)] . \quad (8)$$

The reason we work with  $\mathcal{R}'_{class}$  is because the preference function  $\tilde{h}$  over  $U$  obtaining the min in the definition of  $\mathcal{R}'_{class}$  can be determined locally for any  $u, v \in U$  by

$$\tilde{h}(u, v) = \begin{cases} 1 & e(u, v) > e(v, u) \\ 0 & e(v, u) > e(u, v) \\ \mathbf{1}_{u > v} & \text{otherwise} . \end{cases} \quad (9)$$

Also, equation (3) holds true with  $e$  replacing  $\tau^*$ , and similarly for (4) if  $D$  is bipartite (by linearity of expectation). We cannot do a similar thing when working with the strong regret function  $\mathcal{R}_{class}$ .

The reason we work with weak ranking regret is for compatibility with our choice of weak classification regret, although our upper bounds on  $\mathcal{R}'_{rank}$  trivially apply to  $\mathcal{R}_{rank}$  in virtue of (7).

In Section 5.4, we will discuss certain assumptions under which our results work for the notion of strong regret as well. Note that Balcan et al. (2007) also implicitly use such an assumption in deriving their regret bounds. Our regret bounds (second part of Theorem 2) hold under the same assumption. Our result is thus exactly comparable with theirs.

## 3 Algorithm for Ranking Using a Preference Function

This section describes and analyzes an algorithm for obtaining a global ranking of a subset using a prelearned preference function  $h$ , which corresponds to the second stage of the preference-based setting. Our bound on the loss will be derived using conditional expectation on the preference loss assuming a fixed subset  $V \subseteq U$ , and fixed ground truth  $\tau^*$ .

To further simplify the analysis, we assume that  $h$  is binary, that is  $h(u, v) \in \{0, 1\}$  for all  $u, v \in U$ .

### 3.1 Description

One simple idea to obtain a global ranking of the points in  $V$  consists of using a standard comparison-based sorting algorithm where the comparison operation is based on the preference function. However, since in general the preference function is not transitive, the property of the resulting permutation obtained is unclear.

This section shows however that the permutation generated by the standard QuickSort algorithm provides excellent guarantees.<sup>1</sup> Thus, the algorithm we suggest is the following. Pick a random *pivot* element  $u$  uniformly at random from  $V$ . For each  $v \neq u$ , place  $v$  on the left<sup>2</sup> of  $u$  if  $h(v, u) = 1$ , and to its right otherwise. Proceed recursively with the array to the left of  $u$  and the one to its right and return the concatenation of the permutation returned by the left recursion,  $u$ , and the permutation returned by the right recursion.

We will denote by  $Q_s^h(V)$  the permutation resulting in running QuickSort on  $V$  using preference function  $h$ , where  $s$  is the random stream of bits used by QuickSort for the selection of the pivots. As we shall see in the next two sections, this algorithm produces high-quality global rankings in a time-efficient manner.

### 3.2 Ranking Quality Guarantees

The following theorems bound the ranking quality of the algorithm described, for both loss and regret, in the general and bipartite cases.

**Theorem 1 (Loss bounds in general case)** *For any fixed subset  $V \subseteq U$ , preference function  $h$  on  $V$ , and generalized ranking  $\tau^*$  on  $V$ , the following bound holds:*

$$E_s[L(Q_s^h(V), \tau^*)] \leq 2L(h, \tau^*) . \quad (10)$$

Taking the expectation of both sides, this implies immediately that

$$E_{V, \tau^*, s}[L(Q_s^h(V), \tau^*)] \leq 2E_{V, \tau^*}[L(h, \tau^*)], \quad (11)$$

where  $h$  could depend on  $V$ .

**Theorem 2 (Loss and regret bounds in bipartite case)** *For any fixed  $V \subseteq U$ , preference function  $h$  over  $V$ , and bipartite generalized ranking  $\tau^*$ , the following bound holds:*

$$E_s[L(Q_s^h(V), \tau^*)] = L(h, \tau^*) \quad (12)$$

$$\mathcal{R}'_{rank}(Q_s^h(\cdot), D) \leq \mathcal{R}'_{class}(h, D) . \quad (13)$$

Taking the expectation of both sides of Equation 12, this implies immediately that if  $(V, \tau^*)$  is drawn from a bipartite distribution  $D$ , then

$$E_{V, \tau^*, s}[L(Q_s^h(V), \tau^*)] = E_{V, \tau^*}[L(h, \tau^*)], \quad (14)$$

where  $h$  can depend on  $V$ .

To present the proof of these theorems, we need some tools helpful in the analysis of QuickSort, similar to those originally developed by Ailon et al. (2005). The next section introduces these tools.

<sup>1</sup>We are not assuming here transitivity as in standard textbook presentations of QuickSort.

<sup>2</sup>We will use the convention that ranked items are written from left to right, starting with the most preferred ones.

### 3.3 Analysis of QuickSort

Assume  $V$  is fixed, and let  $Q_s = Q_s^h(V)$  be the (random) ranking output by QuickSort on  $V$  using the preference function  $h$ . During the execution of QuickSort, the order between two elements  $u, v \in V$  is determined in one of two ways:

- Directly:  $u$  (or  $v$ ) was selected as the pivot with  $v$  (resp.  $u$ ) present in the same sub-array in a recursive call to QuickSort. We denote by  $p_{uv} = p_{vu}$  the probability of that event. In that case, the algorithm orders  $u$  and  $v$  according to the preference function  $h$ .
- Indirectly: a third element  $w \in V$  is selected as pivot with  $w, u, v$  all present in the same sub-array in a recursive call to QuickSort,  $u$  is assigned to the left sub-array and  $v$  to the right (or vice-versa).

Let  $p_{uvw}$  denote the probability of the event that  $u, v$ , and  $w$  are present in the same array in a recursive call to QuickSort and that one of them is selected as pivot. Note that conditioned on that event, each of these three elements is equally likely to be selected as a pivot since the pivot selection is based on a uniform distribution.

If (say)  $w$  is selected among the three, then  $u$  will be placed on the left of  $v$  if  $h(u, w) = h(w, v) = 1$ , and to its right if  $h(v, w) = h(w, u) = 1$ . In all other cases, the order between  $u, v$  will be determined only in a deeper nested call to QuickSort.

Let  $X, Y: V \times V \rightarrow \mathbb{R}$  be any two functions on ordered pairs  $u, v \in V$ , and let  $Z: \binom{V}{2} \rightarrow \mathbb{R}$  be a function on unordered pairs. We define three functions  $\alpha[X, Y]: \binom{V}{2} \rightarrow \mathbb{R}$ ,  $\beta[X]: \binom{V}{3} \rightarrow \mathbb{R}$  and  $\gamma[Z]: \binom{V}{3} \rightarrow \mathbb{R}$  as follows:

$$\alpha[X, Y]_{uv} = X(u, v)Y(v, u) + X(v, u)Y(u, v),$$

$$\beta[X]_{uvw} =$$

$$\frac{1}{3}(h(u, v)h(v, w)X(w, u) + h(w, v)h(v, u)X(u, w)) +$$

$$\frac{1}{3}(h(v, u)h(u, w)X(w, v) + h(w, u)h(u, v)X(v, w)) +$$

$$\frac{1}{3}(h(u, w)h(w, v)X(v, u) + h(v, w)h(w, u)X(u, v)),$$

$$\gamma[Z]_{uvw} =$$

$$\frac{1}{3}(h(u, v)h(v, w) + h(w, v)h(v, u))Z_{uw} +$$

$$\frac{1}{3}(h(v, u)h(u, w) + h(w, u)h(u, v))Z_{vw} +$$

$$\frac{1}{3}(h(u, w)h(w, v) + h(v, w)h(w, u))Z_{uv} .$$

#### Lemma 3 (QuickSort Decomposition)

1. For any  $Z: \binom{V}{2} \rightarrow \mathbb{R}$ ,

$$\sum_{u < v} Z_{uv} = \sum_{u < v} p_{uv} Z_{uv} + \sum_{u < v < w} p_{uvw} \gamma[Z]_{uvw} .$$

2. For any  $X: V \times V \rightarrow \mathbb{R}$ ,

$$E_s[\sum_{u < v} \alpha[Q_s, X]_{uv}] =$$

$$\sum_{u < v} p_{uv} \alpha[h, X]_{uv} + \sum_{u < v < w} p_{uvw} \beta[X]_{uvw} .$$

**Proof:** To see the first part, notice that for every unordered pair  $u < v$  the expression  $Z_{uv}$  is accounted for on the RHS of the equation with total coefficient:

$$p_{uv} + \sum_{w \notin \{u,v\}} \frac{1}{3} p_{uvw} (h(u, w)h(w, v) + h(v, w)h(w, u)) .$$

Now,  $p_{uv}$  is the probability that the order of  $(u, v)$  is determined directly (by definition), and

$$\frac{1}{3} p_{uvw} (h(u, w)h(w, v) + h(v, w)h(w, u))$$

is the probability that their order is determined indirectly via  $w$  as pivot. Since each pair's ordering is accounted for exactly once, these probabilities are for pairwise disjoint events that cover the probability space. Thus, the total coefficient of  $Z_{uv}$  on the RHS is 1, as is on the LHS. The second part is proved similarly. ■

### 3.4 Loss Bounds

This section proves Theorem 1 and the first part of Theorem 2. For a fixed  $\tau^*$ , the loss incurred by QuickSort is  $L(Q_s, \tau^*) = \binom{n}{2}^{-1} \sum_{u < v} \alpha[Q_s, \tau^*]_{uv}$ . By the second part of Lemma 3, the expected loss is therefore

$$\mathbb{E}_s[L(Q_s, \tau^*)] = \binom{n}{2}^{-1} \left( \sum_{u < v} p_{uv} \alpha[h, \tau^*]_{uv} + \sum_{u < v < w} p_{uvw} \beta[\tau^*]_{uvw} \right) .$$

Also, the following holds by definition of  $L$ :

$$L(h, \tau^*) = \binom{n}{2}^{-1} \sum_{u < v} \alpha[h, \tau^*]_{uv} .$$

Thus, by the first part of Lemma 3,

$$L(h, \tau^*) = \binom{n}{2}^{-1} \left( \sum_{u < v} p_{uv} \alpha[h, \tau^*]_{uv} + \sum_{u < v < w} \gamma[\alpha[h, \tau^*]]_{uvw} \right) .$$

To complete the proof, it suffices to show that for all  $u, v, w$ ,

$$\beta[\tau^*]_{uvw} \leq 2\gamma[\alpha[h, \tau^*]]_{uvw} , \quad (15)$$

and that if  $\tau^*$  is bipartite, then

$$\beta[\tau^*]_{uvw} = \gamma[\alpha[h, \tau^*]]_{uvw} . \quad (16)$$

Up to symmetry, there are two cases to consider. The first case assumes that  $h$  induces a cycle on  $u, v, w$ , the second assumes that it doesn't.

1. Without loss of generality, assume  $h(u, v) = h(v, w) = h(w, u) = 1$ . Plugging in the definitions leads to

$$\beta[\tau^*]_{uvw} = \frac{1}{3} (\tau^*(u, v) + \tau^*(v, w) + \tau^*(w, u)), \text{ and } (17)$$

$$\gamma[\alpha[h, \tau^*]]_{uvw} = \frac{1}{3} (\tau^*(v, u) + \tau^*(w, v) + \tau^*(u, w)) . \quad (18)$$

If  $\tau^*$  is bipartite, then by (4) the right hand sides of (17) and (18) are equal, giving (16). Otherwise we use (3) to derive

$$\tau^*(u, v) \leq \tau^*(u, w) + \tau^*(w, v)$$

$$\tau^*(v, w) \leq \tau^*(v, u) + \tau^*(u, w)$$

$$\tau^*(w, u) \leq \tau^*(w, v) + \tau^*(v, u)$$

Summing up the three equations, this implies (15).

2. Without loss of generality, assume  $h(u, v) = h(v, w) = h(u, w) = 1$ . Plugging in the definitions gives

$$\beta[\tau^*]_{uvw} = \gamma[\alpha[h, \tau^*]]_{uvw} = \tau^*(w, u)$$

as required. ■

We now examine a consequence of Theorem 1 for QuickSort that can be compared with the bound given by Cohen et al. (1999) for a greedy algorithm based on the tournament degree. Let  $\sigma_{optimal}$  be the ranking with the least amount of pairwise disagreement with  $h$ :

$$\sigma_{optimal} = \underset{\sigma}{\operatorname{argmin}} L(h, \sigma) .$$

Then, the following corollary bounds the expected pairwise disagreement of QuickSort with respect to  $\sigma_{optimal}$  by twice that of the preference function with respect to  $\sigma_{optimal}$ .

**Corollary 4** For any  $V \subseteq U$  and preference function  $h$  over  $V$ , the following bound holds:

$$\mathbb{E}_s[L(Q_s^h(V), \sigma_{optimal})] \leq 2L(h, \sigma_{optimal}) . \quad (19)$$

The corollary is immediate since technically any ranking, in particular  $\sigma_{optimal}$ , can be taken as  $\sigma^*$  in the proof of Theorem 1.

**Corollary 5** Let  $V \subseteq U$  be an arbitrary subset of  $U$  and let  $\sigma_{optimal}$  be as above. Then, the following bound holds for the pairwise disagreement of the ranking  $Q_s^h(V)$  with respect to  $h$ :

$$\mathbb{E}_s[L(h, Q_s^h(V))] \leq 3L(h, \sigma_{optimal}) . \quad (20)$$

**Proof:** The result follows directly Corollary 4 and the application of the triangle inequality. ■

This result is in fact known from previous work (Ailon et al. 2005; 2007) where it is proven directly without resorting to the intermediate inequality (19). In fact, a better factor of 2.5 is known to be achievable using a more complicated algorithm, which gives hope for a 1.5 bound improving Theorem 1.

### 3.5 Regret Bounds for Bipartite case

This section proves the second part of Theorem 2, that is the regret bound. Since in the definition of  $\mathcal{R}'_{rank}$  and  $\mathcal{R}'_{class}$  the expectation over  $V$  is outside the min operator, we may continue to fix  $V$ . Let  $D_V$  denote the distribution over the bipartite  $\tau^*$  conditioned on  $V$ . By the definitions of  $\mathcal{R}'_{rank}$  and  $\mathcal{R}'_{class}$ , it is now sufficient to prove that

$$\begin{aligned} & \mathbb{E}_{\tau^*|V, s} [L(Q_s^h, \tau^*)] - \min_{\tilde{\sigma}} \mathbb{E}_{\tau^*|V} [L(\tilde{\sigma}, \tau^*)] \\ & \leq \mathbb{E}_{\tau^*|V} [L(h, \tau^*)] - \min_{\tilde{h}} \mathbb{E}_{\tau^*|V} [L(\tilde{h}, \tau^*)] . \end{aligned} \quad (21)$$

We let  $e(u, v)$  denote  $E_{\tau^*|V}[\tau^*(u, v)]$ , then by the linearity of expectation,  $E_{\tau^*|V}[L(\tilde{\sigma}, \tau^*)] = L(\tilde{\sigma}, e)$  and similarly  $E_{\tau^*|V}[L(\tilde{h}, \tau^*)] = L(\tilde{h}, e)$ . Thus, inequality 21 can be rewritten as

$$E[L(Q_s^h, e)] - \min_{\tilde{\sigma}} L(\tilde{\sigma}, e) \leq L(h, e) - \min_{\tilde{h}} L(\tilde{h}, e). \quad (22)$$

Now let  $\tilde{\sigma}$  and  $\tilde{h}$  be the minimizers of the min operators on the left and right sides, respectively. Recall that for all  $u, v \in V$ ,  $\tilde{h}(u, v)$  can be taken greedily as a function of  $e(u, v)$  and  $e(v, u)$ , as in (9):

$$\tilde{h}(u, v) = \begin{cases} 1 & e(u, v) > e(v, u) \\ 0 & e(u, v) < e(v, u) \\ \mathbf{1}_{u>v} & \text{otherwise (equality)} \end{cases} \quad (23)$$

Using Lemma 3 and linearity, the LHS of (22) can be rewritten as:

$$\binom{n}{2}^{-1} \left( \sum_{u<v} p_{uv} \alpha[h - \tilde{\sigma}, e]_{uv} + \sum_{u<v<w} p_{uvw} (\beta[e] - \gamma[\alpha[\tilde{\sigma}, e]])_{uvw} \right),$$

and the RHS of (22) as:

$$\binom{n}{2}^{-1} \left( \sum_{u<v} p_{uv} \alpha[h - \tilde{h}, e]_{uv} + \sum_{u<v<w} p_{uvw} \gamma[\alpha[h - \tilde{h}, e]]_{uvw} \right).$$

Now, clearly, for all  $(u, v)$  by construction of  $\tilde{h}$ , we must have  $\alpha[h - \tilde{\sigma}, e]_{uv} \leq \alpha[h - \tilde{h}, e]_{uv}$ . To conclude the proof of the theorem, we define  $F: \binom{n}{3} \rightarrow \mathbb{R}$  as follows:

$$F = \beta[e] - \gamma[\alpha[\tilde{\sigma}, e]] - (\gamma[\alpha[h, e]] - \gamma[\alpha[\tilde{h}, e]]) \quad (24)$$

It now suffices to prove that  $F_{uvw} \leq 0$  for all  $u, v, w \in V$ . Clearly  $F$  is a function of the values of

$$\begin{aligned} e(a, b) &: \{a, b\} \subseteq \{u, v, w\} \\ h(a, b) &: \{a, b\} \subseteq \{u, v, w\} \\ \tilde{\sigma}(a, b) &: \{a, b\} \subseteq \{u, v, w\}. \end{aligned} \quad (25)$$

Recall that  $\tilde{h}$  depends on  $e$ . By (3) and (4), the  $e$ -variables can take values satisfying the following constraints for all  $u, v, w \in V$ :

$$\forall \{a, b, c\} = \{u, v, w\}, e(a, c) \leq e(a, b) + e(b, c) \quad (26)$$

$$e(u, v) + e(v, w) + e(w, u) = e(v, u) + e(w, v) + e(u, w) \quad (27)$$

$$\forall a, b \in \{u, v, w\}, e(a, b) \geq 0 \quad (28)$$

Let  $P \subseteq \mathbb{R}^6$  denote the polytope defined by (26-28) in the variables  $e(a, b)$  for  $\{a, b\} \subseteq \{u, v, w\}$ . We subdivide  $P$  into smaller subpolytopes on which the  $\tilde{h}$  variables are constant. Up to symmetries, we can consider only two cases: (i)  $\tilde{h}$  induces a cycle on  $u, v, w$  and (ii)  $\tilde{h}$  is cycle-free on  $u, v, w$ .

(i) Without loss of generality, assume  $\tilde{h}(u, v) = \tilde{h}(v, w) = \tilde{h}(w, u) = 1$ . But this implies that  $e(u, v) \geq e(v, u)$ ,  $e(v, w) \geq e(w, v)$  and  $e(w, u) \geq e(u, w)$ . Together with (27) and (28), this implies that  $e(u, v) = e(v, u)$ ,  $e(v, w) = e(w, v)$ , and  $e(w, u) = e(u, w)$ . Consequently,

$$\begin{aligned} \beta[e]_{uvw} &= \gamma[\alpha[\tilde{\sigma}, e]]_{uvw} \\ &= \gamma[\alpha[h, e]]_{uvw} = \gamma[\alpha[\tilde{h}, e]]_{uvw} \\ &= \frac{1}{3}(e(u, v) + e(v, w) + e(w, u)) \end{aligned} \quad ,$$

and  $F_{uvw} = 0$ , as required.

(ii) Without loss of generality, assume  $\tilde{h}(u, v) = \tilde{h}(v, w) = \tilde{h}(u, w) = 1$ . This implies that

$$\begin{aligned} e(u, v) &\geq e(v, u) \\ e(v, w) &\geq e(w, v) \\ e(u, w) &\geq e(w, u) \end{aligned} \quad (29)$$

Let  $\tilde{P} \subseteq P$  denote the polytope defined by (29) and (26)-(28). Clearly,  $F$  is linear in the 6  $e$  variables when all the other variables are fixed. Since  $F$  is also homogenous in the  $e$  variables, it suffices to prove that  $F \leq 0$  for  $e$  taking values in  $\tilde{P}' \subseteq \tilde{P}$ , which is defined by adding the constraint, say,

$$\sum_{a, b \in \{u, v, w\}} e(a, b) = 2 \quad .$$

It is now enough to prove that  $F \leq 0$  for  $\tau^*$  being a vertex of of  $\tilde{P}'$ . This finite set of cases can be easily checked to be:

$$\begin{aligned} (e(u, v), e(v, u), e(u, w), \\ e(w, u), e(w, v), e(v, w)) \in A \cup B \end{aligned} \quad ,$$

where

$$\begin{aligned} A &= \{(0, 0, 1, 0, 0, 1), (1, 0, 1, 0, 0, 0)\} \\ B &= \{(.5, .5, .5, .5, 0, 0), (.5, .5, 0, 0, .5, .5), \\ &\quad (0, 0, .5, .5, .5, .5)\} \end{aligned} \quad .$$

The points in  $B$  were already checked in case (i), which is, geometrically, a boundary of case (ii). It remains to check the two points in  $A$ .

• case  $(0, 0, 1, 0, 0, 1)$ : plugging in the definitions, one checks that:

$$\begin{aligned} \beta[e]_{uvw} &= \frac{1}{3}(h(w, v)h(v, u) + h(w, u)h(u, v)) \\ \gamma[\alpha[h, e]]_{uvw} &= \\ &\frac{1}{3}((h(u, v)h(v, w) + h(w, v)h(v, u))h(w, u) \\ &\quad + (h(v, u)h(u, w) + h(w, u)h(u, v))h(w, v)) \\ \gamma[\alpha[\tilde{h}, e]]_{uvw} &= 0 \end{aligned} \quad .$$

Clearly  $F$  could be positive only of  $\beta_{uvw} = 1$ , which happens if and only if either  $h(w, v)h(v, u) =$

1 or  $h(w, u)h(u, v) = 1$ . In the former case, we obtain that

$$\text{either } h(w, v)h(v, u)h(w, u) = 1 \quad (30)$$

$$\text{or } h(v, u)h(u, w)h(w, v) = 1, \quad (31)$$

both implying that  $\gamma[\alpha[h, e]]_{uvw} \geq 1$ , thus  $F \leq 0$ . In the latter case,

$$\text{either } h(w, u)h(u, v)h(w, v) = 1 \quad (32)$$

$$\text{or } h(u, v)h(v, w)h(w, u) = 1, \quad (33)$$

both implying again that  $\gamma[\alpha[h, e]]_{uvw} \geq 1$  and thus  $F \leq 0$ .

- case  $(1, 0, 1, 0, 0)$ : plugging in the definitions, one checks that:

$$\beta[e]_{uvw} = \frac{1}{3}(h(w, v)h(v, u) + h(v, w)h(w, u))$$

$$\gamma[\alpha[h, e]]_{uvw} =$$

$$\frac{1}{3}((h(u, v)h(v, w) + h(w, v)h(v, u))h(w, u)$$

$$+ (h(u, w)h(w, v) + h(v, w)h(w, u))h(v, u)) .$$

$$\gamma[\alpha[\tilde{h}, e]]_{uvw} = 0 .$$

Now  $F$  could be positive if and only if

$$\text{either } h(w, v)h(v, u) = 1 \quad (34)$$

$$\text{or } h(v, w)h(w, u) = 1 . \quad (35)$$

In the former case, we obtain that

$$\text{either } h(w, v)h(v, u)h(w, u) = 1 \quad (36)$$

$$\text{or } h(v, u)h(u, w)h(w, v) = 1, \quad (37)$$

both implying that  $\gamma[\alpha[h, e]]_{uvw} \geq 1$ , and thus  $F \leq 0$ . In the latter case,

$$\text{either } h(v, w)h(w, u)h(v, u) = 1 \quad (38)$$

$$\text{or } h(u, v)h(v, w)h(w, u) = 1, \quad (39)$$

both implying again that  $\gamma[\alpha[h, e]]_{uvw} \geq 1$  and thus  $F \leq 0$ .

This concludes the proof of the second part of Theorem 2. ■

### 3.6 Time Complexity

Running QuickSort does not entail  $\Omega(|V|^2)$  accesses to  $h_{u,v}$ . The following bound on the running time is proven in Section 3.6.

**Theorem 6** *The expected number of times QuickSort accesses to the preference function  $h$  is at most  $O(n \log n)$ . Moreover, if only the top  $k$  elements are sought then the bound is reduced to  $O(k \log k + n)$  by pruning the recursion.*

It is well known that QuickSort on cycle-free tournaments runs in time  $O(n \log n)$ , where  $n$  is the size of the set we wish to sort. That this holds for QuickSort on general tournaments is a simple extension (communicated by Heikki Mannila) which we present it here to keep this presentation self-contained. The second part of the theorem requires some more work.

**Proof:** Let  $T(n)$  be the maximum expected running time of QuickSort on a possibly cyclic tournament on  $n$  vertices in terms of number of comparisons. Let  $G = (V, A)$  denote a tournament. The main observation is that each vertex  $v \in V$  is assigned to the left recursion with probability exactly  $\text{outdeg}(v)/n$  and to the right with probability  $\text{indeg}(v)/n$ , over the choice of the pivot. Therefore, the expected size of both the left and right recursions is exactly  $(n-1)/2$ . The separation itself costs  $n-1$  comparisons. The resulting recursion formula  $T(n) \leq n-1 + 2T((n-1)/2)$  clearly solves to  $T(n) = O(n \log n)$ .

Assume now that only the  $k$  first elements of the output are sought, that is, we are interested in outputting only elements in positions  $1, \dots, k$ . The algorithm which we denote by  $k$ -QuickSort is clear: recurse with  $\min\{k, n_L\}$ -QuickSort on the left side and  $\max\{0, k - n_L - 1\}$ -QuickSort on the right side, where  $n_L, n_R$  are the sizes of the left and right recursions respectively and 0-QuickSort takes 0 steps by assumption. To make the analysis simpler, we will assume that whenever  $k \geq n/8$ ,  $k$ -QuickSort simply returns the output of the standard QuickSort, which runs in expected time  $O(n \log n) = O(n + k \log k)$ , within the sought bound. Fix a tournament  $G$  on  $n$  vertices, and let  $t_k(G)$  denote the running time of  $k$ -QuickSort on  $G$ , where  $k < n/8$ . Denote the (random) left and right sub-tournaments by  $G_L$  and  $G_R$  respectively, and let  $n_L = |G_L|, n_R = |G_R|$  denote their sizes in terms of number of vertices. Then, clearly,

$$t_k(G) = n - 1 + t_{\min\{k, n_L\}}(G_L) + t_{\max\{0, k - n_L - 1\}}(G_R). \quad (40)$$

Assume by structural induction that for all  $\{k', n': k' \leq n' < n\}$  and for all tournaments  $G'$  on  $n'$  vertices,

$$\mathbb{E}[t_{k'}(G')] \leq cn' + c'k' \log k'$$

for some global  $c, c' > 0$ . Then, by conditioning on  $G_L, G_R$ , taking expectations on both sides of (40) and by induction,

$$\mathbb{E}[t_k(G) \mid G_L, G_R] \leq n - 1 + cn_L +$$

$$c' \min\{k, n_L\} \log \min\{k, n_L\} + cn_R \mathbf{1}_{n_L < k-1} +$$

$$c' \max\{k - n_L - 1, 0\} \log \max\{k - n_L - 1, 0\}.$$

By convexity of the function  $x \mapsto x \log x$ ,

$$\min\{k, n_L\} \log \min\{k, n_L\} +$$

$$\max\{k - n_L - 1, 0\} \log \max\{k - n_L - 1, 0\}$$

$$\leq k \log k. \quad (41)$$

Thus,

$$\mathbb{E}[t_k(G) \mid G_L, G_R] \leq n - 1 + cn_L +$$

$$cn_R \mathbf{1}_{n_L < k-1} + c'k \log k. \quad (42)$$

By conditional expectation,

$$\mathbb{E}[t_k(G)] \leq n - 1 + c(n-1)/2 + c'k \log k + c \mathbb{E}[n_R \mathbf{1}_{n_L < k-1}].$$

To complete the inductive hypothesis, we need to bound the quantity  $\mathbb{E}[n_R \mathbf{1}_{n_L < k-1}]$ , which is bounded by  $n \Pr[n_L < k-1]$ . The event  $\{n_L < k-1\}$ , equivalent to  $\{n_R > n-k\}$ , occurs when a vertex of out-degree at least  $n-k \geq 7n/8$  is chosen as pivot. For a random pivot  $v \in V$ , where  $V$  is the vertex set of  $G$ ,  $\mathbb{E}[\text{outdeg}(v)^2] \leq n^2/3 + n/2 \leq n^2/2.9$ .

Indeed, each pair of edges  $(v, u_1) \in A$  and  $(v, u_2) \in A$  for  $u_1 \neq u_2$  gives rise to a triangle which is counted exactly twice in the cross-terms, hence  $n^2/3$  which upper-bounds  $2\binom{n}{3}/n$ ;  $n/2$  bounds the diagonal. Thus,  $\Pr[\text{outdeg}(v) \geq 7n/8] = \Pr[\text{outdeg}(v)^2 \geq 49n^2/64] \leq 0.46$  (by Markov). Plugging in this value into our last estimate yields

$E[t_k(G)] \leq n - 1 + c(n - 1)/2 + c'k \log k + 0.46 \times cn$ , which is at most  $cn + c'k \log k$  for  $c \geq 30$ , as required. ■

## 4 Lower Bounds

Let  $r$  denote the classification regret. Balcan et al. (2007) proved a lower bound of  $2r$  for the regret of the algorithm MFAT defined as the solution to the minimum feedback arc-set problem on the tournament  $V$  with an edge  $(u, v)$  when  $h(u, v) = 1$ . More precisely, they showed an example of fixed  $V, h$ , and bipartite generalized ranking  $\tau^*$  on  $V$ , such that the classification regret of  $h$  tends to  $1/2$  of the ranking regret of MFAT on  $V, h$ . Note that in this case, since  $\tau^*$  is a fixed function of  $V$ , the regret and loss coincide both for classification and for ranking.

Here we give a simple proof of a more general theorem stating that same bound holds for *any* deterministic algorithm, including of course MFAT.

**Theorem 7** *For any deterministic algorithm  $A$  taking as input  $V \subseteq U$  and a preference function  $h$  on  $V$  and outputting a ranking  $\sigma \in S(V)$ , there exists a bipartite distribution  $D$  on  $(V, \tau^*)$  such that*

$$\mathcal{R}_{\text{rank}}(A, D) \geq 2 \mathcal{R}_{\text{class}}(h, D). \quad (43)$$

Note that the theorem implies that, in the bipartite case, no deterministic algorithm converting a preference function into a linear ranking can do better than a randomized algorithm, on expectation. Thus, randomization is essentially necessary in this setting.

The proof is based on an adversarial argument. In our construction, the support of  $D$  is reduced to a single pair  $(V, \tau^*)$  (deterministic input), thus the loss and both the weak and strong regrets coincide and a similar argument applies to the loss function and the weak regret functions.

**Proof:** Fix  $V = \{u, v, w\}$ , and let the support of  $D$  be reduced to  $(V, \tau^*)$ , where the bipartite generalized ranking  $\tau^*$  is one that we will select adversarially. Assume a cycle:  $h(u, v) = h(v, w) = h(w, u) = 1$ . Up to symmetry, there are two options for the output  $\sigma$  of  $A$  on  $V, h$ .

1.  $\sigma(u) < \sigma(v) < \sigma(w)$ : in this case, the adversary can choose  $\tau^*$  corresponding to the partition  $V^+ = \{w\}$  and  $V^- = \{u, v\}$ . Clearly,  $\mathcal{R}_{\text{class}}(h, D)$  now equals  $1/2$  since  $h$  is penalized only for misranking the pair  $(v, w)$ , but  $\mathcal{R}_{\text{rank}}(A, D) = 1$  since  $\sigma$  is misordering both  $(u, w)$  and  $(v, w)$ .
2.  $\sigma(w) < \sigma(v) < \sigma(u)$ : in this case, the adversary can choose  $\tau^*$  corresponding to the partition  $V^+ = \{u\}$  and  $V^- = \{v, w\}$ . Similarly,  $\mathcal{R}_{\text{class}}(h, D)$  now equals  $1/2$  since  $h$  is penalized only for misranking the pair  $(u, w)$ , while  $\mathcal{R}_{\text{rank}}(A, D) = 1$  since  $\sigma$  is misordering both  $(u, v)$  and  $(u, w)$ . ■

## 5 Discussion

### 5.1 History of QuickSort

The textbook algorithm, by now standard, was originally discovered by Hoare (1961). Montague and Aslam (Montague & Aslam, 2002) experimented with QuickSort for information retrieval (IR) by aggregating rankings from different sources of retrieval. They claimed an  $O(n \log n)$  time bound on the number of comparisons, although the proof seemed to rely on the folklore QuickSort proof without addressing the non-transitivity problem. They proved certain combinatorial bounds on the output of QuickSort and provided an empirical justification of its IR merits. Ailon et al. (2005) also considered the rank aggregation problem and proved theoretical cost bounds for many ranking problems on weighted tournaments. They strengthened these bounds by considering non-deterministic pivoting rules arising from solutions to certain ranking LP's. This work was later extended by Ailon (2007) to deal with rankings with ties, in particular, top- $k$  rankings. Hedge et al. (2007) and Williamson and van Zuylen (2007) derandomized the random pivot selection step in QuickSort for many of the combinatorial optimization problems studied by Ailon et al.

### 5.2 The decomposition technique

The technique developed in Lemma 3 is very general and can be used for a wide variety of loss functions and variants of QuickSort involving non-deterministic ordering rules (Ailon et al. 2005; 2007). Such results would typically amount to bounding  $\beta[X]_{uvw}/\gamma[Z]_{uvw}$  for some carefully chosen functions  $X, Z$  depending on the application.

### 5.3 Combinatorial Optimization vs. Learning of Ranking

QuickSort, sometimes referred to as FAS-Pivot in that context, was used by Ailon et al. (2005; 2007) to approximate certain NP-Hard weighted instances of the problem of minimum feedback arcset in tournaments (Alon, 2006). There is much similarity between the techniques used in that work and those of the analyses of this work, but there is also a significant difference that should be noted.

In the minimum feedback arc-set problem, we are given a tournament  $G$  and wish to find an acyclic tournament  $H$  on the same vertex set minimizing  $\Delta(G, H)$ , where  $\Delta$  counts the number of edges pointing in opposite directions between  $G, H$  (or a weighted version thereof). However, the cost we are considering is  $\Delta(G, H_\sigma)$  for some fixed acyclic tournament  $H_\sigma$  induced by some permutation  $\sigma$  (the ground truth). In this work, we showed in fact that if  $G'$  is obtained from  $G$  using QuickSort, then  $E[\Delta(G', H_\sigma)] \leq 2\Delta(G, H_\sigma)$  for *any*  $\sigma$  (Theorem 1). If  $H$  is the optimal solution to the (weighted) minimum feedback arc-set problem corresponding to  $G$ , then it is easy to see that  $\Delta(H, H_\sigma) \leq \Delta(G, H) + \Delta(G, H_\sigma) \leq 2\Delta(G, H_\sigma)$ . However, recovering  $G$  is NP-Hard in general. Approximating  $\Delta(G, H)$  modulo a constant factor  $1 + \varepsilon$  using an acyclic tournament  $H'$ , as in the combinatorial optimization world, only guarantees a constant factor of  $2 + \varepsilon$ :

$$\Delta(H', H_\sigma) \leq \Delta(G, H') + \Delta(G, H_\sigma) \leq (1 + \varepsilon)\Delta(G, H) + \Delta(G, H_\sigma) \leq (2 + \varepsilon)\Delta(G, H_\sigma) .$$

Thus, this work also adds a significant contribution to (Ailon et al., 2005; Ailon, 2007; Kenyon-Mathieu & Schudy, 2007).

#### 5.4 Weak vs. Strong Regret Functions

For the proof of the regret bound of Theorem 2 we used the fact that the minimizer  $\tilde{h}$  in the definition (5-6) of  $\mathcal{R}'_{class}$  could be determined independently for each pair  $u, v \in U$ , using (9). This could also be done for strong regrets if the distribution  $D$  on  $V, \tau^*$  satisfied the following pairwise IIA condition.

**Definition 8** A distribution  $D$  on subsets  $V \subseteq U$  and generalized rankings  $\tau^*$  on  $V$  satisfies the pairwise independence on irrelevant alternatives (pairwise IIA) if for all  $u, v \in U$  and for any two subsets  $V_1, V_2 \supseteq \{u, v\}$ ,

$$E_{\tau^*|V_1}[\tau^*(u, v)] = E_{\tau^*|V_2}[\tau^*(u, v)] .$$

Note: We chose the terminology IIA to match that used in Arrow’s seminal work (Arrow, 1950) to describe a similar notion.

When pairwise IIA holds, the average ground truth relation between  $u$  and  $v$ , conditioned on  $u, v$  included in  $V$ , is independent of  $V$ .

Recall that a bipartite  $\tau^*$  is derived from a pair  $\sigma^*, \omega$ , where  $\omega$  is defined using a term  $1/m^- m^+$ , for compatibility with the definition of AUC. The numbers  $m^+$  and  $m^-$  depend on the underlying size of the positive and negative sets partitioning of  $V$  and therefore cannot be inferred from  $(u, v)$  alone. Thus, in the standard bipartite case, the pairwise IIA assumption is not natural. If, however, we replaced our definitions in the bipartite case and used the following:

$$\omega(i, j) = \begin{cases} 1 & (i \leq m^+) \wedge (j > m^+) \\ 1 & (j \leq m^+) \wedge (i > m^+) \\ 0 & \text{otherwise,} \end{cases} \quad (44)$$

instead of (2), then it would be reasonable to believe that pairwise IIA does hold in the bipartite case. In fact, it would be reasonable to make the stronger assumption that for any fixed  $u, v \in U$  and  $V_1, V_2 \supseteq \{u, v\}$  the distribution of the random variables  $\tau^*(u, v)|V_1$  and  $\tau^*(u, v)|V_2$  are equal. This corresponds to the intuition that when comparing a pair  $u, v$  in a context of a set  $V$  containing them, human labelers are not as influenced by the irrelevant information  $V \setminus \{u, v\}$  as they would be by  $V \setminus \{u\}$  if asked to evaluate single elements  $u$ . The irrelevant information in  $V$  is often referred to as *anchor* in experimental psychology and economics (Ariely et al., 2008).

Our regret bounds would still hold if we used (44), but we chose (2) to present our results in terms of the familiar average pairwise misranking error or AUC loss.

Another possible assumption allowing usage of strong regrets is to let the preference function learned in the first stage depend on  $V$ . This is the assumption implicitly made by Balcan et al. (2007) (based on our private communication). We do not further elaborate on this assumption.

## 6 Conclusion

We described a reduction of the learning problem of ranking to classification. The efficiency of this reduction makes

it practical for large-scale information extraction and search engine applications. A finer analysis of QuickSort is likely to further improve our reduction bound by providing a concentration inequality for the algorithm’s deviation from its expected behavior using the confidence scores output by the classifier. Our reduction leads to a competitive ranking algorithm that can be viewed as an alternative to the algorithms previously designed for the score-based setting.

## 7 Acknowledgments

We thank Alina Beygelzimer and John Langford for helpful discussions. Mehryar Mohri’s work was partially funded by the New York State Office of Science Technology and Academic Research (NYSTAR).

## References

- Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., & Roth, D. (2005). Generalization bounds for the area under the roc curve. *Journal of Machine Learning Research*, 6, 393–425.
- Agarwal, S., & Niyogi, P. (2005). Stability and generalization of bipartite ranking algorithms. *COLT* (pp. 32–47).
- Ailon, N. (2007). Aggregation of partial rankings, p-ratings and top-m lists. *SODA*.
- Ailon, N., Charikar, M., & Newman, A. (2005). Aggregating inconsistent information: ranking and clustering. *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005* (pp. 684–693). ACM.
- Alon, N. (2006). Ranking tournaments. *SIAM J. Discrete Math.*, 20, 137–142.
- Ariely, D., Loewenstein, G., & Prelec, D. (2008). Coherent arbitrariness: Stable demand curves without stable preferences. *The Quarterly Journal of Economics*, 118, 73–105.
- Arrow, K. J. (1950). A difficulty in the concept of social welfare. *Journal of Political Economy*, 58, 328–346.
- Balcan, M.-F., Bansal, N., Beygelzimer, A., Coppersmith, D., Langford, J., & Sorkin, G. B. (2007). Robust reductions from ranking to classification. *COLT* (pp. 604–619). Springer.
- Cléménçon, S., & Vayatis, N. (2007). Ranking the best instances. *Journal of Machine Learning Research*, 8, 2671–2699.
- Cohen, W. W., Schapire, R. E., & Singer, Y. (1999). Learning to order things. *J. Artif. Intell. Res. (JAIR)*, 10, 243–270.
- Coppersmith, D., Fleischer, L., & Rudra, A. (2006). Ordering by weighted number of wins gives a good ranking for weighted tournaments. *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

- Cortes, C., Mohri, M., & Rastogi, A. (2007a). An Alternative Ranking Problem for Search Engines. *Proceedings of the 6th Workshop on Experimental Algorithms (WEA 2007)* (pp. 1–21). Rome, Italy: Springer-Verlag, Heidelberg, Germany.
- Cortes, C., Mohri, M., & Rastogi, A. (2007b). Magnitude-Preserving Ranking Algorithms. *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML 2007)*. Oregon State University, Corvallis, OR.
- Cossock, D., & Zhang, T. (2006). Subset ranking using regression. *COLT* (pp. 605–619).
- Crammer, K., & Singer, Y. (2001). Pranking with ranking. *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]* (pp. 641–647). MIT Press.
- Freund, Y., Iyer, R. D., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4, 933–969.
- Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*.
- Hedge, R., Jain, K., Williamson, D. P., & van Zuylen, A. (2007). "deterministic pivoting algorithms for constrained ranking and clustering problems". *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*.
- Hoare, C. (1961). Quicksort: Algorithm 64. *Comm. ACM*, 4, 321–322.
- Joachims, T. (2002). Optimizing search engines using click-through data. *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 133–142). New York, NY, USA: ACM Press.
- Kenyon-Mathieu, C., & Schudy, W. (2007). How to rank with few errors. *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing* (pp. 95–103). New York, NY, USA: ACM Press.
- Lehmann, E. L. (1975). *Nonparametrics: Statistical methods based on ranks*. San Francisco, California: Holden-Day.
- Montague, M. H., & Aslam, J. A. (2002). Condorcet fusion for improved retrieval. *Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, November 4-9, 2002* (pp. 538–548). ACM.
- Rudin, C., Cortes, C., Mohri, M., & Schapire, R. E. (2005). Margin-based ranking meets boosting in the middle. *Learning Theory, 18th Annual Conference on Learning Theory, COLT 2005, Bertinoro, Italy, June 27-30, 2005, Proceedings* (pp. 63–78). Springer.
- Williamson, D. P., & van Zuylen, A. (2007). "deterministic algorithms for rank aggregation and other ranking and clustering problems". *Proceedings of the 5th Workshop on Approximation and Online Algorithms (WAOA) (to appear)*.



---

# Learning from Collective Behavior

---

**Michael Kearns**

Computer and Information Science  
University of Pennsylvania  
mkearns@cis.upenn.edu

**Jennifer Wortman**

Computer and Information Science  
University of Pennsylvania  
wortmanj@seas.upenn.edu

## Abstract

Inspired by longstanding lines of research in sociology and related fields, and by more recent large-population human subject experiments on the Internet and the Web, we initiate a study of the computational issues in learning to model collective behavior from observed data. We define formal models for efficient learning in such settings, and provide both general theory and specific learning algorithms for these models.

## 1 Introduction

Collective behavior in large populations has been a subject of enduring interest in sociology and economics, and a more recent topic in fields such as physics and computer science. There is consequently now an impressive literature on mathematical models for collective behavior in settings as diverse as the diffusion of fads or innovation in social networks [10, 1, 2, 18], voting behavior [10], housing choices and segregation [22], herding behaviors in financial markets [27, 8], Hollywood trends [25, 24], critical mass phenomena in group activities [22], and many others. The advent of the Internet and the Web have greatly increased the number of both controlled experiments [7, 17, 20, 21, 8] and open-ended systems (such as Wikipedia and many other instances of “human peer-production”) that permit the logging and analysis of detailed collective behavioral data. It is natural to ask if there are learning methods specifically tailored to such models and data.

The mathematical models of the collective behavior literature differ from one another in important details, such as the extent to which individual agents are assumed to act according to traditional notions of rationality, but they generally share the significant underlying assumption that each agent’s current behavior is entirely or largely determined by the recent behavior of the other agents. Thus the collective behavior is a *social* phenomenon, and the population evolves over time according to its own internal dynamics — there is no exogenous “Nature” being reacted to, or injecting shocks to the collective.

In this paper, we introduce a computational theory of learning from collective behavior, in which the goal is to accurately model and predict the future behavior of a large

population after observing their interactions during a training phase of polynomial length. We assume that each agent  $i$  in a population of size  $N$  acts according to a fixed but unknown strategy  $c_i$  drawn from a known class  $\mathcal{C}$ . A strategy probabilistically maps the current population state to the next state or action for that agent, and each agent’s strategy may be different. As is common in much of the literature cited above, there may also be a network structure governing the population interaction, in which case strategies may map the local neighborhood state to next actions.

Learning algorithms in our model are given training data of the population behavior, either as repeated finite-length trajectories from multiple initial states (an *episodic* model), or in a single unbroken trajectory from a fixed start state (a *no-reset* model). In either case, they must efficiently (polynomially) learn to accurately predict or simulate (properties of) the future behavior of the same population. Our framework may be viewed as a computational model for learning the dynamics of an unknown Markov process — more precisely, a dynamic Bayes net — in which our primary interest is in Markov processes inspired by simple models for social behavior.

As a simple, concrete example of the kind of system we have in mind, consider a population in which each agent makes a series of choices from a fixed set over time (such as what restaurant to go to, or what political party to vote for). Like many previously studied models, we consider agents who have a desire to behave like the rest of the population (because they want to visit the popular restaurants, or want to vote for “electable” candidates). On the other hand, each agent may also have different and unknown intrinsic preferences over the choices as well (based on cuisine and decor, or the actual policies of the candidates). We consider models in which each agent balances or integrates these two forces in deciding how to behave at each step [12]. Our main question is: Can a learning algorithm watching the collective behavior of such a population for a short period produce an accurate model of their future choices?

The assumptions of our model fit nicely with the literature cited in the first paragraph, much of which indeed proposes simple stochastic models for how individual agents react to the current population state. We emphasize from the outset the difference between our interests and those common in multiagent systems and learning in games. In those fields, it is often the case that the agents themselves are acting according to complex and fairly general learning al-

gorithms (such as Q-learning [26], no-regret learning [9], fictitious play [3], and so on), and the central question is whether and when the population converges to particular, “nice” states (such as Nash or correlated equilibria). In contrast, while the agent strategies we consider are certainly “adaptive” in a reactive sense, they are much simpler than general-purpose learning algorithms, and we are interested in learning algorithms that *model* the full collective behavior no matter what its properties; there is no special status given either to particular states nor to any notion of convergence. Thus our interest is not in learning by the agents themselves, but at the higher level of an observer of the population.

Our primary contributions are:

- The introduction of a computational model for learning from collective behavior.
- The development of some general theory for this model, including a polynomial-time reduction of learning from collective behavior to learning in more traditional, single-target I.I.D. settings, and a separation between efficient learnability in collective models in which the learner does and does not see all intermediate population states.
- The definition of specific classes of agent strategies, including variants of the “crowd affinity” strategies sketched above, and complementary “crowd aversion” classes.
- Provably efficient algorithms for learning from collective behavior for these same classes.

The outline of the paper is as follows. In Section 2, we introduce our main model for learning from collective behavior, and then discuss two natural variants. Section 3 introduces and motivates a number of specific agent strategy classes that are broadly inspired by earlier sociological models, and provides brief simulations of the collective behaviors they can generate. Section 4 provides a general reduction of learning from collective behavior to a generalized PAC-style model for learning from I.I.D. data, which is used subsequently in Section 5, where we give provably efficient algorithms for learning some of the strategy classes introduced in Section 3. Brief conclusions and topics for further research are given in Section 6.

## 2 The Model

In this section we describe a learning model in which the observed data is generated from observations of trajectories (defined shortly) of the collective behavior of  $N$  interacting agents. The key feature of the model is the fact that each agent’s next state or action is always *determined by the recent actions of the other agents*, perhaps combined with some intrinsic “preferences” or behaviors of the particular agent. As we shall see, we can view our model as one for learning certain kinds of factored Markov processes that are inspired by models common in sociology and related fields.

Each agent may follow a different and possibly probabilistic strategy. We assume that the strategy followed by each agent is constrained to lie in a known (and possibly

large) class, but is otherwise unknown. The learner’s ultimate goal is not to discover each individual agent strategy per se, but rather to make accurate predictions of the *collective* behavior in novel situations.

### 2.1 Agent Strategies and Collective Trajectories

We now describe the main components of our framework:

- **State Space.** At each time step, each agent  $i$  is in some state  $s_i$  chosen from a known, finite set  $\mathcal{S}$  of size  $K$ . We often think of  $K$  as being large, and thus want algorithms whose running time scales polynomially in  $K$  and other parameters. We view  $s_i$  as the *action* taken by agent  $i$  in response to the recent population behavior. The joint action vector  $\vec{s} \in \mathcal{S}^N$  describes the current global state of the collective.
- **Initial State Distribution.** We assume that the initial population state  $\vec{s}^0$  is drawn according to a fixed but unknown distribution  $P$  over  $\mathcal{S}^N$ . During training, the learner is able to see trajectories of the collective behavior in which the initial state is drawn from  $P$ , and as in many standard learning models, must generalize with respect to this same distribution. (We also consider a no-reset variant of our model in Section 2.3.)
- **Agent Strategy Class.** We assume that each agent’s strategy is drawn from a known class  $\mathcal{C}$  of (typically probabilistic) mappings from the recent collective behavior into the agent’s next state or action in  $\mathcal{S}$ . We mainly consider the case in which  $c_i \in \mathcal{C}$  probabilistically maps the current global state  $\vec{s}$  into agent  $i$ ’s next state. However, much of the theory we develop applies equally well to more complex strategies that might incorporate a longer history of the collective behavior on the current trajectory, or might depend on summary statistics of that history.

Given these components, we can now define what is meant by a *collective trajectory*.

**Definition 1** Let  $\vec{c} \in \mathcal{C}^N$  be the vector of strategies for the  $N$  agents,  $P$  be the initial state distribution, and  $T \geq 1$  be an integer. A  $T$ -trajectory of  $\vec{c}$  with respect to  $P$  is a random variable  $\langle \vec{s}^0, \dots, \vec{s}^T \rangle$  in which the initial state  $\vec{s}^0 \in \mathcal{S}^N$  is drawn according to  $P$ , and for each  $t \in \{1, \dots, T\}$ , the component  $s_i^t$  of the joint state  $\vec{s}^t$  is obtained by applying the strategy  $c_i$  to  $\vec{s}^{t-1}$ . (Again, more generally we may also allow the strategies  $c_i$  to depend on the full sequence  $\vec{s}^0, \dots, \vec{s}^{t-1}$ , or on summary statistics of that history.)

Thus, a collective trajectory in our model is simply a Markovian sequence of states that *factors* according to the  $N$  agent strategies — that is, a dynamic Bayes net [19]. Our interest is in cases in which this Markov process is generated by particular models of social behavior, some of which are discussed in Section 3.

### 2.2 The Learning Model

We now formally define the learning model we study. In our model, learning algorithms are given access to an oracle  $\mathcal{O}_{\text{EXP}}(\vec{c}, P, T)$  that returns a  $T$ -trajectory  $\langle \vec{s}^0, \dots, \vec{s}^T \rangle$  of

$\vec{c}$  with respect to  $P$ . This is thus an *episodic* or *reset* model, in which the learner has the luxury of repeatedly observing the population behavior from random initial conditions. It is most applicable in (partially) controlled, experimental settings [7, 17, 20, 21, 8] where such “population resets” can be implemented or imposed. In Section 2.3 below we define a perhaps more broadly applicable variant of the model in which resets are not available; the algorithms we provide can be adapted for this model as well (Section 5.3).

The goal of the learner is to find a *generative model* that can efficiently produce trajectories from a distribution that is arbitrarily close to that generated by the true population. Thus, let  $\hat{M}(\vec{s}^0, T)$  be a (randomized) model output by a learning algorithm that takes as input a start state  $\vec{s}^0$  and time horizon  $T$ , and outputs a random  $T$ -trajectory, and let  $Q_{\hat{M}}$  denote the distribution over trajectories generated by  $\hat{M}$  when the start state is distributed according to  $P$ . Similarly, let  $Q_{\vec{c}}$  denote the distribution over trajectories generated by  $\mathcal{O}_{\text{EXP}}(\vec{c}, P, T)$ . Then the goal of the learning algorithm is to find a model  $\hat{M}$  making the  $\mathcal{L}_1$  distance  $\varepsilon(Q_{\hat{M}}, Q_{\vec{c}})$  between  $Q_{\hat{M}}$  and  $Q_{\vec{c}}$  small, where

$$\varepsilon(Q_{\hat{M}}, Q_{\vec{c}}) \equiv \sum_{\langle \vec{s}^0, \dots, \vec{s}^T \rangle} |Q_{\hat{M}}(\langle \vec{s}^0, \dots, \vec{s}^T \rangle) - Q_{\vec{c}}(\langle \vec{s}^0, \dots, \vec{s}^T \rangle)|.$$

A couple of remarks are in order here. First, note that we have defined the output of the learning algorithm to be a “black box” that simply produces trajectories from initial states. Of course, it would be natural to expect that this black box operates by having good approximations to every agent strategy in  $\vec{c}$ , and using collective simulations of these to produce trajectories, but we choose to define the output  $\hat{M}$  in a more general way since there may be other approaches. Second, we note that our learning criteria is both strong (see below for a discussion of weaker alternatives) and useful, in the sense that if  $\varepsilon(Q_{\hat{M}}, Q_{\vec{c}})$  is smaller than  $\epsilon$ , then we can sample  $\hat{M}$  to obtain  $O(\epsilon)$ -good approximations to the expectation of any (bounded) *function* of trajectories. Thus, for instance, we can use  $\hat{M}$  to answer questions like “What is the expected number of agents playing the plurality action after  $T$  steps?” or “What is the probability the entire population is playing the same action after  $T$  steps?” (In Section 2.4 below we discuss a weaker model in which we care only about one *fixed* outcome function.)

Our algorithmic results consider cases in which the agent strategies may themselves already be rather rich, in which case the learning algorithm should be permitted resources commensurate with this complexity. For example, the crowd affinity models have a number of parameters that scales with the number of actions  $K$ . More generally, we use  $\dim(\mathcal{C})$  to denote the complexity or dimension of  $\mathcal{C}$ ; in all of our imagined applications  $\dim(\cdot)$  is either the VC dimension for deterministic classes, or one of its generalizations to probabilistic classes (such as pseudo-dimension [11], fat-shattering dimension [15], combinatorial dimension [11], etc.).

We are now ready to define our learning model.

**Definition 2** *Let  $\mathcal{C}$  be an agent strategy class over actions  $S$ . We say that  $\mathcal{C}$  is **polynomially learnable from collective***

**behavior** if there exists an algorithm  $A$  such that for any population size  $N \geq 1$ , any  $\vec{c} \in \mathcal{C}^N$ , any time horizon  $T$ , any distribution  $P$  over  $\mathcal{S}^N$ , and any  $\epsilon > 0$  and  $\delta > 0$ , given access to the oracle  $\mathcal{O}_{\text{EXP}}(\vec{c}, P, T)$ , algorithm  $A$  runs in time polynomial in  $N$ ,  $T$ ,  $\dim(\mathcal{C})$ ,  $1/\epsilon$ , and  $1/\delta$ , and outputs a polynomial-time model  $\hat{M}$  such that with probability at least  $1 - \delta$ ,  $\varepsilon(Q_{\hat{M}}, Q_{\vec{c}}) \leq \epsilon$ .

We now discuss two reasonable variations on the model we have presented.

### 2.3 A No-Reset Variant

The model above assumes that learning algorithms are given access to repeated, independent trajectories via the oracle  $\mathcal{O}_{\text{EXP}}$ , which is analogous to the *episodic* setting of reinforcement learning. As in that field, we may also wish to consider an alternative “no-reset” model in which the learner has access only to a *single*, unbroken trajectory of states generated by the Markov process. To do so we must formulate an alternative notion of generalization, since on the one hand, the (distribution of the) initial state may quickly become irrelevant as the collective behavior evolves, but on the other, the state space is exponentially large and thus it is unrealistic to expect to model the dynamics from an *arbitrary* state in polynomial time.

One natural formulation allows the learner to observe any polynomially long prefix of a trajectory of states for training, and then to announce its readiness for the test phase. If  $\vec{s}$  is the final state of the training prefix, we can simply ask that the learner output a model  $\hat{M}$  that generates accurate  $T$ -step trajectories *forward* from the current state  $\vec{s}$ . In other words,  $\hat{M}$  should generate trajectories from a distribution close to the distribution over  $T$ -step trajectories that would be generated if each agent continued choosing actions according to his strategy. The length of the training prefix is allowed to be polynomial in  $T$  and the other parameters.

While aspects of the general theory described below are particular to our main (episodic) model, we note here that the algorithms we give for specific classes can in fact be adapted to work in the no-reset model as well. Such extensions are discussed briefly in Section 5.3.

### 2.4 Weaker Criteria for Learnability

We have chosen to formulate learnability in our model using a rather strong success criterion — namely, the ability to (approximately) simulate the full dynamics of the unknown Markov process induced by the population strategy  $\vec{c}$ . In order to meet this strong criterion, we have also allowed the learner access to a rather strong oracle, which returns all *intermediate* states of sampled trajectories.

There may be natural scenarios, however, in which we are interested only in specific *fixed* properties of collective behavior, and thus a weaker data source may suffice. For instance, suppose we have a fixed, real-valued *outcome function*  $F(\vec{s}^T)$  of final states (for instance, the fraction of agents playing the plurality action at time  $T$ ), with our goal being to simply learn a function  $G$  that maps initial states  $\vec{s}^0$  and a time horizon  $T$  to real values, and approximately minimizes

$$\mathbb{E}_{\vec{s}^0 \sim P} [ |G(\vec{s}^0, T) - \mathbb{E}_{\vec{s}^T} [F(\vec{s}^T)]| ]$$

where  $\vec{s}^T$  is a random variable that is the final state of a  $T$ -trajectory of  $\vec{c}$  from the initial state  $\vec{s}^0$ . Clearly in such a model, while it certainly would suffice, there may be no need to directly learn a full dynamical model. It may be feasible to satisfy this criterion without even observing intermediate states, but only seeing initial state and final outcome pairs  $\langle \vec{s}^0, F(\vec{s}^T) \rangle$ , closer to a traditional regression problem.

It is not difficult to define simple agent strategy classes for which learning from only  $\langle \vec{s}^0, F(\vec{s}^T) \rangle$  pairs is provably intractable, yet efficient learning is possible in our model. This idea is formalized in Theorem 3 below. Here the population forms a rather powerful computational device mapping initial states to final states. In particular, it can be thought of as a circuit of depth  $T$  with “gates” chosen from  $\mathcal{C}$ , with the only real constraint being that each layer of the circuit is an identical sequence of  $N$  gates which are applied to the outputs of the previous layer. Intuitively, if only initial states and final outcomes are provided to the learner, learning should be as difficult as a corresponding PAC-style problem. On the other hand, by observing intermediate state vectors we can build arbitrarily accurate models for each agent, which in turn allows us to accurately simulate the full dynamical model.

**Theorem 3** *Let  $\mathcal{C}$  be the class of 2-input AND and OR gates, and one-input NOT gates. Then  $\mathcal{C}$  is polynomially learnable from collective behavior, but there exists a binary outcome function  $F$  such that learning an accurate mapping from start states  $\vec{s}^0$  to outcomes  $F(\vec{s}^T)$  without observing intermediate state data is intractable.*

**Proof:** (Sketch) We first sketch the hardness construction. Let  $\mathcal{H}$  be any class of Boolean circuits (that is, with gates in  $\mathcal{C}$ ) that is not polynomially learnable in the standard PAC model; under standard cryptographic assumptions, such a class exists. Let  $D$  be a hard distribution for PAC learning  $\mathcal{H}$ . Let  $h \in \mathcal{H}$  be a Boolean circuit with  $R$  inputs,  $S$  gates, and depth  $D$ . To embed the computation by  $h$  in a collective problem, we let  $N = R + S$  and  $T = D$ . We introduce an agent for each of the  $R$  inputs to  $h$ , whose value after the initial state is set according to an arbitrary AND, OR, or NOT gate. We additionally introduce one agent for every gate  $g$  in  $h$ . If a gate  $g$  in  $h$  takes as its inputs the outputs of gates  $g'$  and  $g''$ , then at each time step the agent corresponding to  $g$  computes the corresponding function of the states of the agents corresponding to  $g'$  and  $g''$  at the previous time step. Finally, by convention we always have the  $N$ th agent be the agent corresponding to the output gate of  $h$ , and define the output function as  $F(\vec{s}) = s_N$ . The distribution  $P$  over initial states of the  $N$  agents is identical to  $D$  on the  $R$  agents corresponding to the inputs of  $h$ , and arbitrary (e.g., independent and uniform) on the remaining  $S$  agents.

Despite the fact that this construction introduces a great deal of spurious computation (for instance, at the first time step, many or most gates may simply be computing Boolean functions of the random bits assigned to non-input agents), it is clear that if gate  $g$  is at depth  $d$  in  $h$ , then at time  $d$  in the collective simulation of the agents, the corresponding agent has exactly the value computed by  $g$  under the inputs to  $h$  (which are distributed according to  $D$ ). Because the outcome function is the value of the agent corresponding to the output

gate of  $h$  at time  $T = D$ , pairs of the form  $\langle \vec{s}^0, F(\vec{s}^T) \rangle$  provide exactly the same data as the PAC model for  $h$  under  $D$ , and thus must be equally hard.

For the polynomial learnability of  $\mathcal{C}$  from collective behavior, we note that  $\mathcal{C}$  is clearly PAC learnable, since it is just Boolean combinations of 1 or 2 inputs. In Section 4 we give a general reduction from collective learning of any agent strategy class to PAC learning the class, thus giving the claimed result. ■

Conversely, it is also not difficult to concoct cases in which learning the full dynamics in our sense is intractable, but we can learn to approximate a specific outcome function from only  $\langle \vec{s}^0, F(\vec{s}^T) \rangle$  pairs. Intuitively, if each agent strategy is very complex but the outcome function applied to final states is sufficiently simple (e.g., constant), we cannot but do not need to model the full dynamics in order to learn to approximate the outcome.

We note that there is an analogy here to the distinction between *direct* and *indirect* approaches to reinforcement learning [16]. In the former, one learns a policy that is specific to a fixed reward function without learning a model of next-state dynamics; in the latter, at possibly greater cost, one learns an accurate dynamical model, which can in turn be used to compute good policies for any reward function. For the remainder of this paper, we focus on the model as we formalized it in Definition 2, and leave for future work the investigation of such alternatives.

### 3 Social Strategy Classes

Before providing our general theory, including the reduction from collective learning to I.I.D. learning, we first illustrate and motivate the definitions so far with some concrete examples of social strategy classes, some of which we analyze in detail in Section 5.

#### 3.1 Crowd Affinity: Mixture Strategies

The first class of agent strategies we discuss are meant to model settings in which each individual wishes to balance their intrinsic personal preferences with a desire to “follow the crowd.” We broadly refer to strategies of this type as *crowd affinity* strategies (in contrast to the *crowd aversion* strategies discussed shortly), and examine a couple of natural variants.

As a motivating example, imagine that there are  $K$  restaurants, and each week, every member of a population chooses one of the restaurants in which to dine. On the one hand, each agent has personal preferences over the restaurants based on the cuisine, service, ambiance, and so on. On the other, each agent has some desire to go to the currently “hot” restaurants — that is, where many or most other agents have been recently. To model this setting, let  $\mathcal{S}$  be the set of  $K$  restaurants, and suppose  $\vec{s} \in \mathcal{S}^N$  is the population state vector indicating where each agent dined last week. We can summarize the population behavior by the vector or distribution  $\vec{f} \in [0, 1]^K$ , where  $f_a$  is the fraction of agents dining in restaurant  $a$  in  $\vec{s}$ . Similarly, we might represent the personal preferences of a specific agent by another distribution  $\vec{w} \in [0, 1]^K$  in which  $w_a$  represents the probability this agent would attend restaurant  $a$  in the absence of any information

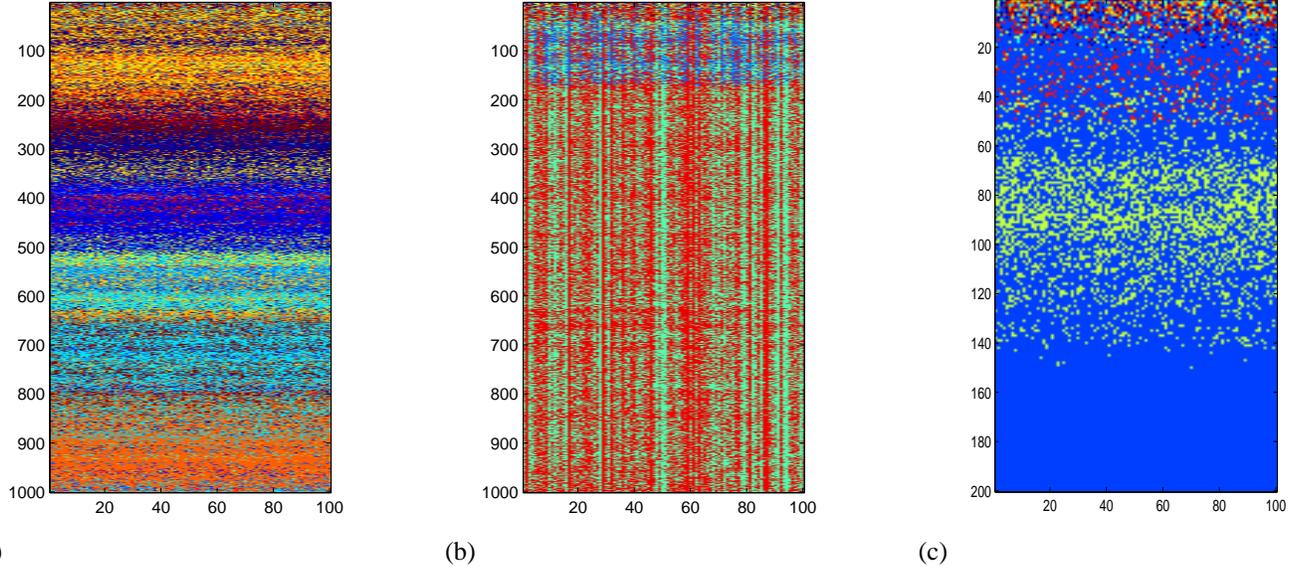


Figure 1: Sample simulations of the (a) crowd affinity mixture model; (b) crowd affinity multiplicative model; (c) agent affinity model. Horizontal axis is population state; vertical axis is simulation time. See text for details.

about what the population is doing. One natural way for the agent to balance their preferences with the population behavior would be to choose a restaurant according to the mixture distribution  $(1 - \alpha)\vec{f} + \alpha\vec{w}$  for some agent-dependent mixture coefficient  $\alpha$ . Such models have been studied in the sociology literature [12] in the context of belief formation.

We are interested in collective systems in which every agent  $i$  has some unknown preferences  $\vec{w}_i$  and mixture coefficient  $\alpha_i$ , and in each week  $t$  chooses its next restaurant according to  $(1 - \alpha_i)\vec{f}^t + \alpha_i\vec{w}_i$ , which thus probabilistically yields the next population distribution  $\vec{f}^{t+1}$ . How do such systems behave? And how can we learn to model their macroscopic properties from only observed behavior, especially when the number of choices  $K$  is large?

An illustration of the rich collective behavior that can already be generated from such simple strategies is shown in Figure 1(a). Here we show a single but typical 1000-step simulation of collective behavior under this model, in which  $N = 100$  and each agent’s individual preference vector  $\vec{w}$  puts all of its weight on just one of 10 possible actions (represented as colors); this action was selected independently at random for each agent. All agents have an  $\alpha$  value of just 0.01, and thus are selecting from the population distribution 99% of the time. Each row shows the population state at a given step, with time increasing down the horizontal axis of the image. The initial state was chosen uniformly at random.

It is interesting to note the dramatic difference between  $\alpha = 0$  (in which rapid convergence to a common color is certain) and this small value for  $\alpha$ ; despite the fact that almost all agents play the population distribution at every step, revolving horizontal waves of near-consensus to different choices are present, with no final convergence in sight. The slight “personalization” of population-only behavior is enough to dramatically change the collective be-

havior. Broadly speaking, it is such properties we would like a learning algorithm to model and predict from sufficient observations.

### 3.2 Crowd Affinity: Multiplicative Strategies

One possible objection to the crowd affinity mixture strategies described above is that each agent can be viewed as *randomly* choosing whether to *entirely* follow the population distribution (with probability  $1 - \alpha$ ) or to *entirely* follow their personal preferences (with probability  $\alpha$ ) at each time step. A more realistic model might have each agent truly *combine* the population behavior with their preferences at every step.

Consider, for instance, how an American citizen might alter their anticipated presidential voting decision over time in response to recent primary or polling news. If their first choice of candidate — say, an Independent or Libertarian candidate — appears over time to be “unelectable” in the general election due to their inability to sway large numbers of Democratic and Republican voters, a natural and typical response is for the citizen to shift their intended vote to whichever of the front-runners they most prefer or least dislike. In other words, the low popularity of their first choice causes that choice to be dampened or eradicated; unlike the mixture model above, where weight  $\alpha$  is always given to personal preferences, here there may remain *no* weight on this candidate.

One natural way of defining a general such class of strategies is as follows. As above, let  $\vec{f} \in [0, 1]^K$ , where  $f_a$  is the fraction of agents dining in restaurant  $a$  in the current state  $\vec{s}$ . Similar to the mixture strategies above, let  $\vec{w}_i \in [0, 1]^K$  be a vector of *weights* representing the intrinsic preferences of agent  $i$  over actions. Then define the probability that agent  $i$  plays action  $a$  to be  $f_a \cdot w_{i,a} / Z(\vec{f}, \vec{w}_i)$ , where the normalizing factor is  $Z(\vec{f}, \vec{w}_i) = \sum_{b \in S} f_b \cdot w_{i,b}$ .

Thus, in such *multiplicative* crowd affinity models, the probability the agent takes an action is always proportional to the product of their preference for it and its current popularity.

Despite their similar motivation, the mixture and multiplicative crowd affinity strategies can lead to dramatically different collective behavior. Perhaps the most obvious difference is that in the mixture case, if agent  $i$  has a strong preference for action  $a$  there is *always* some minimum probability ( $\alpha_i w_{i,a}$ ) they take this action, whereas in the multiplicative case even a strong preference can be eradicated from expression by small or zero values for the popularity  $f_a$ .

In Figure 1(b), we again show a single but typical 1000-step,  $N = 100$  simulation for the multiplicative model in which agent’s individual preference distributions  $\vec{w}$  are chosen to be random normalized vectors over 10 actions. The dynamics are now quite different than for the additive crowd affinity model. In particular, now there is never near-consensus but a gradual dwindling of the colors represented in the population — from the initial full diversity down to 3 colors remaining at approximately  $t = 100$ , until by  $t = 200$  there is a stand-off in the population between red and light green. Unlike the additive models, colors die out in the population permanently. There is also clear vertical structure corresponding to strong conditional preferences of the agents once the stand-off emerges.

### 3.3 Crowd Aversion and Other Variants

It is easy to transform the mixture or multiplicative crowd affinity strategies into *crowd aversion* strategies — that is, in which agents wish to balance or combine their personal preferences with a desire to act *differently* than the population at large. This can be accomplished in a variety of simple ways. For instance, if  $\vec{f}$  is the current distributions over actions in the population, we can simply define a kind of “inverse” to the distribution by letting  $g_a = (1 - f_a)/(K - 1)$ , where  $K - 1 = \sum_{b \in \mathcal{S}} (1 - f_b)$  is the normalizing factor, and applying the strategies above to  $\vec{g}$  rather than  $\vec{f}$ . Now each agent exhibits a tendency to “avoid the crowd”, moderated as before by their own preferences.

Of course, there is no reason to assume that the entire population is crowd-seeking, or crowd-avoiding; more generally we would allow there to be both types of individuals present. Furthermore, we might entertain other transforms of the population distribution than just  $g_a$  above. For instance, we might wish to still consider crowd affinity, but to first “sharpen” the distribution by replacing each  $f_a$  with  $f_a^2$  and normalizing, then applying the models discussed above to the resulting vector. This has the effect of magnifying the attraction to the most popular actions. In general our algorithmic results are robust to a wide range of such variations.

### 3.4 Agent Affinity and Aversion Strategies

In the two versions of crowd affinity strategies discussed above, an agent has personal preferences over actions, and also reacts to the current population behavior, but only in an aggregate fashion. An alternative class of strategies that we call *agent affinity* strategies instead allows agents to prefer to agree (or disagree) in their choice with specific other agents.

For a fixed agent, such a strategy can be modeled by a weight vector  $\vec{w} \in [0, 1]^N$ , with one weight for each *agent* in the population rather than each action. We define the probability that this agent takes action  $a$  if the current global state is  $\vec{s} \in \mathcal{S}^N$  to be proportional to  $\sum_{i: s_i = a} w_i$ . In this class of strategies, the strength of the agent’s desire to take the same action as agent  $i$  is determined by how large the weight  $w_i$  is. The overall behavior of this agent is then probabilistically determined by summing over all agents in the fashion above.

In Figure 1(c), we show a single but typical simulation, again with  $N = 100$  but now with a much shorter time horizon of 200 steps and a much larger set of 100 actions. All agents have random distributions as their preferences over other agents; this model is similar to traditional diffusion dynamics in a dense, random (weighted) network, and quickly converges to global consensus.

We leave the analysis of this strategy class to future work, but remark that in the simple case in which  $K = 2$ , learning this class is closely related to the problem of learning perceptrons under certain noise models in which the intensity of the noise increases with proximity to the separator [5, 4] and seems at least as difficult.

### 3.5 Incorporating Network Structure

Many of the social models inspiring this work involve a network structure that dictates or restricts the interactions between agents [18]. It is natural to ask if the strategy classes discussed here can be extended to the scenario in which each agent is influenced only by his neighbors in a given network. Indeed, it is straightforward to extend each of the strategy classes introduced in this section to a network setting. For example, to adapt the crowd affinity and aversion strategy classes, it suffices to redefine  $f_a$  for each agent  $i$  to be the fraction of agents in the local neighborhood of agent  $i$  choosing action  $a$ . To adapt the agent affinity and aversion classes, it is necessary only to require that  $w_j = 0$  for every agent  $j$  outside the local neighborhood of agent  $i$ . By making these simple modifications, the learning algorithms discussed in Section 5 can immediately be applied to settings in which a network structure is given.

## 4 A Reduction to I.I.D. Learning

Since algorithms in our framework are attempting to learn to model the dynamics of a factored Markov process in which each component is known to lie in the class  $\mathcal{C}$ , it is natural to investigate the relationship between learning just a single strategy in  $\mathcal{C}$  and the entire Markovian dynamics. One main concern might be effects of dynamic instability — that is, that even small errors in models for each of the  $N$  components could be amplified exponentially in the overall population model.

In this section we show that this can be avoided. More precisely, we prove that if the component errors are all small compared to  $1/(NT)^2$ , the population model also has small error. Thus fast rates of learning for individual components are polynomially preserved in the resulting population model.

To show this, we give a reduction showing that if a class  $\mathcal{C}$  of (possibly probabilistic) strategies is polynomially learnable (in a sense that we describe shortly) from I.I.D. data,

then  $\mathcal{C}$  is also polynomially learnable from collective behavior. The key step in the reduction is the introduction of the experimental distribution, defined below. Intuitively, the experimental distribution is meant to capture the distribution over states that are encountered in the collective setting over repeated trials. Polynomial I.I.D. learning on this distribution leads to polynomial learning from the collective.

#### 4.1 A Reduction for Deterministic Strategies

In order to illustrate some of the key ideas we use in the more general reduction, we begin by examining the simple case in which the number of actions  $K = 2$  and each strategy  $c \in \mathcal{C}$  is deterministic. We show that if  $\mathcal{C}$  is polynomially learnable in the (distribution-free) PAC model, then  $\mathcal{C}$  is polynomially learnable from collective behavior.

In order to exploit the fact that  $\mathcal{C}$  is PAC learnable, it is first necessary to define a single distribution over states on which we would like to learn.

**Definition 4** For any initial state distribution  $P$ , strategy vector  $\vec{c}$ , and sequence length  $T$ , the **experimental distribution**  $D_{P, \vec{c}, T}$  is the distribution over state vectors  $\vec{s}$  obtained by querying  $\mathcal{O}_{\text{EXP}}(\vec{c}, P, T)$  to obtain  $\langle \vec{s}^0, \dots, \vec{s}^T \rangle$ , choosing  $t$  uniformly at random from  $\{0, \dots, T-1\}$ , and setting  $\vec{s} = \vec{s}^t$ .

We denote this distribution simply as  $D$  when  $P$ ,  $\vec{c}$ , and  $T$  are clear from context. Given access to the oracle  $\mathcal{O}_{\text{EXP}}$ , we can sample pairs  $\langle \vec{s}, c_i(\vec{s}) \rangle$  where  $\vec{s}$  is distributed according to  $D$  using the following procedure:

1. Query  $\mathcal{O}_{\text{EXP}}(\vec{c}, P, T)$  to obtain  $\langle \vec{s}^0, \dots, \vec{s}^T \rangle$ .
2. Choose  $t \in \{0, \dots, T-1\}$  uniformly at random.
3. Return  $\langle \vec{s}^t, s_i^{t+1} \rangle$ .

If  $\mathcal{C}$  is polynomially learnable in the PAC model, then by definition, with access to the oracle  $\mathcal{O}_{\text{EXP}}$ , for any  $\delta, \epsilon > 0$ , it is possible to learn a model  $\hat{c}_i$  such that with probability  $1 - (\delta/N)$ ,

$$\Pr_{\vec{s} \sim D}[\hat{c}_i(\vec{s}) \neq c_i(\vec{s})] \leq \frac{\epsilon}{NT}$$

in time polynomial in  $N, T, 1/\epsilon, 1/\delta$ , and the VC dimension of  $\mathcal{C}$  using the sampling procedure above; the dependence on  $N$  and  $T$  come from the fact that we are requesting a confidence of  $1 - (\delta/N)$  and an accuracy of  $\epsilon/(TN)$ . We can learn a set of such strategies  $\hat{c}_i$  for all agents  $i$  at the cost of an additional factor of  $N$ .

Consider a new sequence  $\langle \vec{s}^0, \dots, \vec{s}^T \rangle$  returned by the oracle  $\mathcal{O}_{\text{EXP}}$ . By the union bound, with probability  $1 - \delta$ , the probability that there exists any agent  $i$  and any  $t \in \{0, \dots, T-1\}$ , such that  $\hat{c}_i(\vec{s}^t) \neq c_i(\vec{s}^t)$  is less than  $\epsilon$ . If this is not the case (i.e., if  $\hat{c}_i(\vec{s}^t) = c_i(\vec{s}^t)$  for all  $i$  and  $t$ ) then the same sequence of states would have been reached if we had instead started at state  $\vec{s}^0$  and generated each additional state  $\vec{s}^t$  by letting  $s_i^t = c_i(\vec{s}^{t-1})$ . This implies that with probability  $1 - \delta$ ,  $\varepsilon(Q_{\vec{M}}, Q_{\vec{c}}) \leq \epsilon$ , and  $\mathcal{C}$  is polynomially learnable from collective behavior.

#### 4.2 A General Reduction

Multiple analogs of the definition of learnability in the PAC model have been proposed for distribution learning settings. The probabilistic concept model [15] presents a definition for learning conditional distributions over binary outcomes, while later work [13] proposes a definition for learning unconditional distributions over larger outcome spaces. We combine the two into a single PAC-style model for learning conditional distributions over large outcome spaces from I.I.D. data as follows.

**Definition 5** Let  $\mathcal{C}$  be a class of probabilistic mappings from an input  $\vec{x} \in \mathcal{X}$  to an output  $y \in \mathcal{Y}$  where  $\mathcal{Y}$  is a finite set. We say that  $\mathcal{C}$  is **polynomially learnable** if there exists an algorithm  $A$  such that for any  $c \in \mathcal{C}$  and any distribution  $D$  over  $\mathcal{X}$ , if  $A$  is given access to an oracle producing pairs  $\langle \vec{x}, c(\vec{x}) \rangle$  with  $x$  distributed according to  $D$ , then for any  $\epsilon, \delta > 0$ , algorithm  $A$  runs in time polynomial in  $1/\epsilon, 1/\delta$ , and  $\dim(\mathcal{C})$  and outputs a function  $\hat{c}$  such that with probability  $1 - \delta$ ,

$$\mathbb{E}_{\vec{x} \sim D} \left[ \sum_{y \in \mathcal{Y}} |\Pr(c(\vec{x}) = y) - \Pr(\hat{c}(\vec{x}) = y)| \right] \leq \epsilon.$$

We could have chosen instead to require that the expected KL divergence between  $c$  and  $\hat{c}$  be bounded. Using Jensen's inequality and Lemma 12.6.1 of Cover and Thomas [6], it is simple to show that if the expected KL divergence between two distributions is bounded by  $\epsilon$ , then the expected  $\mathcal{L}_1$  distance is bounded by  $\sqrt{2 \ln(2)} \epsilon$ . Thus any class that is polynomially learnable under this alternate definition is also polynomially learnable under ours.

**Theorem 6** For any class  $\mathcal{C}$ , if  $\mathcal{C}$  is polynomially learnable according to Definition 5, then  $\mathcal{C}$  is polynomially learnable from collective behavior.

**Proof:** This proof is very similar in spirit to the proof of the reduction for the deterministic case. However, several tricks are needed to deal with the fact that trajectories are now random variables, even given a fixed start state. In particular, it is no longer the case that we can argue that starting at a given start state and executing a set of strategies that are “close to” the true strategy vector usually yields *the same* full trajectory we would have obtained by executing the true strategies of each agent. Instead, due to the inherent randomness in the strategies, we must argue that the *distribution* over trajectories is similar when the estimated strategies are sufficiently close to the true strategies.

To make this argument, we begin by introducing the idea of sampling from a distribution  $P_1$  using a “filtered” version of a second distribution  $P_2$  as follows. First, draw an outcome  $\omega \in \Omega$  according to  $P_2$ . If  $P_1(\omega) \geq P_2(\omega)$ , output  $\omega$ . Otherwise, output  $\omega$  with probability  $P_1(\omega)/P_2(\omega)$ , and with probability  $1 - P_1(\omega)/P_2(\omega)$ , output an alternate action drawn according to a third distribution  $P_3$ , where

$$P_3(\omega) = \frac{P_1(\omega) - P_2(\omega)}{\sum_{\omega': P_2(\omega') < P_1(\omega')} P_1(\omega') - P_2(\omega')}$$

if  $P_1(\omega) > P_2(\omega)$ , and  $P_3(\omega) = 0$  otherwise.

It is easy to verify that the output of this filtering algorithm is indeed distributed according to  $P_1$ . Additionally, notice that the probability that the output is “filtered” is

$$\sum_{\omega: P_2(\omega) > P_1(\omega)} P_2(\omega) \left(1 - \frac{P_1(\omega)}{P_2(\omega)}\right) = \frac{1}{2} \|P_2 - P_1\|_1. \quad (1)$$

As in the deterministic case, we make use of the experimental distribution  $D$  as defined in Definition 4. If  $\mathcal{C}$  is polynomially learnable as in Definition 5, then with access to the oracle  $\mathcal{O}_{\text{EXP}}$ , for any  $\delta, \epsilon > 0$ , it is possible to learn a model  $\hat{c}_i$  such that with probability  $1 - (\delta/N)$ ,

$$\mathbb{E}_{\vec{s} \sim D} \left[ \sum_{s \in \mathcal{S}} |\Pr(c_i(\vec{s}) = s) - \Pr(\hat{c}_i(\vec{s}) = s)| \right] \leq \left(\frac{\epsilon}{NT}\right)^2 \quad (2)$$

in time polynomial in  $N, T, 1/\epsilon, 1/\delta$ , and  $\dim(\mathcal{C})$  using the three-step sampling procedure described in the deterministic case; as before, the dependence on  $N$  and  $T$  stem from the fact that we are requesting a confidence of  $1 - (\delta/N)$  and an accuracy that is polynomial in both  $N$  and  $T$ . It is possible to learn a set of such strategies  $\hat{c}_i$  for all agents  $i$  at the cost of an additional factor of  $N$ .

If Equation 2 is satisfied for agent  $i$ , then for any  $\tau \geq 1$ , the probability of drawing a state  $\vec{s}$  from  $D$  such that

$$\sum_{s \in \mathcal{S}} |\Pr(c_i(\vec{s}) = s) - \Pr(\hat{c}_i(\vec{s}) = s)| \geq \tau \left(\frac{\epsilon}{NT}\right)^2 \quad (3)$$

is no more than  $1/\tau$ .

Consider a new sequence  $\langle \vec{s}^0, \dots, \vec{s}^T \rangle$  returned by the oracle  $\mathcal{O}_{\text{EXP}}$ . For each  $\vec{s}^t$ , consider the action  $s_i^{t+1}$  chosen by agent  $i$ . This action was chosen according to the distribution  $c_i$ . Suppose instead we would like to choose this action according to the distribution  $\hat{c}_i$  using a filtered version of  $c_i$  as described above. By Equation 1, the probability that the action choice of  $c_i$  is “filtered” (and thus not equal to  $s_i^{t+1}$ ) is half the  $\mathcal{L}_1$  distance between  $c_i(\vec{s}^t)$  and  $\hat{c}_i(\vec{s}^t)$ . From Equation 3, we know that for any  $\tau \geq 1$ , with probability at least  $1 - 1/\tau$ , this probability is less than  $\tau(\epsilon/(NT))^2$ , so the probability of the new action being different from  $s_i^{t+1}$  is less than  $\tau(\epsilon/(NT))^2 + 1/\tau$ . This is minimized when  $\tau = 2NT/\epsilon$ , giving us a bound of  $\epsilon/(NT)$ .

By the union bound, with probability  $1 - \delta$ , the probability that there exists any agent  $i$  and any  $t \in \{1, \dots, T\}$ , such that  $s_i^{t+1}$  is not equal to the action we get by sampling  $\hat{c}_i(\vec{s}^t)$  using the filtered version of  $c_i$  must then be less than  $\epsilon$ . As in the deterministic version, if this is *not* the case, then the same sequence of states would have been reached if we had instead started at state  $\vec{s}^0$  and generated each additional state  $\vec{s}^t$  by letting  $s_i^t = \hat{c}_i(\vec{s}^{t-1})$  filtered using  $c_i$ . This implies that with probability  $1 - \delta$ ,  $\varepsilon(Q_{\vec{M}}, Q_{\vec{e}}) \leq \epsilon$ , and  $\mathcal{C}$  is polynomially learnable from collective behavior. ■

## 5 Learning Social Strategy Classes

We now turn our attention to efficient algorithms for learning some of the specific social strategy classes introduced in Section 3. We focus on the two crowd affinity model classes. Recall that these classes are designed to model the scenario

in which each agent has an intrinsic set of preferences over actions, but simultaneously would prefer to choose the same actions chosen by other agents. Similar techniques can be applied to learn the crowd aversion strategies.

Formally, let  $\vec{f}$  be a vector representing the distribution over current states of the agents; if  $\vec{s}$  is the current state, then for each action  $a$ ,  $f_a = |\{i : s_i = a\}|/N$  is the fraction of the population currently choosing action  $a$ . (Alternately, if there is a network structure governing interaction among agents,  $f_a$  can be defined as the fraction of nodes in an agent’s local neighborhood choosing action  $a$ .) We denote by  $D^f$  the distribution over vectors  $\vec{f}$  induced by the experimental distribution  $D$  over state vectors  $\vec{s}$ . In other words, the probability of a vector  $\vec{f}$  under  $D^f$  is the sum over all state vectors  $\vec{s}$  mapping to  $\vec{f}$  of the probability of  $\vec{s}$  under  $D$ .

We focus on the problem of learning the parameters of the strategy of a single agent  $i$  in each of the models. We assume that we are presented with a set of samples  $\mathcal{M}$ , where each instance  $\mathcal{I}_m \in \mathcal{M}$  consists of a pair  $\langle \vec{f}_m, a_m \rangle$ . Here  $\vec{f}_m$  is the distribution over states of the agents and  $a_m$  is the next action chosen by agent  $i$ . We assume that the state distributions  $\vec{f}_m$  of these samples are distributed according to  $D^f$ . Given access to the oracle  $\mathcal{O}_{\text{EXP}}$ , such samples could be collected, for example, using a three-step procedure like the one in Section 4.1. We show that each class is polynomially learnable with respect to the distribution  $D^f$  induced by *any* distribution  $D$  over states, and so by Theorem 6, also polynomially learnable from collective behavior.

While it may seem wasteful to gather only one data instance for each agent  $i$  from each  $T$ -trajectory, we remark that only small, isolated pieces of the analysis presented in this section rely on the assumption that the state distributions of the samples are distributed according to  $D^f$ . In practice, the entire trajectories could be used for learning with no impact on the structure of the algorithms. Additionally, while the analysis here is geared towards learning under the experimental distribution, the algorithms we present can be applied without modification in the no-reset variant of the model introduced in Section 2.3. We briefly discuss how to extend the analysis to the no-reset variant in Section 5.3.

### 5.1 Learning Crowd Affinity Mixture Models

In Section 3.1, we introduced the class of crowd affinity mixture model strategies. Such strategies are parameterized by a (normalized) weight vector  $\vec{w}$  and parameter  $\alpha \in [0, 1]$ . The probability that agent  $i$  chooses action  $a$  given that the current state distribution is  $\vec{f}$  is then  $\alpha f_a + (1 - \alpha)w_a$ . In this section, we show that this class of strategies is polynomially learnable from collective behavior and sketch an algorithm for learning estimates of the parameters  $\alpha$  and  $\vec{w}$ .

Let  $I(x)$  be the indicator function that is 1 if  $x$  is true and 0 otherwise. From the definition of the model it is easy to see that for any  $m$  such that  $\mathcal{I}_m \in \mathcal{M}$ , for any action  $a \in \mathcal{S}$ ,  $\mathbb{E}[I(a_m = a)] = \alpha f_a + (1 - \alpha)w_a$ , where the expectation is over the randomness in the agent’s strategy. By linearity of expectation,

$$\mathbb{E} \left[ \sum_{m: \mathcal{I}_m \in \mathcal{M}} I(a_m = a) \right] = \alpha \sum_{m: \mathcal{I}_m \in \mathcal{M}} f_{m,a} + (1 - \alpha)w_a |\mathcal{M}|. \quad (4)$$

Standard results from uniform convergence theory say that we can approximate the left-hand side of this equation arbitrarily well given a sufficiently large data set  $\mathcal{M}$ . Replacing the expectation with this approximation in Equation 4 yields a single equation with two unknown variables,  $\alpha$  and  $w_a$ . To solve for these variables, we must construct a *pair* of equations with two unknown variables. We do so by splitting the data into instances where  $f_{m,a}$  is “high” and instances where it is “low.”

Specifically, let  $M = |\mathcal{M}|$ . For convenience of notation, assume without loss of generality that  $M$  is even; if  $M$  is odd, simply discard an instance at random. Define  $\mathcal{M}_a^{low}$  to be the set containing the  $M/2$  instances in  $\mathcal{M}$  with the lowest values of  $f_{m,a}$ . Similarly, define  $\mathcal{M}_a^{high}$  to be the set containing the  $M/2$  instances with the highest values of  $f_{m,a}$ . Replacing  $\mathcal{M}$  with  $\mathcal{M}_a^{low}$  and  $\mathcal{M}_a^{high}$  respectively in Equation 4 gives us two linear equations with two unknowns. As long as these two equations are linearly independent, we can solve the system of equations for  $\alpha$ , giving us

$$\alpha = \frac{\mathbb{E} \left[ \sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{high}} \mathbb{I}(a_m = a) - \sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{low}} \mathbb{I}(a_m = a) \right]}{\sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{high}} f_{m,a} - \sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{low}} f_{m,a}}.$$

We can approximate  $\alpha$  from data in the natural way, using

$$\hat{\alpha} = \frac{\sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{high}} \mathbb{I}(a_m = a) - \sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{low}} \mathbb{I}(a_m = a)}{\sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{high}} f_{m,a} - \sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{low}} f_{m,a}}. \quad (5)$$

By Hoeffding’s inequality and the union bound, for any  $\delta > 0$ , with probability  $1 - \delta$ ,

$$\begin{aligned} |\alpha - \hat{\alpha}| &\leq \frac{\sqrt{\ln(4/\delta)M}}{\sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{high}} f_{m,a} - \sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{low}} f_{m,a}} \\ &= (1/Z_a) \sqrt{\ln(4/\delta)/M}, \end{aligned} \quad (6)$$

where

$$Z_a = \frac{1}{M/2} \sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{high}} f_{m,a} - \frac{1}{M/2} \sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{low}} f_{m,a}.$$

The quantity  $Z_a$  measures the difference between the mean value of  $f_{m,a}$  among instances with “high” values of  $f_{m,a}$  and the mean value of  $f_{m,a}$  among instances with “low” values. While this quantity is data-dependent, standard uniform convergence theory tells us that it is stable once the data set is large. From Equation 6, we know that if there is an action  $a$  for which this difference is sufficiently high, then it is possible to obtain an accurate estimate of  $\alpha$  given enough data. If, on the other hand, no such  $a$  exists, it follows that there is very little variance in the population distribution over the sample. We argue below that it is not necessary to learn  $\alpha$  in order to mimic the behavior of an agent  $i$  if this is the case.

For now, assume that  $Z_a$  is sufficiently large for at least one value of  $a$ , and call this value  $a^*$ . We can use the estimate of  $\alpha$  to obtain estimates of the weights for each action. From Equation 4, it is clear that for any  $a$ ,

$$w_a = \frac{\mathbb{E} \left[ \sum_{m: \mathcal{I}_m \in \mathcal{M}} \mathbb{I}(a_m = a) \right] - \alpha \sum_{m: \mathcal{I}_m \in \mathcal{M}} f_{m,a}}{(1 - \hat{\alpha})M}.$$

We estimate this weight using

$$\hat{w}_a = \frac{\sum_{m: \mathcal{I}_m \in \mathcal{M}} \mathbb{I}(a_m = a) - \hat{\alpha} \sum_{m: \mathcal{I}_m \in \mathcal{M}} f_{m,a}}{(1 - \hat{\alpha})M}. \quad (7)$$

The following lemma shows that given sufficient data, the error in these estimates is small when  $Z_{a^*}$  is large.

**Lemma 7** *Let  $a^* = \operatorname{argmax}_{a \in \mathcal{S}} Z_a$ , and let  $\hat{\alpha}$  be calculated as in Equation 5 with  $a = a^*$ . For each  $a \in \mathcal{S}$ , let  $\hat{w}_a$  be calculated as in Equation 7. For sufficiently large  $M$ , for any  $\delta > 0$ , with probability  $1 - \delta$ ,*

$$|\alpha - \hat{\alpha}| \leq (1/Z_{a^*}) \sqrt{\ln((4 + 2K)/\delta)/M},$$

and for all actions  $a$ ,

$$\begin{aligned} |w_a - \hat{w}_a| &\leq \frac{((1 - \hat{\alpha})Z_{a^*}/\sqrt{2} + 2) \sqrt{\ln((4 + 2K)/\delta)}}{Z_{a^*}(1 - \hat{\alpha})^2 \sqrt{M} - (1 - \hat{\alpha}) \sqrt{\ln((4 + 2K)/\delta)}}. \end{aligned}$$

The proof of this lemma, which is in the appendix,<sup>1</sup> relies heavily on the following technical lemma for bounding the error of estimated ratios, which is used frequently throughout the remainder of the paper.

**Lemma 8** *For any positive  $u, \hat{u}, v, \hat{v}, k$ , and  $\epsilon$  such that  $\epsilon k < v$ , if  $|u - \hat{u}| \leq \epsilon$  and  $|v - \hat{v}| \leq \epsilon k$ , then*

$$\left| \frac{u}{v} - \frac{\hat{u}}{\hat{v}} \right| \leq \frac{\epsilon(v + uk)}{v(v - \epsilon k)}.$$

Now that we have bounds on the error of the estimated parameters, we can bound the expected  $\mathcal{L}_1$  distance between the estimated model and the real model.

**Lemma 9** *For sufficiently large  $M$ ,*

$$\begin{aligned} \mathbb{E}_{\vec{f} \sim D^f} \sum_{a \in \mathcal{S}} |(\alpha f_a + (1 - \alpha)w_a) - (\hat{\alpha} f_a + (1 - \hat{\alpha})\hat{w}_a)| &\leq \frac{2\sqrt{\ln((4 + 2K)/\delta)}}{Z_{a^*} \sqrt{M}} \\ &+ \min \left\{ \frac{K(Z_{a^*}/\sqrt{2} + 2) \sqrt{\ln((4 + 2K)/\delta)}}{Z_{a^*}(1 - \hat{\alpha}) \sqrt{M} - \sqrt{\ln((4 + 2K)/\delta)}}, \right. \\ &\left. 2(1 - \hat{\alpha}) \right\}. \end{aligned}$$

In this proof of this lemma, which appears in the appendix, the quantity

$$\sum_{a \in \mathcal{S}} |(\alpha f_a + (1 - \alpha)w_a) - (\hat{\alpha} f_a + (1 - \hat{\alpha})\hat{w}_a)|$$

is bounded *uniformly* for all  $\vec{f}$  using the error bounds. The bound on the expectation follows immediately.

It remains to show that we can still bound the error when  $Z_{a^*}$  is zero or very close to zero. We present a light sketch of the argument here; more details appear in the appendix.

<sup>1</sup>An appendix containing omitted proofs can be found in the long version of this paper available on the authors’ websites.

Let  $\eta_a$  and  $\mu_a$  be the true median and mean of the distribution from which the random variables  $f_{m,a}$  are drawn. Let  $f_a^{high}$  be the mean value of the distribution over  $f_{m,a}$  conditioned on  $f_{m,a} > \eta_a$ . Let  $\bar{f}_a^{high}$  be the empirical average of  $f_{m,a}$  conditioned on  $f_{m,a} > \eta_a$ . Finally, let  $\hat{f}_a^{high} = (2/M) \sum_{m: \mathcal{I}_m \in \mathcal{M}_a^{high}} f_{m,a}$  be the empirical average of  $f_{m,a}$  conditioned on  $f_{m,a}$  being greater than the empirical median. We can calculate  $\hat{f}_a^{high}$  from data.

We can apply standard arguments from uniform convergence theory to show that  $f_a^{high}$  is close to  $\bar{f}_a^{high}$ , and in turn that  $\bar{f}_a^{high}$  is close to  $\hat{f}_a^{high}$ . Similar statements can be made for the analogous quantities  $f_a^{low}$ ,  $\bar{f}_a^{low}$ , and  $\hat{f}_a^{low}$ . By noting that  $Z_a = \hat{f}_a^{high} - \hat{f}_a^{low}$  this implies that if  $Z_a$  is small, then the probability that a random value of  $f_{m,a}$  is far from the mean  $\mu_a$  is small. When this is the case, it is not necessary to estimate  $\alpha$  directly. Instead, we set  $\hat{\alpha} = 0$  and

$$\hat{w}_a = \frac{1}{M} \sum_{m: \mathcal{I}_m \in \mathcal{M}} I(a_m = a).$$

Applying Hoeffding's inequality again, it is easy to show that for each  $a$ ,  $\hat{w}_a$  is very close to  $\alpha\mu_a + (1-\alpha)w_a$ , and from here it can be argued that the  $\mathcal{L}_1$  distance between the estimated model and the real model is small.

Thus for any distribution  $D$  over state vectors, regardless of the corresponding value of  $Z_{a^*}$ , it is possible to build an accurate model for the strategy of agent  $i$  in polynomial time. By Theorem 6, this implies that the class is polynomially learnable from collective behavior.

**Theorem 10** *The class of crowd affinity mixture model strategies is polynomially learnable from collective behavior.*

## 5.2 Learning Crowd Affinity Multiplicative Models

In Section 3.2, we introduced the crowd affinity multiplicative model. In this model, strategies are parameterized only by a weight vector  $\vec{w}$ . The probability that agent  $i$  chooses action  $a$  is simply  $f_a w_a / \sum_{b \in \mathcal{S}} f_b w_b$ .

Although the motivation for this model is similar to that for the mixture model, the dynamics of the system are quite different (see the simulations and discussion in Section 3), and a very different algorithm is necessary to learn individual strategies. In this section, we show that this class is polynomially learnable from collective behavior, and sketch the corresponding learning algorithm. The algorithm we present is based on a simple but powerful observation. In particular, consider the following random variable:

$$\chi_a^m = \begin{cases} 1/f_{m,a} & \text{if } f_{m,a} > 0 \text{ and } a_m = a, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose that for all  $m$  such that  $\mathcal{I}_m \in \mathcal{M}$ , it is the case that  $f_{m,a} > 0$ . Then by the definition of the strategy class and linearity of expectation,

$$\begin{aligned} \mathbb{E} \left[ \sum_{m: \mathcal{I}_m \in \mathcal{M}} \chi_a^m \right] &= \sum_{m: \mathcal{I}_m \in \mathcal{M}} \frac{1}{f_{m,a}} \left( \frac{f_{m,a} w_a}{\sum_{s \in \mathcal{S}} f_{m,s} w_s} \right) \\ &= w_a \sum_{m: \mathcal{I}_m \in \mathcal{M}} \frac{1}{\sum_{s \in \mathcal{S}} f_{m,s} w_s}, \end{aligned}$$

where the expectation is over the randomness in the agent's strategy. Notice that this expression is the product of two terms. The first,  $w_a$ , is precisely the value we would like to calculate. The second term is something that depends on the set of instances  $\mathcal{M}$ , but *does not* depend on action  $a$ . This leads to the key observation at the core of our algorithm. Specifically, if we have a second action  $b$  such that  $f_{m,b} > 0$  for all  $m$  such that  $\mathcal{I}_m \in \mathcal{M}$ , then

$$\frac{w_a}{w_b} = \frac{\mathbb{E} \left[ \sum_{m: \mathcal{I}_m \in \mathcal{M}} \chi_a^m \right]}{\mathbb{E} \left[ \sum_{m: \mathcal{I}_m \in \mathcal{M}} \chi_b^m \right]}.$$

Although we do not know the values of these expectations, we can approximate them arbitrarily well given enough data. Since we have assumed (so far) that  $f_{m,a} > 0$  for all  $m \in \mathcal{M}$ , and we know that  $f_{m,a}$  represents a fraction of the population, it must be the case that  $f_{m,a} \geq 1/N$  and  $\chi_a^m \in [0, N]$  for all  $m$ . By a standard application of Hoeffding's inequality and the union bound, we see that for any  $\delta > 0$ , with probability  $1 - \delta$ ,

$$\left| \sum_{m: \mathcal{I}_m \in \mathcal{M}} \chi_a^m - \mathbb{E} \left[ \sum_{m: \mathcal{I}_m \in \mathcal{M}} \chi_a^m \right] \right| \leq \sqrt{\frac{N \ln(2/\delta)}{2|\mathcal{M}|}}. \quad (8)$$

This leads to the following lemma. We note that the role of  $\beta$  in this lemma may appear somewhat mysterious. It comes from the fact that we are bounding the error of a ratio of two terms; an application of Lemma 8 using the bound in Equation 8 gives us a factor of  $\chi_{a,b} + \chi_{b,a}$  in the numerator and a factor of  $\chi_{b,a}$  in the denominator. This is problematic only when  $\chi_{a,b}$  is significantly larger than  $\chi_{b,a}$ . The full proof appears in the appendix.

**Lemma 11** *Suppose that  $f_{m,a} > 0$  and  $f_{m,b} > 0$  for all  $m$  such that  $\mathcal{I}_m \in \mathcal{M}$ . Then for any  $\delta > 0$ , with probability  $1 - \delta$ , for any  $\beta > 0$ , if  $\chi_{a,b} \leq \beta \chi_{b,a}$  and  $\chi_{b,a} \geq 1$ , then if  $|\mathcal{M}| \geq N \ln(2/\delta)/2$ , then*

$$\left| \frac{w_a}{w_b} - \frac{\sum_{m: \mathcal{I}_m \in \mathcal{M}} \chi_a^m}{\sum_{m: \mathcal{I}_m \in \mathcal{M}} \chi_b^m} \right| \leq \frac{(1 + \beta) \sqrt{N \ln(2/\delta)}}{\sqrt{2|\mathcal{M}|} - \sqrt{N \ln(2/\delta)}}.$$

If we are fortunate enough to have a sufficient number of data instances for which  $f_{m,a} > 0$  for all  $a \in \mathcal{S}$ , then this lemma supplies us with a way of approximating the ratios between all pairs of weights and subsequently approximating the weights themselves. In general, however, this may not be the case. Luckily, it is possible to estimate the ratio of the weights of each pair of actions  $a$  and  $b$  that are used together frequently by the population using only those data instances in which at least one agent is choosing each. Formally, define

$$\mathcal{M}_{a,b} = \{\mathcal{I}_m \in \mathcal{M} : f_{m,a} > 0, f_{m,b} > 0\}.$$

Lemma 11 tells us that if  $\mathcal{M}_{a,b}$  is sufficiently large, and there is at least one instance  $\mathcal{I}_m \in \mathcal{M}_{a,b}$  for which  $a_m = b$ , then we can approximate the ratio between  $w_a$  and  $w_b$  well.

What if one of these assumptions does not hold? If we are not able to collect sufficiently many instances in which  $f_{m,a} > 0$  and  $f_{m,b} > 0$ , then standard uniform convergence results can be used to show that it is very unlikely that we see a new instance for which  $f_a > 0$  and  $f_b > 0$ . This idea is formalized in the following lemma, the proof of which is in the appendix.

**Lemma 12** For any  $M < |\mathcal{M}|$ , for any  $\delta \in (0, 1)$ , with probability  $1 - \delta$ ,

$$\Pr_{\vec{f} \sim D^f} [\exists a, b \in \mathcal{S} : f_a > 0, f_b > 0, |\mathcal{M}_{a,b}| < M] \leq \frac{K^2}{2} \left( \frac{M}{|\mathcal{M}|} + \sqrt{\frac{\ln(K^2/(2\delta))}{2|\mathcal{M}|}} \right).$$

Similarly, if  $\chi_{a,b} = \chi_{b,a} = 0$ , then a standard uniform convergence argument can be used to show that it is unlikely that agent  $i$  would ever select action  $a$  or  $b$  when  $f_{m,a} > 0$  and  $f_{m,b} > 0$ . We will see that in this case, it is not important to learn the ratio between these two weights.

Using these observations, we can accurately model the behavior of agent  $i$ . The model consists of two phases. First, as a preprocessing step, we calculate a quantity

$$\chi_{a,b} = \sum_{m: \mathcal{I}_m \in \mathcal{M}_{a,b}} \chi_{a,b}^m$$

for each pair  $a, b \in \mathcal{S}$ . Then, each time we are presented with a state  $\vec{f}$ , we calculate a set of weights for all actions  $a$  with  $f_a > 0$  on the fly.

For a fixed  $\vec{f}$ , let  $\mathcal{S}'$  be the set of actions  $a \in \mathcal{S}$  such that  $f_a > 0$ . By Lemma 12, if the data set is sufficiently large, then we know that with high probability, it is the case that for all  $a, b \in \mathcal{S}'$ ,  $|\mathcal{M}_{a,b}| \geq M$  for some threshold  $M$ .

Now, let  $a^* = \operatorname{argmax}_{a \in \mathcal{S}'} |\{b : b \in \mathcal{S}', \chi_{a,b} \geq \chi_{b,a}\}|$ . Intuitively, if there is sufficient data,  $a^*$  should be the action in  $\mathcal{S}'$  with the highest weight, or have a weight arbitrarily close to the highest. Thus for any  $a \in \mathcal{S}'$ , Lemma 11 can be used to bound our estimate of  $w_a/w_{a^*}$  with a value of  $\beta$  arbitrarily close to 1. Noting that

$$\frac{w_a}{\sum_{s \in \mathcal{S}'} w_s} = \frac{w_a/w_{a^*}}{\sum_{s \in \mathcal{S}'} w_s/w_{a^*}},$$

we approximate the *relative* weight of action  $a \in \mathcal{S}'$  with respect to the other actions in  $\mathcal{S}'$  using

$$\hat{w}_a = \frac{\chi_{a,a^*}/\chi_{a^*,a}}{\sum_{s \in \mathcal{S}'} \chi_{s,a^*}/\chi_{a^*,s}},$$

and simply let  $\hat{w}_a = 0$  for any  $a \notin \mathcal{S}'$ . Applying Lemma 8, we find that for all  $a \in \mathcal{S}'$ , with high probability,

$$\left| \frac{w_a}{\sum_{s \in \mathcal{S}'} w_s} - \hat{w}_a \right| \leq \frac{(1 + \beta)K \sqrt{N \ln(2K^2/\delta)}}{\sqrt{2M} - (1 + \beta)K \sqrt{N \ln(2K^2/\delta)}}, \quad (9)$$

where  $M$  is the lower bound on  $|\mathcal{M}_{a,b}|$  for all  $a, b \in \mathcal{S}'$ , and  $\beta$  is close to 1. With this bound in place, it is straightforward to show that we can apply Lemma 8 once more to bound the expected  $\mathcal{L}_1$ ,

$$\mathbb{E}_{\vec{f} \sim D^f} \left[ \sum_{a \in \mathcal{S}} \left| \frac{w_a f_a}{\sum_{s \in \mathcal{S}} w_s f_s} - \frac{\hat{w}_a f_a}{\sum_{s \in \mathcal{S}} \hat{w}_s f_s} \right| \right],$$

and that the bound goes to 0 at a rate of  $O(1/\sqrt{M})$  as the threshold  $M$  grows. More details are given in the appendix.

Since it is possible to build an accurate model of the strategy of agent  $i$  in polynomial time under any distribution  $D$  over state vectors, we can again apply Theorem 6 to see that this class is polynomially learnable from collective behavior.

**Theorem 13** The class of crowd affinity multiplicative model strategies is polynomially learnable from collective behavior.

### 5.3 Learning Without Resets

Although the analyses in the previous subsections are tailored to learnability in the sense of Definition 2, they can easily be adapted to hold in the alternate setting in which the learner has access only to a single, unbroken trajectory of states. In this alternate model, the learning algorithm observes a polynomially long prefix of a trajectory of states for training, and then must produce a generative model which results in a distribution over the values of the subsequent  $T$  states close to the true distribution.

When learning individual crowd affinity models for each agent in this setting, we again assume that we are presented with a set of samples  $\mathcal{M}$ , where each instance  $\mathcal{I}_m \in \mathcal{M}$  consists of a pair  $\langle \vec{f}_m, a_m \rangle$ . However, instead of assuming that the state distributions  $\vec{f}_m$  are distributed according to  $D^f$ , we now assume that the state and action pairs represent a single trajectory. As previously noted, the majority of the analysis for both the mixture and multiplicative variants of the crowd affinity model does not depend on the particular way in which state distribution vectors are distributed, and thus carries over to this setting as is. Here we briefly discuss the few modifications that are necessary.

The only change required in the analysis of the crowd affinity mixture model relates to handling the case in which  $Z_a$  is small for all  $a$ . Previously we argued that when this is the case, the distribution  $D^f$  must be concentrated so that for all  $a$ ,  $f_a$  falls within a very small range with high probability. Thus it is not necessary to estimate the parameter  $\alpha$  directly, and we can instead learn a single probability for each action that is used regardless of  $\vec{f}$ . A similar argument holds in the no-reset variant. If it is the case that  $Z_a$  is small for all  $a$ , then it must be the case that for each  $a$ , the value of  $f_a$  has fallen into the same small range for the entire observed trajectory. A standard uniform convergence argument says that the probability that  $f_a$  suddenly changes dramatically is very small, and thus again it is sufficient to learn a single probability for each action that is used regardless of  $\vec{f}$ .

To adapt the analysis of the crowd affinity multiplicative model, it is first necessary to replace Lemma 12. Recall that the purpose of this lemma was to show that when the data set does not contain sufficient samples in which  $f_a > 0$  and  $f_b > 0$  for a pair of actions  $a$  and  $b$ , the chance of observing a new state distribution  $\vec{f}$  with  $f_a > 0$  and  $f_b > 0$  is small. This argument is actually much more straightforward in the no-reset case. By the definition of the model, it is easy to see that if  $f_a > 0$  for some action  $a$  at time  $t$  in a trajectory, then it must be the case that  $f_a > 0$  at all previous points in the trajectory. Thus if  $f_a > 0$  on any test instance, then  $f_a$  must have been non-negative on *every* training instance, and we do not have to worry about the case in which there is insufficient data to compare the weights of a particular pair of actions.

One additional, possibly more subtle, modification is necessary in the analysis of the multiplicative model to handle the case in which  $\chi_{a,b} = \chi_{b,a} = 0$  for all “active” pairs of actions  $a, b \in \mathcal{S}'$ . This can happen only if agent  $i$  has

extremely small weights for every action in  $\mathcal{S}'$ , and had previously been choosing an alternate action that is no longer available, i.e., an action  $s$  for which  $f_s$  had previously been non-negative but suddenly is not. However, in order for  $f_s$  to become 0, it must be the case that agent  $i$  himself chooses an alternate action (say, action  $a$ ) instead of  $s$ , which cannot happen since the estimated weight of action  $a$  used by the model is 0. Thus this situation can never occur in the no-reset variant.

## 6 Conclusions and Future Work

We have introduced a computational model for learning from collective behavior, and populated it with some initial general theory and algorithmic results for crowd affinity models. In addition to positive or negative results for further agent strategy classes, there are a number of other general directions of interest for future research. These include extension of our model to agnostic [14] settings, in which we relax the assumption that every agent strategy falls in a known class, and to reinforcement learning [23] settings, in which the learning algorithm may itself be a member of the population being modeled, and wishes to learn an optimal policy with respect to some reward function.

## Acknowledgments

We thank Nina Balcan and Eyal Even-Dar for early discussions on models of social learning, and Duncan Watts for helpful conversations and pointers to relevant literature.

## References

- [1] S. Bikhchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy*, 100:992–1026, 1992.
- [2] S. Bikhchandani, D. Hirshleifer, and I. Welch. Learning from the behavior of others: Conformity, fads, and informational cascades. *The Journal of Economic Perspectives*, 12:151–170, 1998.
- [3] G. W. Brown. Iterative solutions of games by fictitious play. In T.C. Koopmans, editor, *Activity Analysis of Production and Allocation*. Wiley, 1951.
- [4] T. Bylander. Learning noisy linear threshold functions. Technical Report, 1998.
- [5] E. Cohen. Learning noisy perceptrons by a perceptron in polynomial time. In *38th IEEE Annual Symposium on Foundations of Computer Science*, 1997.
- [6] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, NY, 1991.
- [7] P. Dodds, R. Muhamad, and D. J. Watts. An experimental study of search in global social networks. *Science*, 301:828–829, August 2003.
- [8] M. Drehmann, J. Oechssler, and A. Roeder. Herding and contrarian behavior in financial markets: An Internet experiment. *American Economic Review*, 95(5):1403–1426, 2005.
- [9] D. Foster and R. Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29:7–35, 1999.
- [10] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 61:1420–1443, 1978.
- [11] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.
- [12] P. Hedstrom. Rational imitation. In P. Hedstrom and R. Swedberg, editors, *Social Mechanisms: An Analytical Approach to Social Theory*. Cambridge University Press, 1998.
- [13] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R.E. Schapire, and L. Sellie. On the learnability of discrete distributions. In *26th Annual ACM Symposium on Theory of Computing*, pages 273–282, 1994.
- [14] M. Kearns, R. Schapire, and L. Sellie. Towards efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.
- [15] M. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.
- [16] M. Kearns and S. Singh. Finite-sample rates of convergence for Q-learning and indirect methods. In *Advances in Neural Information Processing Systems 11*, 1999.
- [17] M. Kearns, S. Suri, and N. Montfort. A behavioral study of the coloring problem on human subject networks. *Science*, 313(5788):824–827, 2006.
- [18] J. Kleinberg. Cascading behavior in networks: Algorithmic and economic issues. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [19] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.
- [20] M. Salganik, P. Dodds, and D. J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 331(5762):854–856, 2006.
- [21] M. Salganik and D. J. Watts. Social influence, manipulation, and self-fulfilling prophecies in cultural markets. Preprint, 2007.
- [22] T. Schelling. *Micromotives and Macrobehavior*. Norton, New York, NY, 1978.
- [23] R. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, 1998.
- [24] A. De Vany. *Hollywood Economics: How Extreme Uncertainty Shapes the Film Industry*. Routledge, London, 2004.
- [25] A. De Vany and C. Lee. Quality signals in information cascades and the dynamics of the distribution of motion picture box office revenues. *Journal of Economic Dynamics and Control*, 25:593–614, 2001.
- [26] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.
- [27] I. Welch. Herding among security analysts. *Journal of Financial Economics*, 58:369–396, 2000.

---

# Injective Hilbert Space Embeddings of Probability Measures

---

Bharath K. Sriperumbudur<sup>1\*</sup>, Arthur Gretton<sup>2</sup>, Kenji Fukumizu<sup>3</sup>, Gert Lanckriet<sup>1</sup> and Bernhard Schölkopf<sup>2</sup>

<sup>1</sup>Department of ECE, UC San Diego, La Jolla, CA 92093, USA.

<sup>2</sup>MPI for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany.

<sup>3</sup>Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106-8569, Japan.

bharathsv@ucsd.edu, {arthur,bernhard.schoelkopf}@tuebingen.mpg.de  
fukumizu@ism.ac.jp, gert@ece.ucsd.edu

## Abstract

A Hilbert space embedding for probability measures has recently been proposed, with applications including dimensionality reduction, homogeneity testing and independence testing. This embedding represents any probability measure as a mean element in a reproducing kernel Hilbert space (RKHS). The embedding function has been proven to be injective when the reproducing kernel is universal. In this case, the embedding induces a metric on the space of probability distributions defined on compact metric spaces.

In the present work, we consider more broadly the problem of specifying characteristic kernels, defined as kernels for which the RKHS embedding of probability measures is injective. In particular, characteristic kernels can include non-universal kernels. We restrict ourselves to translation-invariant kernels on Euclidean space, and define the associated metric on probability measures in terms of the Fourier spectrum of the kernel and characteristic functions of these measures. The support of the kernel spectrum is important in finding whether a kernel is characteristic: in particular, the embedding is injective if and only if the kernel spectrum has the entire domain as its support. Characteristic kernels may nonetheless have difficulty in distinguishing certain distributions on the basis of finite samples, again due to the interaction of the kernel spectrum and the characteristic functions of the measures.

## 1 Introduction

The concept of distance between probability measures is a fundamental one and has many applications in probability theory and statistics. In probability theory, this notion is

---

\*The author wishes to acknowledge the support from the Max Planck Institute (MPI) for Biological Cybernetics, National Science Foundation (grant DMS-MSPA 0625409), the Fair Isaac Corporation and the University of California MICRO program. Part of this work was done while the author was an intern at MPI. The authors thank anonymous reviewers for their comments to improve the paper.

used to metrize the weak convergence (convergence in distribution) of probability measures defined on a metric space. Formally, let  $\mathfrak{S}$  be the set of all Borel probability measures defined on a metric measurable space  $(M, \rho, \mathcal{M}_\rho)$  and let  $\gamma$  be its metric, i.e.,  $(\mathfrak{S}, \gamma)$  is a metric space. Then  $P_n$  is said to converge weakly to  $P$  if and only if  $\gamma(P_n, P) \xrightarrow{n \rightarrow \infty} 0$ , where  $P, \{P_n\}_{n \geq 1} \in \mathfrak{S}$ . When  $M$  is separable, examples for  $\gamma$  include the *Lévy-Prohorov distance* and the *dual-bounded Lipschitz distance (Dudley metric)* [Dud02, Chapter 11]. Other popular examples for  $\gamma$  include the *Monge-Wasserstein distance*, *total variation distance* and the *Hellinger distance*, which yield a stronger notion of convergence of probability measures [Sho00, Chapter 19].

In statistics, the notion of distance between probability measures is used in a variety of applications, including homogeneity tests (the two-sample problem), independence tests, and goodness-of-fit tests. The two-sample problem involves testing the null hypothesis  $H_0 : P = Q$  versus the alternative  $H_1 : P \neq Q$ , using random samples  $\{X_l\}_{l=1}^m$  and  $\{Y_l\}_{l=1}^n$  drawn i.i.d. from distributions  $P$  and  $Q$  on a measurable space  $(M, \mathcal{M})$ . If  $\gamma$  is a metric (or more generally a semi-metric<sup>1</sup>) on  $\mathfrak{S}$ , then  $\gamma(P, Q)$  can be used as a test statistic to address the two-sample problem. This is because  $\gamma(P, Q)$  takes the unique and distinctive value of zero only when  $P = Q$ . Thus, the two-sample problem can be reduced to testing  $H_0 : \gamma(P, Q) = 0$  versus  $H_1 : \gamma(P, Q) > 0$ . The problems of testing independence and goodness-of-fit can be posed in an analogous form.

Several recent studies on kernel methods have focused on applications in distribution comparison: the advantage being that kernels represent a linear way of dealing with higher order statistics. For instance, in homogeneity testing, differences in higher order moments are encoded in mean differences computed in the right reproducing kernel Hilbert space (RKHS) [GBR<sup>+</sup>07]; in kernel ICA [BJ02, GHS<sup>+</sup>05], general nonlinear dependencies show up as linear correlations once they are computed in a suitable RKHS. Instrumental to these studies is the notion of a Hilbert space embedding for probability measures [SGSS07], which involves representing any probability measure as a mean element in an RKHS  $(\mathcal{H}, k)$ , where  $k$  is the reproducing kernel [Aro50,

---

<sup>1</sup>Given a set  $M$ , a *metric* for  $M$  is a function  $\rho : M \times M \rightarrow \mathbb{R}_+$  such that (i)  $\forall x, \rho(x, x) = 0$ , (ii)  $\forall x, y, \rho(x, y) = \rho(y, x)$ , (iii)  $\forall x, y, z, \rho(x, z) \leq \rho(x, y) + \rho(y, z)$ , and (iv)  $\rho(x, y) = 0 \Rightarrow x = y$  [Dud02, Chapter 2]. A semi-metric only satisfies (i), (ii) and (iv).

SS02]. For this reason, the RKHSs used have to be “sufficiently large” to capture all nonlinearities that are relevant to the problem at hand, so that differences in embeddings correspond to differences of interest in the distributions. The question of how to choose such RKHSs is the central focus of the present paper.

Recently, Fukumizu *et al.* [FGSS08] introduced the concept of a *characteristic kernel*, this being an RKHS kernel for which the mapping  $\Pi : \mathfrak{S} \rightarrow \mathcal{H}$  from the space of Borel probability measures  $\mathfrak{S}$  to the associated RKHS  $\mathcal{H}$  is injective ( $\mathcal{H}$  is denoted as a characteristic RKHS). Clearly, a characteristic RKHS is sufficiently large in the sense we have described: in this case  $\gamma(P, Q) = 0$  implies  $P = Q$ , where  $\gamma$  is the induced metric on  $\mathfrak{S}$  by  $\Pi$ , defined as the RKHS distance between the mappings of  $P$  and  $Q$ . Under what conditions, then, is  $\Pi$  injective? As discussed in [GBR<sup>+</sup>07, SGSS07], when  $M$  is compact, the RKHS is characteristic when its kernel is universal in the sense of Steinwart [Ste02, Definition 4]: the induced RKHS should be dense in the Banach space of bounded continuous functions with respect to the supremum norm (examples include the Gaussian and Laplacian kernels). Fukumizu *et al.* [FGSS08, Lemma 1] considered injectivity for non-compact  $M$ , and showed  $\Pi$  to be injective if the direct sum of  $\mathcal{H}$  and  $\mathbb{R}$  is dense in the Banach space of  $p$ -power ( $p \geq 1$ ) integrable functions (we denote RKHSs satisfying this criterion as  $F$ -characteristic). In addition, for  $M = \mathbb{R}^d$ , Fukumizu *et al.* provide sufficient conditions on the Fourier spectrum of a translation-invariant kernel for it to be characteristic [FGSS08, Theorem 2]. Using this result, popular kernels like Gaussian and Laplacian can be shown to be characteristic on all of  $\mathbb{R}^d$ .

In the present study, we provide an alternative means of determining whether kernels are characteristic, for the case of translation-invariant kernels on  $\mathbb{R}^d$ . This addresses several limitations of the previous work: in particular, it can be difficult to verify the conditions that a universal or  $F$ -characteristic kernel must satisfy; and universality is in any case an overly restrictive condition because universal kernels assume  $M$  to be compact. In other words, they induce a metric only on the space of probability measures that are compactly supported on  $M$ . In addition, there are compactly supported kernels which are not universal, e.g.  $B_{2n+1}$ -splines, which can be shown to be characteristic. We provide simple verifiable rules in terms of the Fourier spectrum of the kernel that characterize the injective behavior of  $\Pi$ , and derive a relationship between the family of kernels and the family of probability measures for which  $\gamma(P, Q) = 0$  implies  $P = Q$ . In particular, we show that a translation-invariant kernel on  $\mathbb{R}^d$  is characteristic if and only if its Fourier spectrum has the entire domain as its support.

We begin our presentation in §2 with an overview of terminology and notation. In §3, we briefly describe the approach of Hilbert space embedding of probability measures. Assuming the kernel to be translation-invariant in  $\mathbb{R}^d$ , in §4, we deduce conditions on the kernel and the set of probability measures for which the RKHS is characteristic. We show that the support of the kernel spectrum is crucial:  $\mathcal{H}$  is characteristic if and only if the kernel spectrum has the entire domain as its support. We note, however, that even using such a kernel does not guarantee that one can easily distinguish dis-

tributions based on finite samples. In particular, we provide two illustrations in §5 where interactions between the kernel spectrum and the characteristic functions of the probability measures can result in an arbitrarily small  $\gamma(P, Q) = \epsilon > 0$  for non-trivial differences in distributions  $P \neq Q$ . Proofs of the main theorems and related lemmas are provided in §6. The results presented in this paper use tools from *distribution theory* and Fourier analysis: the related technical results are collected in Appendix A.

## 2 Notation

For  $M \subset \mathbb{R}^d$  and  $\mu$  a Borel measure on  $M$ ,  $L^p(M, \mu)$  denotes the Banach space of  $p$ -power ( $p \geq 1$ )  $\mu$ -integrable functions. We will also use  $L^p(M)$  for  $L^p(M, \mu)$  and  $dx$  for  $d\mu(x)$  if  $\mu$  is the Lebesgue measure on  $M$ .  $C_b(M)$  denotes the space of all bounded, continuous functions on  $M$ . The space of all  $q$ -continuously differentiable functions on  $M$  is denoted by  $C^q(M)$ ,  $0 \leq q \leq \infty$ . For  $x \in \mathbb{C}$ ,  $\bar{x}$  represents the complex conjugate of  $x$ . We denote as  $i$  the complex number  $\sqrt{-1}$ .

The set of all compactly supported functions in  $C^\infty(\mathbb{R}^d)$  is denoted by  $\mathcal{D}_d$  and the space of rapidly decreasing functions in  $\mathbb{R}^d$  is denoted by  $\mathcal{S}_d$ . For an open set  $U \subset \mathbb{R}^d$ ,  $\mathcal{D}_d(U)$  denotes the subspace of  $\mathcal{D}_d$  consisting of the functions with support contained in  $U$ . The space of linear continuous functionals on  $\mathcal{D}_d$  (resp.  $\mathcal{S}_d$ ) is denoted by  $\mathcal{D}'_d$  (resp.  $\mathcal{S}'_d$ ) and an element of such a space is called as a *distribution* (resp. *tempered distribution*).  $m_d$  denotes the normalized Lebesgue measure defined by  $dm_d(x) = (2\pi)^{-\frac{d}{2}} dx$ .  $\hat{f}$  and  $\check{f}$  represent the Fourier transform and inverse Fourier transform of  $f$  respectively.

For a measurable function  $f$  and a signed measure  $P$ ,  $Pf := \int f dP = \int_M f(x) dP(x)$ .  $\delta_x$  represents the Dirac measure at  $x$ . The symbol  $\delta$  is overloaded to represent the Dirac measure, the Dirac-delta function, and the Kronecker-delta, which should be distinguishable from the context.

## 3 Maximum Mean Discrepancy

We briefly review the theory of RKHS embedding of probability measures proposed by Smola *et al.* [SGSS07]. We lead to these embeddings by first introducing the maximum mean discrepancy (MMD), which is based on the following result [Dud02, Lemma 9.3.2], related to the weak convergence of probability measures on metric spaces.

**Lemma 1 ([Dud02])** *Let  $(M, \rho)$  be a metric space with Borel probability measures  $P$  and  $Q$  defined on  $M$ . Then  $P = Q$  if and only if  $Pf = Qf, \forall f \in C_b(M)$ .*

Originally, Gretton *et al.* [GBR<sup>+</sup>07] defined the maximum mean discrepancy as follows.

**Definition 2 (Maximum Mean Discrepancy)** *Let  $\mathcal{F} = \{f \mid f : M \rightarrow \mathbb{R}\}$  and let  $P, Q$  be Borel probability measures defined on  $(M, \rho)$ . Then the maximum mean discrepancy is defined as*

$$\gamma_{\mathcal{F}}(P, Q) = \sup_{f \in \mathcal{F}} |Pf - Qf|. \quad (1)$$

With this definition, one can derive various metrics (mentioned in §1) that are used to define the weak convergence of probability measures on metric spaces. To start with, it is easy to verify that, independent of  $\mathcal{F}$ ,  $\gamma_{\mathcal{F}}$  in Eq. (1) is a pseudometric<sup>2</sup> on  $\mathfrak{S}$ . Therefore, the choice of  $\mathcal{F}$  determines whether or not  $\gamma_{\mathcal{F}}(P, Q) = 0$  implies  $P = Q$ . In other words,  $\mathcal{F}$  determines the metric property of  $\gamma_{\mathcal{F}}$  on  $\mathfrak{S}$ . By Lemma 1,  $\gamma_{\mathcal{F}}$  is a metric on  $\mathfrak{S}$  when  $\mathcal{F} = C_b(M)$ . When  $\mathcal{F}$  is the set of bounded,  $\rho$ -uniformly continuous functions on  $M$ , by the Portmanteau theorem [Sho00, Chapter 19, Theorem 1.1],  $\gamma_{\mathcal{F}}$  is not only a metric on  $\mathfrak{S}$  but also metrizes the weak topology on  $\mathfrak{S}$ .  $\gamma_{\mathcal{F}}$  is a *Dudley metric* [Sho00, Chapter 19, Definition 2.2] when  $\mathcal{F} = \{f : \|f\|_{BL} \leq 1\}$  where  $\|f\|_{BL} = \|f\|_{\infty} + \|f\|_L$  with  $\|f\|_{\infty} := \sup\{|f(x)| : x \in M\}$  and  $\|f\|_L := \sup\{|f(x) - f(y)|/\rho(x, y) : x \neq y \text{ in } M\}$ .  $\|f\|_L$  is called the Lipschitz seminorm of a real-valued function  $f$  on  $M$ . By the Kantorovich-Rubinstein theorem [Dud02, Theorem 11.8.2], when  $(M, \rho)$  is separable,  $\gamma_{\mathcal{F}}$  equals the *Monge-Wasserstein distance* for  $\mathcal{F} = \{f : \|f\|_L \leq 1\}$ .  $\gamma_{\mathcal{F}}$  is the *total variation metric* when  $\mathcal{F} = \{f : \|f\|_{\infty} \leq 1\}$  while it is the *Kolmogorov distance* when  $\mathcal{F} = \{\mathbb{1}_{(-\infty, t]} : t \in \mathbb{R}^d\}$ . If  $\mathcal{F} = \{e^{i\langle \omega, \cdot \rangle} : \omega \in \mathbb{R}^d\}$ , then  $\gamma_{\mathcal{F}}(P, Q)$  reduces to finding the maximal difference between the characteristic functions of  $P$  and  $Q$ . By the uniqueness theorem for characteristic functions [Dud02, Theorem 9.5.1], we have  $\gamma_{\mathcal{F}}(P, Q) = 0 \Leftrightarrow \phi_P = \phi_Q \Leftrightarrow P = Q$ , where  $\phi_P$  and  $\phi_Q$  represent the characteristic functions of  $P$  and  $Q$ , respectively.<sup>3</sup> Therefore, the function class  $\mathcal{F} = \{e^{i\langle \omega, \cdot \rangle} : \omega \in \mathbb{R}^d\}$  induces a metric on  $\mathfrak{S}$ . Gretton *et al.* [GBR<sup>+</sup>07, Theorem 3] showed  $\gamma_{\mathcal{F}}$  to be a metric on  $\mathfrak{S}$  when  $\mathcal{F}$  is chosen to be a unit ball in a universal RKHS  $\mathcal{H}$ . This choice of  $\mathcal{F}$  yields an injective map,  $\Pi : \mathfrak{S} \rightarrow \mathcal{H}$ , as proposed by Smola *et al.* [SGSS07]. A similar injective map can also be obtained by choosing  $\mathcal{F}$  to be a unit ball in an RKHS induced by kernels satisfying the criteria in [FGSS08, Lemma 1, Theorem 2] (which we denote  $F$ -characteristic kernels).

We henceforth assume  $\mathcal{F}$  to be a unit ball in an RKHS  $(\mathcal{H}, k)$  (not necessarily universal or  $F$ -characteristic) defined on  $(M, \mathcal{M})$  with  $k : M \times M \rightarrow \mathbb{R}$ , i.e.,  $\mathcal{F} = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq 1\}$ . The following result provides a different representation for  $\gamma_{\mathcal{F}}$  defined in Eq. (1) by exploiting the reproducing property of  $\mathcal{H}$ , and will be used later in deriving our main results.

**Theorem 3** *Let  $\mathcal{F}$  be a unit ball in an RKHS  $(\mathcal{H}, k)$  defined on a measurable space  $(M, \mathcal{M})$  with  $k$  measurable and bounded. Then*

$$\gamma_{\mathcal{F}}(P, Q) = \|Pk - Qk\|_{\mathcal{H}}, \quad (2)$$

where  $\|\cdot\|_{\mathcal{H}}$  represents the RKHS norm.

**Proof:** Let  $T_P : \mathcal{H} \rightarrow \mathbb{R}$  be a linear functional defined as  $T_P[f] := \int_M f(x) dP(x)$  with  $\|T_P\| := \sup_{f \in \mathcal{H}} \frac{|T_P[f]|}{\|f\|_{\mathcal{H}}}$ .

<sup>2</sup>A pseudometric only satisfies (i)-(iii) of the properties of a metric (see footnote 1). Unlike a metric space  $(M, \rho)$ , points in a pseudometric space need not be distinguishable: one may have  $\rho(x, y) = 0$  for  $x \neq y$  [Dud02, Chapter 2].

<sup>3</sup>The characteristic function of a probability measure,  $P$  on  $\mathbb{R}^d$  is defined as  $\phi(\omega) := \int_{\mathbb{R}^d} e^{i\omega^T x} dP(x)$ ,  $\forall \omega \in \mathbb{R}^d$ .

Consider

$$\begin{aligned} |T_P[f]| &= \left| \int_M f(x) dP(x) \right| \leq \int_M |f(x)| dP(x) \\ &= \int_M |\langle f, k(\cdot, x) \rangle_{\mathcal{H}}| dP(x) \leq \sqrt{C} \|f\|_{\mathcal{H}}, \end{aligned}$$

where we have exploited the reproducing property and boundedness of the kernel to show  $T_P$  is a bounded linear functional on  $\mathcal{H}$ . Here,  $C > 0$  is the bound on  $k$ , i.e.,  $|k(x, y)| \leq C < \infty$ ,  $\forall x, y \in M$ . Therefore, by the Riesz representation theorem [RS72, Theorem II.4], there exists a unique  $\lambda_P \in \mathcal{H}$  such that  $T_P[f] = \langle f, \lambda_P \rangle_{\mathcal{H}}$ ,  $\forall f \in \mathcal{H}$ . Let  $f = k(\cdot, u)$  for some  $u \in M$ . Then,  $T_P[k(\cdot, u)] = \langle k(\cdot, u), \lambda_P \rangle_{\mathcal{H}} = \lambda_P(u)$ , which implies  $\lambda_P = T_P[k] = Pk = \int_M k(\cdot, x) dP(x)$ . Therefore, with  $|Pf - Qf| = |\langle f, \lambda_P - \lambda_Q \rangle_{\mathcal{H}}|$ , we have  $\gamma_{\mathcal{F}}(P, Q) = \sup_{\|f\|_{\mathcal{H}} \leq 1} |Pf - Qf| = \|\lambda_P - \lambda_Q\|_{\mathcal{H}} = \|Pk - Qk\|_{\mathcal{H}}$ . ■

The representation of  $\gamma_{\mathcal{F}}$  in Eq. (2) yields the embedding,  $\Pi[P] = \int_M k(\cdot, x) dP(x)$  as proposed in [SGSS07, FGSS08], which is injective when  $k$  is characteristic. While the representation of  $\gamma_{\mathcal{F}}$  in Eq. (2) holds irrespective of the characteristic property of  $k$ , it need not be a metric on  $\mathfrak{S}$ , as  $\Pi$  is not guaranteed to be injective. The obvious question to ask is ‘‘For what class of kernels is  $\Pi$  injective?’’. To understand this in detail, we are interested in the following questions which we address in this paper.

- Q1. Let  $\mathfrak{D} \subsetneq \mathfrak{S}$  be a set of Borel probability measures defined on  $(M, \mathcal{M})$ . Let  $\mathcal{K}$  be a family of positive definite kernels defined on  $M$ . What are the conditions on  $\mathfrak{D}$  and  $\mathcal{K}$  for which  $\Pi : \mathfrak{D} \rightarrow \mathcal{H}_k$ ,  $P \mapsto \int_M k(\cdot, x) dP(x)$  is injective, i.e.,  $\gamma_{\mathcal{F}}(P, Q) = 0 \Leftrightarrow P = Q$  for  $P, Q \in \mathfrak{D}$ ? Here,  $\mathcal{H}_k$  represents the RKHS induced by  $k \in \mathcal{K}$ .
- Q2. What are the conditions on  $\mathcal{K}$  so that  $\Pi$  is injective on  $\mathfrak{S}$ ?

Note that Q1 is a restriction of Q2 to  $\mathfrak{D}$ . The idea is that the kernels that do not make  $\gamma_{\mathcal{F}}$  as a metric on  $\mathfrak{S}$  may make it as a metric on some restricted class of probability measures,  $\mathfrak{D} \subsetneq \mathfrak{S}$ . Our next step, therefore, is to characterize the relationship between classes of kernels and probability measures, which is addressed in the following section.

## 4 Characteristic Kernels & Main Theorems

In this section, we present main results related to the behavior of MMD. We start with the following definition of characteristic kernels, which was recently introduced by Fukumizu *et al.* [FGSS08] in the context of measuring conditional (in)dependence using positive definite kernels.

**Definition 4 (Characteristic kernel)** *A positive definite kernel  $k$  is characteristic to a set  $\mathfrak{D}$  of probability measures defined on  $(M, \mathcal{M})$  if  $\gamma_{\mathcal{F}}(P, Q) = 0 \Leftrightarrow P = Q$  for  $P, Q \in \mathfrak{D}$ .*

**Remark 5** *Equivalently,  $k$  is said to be characteristic to  $\mathfrak{D}$  if the map,  $\Pi : \mathfrak{D} \rightarrow \mathcal{H}$ ,  $P \mapsto \int_M k(\cdot, x) dP(x)$ , is injective. When  $M = \mathbb{R}^d$ , the notion of characteristic kernel is a generalization of the characteristic function,  $\phi_P(\omega) = \int_{\mathbb{R}^d} e^{i\omega^T x} dP(x)$ ,  $\forall \omega \in \mathbb{R}^d$ , which is the expectation of the*

complex-valued positive definite kernel,  $k(\omega, x) = e^{i\omega^T x}$ . Thus, the definition of a characteristic kernel generalizes the well-known property of the characteristic function that  $\phi_P$  uniquely determines a Borel probability measure  $P$  on  $\mathbb{R}^d$ . See [FGSS08] for more details.

It is obvious from Definition 4 that universal kernels defined on a compact  $M$  and  $F$ -characteristic kernels on  $M$  are characteristic to the family of all probability measures defined on  $(M, \mathcal{M})$ . The characteristic property of the kernel relates the family of positive definite kernels and the family of probability measures. We would like to characterize the positive definite kernels that are characteristic to  $\mathfrak{S}$ . Among the kernels that are not characteristic to  $\mathfrak{S}$ , we would like to determine those kernels that are characteristic to some appropriately chosen subset  $\mathfrak{D}$ , of  $\mathfrak{S}$ . Intuitively, the smaller the set  $\mathfrak{D}$ , larger is the family of kernels that are characteristic to  $\mathfrak{D}$ . To this end, we make the following assumption.

**Assumption 1**  $k(x, y) = \psi(x - y)$  where  $\psi$  is a bounded continuous real-valued positive definite function<sup>4</sup> on  $M = \mathbb{R}^d$ .

The above assumption means that  $k$  is translation-invariant in  $\mathbb{R}^d$ . A whole family of such kernels can be generated as the Fourier transform of a finite non-negative Borel measure, given by the following result due to Bochner, which we quote from [Wen05, Theorem 6.6].

**Theorem 6 (Bochner)** A continuous function  $\psi : \mathbb{R}^d \rightarrow \mathbb{C}$  is positive definite if and only if it is the Fourier transform of a finite nonnegative Borel measure  $\Lambda$  on  $\mathbb{R}^d$ , i.e.

$$\psi(x) = \int_{\mathbb{R}^d} e^{-ix^T \omega} d\Lambda(\omega), \quad \forall x \in \mathbb{R}^d. \quad (3)$$

Since the translation-invariant kernels in  $\mathbb{R}^d$  are characterized by the Bochner's theorem, it is theoretically interesting to ask which subset in the Fourier images gives characteristic kernels. Before we describe such kernels  $k$  that are characteristic to  $\mathfrak{S}$ , in the following example, we show that there exist kernels that are not characteristic to  $\mathfrak{S}$ . Here,  $\mathfrak{S}$  represents the family of all Borel probability measures defined on  $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ , where  $\mathcal{B}(\mathbb{R}^d)$  represents the Borel  $\sigma$ -algebra defined by open sets in  $\mathbb{R}^d$  (see Assumption 1).

**Example 1 (Trivial kernel)** Let  $k(x, y) = \psi(x - y) = C$ ,  $\forall x, y \in \mathbb{R}^d$  with  $C > 0$ . It can be shown that  $\psi$  is the Fourier transform of  $\Lambda = C\delta_0$  with support  $\{0\}$ .

Consider  $Pk = \int_{\mathbb{R}^d} k(\cdot, x) dP(x) = C \int_{\mathbb{R}^d} dP(x) = C$ . Since  $Pk = C$  irrespective of  $P \in \mathfrak{S}$ , the map  $\Pi$  is not injective. In addition,  $\gamma_{\mathcal{F}}(P, Q) = 0$  for any  $P, Q \in \mathfrak{S}$ . Therefore, the trivial kernel,  $k$  is not characteristic to  $\mathfrak{S}$ .

#### 4.1 Main theorems

The following theorem characterizes all translation-invariant kernels in  $\mathbb{R}^d$  that are characteristic to  $\mathfrak{S}$ .

<sup>4</sup>Let  $M$  be a nonempty set. A function  $\psi : M \rightarrow \mathbb{R}$  is called positive definite if and only if  $\sum_{j,l=1}^n c_j c_l \psi(x_j - x_l) \geq 0$ ,  $\forall x_j \in M$ ,  $\forall c_j \in \mathbb{R}$ ,  $\forall n \in \mathbb{N}$ .

**Theorem 7** Let  $\mathcal{F}$  be a unit ball in an RKHS  $(\mathcal{H}, k)$  defined on  $\mathbb{R}^d$ . Suppose  $k$  satisfies Assumption 1. Then  $k$  is a characteristic kernel to the family,  $\mathfrak{S}$ , of all probability measures defined on  $\mathbb{R}^d$  if and only if  $\text{supp}(\Lambda) = \mathbb{R}^d$ .

We provide a sketch of the proof of the above theorem, which is proved in §6.2.1 using a number of intermediate lemmas. The first step is to derive an alternate representation for  $\gamma_{\mathcal{F}}$  in Eq. (2) under Assumption 1. Lemma 13 provides the Fourier representation of  $\gamma_{\mathcal{F}}$  in terms of the kernel spectrum,  $\Lambda$  and the characteristic functions of  $P$  and  $Q$ . The advantage of this representation over the one in Eq. (2) is that it is easy to obtain necessary and sufficient conditions for the existence of  $P \neq Q$ ,  $P, Q \in \mathfrak{S}$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ , which are captured in Lemma 15. We then show that if  $\text{supp}(\Lambda) = \mathbb{R}^d$ , the conditions mentioned in Lemma 15 are violated, meaning  $\nexists P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ , thereby proving the sufficient condition in Theorem 7. Proving the converse is equivalent to proving that  $k$  is not characteristic to  $\mathfrak{S}$  when  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$ . So, when  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$ , the result is proved using Lemma 19, which shows the existence of  $P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ .

Theorem 7 shows that the embedding function  $\Pi$ , associated with a positive definite translation-invariant kernel in  $\mathbb{R}^d$  is injective if and only if the kernel spectrum has the entire domain as its support. Therefore, this result provides a simple verifiable rule for  $\Pi$  to be injective, unlike the results in [SGSS07, FGSS08] where the universality and  $F$ -characteristic properties of a given kernel are not easy to verify. In addition, the universality and  $F$ -characteristic properties are sufficient conditions for a kernel to induce an injective map  $\Pi$ , whereas Theorem 7 provides  $\text{supp}(\Lambda) = \mathbb{R}^d$  as the necessary and sufficient condition. Therefore, we have answered question Q2 posed in §3. Examples of kernels that are characteristic to  $\mathfrak{S}$  include the Gaussian, Laplacian and  $B_{2n+1}$ -splines. In fact, the whole family of compactly supported translation-invariant kernels on  $\mathbb{R}^d$  are characteristic to  $\mathfrak{S}$ , as shown by the following corollary of Theorem 7.

**Corollary 8** Let  $\mathcal{F}$  be a unit ball in an RKHS  $(\mathcal{H}, k)$  defined on  $\mathbb{R}^d$ . Suppose  $k$  satisfies Assumption 1 and  $\text{supp}(\psi)$  is compact. Then  $k$  is a characteristic kernel to  $\mathfrak{S}$ .

**Proof:** Since  $\text{supp}(\psi)$  is compact in  $\mathbb{R}^d$ , by Lemma 25, which is a corollary of the Paley-Wiener theorem (see also [GW99, Theorem 31.5.2, Proposition 31.5.4]), we deduce that  $\text{supp}(\Lambda) = \mathbb{R}^d$ . Therefore, the result follows from Theorem 7. ■

The above result is interesting in practice because of the computational advantage in dealing with compactly supported kernels. By Theorem 7, it is clear that kernels with  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  are not characteristic to  $\mathfrak{S}$ . However, they can be characteristic to some  $\mathfrak{D} \subsetneq \mathfrak{S}$  (see Q1 in §3). The following result addresses this setting.

**Theorem 9** Let  $\mathcal{F}$  be a unit ball in an RKHS  $(\mathcal{H}, k)$  defined on  $\mathbb{R}^d$ . Let  $\mathfrak{D}$  be the set of all compactly supported probability measures on  $\mathbb{R}^d$  with characteristic functions in  $L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)$ . Suppose  $k$  satisfies Assumption 1 and  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  has a non-empty interior. Then  $k$  is a characteristic kernel to  $\mathfrak{D}$ .

$\psi(x), \Omega = \text{supp}(\Lambda)$	$\mathfrak{D}$	Characteristic	$\gamma_{\mathcal{F}}$	Reference
$\Omega = \mathbb{R}^d$	$\mathfrak{S}$	Yes	Metric	Theorem 7
$\text{supp}(\psi)$ is compact	$\mathfrak{S}$	Yes	Metric	Corollary 8
$\Omega \subsetneq \mathbb{R}^d$ has a non-empty interior	$\{P : \text{supp}(P) \text{ is compact, } \phi_P \in L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)\}$	Yes	Metric	Theorem 9
$\Omega \subsetneq \mathbb{R}^d$	$\mathfrak{S}$	No	Pseudometric	Theorem 7

Table 1:  $k$  satisfies Assumption 1 and is the Fourier transform of a finite nonnegative Borel measure  $\Lambda$  on  $\mathbb{R}^d$ .  $\mathfrak{S}$  is the set of all probability measures defined on  $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ .  $P$  represents a probability measure in  $\mathbb{R}^d$  and  $\phi_P$  is its characteristic function. If  $k$  is characteristic to  $\mathfrak{S}$ , then  $(\mathfrak{S}, \gamma_{\mathcal{F}})$  is a metric space, where  $\mathcal{F}$  is a unit ball in an RKHS  $(\mathcal{H}, k)$ .

The proof is given in §6.2.2 and the strategy is similar to that of Theorem 7, where the Fourier representation of  $\gamma_{\mathcal{F}}$  (see Lemma 13) is used to derive necessary and sufficient conditions for the existence of  $P \neq Q, P, Q \in \mathfrak{D}$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$  (see Lemma 17). We then show that if  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  has a non-empty interior, the conditions mentioned in Lemma 17 are violated, which means  $\nexists P \neq Q, P, Q \in \mathfrak{D}$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ , thereby proving the result.

Although, by Theorem 7, the kernels with  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  are not characteristic to  $\mathfrak{S}$ , Theorem 9 shows that there exists  $\mathfrak{D} \subsetneq \mathfrak{S}$  to which a subset of these kernels are characteristic. This type of result is not available for the methods studied in [SGSS07, FGSS08]. An example of a kernel that satisfies the conditions in Theorem 9 is the Sinc kernel,  $\psi(x) = \frac{\sin(\sigma x)}{x}$  which has  $\text{supp}(\Lambda) = [-\sigma, \sigma]$ . The condition that  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  has a non-empty interior is important for Theorem 9 to hold. If  $\text{supp}(\Lambda)$  has an empty interior (examples include periodic kernels), then one can construct  $P \neq Q, P, Q \in \mathfrak{D}$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ . See §6.2.2 for the related discussion and an example.

We have shown that the support of the Fourier spectrum of a positive definite translation-invariant kernel in  $\mathbb{R}^d$  characterizes the injective or non-injective behavior of  $\Pi$ . In particular,  $\text{supp}(\Lambda) = \mathbb{R}^d$  is the necessary and sufficient condition for the map  $\Pi$  to be injective on  $\mathfrak{S}$ , which answers question Q2 posed in §3. We also showed that kernels with  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  can be characteristic to some  $\mathfrak{D} \subsetneq \mathfrak{S}$  even though they are not characteristic to  $\mathfrak{S}$ , which in turn answers question Q1 in §3. A summary of these results is given in Table 1.

#### 4.2 A result on periodic kernels and discrete probability measures

**Proposition 10** *Let  $\mathcal{F}$  be a unit ball in an RKHS  $(\mathcal{H}, k)$  defined on  $\mathbb{R}^d$  where  $k$  satisfies Assumption 1. Let  $\mathfrak{D} = \{P : P = \sum_{n=1}^{\infty} \beta_n \delta_{x_n}, \sum_{n=1}^{\infty} \beta_n = 1, \beta_n \geq 0, \forall n\}$  be the set of probability measures defined on  $M' = \{x_1, x_2, \dots\} \subsetneq \mathbb{R}^d$ . Then  $\exists P \neq Q, P, Q \in \mathfrak{D}$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$  if the following conditions hold:*

(i)  $\psi$  is  $\tau$ -periodic<sup>5</sup> in  $\mathbb{R}^d$ , i.e.,  $\psi(x) = \psi(x + \eta \bullet \tau), \eta \in \mathbb{Z}^d, \tau \in \mathbb{R}_+^d$ ,

(ii)  $x_s - x_t = l_{st} \bullet \tau, l_{st} \in \mathbb{Z}^d, \forall s, t$ ,

where  $\bullet$  represents the Hadamard multiplication.

**Proof:** Let  $\psi$  be  $\tau$ -periodic in  $\mathbb{R}^d$  and  $x_s - x_t = l_{st} \bullet \tau, l_{st} \in \mathbb{Z}^d, \forall s, t$ . Consider  $P, Q \in \mathfrak{D}$  given by  $P = \sum_{n=1}^{\infty} \tilde{p}_n \delta_{x_n}$  and  $Q = \sum_{n=1}^{\infty} \tilde{q}_n \delta_{x_n}$  such that  $\tilde{p}_n, \tilde{q}_n \geq 0, \forall n; \sum_{n=1}^{\infty} \tilde{p}_n = 1, \sum_{n=1}^{\infty} \tilde{q}_n = 1$ . Then  $\gamma_{\mathcal{F}}(P, Q) = \|Pk - Qk\|_{\mathcal{H}} = \|\int_{\mathbb{R}^d} \psi(\cdot - x) d(P - Q)(x)\|_{\mathcal{H}} = \|\sum_{n=1}^{\infty} (\tilde{p}_n - \tilde{q}_n) \psi(\cdot - x_n)\|_{\mathcal{H}} = \|\sum_{n=1}^{\infty} (\tilde{p}_n - \tilde{q}_n) \psi(\cdot - x_1 - l_{n1} \bullet \tau)\|_{\mathcal{H}} = \|\psi(\cdot - x_1) \sum_{n=1}^{\infty} (\tilde{p}_n - \tilde{q}_n)\|_{\mathcal{H}} = 0$ . This holds for any  $P, Q \in \mathfrak{D}$ . ■

The converse of Proposition 10, if true, would make the result more interesting. This is because any non-periodic translation invariant kernel on  $\mathbb{R}^d$  would then be characteristic to the set of discrete probability measures on  $\mathbb{R}^d$ . In order to prove the converse, we would need to show that (i) and (ii) in Proposition 10 hold when  $\gamma_{\mathcal{F}}(P, Q) = 0$  for  $P \neq Q, P, Q \in \mathfrak{D}$ . However, this is not true as the trivial kernel yields  $\gamma_{\mathcal{F}}(P, Q) = 0$  for any  $P, Q \in \mathfrak{S}$  and not just  $P, Q \in \mathfrak{D}$ .

Let us consider  $\gamma_{\mathcal{F}}(P, Q) = 0$  for  $P, Q \in \mathfrak{D}$ . This is equivalent to  $\|\sum_{n=1}^{\infty} (\tilde{p}_n - \tilde{q}_n) \psi(\cdot - x_n)\|_{\mathcal{H}} = 0$ . Squaring on both sides and using the reproducing property of  $k$ , we get  $\sum_{s,t=1}^{\infty} \tilde{r}_s \tilde{r}_t \psi(x_s - x_t) = 0$  where  $\{\tilde{r}_n = \tilde{p}_n - \tilde{q}_n\}_{n=1}^{\infty}$  satisfy  $\sum_{s=1}^{\infty} \tilde{r}_s = 0$  and  $\{\tilde{r}_s\}_{s=1}^{\infty} \in [-1, 1]$ . So, to prove the converse, we need to characterize all  $\psi, \{\tilde{r}_n\}_{n=1}^{\infty}$  and  $\{x_n\}_{n=1}^{\infty}$  that satisfy  $\mathcal{R} = \{\sum_{s,t=1}^{\infty} \tilde{r}_s \tilde{r}_t \psi(x_s - x_t) = 0 : \sum_{s=1}^{\infty} \tilde{r}_s = 0, \{\tilde{r}_s\}_{s=1}^{\infty} \in [-1, 1]\}$ , which is not easy. However, choosing some  $\psi, \{\tilde{r}_n\}_{n=1}^{\infty}$  and  $\{x_n\}_{n=1}^{\infty}$  is easy, as shown in Proposition 10. Suppose there exists a class,  $\mathcal{K}$  of positive definite translation-invariant kernels in  $\mathbb{R}^d$  with  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  and a class,  $\mathfrak{E} \subset \mathfrak{D}$  of probability measures that jointly violate  $\mathcal{R}$ , then any  $k \in \mathcal{K}$  is characteristic to  $\mathfrak{E}$ .

<sup>5</sup>A  $\tau$ -periodic  $\psi$  in  $\mathbb{R}$  is the Fourier transform of  $\Lambda = \sum_{n=-\infty}^{\infty} \alpha_n \delta_{\frac{2\pi n}{\tau}}$ , where  $\delta_{\frac{2\pi n}{\tau}}$  is the Dirac measure at  $\frac{2\pi n}{\tau}, n \in \mathbb{Z}$  with  $\alpha_n \geq 0$  and  $\sum_{n=-\infty}^{\infty} \alpha_n < \infty$ . Thus,  $\text{supp}(\Lambda) = \{\frac{2\pi n}{\tau} : \alpha_n > 0, n \in \mathbb{Z}\} \subsetneq \mathbb{R}$ .  $\{\alpha_n\}_{n=-\infty}^{\infty}$  are called the Fourier series coefficients of  $\psi$ .

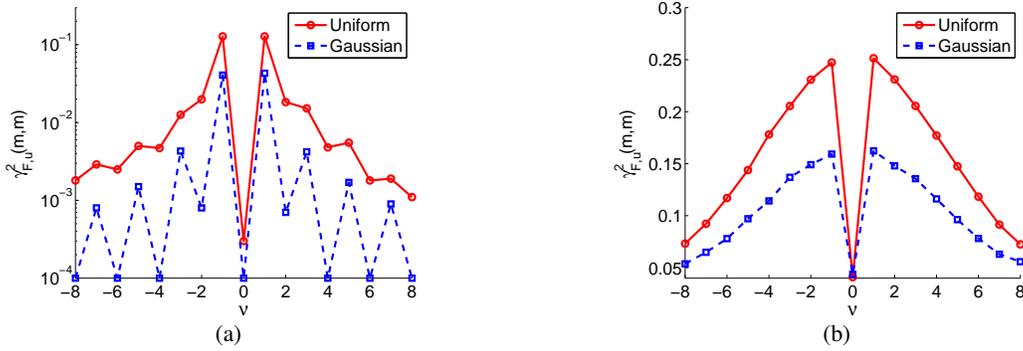


Figure 1: Behavior of the empirical estimate of  $\gamma_{\mathcal{F}}^2(P, Q)$  w.r.t.  $\nu$  for the (a)  $B_1$ -spline kernel and (b) Gaussian kernel.  $P$  is constructed from  $Q$  as defined in Eq. (4). “Uniform” corresponds to  $Q = \mathcal{U}[-1, 1]$  and “Gaussian” corresponds to  $Q = \mathcal{N}(0, 2)$ .  $m = 1000$  samples are generated from  $P$  and  $Q$  to estimate  $\gamma_{\mathcal{F}}^2(P, Q)$  through  $\gamma_{\mathcal{F},u}^2(m, m)$ . See Example 2 for details.

## 5 Dissimilar Distributions with Small Mean Discrepancy

So far, we have studied the behavior of  $\gamma_{\mathcal{F}}$  and have shown that it depends on the support of the spectrum of the kernel. As mentioned in §1, applications like homogeneity testing exploit the metric property of  $\gamma_{\mathcal{F}}$  to distinguish between probability distributions. Since the metric nature of  $\gamma_{\mathcal{F}}$  is guaranteed only for kernels with  $\text{supp}(\Lambda) = \mathbb{R}^d$ , tests based on other kernels can fail to distinguish between different probability distributions. However, in the following, we show that the characteristic kernels, while guaranteeing  $\gamma_{\mathcal{F}}$  to be a metric on  $\mathfrak{S}$ , may nonetheless have difficulty in distinguishing certain distributions on the basis of finite samples. Before proving the result, we motivate it through the following example.

**Example 2** Let  $P$  be defined as

$$p(x) = q(x) + \alpha q(x) \sin(\nu\pi x), \quad (4)$$

where  $q$  is a symmetric probability density function with  $\alpha \in \mathbb{R}$ ,  $\nu \in \mathbb{R} \setminus \{0\}$ . Consider a  $B_1$ -spline kernel on  $\mathbb{R}$  given by  $k(x, y) = \psi(x - y)$  where

$$\psi(x) = \begin{cases} 1 - |x|, & |x| \leq 1 \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

with its Fourier transform given by  $\Psi(\omega) = \frac{2\sqrt{2}}{\sqrt{\pi}} \frac{\sin^2 \frac{\omega}{2}}{\omega^2}$  (see footnote 10 for the definition of  $\Psi$ ). Since  $\psi$  is characteristic to  $\mathfrak{S}$ ,  $\gamma_{\mathcal{F}}(P, Q) > 0$  (see Theorem 7). However, it would be of interest to study the behavior of  $\gamma_{\mathcal{F}}(P, Q)$  as a function of  $\nu$ . We do this through an unbiased, consistent estimator<sup>6</sup> of  $\gamma_{\mathcal{F}}^2(P, Q)$  as proposed by Gretton et al. [GBR<sup>+</sup>07, Lemma 7].

<sup>6</sup>Starting from the expression for  $\gamma_{\mathcal{F}}$  in Eq. (2), we get  $\gamma_{\mathcal{F}}^2(P, Q) = \mathbb{E}_{X, X' \sim P} k(X, X') - 2\mathbb{E}_{X \sim P, Y \sim Q} k(X, Y) + \mathbb{E}_{Y, Y' \sim Q} k(Y, Y')$ , where  $X, X'$  are independent random variables with distribution  $P$  and  $Y, Y'$  are independent random variables with distribution  $Q$ . An unbiased empirical estimate of  $\gamma_{\mathcal{F}}^2$ , denoted as  $\gamma_{\mathcal{F},u}^2(m, m)$  is given by  $\gamma_{\mathcal{F},u}^2(m, m) = \frac{1}{m(m-1)} \sum_{l \neq j}^m h(Z_l, Z_j)$ , which is a one-sample  $U$ -statistic with  $h(Z_l, Z_j) := k(X_l, X_j) + k(Y_l, Y_j) - k(X_l, Y_j) - k(X_j, Y_l)$ , where  $Z_1, \dots, Z_m$  are  $m$  i.i.d. random variables with  $Z_j := (X_j, Y_j)$  (see [GBR<sup>+</sup>07, Lemma 7]).

Figure 1(a) shows the behavior of the empirical estimate of  $\gamma_{\mathcal{F}}^2(P, Q)$  as a function of  $\nu$  for  $q = \mathcal{U}[-1, 1]$  and  $q = \mathcal{N}(0, 2)$  using the  $B_1$ -spline kernel in Eq. (5). Since the Gaussian kernel,  $k(x, y) = e^{-(x-y)^2}$  is also a characteristic kernel, its effect on the behavior of  $\gamma_{\mathcal{F},u}^2(m, m)$  is shown in Figure 1(b) in comparison to that of the  $B_1$ -spline kernel.

From Figure 1, we observe two circumstances under which the mean discrepancy may be small. First,  $\gamma_{\mathcal{F},u}^2(m, m)$  decays with increasing  $|\nu|$ , and can be made as small as desired by choosing a sufficiently large  $|\nu|$ . Second, in Figure 1(a),  $\gamma_{\mathcal{F},u}^2(m, m)$  has troughs at  $\nu = \frac{\omega_0}{\pi}$  where  $\omega_0 = \{\omega : \Psi(\omega) = 0\}$ . Since  $\gamma_{\mathcal{F},u}^2(m, m)$  is a consistent estimate of  $\gamma_{\mathcal{F}}^2(P, Q)$ , one would expect similar behavior from  $\gamma_{\mathcal{F}}^2(P, Q)$ . This means that though the  $B_1$ -spline kernel is characteristic to  $\mathfrak{S}$ , in practice, it becomes harder to distinguish between  $P$  and  $Q$  with finite samples, when  $P$  is constructed as in Eq. (4) with  $\nu = \frac{\omega_0}{\pi}$ . In fact, one can observe from a straightforward spectral argument that the troughs in  $\gamma_{\mathcal{F}}^2(P, Q)$  can be made arbitrarily deep by widening  $q$ , when  $q$  is Gaussian.

For characteristic kernels, although  $\gamma_{\mathcal{F}}(P, Q) > 0$  when  $P \neq Q$ , Example 2 demonstrates that one can construct distributions such that  $\gamma_{\mathcal{F},u}^2(m, m)$  is indistinguishable from zero with high probability, for a given sample size  $m$ . Below, in Theorem 12, we investigate the decay mode of MMD for large  $|\nu|$  (see Example 2) by explicitly constructing  $P \neq Q$  such that  $|P\varphi_l - Q\varphi_l|$  is large for some large  $l$ , but  $\gamma_{\mathcal{F}}(P, Q)$  is arbitrarily small, making it hard to detect a non-zero value of the population MMD on the basis of a finite sample. Here,  $\varphi_l \in L^2(M)$  represents the bounded orthonormal eigenfunctions of a positive definite integral operator<sup>7</sup> associated with  $k$ .

Consider the formulation of MMD in Eq. (1). The construction of  $P$  for a given  $Q$  such that  $\gamma_{\mathcal{F}}(P, Q)$  is small, though not zero, can be intuitively seen by re-writing Eq. (1) as

$$\gamma_{\mathcal{F}}(P, Q) = \sup_{f \in \mathcal{H}} \frac{|Pf - Qf|}{\|f\|_{\mathcal{H}}}. \quad (6)$$

<sup>7</sup>See [SS02, Theorem 2.10] for definition of positive definite integral operator and its corresponding eigenfunctions.

When  $P \neq Q$ ,  $|Pf - Qf|$  can be large for some  $f \in \mathcal{H}$ . However,  $\gamma_{\mathcal{F}}(P, Q)$  can be made small by selecting  $P$  such that the maximization of  $\frac{|Pf - Qf|}{\|f\|_{\mathcal{H}}}$  over  $\mathcal{H}$  requires an  $f$  with large  $\|f\|_{\mathcal{H}}$ . More specifically, higher order eigenfunctions of the kernel ( $\varphi_l$  for large  $l$ ) have large RKHS norms, and so if they are prominent in  $P, Q$  (i.e., highly non-smooth distributions), one can expect  $\gamma_{\mathcal{F}}(P, Q)$  to be small even when there exists an  $l$  for which  $|P\varphi_l - Q\varphi_l|$  is large. To this end, we need the following lemma, which we quote from [GSB<sup>+</sup>04, Lemma 6].

**Lemma 11 ([GSB<sup>+</sup>04])** *Let  $\mathcal{F}$  be a unit ball in an RKHS  $(\mathcal{H}, k)$  defined on compact  $M$ . Let  $\varphi_l \in L^2(M)$  be orthonormal eigenfunctions (assumed to be absolutely bounded), and  $\lambda_l$  be the corresponding eigenvalues (arranged in a decreasing order for increasing  $l$ ) of a positive definite integral operator associated with  $k$ . Assume  $\lambda_l^{-1}$  increases superlinearly with  $l$ . Then for  $f \in \mathcal{F}$  where  $f(x) := \sum_{j=1}^{\infty} \tilde{f}_j \varphi_j(x)$ , we have  $\{\tilde{f}_j\}_{j=1}^{\infty} \in \ell_1$  and for every  $\epsilon > 0$ ,  $\exists l_0 \in \mathbb{N}$  such that  $|\tilde{f}_l| < \epsilon$  if  $l > l_0$ .*

**Theorem 12 ( $P \neq Q$  can give small MMD)** *Assume the conditions in Lemma 11 hold. Then there exists a probability distribution  $P \neq Q$  defined on  $M$  for which  $|P\varphi_l - Q\varphi_l| > \beta - \epsilon$  for some non-trivial  $\beta$  and arbitrarily small  $\epsilon > 0$ , yet for which  $\gamma_{\mathcal{F}}(P, Q) < \eta$  for an arbitrarily small  $\eta > 0$ .*

**Proof:** Let us construct  $p(x) = q(x) + \alpha_l e(x) + \beta \varphi_l(x)$  where  $e(x) = \mathbb{1}_M(x)$ . For  $P$  to be a probability distribution, the following conditions need to be satisfied:

$$\int_M [\alpha_l e(x) + \beta \varphi_l(x)] dx = 0, \quad (7)$$

$$\min_{x \in M} [q(x) + \alpha_l e(x) + \beta \varphi_l(x)] \geq 0. \quad (8)$$

Expanding  $e(x)$  and  $f(x)$  in the orthonormal basis  $\{\varphi_l\}_{l=1}^{\infty}$ , we get  $e(x) = \sum_{l=1}^{\infty} \tilde{e}_l \varphi_l(x)$  and  $f(x) = \sum_{l=1}^{\infty} \tilde{f}_l \varphi_l(x)$ , where  $\tilde{e}_l := \langle e, \varphi_l \rangle_{L^2(M)}$  and  $\tilde{f}_l := \langle f, \varphi_l \rangle_{L^2(M)}$ . Therefore,  $Pf - Qf = \int_M f(x) [\alpha_l e(x) + \beta \varphi_l(x)] dx$  reduces to

$$Pf - Qf = \alpha_l \sum_{j=1}^{\infty} \tilde{e}_j \tilde{f}_j + \beta \tilde{f}_l, \quad (9)$$

where we used the fact that<sup>8</sup>  $\langle \varphi_j, \varphi_t \rangle_{L^2(M)} = \delta_{jt}$ . Rewriting Eq. (7) and substituting for  $e(x)$  gives  $\int_M [\alpha_l e(x) + \beta \varphi_l(x)] dx = \int_M e(x) [\alpha_l e(x) + \beta \varphi_l(x)] dx = \alpha_l \sum_{j=1}^{\infty} \tilde{e}_j^2 + \beta \tilde{e}_l = 0$ , which implies

$$\alpha_l = -\frac{\beta \tilde{e}_l}{\sum_{j=1}^{\infty} \tilde{e}_j^2}. \quad (10)$$

Now, let us consider  $P\varphi_t - Q\varphi_t = \alpha_l \tilde{e}_t + \beta \delta_{tl}$ . Substituting for  $\alpha_l$  gives

$$P\varphi_t - Q\varphi_t = \beta \delta_{tl} - \beta \frac{\tilde{e}_t \tilde{e}_l}{\sum_{j=1}^{\infty} \tilde{e}_j^2} = \beta \delta_{tl} - \beta \tau_{tl}, \quad (11)$$

where  $\tau_{tl} := \frac{\tilde{e}_t \tilde{e}_l}{\sum_{j=1}^{\infty} \tilde{e}_j^2}$ . By Lemma 11,  $\{\tilde{e}_l\}_{l=1}^{\infty} \in \ell_1 \Rightarrow \sum_{j=1}^{\infty} \tilde{e}_j^2 < \infty$ , and choosing large enough  $l$  gives  $|\tau_{tl}| <$

<sup>8</sup>Here  $\delta$  is used in the Kronecker sense.

$\epsilon, \forall t$ , for any arbitrary  $\epsilon > 0$ . Therefore,  $|P\varphi_t - Q\varphi_t| > \beta - \epsilon$  for  $t = l$  and  $|P\varphi_t - Q\varphi_t| < \epsilon$  for  $t \neq l$ . By appealing to Lemma 1, we therefore establish that  $P \neq Q$ . In the following we prove that  $\gamma_{\mathcal{F}}(P, Q)$  can be arbitrarily small, though non-zero.

Recall that  $\gamma_{\mathcal{F}}(P, Q) = \sup_{\|f\|_{\mathcal{H}} \leq 1} |Pf - Qf|$ . Substituting for  $\alpha_l$  in Eq. (9), we have

$$\gamma_{\mathcal{F}}(P, Q) = \sup \left\{ \beta \sum_{j=1}^{\infty} \nu_{jl} \tilde{f}_j : \sum_{j=1}^{\infty} \frac{\tilde{f}_j^2}{\lambda_j} \leq 1 \right\}, \quad (12)$$

where we used the definition of RKHS norm as  $\|f\|_{\mathcal{H}} := \sum_{j=1}^{\infty} \frac{\tilde{f}_j^2}{\lambda_j}$  and  $\nu_{jl} := \delta_{jl} - \tau_{jl}$ . Eq. (12) is a convex quadratic program in  $\{\tilde{f}_j\}_{j=1}^{\infty}$ . Solving the Lagrangian yields  $\tilde{f}_j = \frac{\nu_{jl} \lambda_j}{\sqrt{\sum_{j=1}^{\infty} \nu_{jl}^2 \lambda_j}}$ . Therefore,  $\gamma_{\mathcal{F}}(P, Q) = \beta \sqrt{\sum_{j=1}^{\infty} \nu_{jl}^2 \lambda_j} = \beta \sqrt{\lambda_l - 2\tau_{ll} \lambda_l + \sum_{j=1}^{\infty} \tau_{jl}^2 \lambda_j} \rightarrow 0$  as  $l \rightarrow \infty$  because (i) by choosing sufficiently large  $l$ ,  $|\tau_{jl}| < \epsilon, \forall j$ , for any arbitrary  $\epsilon > 0$ , (ii)  $\lambda_l \rightarrow 0$  as  $l \rightarrow \infty$  [SS02, Theorem 2.10]. ■

## 6 Proofs of the Main Theorems

In this section, we prove the main theorems in Section 4.

### 6.1 Preliminary lemmas

Using the Fourier characterization of  $\psi$  given by Eq. (3), under Assumption 1, we derive the following result that provides the Fourier representation of MMD. This result requires tools from *distribution theory* related to the Fourier transforms of distributions.<sup>9</sup> We refer the reader to [Rud91, Chapters 6,7] for the detailed treatment of distribution theory. Another good and basic reference on distribution theory is [Str03].

**Lemma 13 (Fourier representation of MMD)** *Let  $\mathcal{F}$  be a unit ball in an RKHS  $(\mathcal{H}, k)$  defined on  $\mathbb{R}^d$  with  $k$  satisfying Assumption 1. Let  $\phi_P$  and  $\phi_Q$  be the characteristic functions of probability measures  $P$  and  $Q$  defined on  $\mathbb{R}^d$ . Then*

$$\gamma_{\mathcal{F}}(P, Q) = \|[(\bar{\phi}_P - \bar{\phi}_Q)\Lambda]^{\vee}\|_{\mathcal{H}}, \quad (13)$$

where  $-$  represents complex conjugation,  $\vee$  represents the inverse Fourier transform and  $\Lambda$  represents the finite non-negative Borel measure on  $\mathbb{R}^d$  as defined in Eq. (3).  $(\bar{\phi}_P - \bar{\phi}_Q)\Lambda$  represents a finite Borel measure defined by Eq. (26).

**Proof:** From Theorem 3, we have  $\gamma_{\mathcal{F}}(P, Q) = \|Pk - Qk\|_{\mathcal{H}}$ . Consider  $Pk = \int_{\mathbb{R}^d} k(\cdot, x) dP(x) = \int_{\mathbb{R}^d} \psi(\cdot - x) dP(x)$ . By Eq. (23),  $\int_{\mathbb{R}^d} \psi(\cdot - x) dP(x)$  represents the convolution of  $\psi$  and  $P$ , denoted as  $\psi * P$ . By appealing to the convolution theorem (Theorem 22), we have  $(\psi * P)^{\wedge} = \hat{P}\Lambda$ , where  $\hat{P}(\omega) =$

<sup>9</sup>Here, the term *distribution* should not be confused with probability distributions. In short, distributions refer to generalized functions which cannot be treated as functions in the Lebesgue sense. Classical examples of distributions are the Dirac-delta function and Heaviside's function, for which derivatives and Fourier transforms do not exist in the usual sense.

$\int_{\mathbb{R}^d} e^{-i\omega^T x} dP(x), \forall \omega \in \mathbb{R}^d$  (by Lemma 20). Note that  $\hat{P} = \bar{\phi}_P$ . Therefore,  $\gamma_{\mathcal{F}}(P, Q) = \|\psi * P - \psi * Q\|_{\mathcal{H}} = \|[(\bar{\phi}_P \Lambda)^\vee - (\bar{\phi}_Q \Lambda)^\vee]\|_{\mathcal{H}}$ . Using the linearity of the Fourier inverse, we get the desired result. ■

**Remark 14** (a) If  $\Psi$  is the distributional derivative<sup>10</sup> of  $\Lambda$ , then Eq. (13) can also be written as

$$\gamma_{\mathcal{F}}(P, Q) = \|[(\bar{\phi}_P - \bar{\phi}_Q)\Psi]^\vee\|_{\mathcal{H}}, \quad (14)$$

where the term inside the RKHS norm is the Fourier inverse of a tempered distribution.

(b) By Assumption 1,  $\psi$  is real-valued and symmetric in  $\mathbb{R}^d$ . Therefore, by (ii) in Lemma 20,  $\Lambda$  and  $\Psi$  are real-valued, symmetric tempered distributions.

The representation of MMD in terms of the kernel spectrum as in Eq. (13) will be central to deriving our main theorems. It is easy to see that characteristic kernels can be described indirectly by deriving conditions for the existence of  $P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ . Using the Fourier representation of  $\gamma_{\mathcal{F}}$ , the following result provides necessary and sufficient conditions for the existence of  $P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ .

**Lemma 15** Let  $\mathcal{F}$  be a unit ball in an RKHS  $(\mathcal{H}, k)$  defined on  $\mathbb{R}^d$ , and let  $P, Q$  be probability distributions on  $\mathbb{R}^d$  such that  $P \neq Q$ . Suppose that  $k$  satisfies Assumption 1 and  $\text{supp}(\Lambda) \subset \mathbb{R}^d$ . Then  $\gamma_{\mathcal{F}}(P, Q) = 0$  if and only if there exists  $\theta \in \mathcal{S}'_d$  that satisfies the following conditions:

- (i)  $p - q = \check{\theta}$ ,
- (ii)  $\theta \Lambda = 0$ ,

where  $p$  and  $q$  represent the distributional derivatives of  $P$  and  $Q$  respectively, and  $\theta \Lambda$  represents a finite Borel measure defined by Eq. (26).

**Proof:** The proof follows directly from the formulation of  $\gamma_{\mathcal{F}}$  in Eq. (13).

( $\Rightarrow$ ) Let  $\theta \in \mathcal{S}'_d$  satisfy (i) and (ii). Since  $\theta \in \mathcal{S}'_d$ , we have  $\theta = \hat{\theta} = (p - q)^\wedge = \hat{p} - \hat{q} = \bar{\phi}_P - \bar{\phi}_Q$ . Therefore, by (ii), we have  $\gamma_{\mathcal{F}}(P, Q) = \|[(\bar{\phi}_P - \bar{\phi}_Q)\Lambda]^\vee\|_{\mathcal{H}} = \|[\theta \Lambda]^\vee\|_{\mathcal{H}} = 0$ .

( $\Leftarrow$ ) Let  $\gamma_{\mathcal{F}}(P, Q) = \|[(\bar{\phi}_P - \bar{\phi}_Q)\Lambda]^\vee\|_{\mathcal{H}} = 0$ , which implies  $[(\bar{\phi}_P - \bar{\phi}_Q)\Lambda]^\vee = 0$ . Since  $(\bar{\phi}_P - \bar{\phi}_Q)\Lambda$  is a finite Borel measure as defined by Eq. (26), it is therefore a tempered distribution and so  $(\bar{\phi}_P - \bar{\phi}_Q)\Lambda = [[(\bar{\phi}_P - \bar{\phi}_Q)\Lambda]^\vee]^\wedge = 0$ . Let  $\theta := \bar{\phi}_P - \bar{\phi}_Q$ . Clearly  $\theta \in \mathcal{S}'_d$  as by Lemma 20,  $\bar{\phi}_P, \bar{\phi}_Q \in \mathcal{S}'_d$ . So,  $p - q = (\bar{\phi}_P)^\vee - (\bar{\phi}_Q)^\vee = (\bar{\phi}_P - \bar{\phi}_Q)^\vee = \check{\theta}$ . ■

$\theta = 0$  trivially satisfies (ii) in Lemma 15. However, it violates our assumption of  $P \neq Q$  when it is used in condition

<sup>10</sup>If  $\Lambda$  is absolutely continuous w.r.t. the Lebesgue measure, then  $\Psi$  represents the Radon-Nikodym derivative of  $\Lambda$  w.r.t. the Lebesgue measure. In such a case,  $\psi$  is the Fourier transform of  $\Psi$  in the usual sense; i.e.,  $\psi(x) = \int_{\mathbb{R}^d} e^{-ix^T \omega} \Psi(\omega) dm_d(\omega)$ . On the other hand, if  $\Psi$  is the distributional derivative of  $\Lambda$ , then  $\Psi$  is a symbolic representation of the derivative of  $\Lambda$  and will make sense only under the integral sign.

(i). If we relax this assumption, then the result is trivial as  $P = Q \Rightarrow \gamma_{\mathcal{F}}(P, Q) = 0$ . For the results we derive later, it is important to understand the properties of  $\theta$ , which we present in the following proposition.

**Proposition 16 (Properties of  $\theta$ )**  $\theta$  in Lemma 15 satisfies the following properties:

- (a)  $\theta$  is a conjugate symmetric, bounded and uniformly continuous function on  $\mathbb{R}^d$ .
- (b)  $\theta(0) = 0$ .
- (c)  $\text{supp}(\theta) \subset \overline{\mathbb{R}^d \setminus \Omega}$  where  $\Omega := \text{supp}(\Lambda)$ . In addition, if  $\Omega = \{a_1, a_2, \dots\}$ , then  $\theta(a_j) = 0, \forall a_j \in \Omega$ .

**Proof:** (a) From Lemma 15, we have  $\theta = \bar{\phi}_P - \bar{\phi}_Q$ . Therefore, the result in (a) follows from Lemma 20, which shows that  $\bar{\phi}_P, \bar{\phi}_Q$  are conjugate symmetric, bounded, and uniformly continuous functions on  $\mathbb{R}^d$ .

(b) By Lemma 20,  $\bar{\phi}_P(0) = \bar{\phi}_Q(0) = 1$ . Therefore,  $\theta(0) = \bar{\phi}_P(0) - \bar{\phi}_Q(0) = 0$ .

(c) Let  $W := \{x \in \mathbb{R}^d \mid \theta(x) \neq 0\}$ . It suffices to show that  $W \subset \overline{\mathbb{R}^d \setminus \Omega}$ . Suppose  $W$  is not contained in  $\overline{\mathbb{R}^d \setminus \Omega}$ . Then there is a non-empty open subset  $U$  such that  $U \subset W \cap (\Omega \cup \partial\Omega)$ . Fix further a non-empty open subset  $V$  with  $\bar{V} \subset U$ . Since  $V \subset \Omega$ , there is  $\varphi \in \mathcal{D}_d(V)$  with  $\Lambda(\varphi) \neq 0$ . Take  $h \in \mathcal{D}_d(U)$  such that  $h = 1$  on  $\bar{V}$ , and define a continuous function  $\varrho = \frac{h\varphi}{\theta}$  on  $\mathbb{R}^d$ , which is well-defined from  $\text{supp}(h) \subset U$  and  $\theta \neq 0$  on  $U$ . By (ii) of Lemma 15,  $\theta \Lambda = 0$ , where  $\theta \Lambda$  is a finite Borel measure on  $\mathbb{R}^d$  as defined by Eq. (26). Therefore,

$$\int_{\mathbb{R}^d} \varrho(x)\theta(x) d\Lambda(x) = 0. \quad (15)$$

The left hand side of Eq. (15) simplifies to

$$\begin{aligned} \int_{\mathbb{R}^d} \varrho(x)\theta(x) d\Lambda(x) &= \int_U \frac{h(x)\varphi(x)}{\theta(x)} \theta(x) d\Lambda(x) \\ &= \int_U \varphi(x) d\Lambda(x) = \Lambda(\varphi) \neq 0, \end{aligned}$$

resulting in a contradiction. So,  $\text{supp}(\theta) \subset \overline{\mathbb{R}^d \setminus \Omega}$ .

If  $\Omega = \{a_1, a_2, \dots\}$ , then  $\Lambda = \sum_{a_j \in \Omega} \beta_j \delta_{a_j}, \beta_j > 0$  and  $\sum_j \beta_j < \infty$ .  $\theta \Lambda = 0$  implies  $\int_{\mathbb{R}^d} \chi(x)\theta(x) d\Lambda(x) = \sum_j \beta_j \chi(a_j)\theta(a_j) = 0$  for any continuous function  $\chi$  in  $\mathbb{R}^d$ . This implies  $\theta(a_j) = 0, \forall a_j \in \Omega$ . ■

Lemma 15 provides conditions under which  $\gamma_{\mathcal{F}}(P, Q) = 0$  when  $P \neq Q$ . It shows that the kernel  $k$  cannot distinguish between  $P$  and  $Q$  if  $P$  is related to  $Q$  by condition (i). Condition (ii) in Lemma 15 says that  $\theta$  has to be chosen such that its support is disjoint with that of the kernel spectrum. This is what is precisely captured by (c) in Proposition 16. So, for a given  $Q$ , one can construct  $P$  such that  $P \neq Q$  and  $\gamma_{\mathcal{F}}(P, Q) = 0$  by choosing  $\theta$  that satisfies the properties in Proposition 16. However,  $P$  should be a positive distribution so that it corresponds to a positive measure.<sup>11</sup> Therefore,

<sup>11</sup>A positive distribution is defined to be as the one that takes nonnegative values on nonnegative test functions. So,  $D \in \mathcal{D}'_d(M)$

$\theta$  should also be such that  $q + \check{\theta}$  is a positive distribution. Imposing such a constraint on  $\theta$  is not straightforward, and therefore Lemma 15 does not provide a procedure to construct  $P \neq Q$  given  $Q$ . However, by imposing some conditions on  $P$  and  $Q$ , we obtain the following result wherein the conditions on  $\theta$  can be explicitly specified, yielding a procedure to construct  $P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ .

**Lemma 17** *Let  $\mathcal{F}$  be a unit ball in an RKHS  $(\mathcal{H}, k)$  defined on  $\mathbb{R}^d$ . Let  $\mathfrak{D}$  be the set of probability measures on  $\mathbb{R}^d$  with characteristic functions either absolutely integrable or square integrable, i.e., for any  $P \in \mathfrak{D}$ ,  $\phi_P \in L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)$ . Suppose that  $k$  satisfies Assumption 1 and  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$ . Then for any  $Q \in \mathfrak{D}$ ,  $\exists P \neq Q$ ,  $P \in \mathfrak{D}$  given by*

$$p = q + \check{\theta} \quad (16)$$

such that  $\gamma_{\mathcal{F}}(P, Q) = 0$  if and only if there exists a non-zero function  $\theta : \mathbb{R}^d \rightarrow \mathbb{C}$  that satisfies the following conditions:

- (i)  $\theta \in (L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)) \cap C_b(\mathbb{R}^d)$  is conjugate symmetric,
- (ii)  $\check{\theta} \in L^1(\mathbb{R}^d) \cap (L^2(\mathbb{R}^d) \cup C_b(\mathbb{R}^d))$ ,
- (iii)  $\theta \Lambda = 0$ ,
- (iv)  $\theta(0) = 0$ ,
- (v)  $\inf_{x \in \mathbb{R}^d} \{\check{\theta}(x) + q(x)\} \geq 0$ .

**Proof:** ( $\Rightarrow$ ) Suppose there exists a non-zero function  $\theta$  satisfying (i) – (v). We need to show that  $p = q + \check{\theta}$  is in  $\mathfrak{D}$  for  $q \in \mathfrak{D}$  and  $\gamma_{\mathcal{F}}(P, Q) = 0$ .

For any  $Q \in \mathfrak{D}$ ,  $\phi_Q \in (L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)) \cap C_b(\mathbb{R}^d)$ . When  $\phi_Q \in L^1(\mathbb{R}^d) \cap C_b(\mathbb{R}^d)$ , the Riemann-Lebesgue lemma (Lemma 23) implies that  $q = [\overline{\phi_Q}]^\vee \in L^1(\mathbb{R}^d) \cap C_b(\mathbb{R}^d)$ . When  $\phi_Q \in L^2(\mathbb{R}^d) \cap C_b(\mathbb{R}^d)$ , the Fourier transform in the  $L^2$  sense<sup>12</sup> implies that  $q = [\overline{\phi_Q}]^\vee \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ . Therefore,  $q \in L^1(\mathbb{R}^d) \cap (L^2(\mathbb{R}^d) \cup C_b(\mathbb{R}^d))$ . Define  $p := q + \check{\theta}$ . Clearly  $p \in L^1(\mathbb{R}^d) \cap (L^2(\mathbb{R}^d) \cup C_b(\mathbb{R}^d))$ . In addition,  $\overline{\phi_P} = \hat{p} = \hat{q} + \hat{\check{\theta}} = \overline{\phi_Q} + \theta \in (L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)) \cap C_b(\mathbb{R}^d)$ . Since  $\theta$  is conjugate symmetric,  $\check{\theta}$  is real valued and so is  $p$ . Consider  $\int_{\mathbb{R}^d} p(x) dx = \int_{\mathbb{R}^d} q(x) dx + \int_{\mathbb{R}^d} \check{\theta}(x) dx = 1 + \theta(0) = 1$ . (v) implies that  $p$  is non-negative. Therefore,  $P$  represents a probability measure such that  $P \neq Q$  and  $P \in \mathfrak{D}$ . Since  $P, Q$  are probability measures,  $\gamma_{\mathcal{F}}(P, Q)$  is computed as  $\gamma_{\mathcal{F}}(P, Q) = \|[(\overline{\phi_P} - \overline{\phi_Q})\Lambda]^\vee\|_{\mathcal{H}} = \|[\theta\Lambda]^\vee\|_{\mathcal{H}} = 0$ .

( $\Leftarrow$ ) Suppose that  $P, Q \in \mathfrak{D}$  and  $p = q + \check{\theta}$  gives  $\gamma_{\mathcal{F}}(P, Q) = 0$ . We need to show that  $\theta$  satisfies (i) – (v).

is a positive distribution if  $D(\varphi) \geq 0$  for  $0 \leq \varphi \in \mathcal{D}_d(M)$ . If  $\mu$  is a positive measure that is locally finite, then  $D_\mu(\varphi) = \int_M \varphi d\mu$  defines a positive distribution. Conversely, every positive distribution comes from a locally finite positive measure [Str03, §6.4].

<sup>12</sup>If  $f \in L^2(\mathbb{R}^d)$ , the Fourier transform  $F[f] := \hat{f}$  of  $f$  is defined to be the limit, in the  $L^2$ -norm, of the sequence  $\{\hat{f}_n\}$  of Fourier transforms of any sequence  $\{f_n\}$  of functions belonging to  $\mathcal{S}_d$ , such that  $f_n$  converges in the  $L^2$ -norm to the given function  $f \in L^2(\mathbb{R}^d)$ , as  $n \rightarrow \infty$ . The function  $\hat{f}$  is defined almost everywhere on  $\mathbb{R}^d$  and belongs to  $L^2(\mathbb{R}^d)$ . Thus,  $F$  is a linear operator, mapping  $L^2(\mathbb{R}^d)$  into  $L^2(\mathbb{R}^d)$ .

$P, Q \in \mathfrak{D}$  implies  $\phi_P, \phi_Q \in (L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)) \cap C_b(\mathbb{R}^d)$  and  $p, q \in L^1(\mathbb{R}^d) \cap (L^2(\mathbb{R}^d) \cup C_b(\mathbb{R}^d))$ . Therefore,  $\theta = \overline{\phi_P} - \overline{\phi_Q} \in (L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)) \cap C_b(\mathbb{R}^d)$  and  $\check{\theta} = p - q \in L^1(\mathbb{R}^d) \cap (L^2(\mathbb{R}^d) \cup C_b(\mathbb{R}^d))$ . By Lemma 20,  $\phi_P$  and  $\phi_Q$  are conjugate symmetric and so is  $\theta$ . Therefore  $\theta$  satisfies (i) and  $\check{\theta}$  satisfies (ii).  $\theta$  satisfies (iv) as  $\theta(0) = \int_{\mathbb{R}^d} \check{\theta}(x) dx = \int_{\mathbb{R}^d} (p(x) - q(x)) dx = 0$ . Non-negativity of  $p$  yields (v).  $\gamma_{\mathcal{F}}(P, Q) = 0$  implies (iii), with a proof similar to that of Lemma 15. ■

**Remark 18** *Conditions (iii) and (iv) in Lemma 17 are the same as those of Proposition 16. Conditions (i) and (ii) are required to satisfy our assumption  $P, Q \in \mathfrak{D}$  and Eq. (16). Condition (v) ensures that  $P$  is a positive measure, which was the condition difficult to impose in Lemma 15.*

In the above result, we restricted ourselves to probability measures  $P$  with characteristic functions  $\phi_P$  in  $L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)$ . This ensures that the inverse Fourier transform of  $\phi_P$  exists in the  $L^1$  or  $L^2$  sense. Without this assumption,  $\phi_P$  is not guaranteed to have a Fourier transform in the  $L^1$  or  $L^2$  sense, and therefore has to be treated as a tempered distribution for the purpose of computing its Fourier transform. This implies  $\theta = \overline{\phi_P} - \overline{\phi_Q}$  has to be treated as a tempered distribution, which is the setting in Lemma 15. Since we wanted to avoid dealing with distributions where the required positivity constraint is difficult to impose, we restricted ourselves to  $\mathfrak{D}$ .<sup>13</sup> Though this result explicitly captures the conditions on  $\theta$ , it is a very restricted result as it only deals with continuous (a.e.) probability measures. However, we use this result in Lemma 19 to construct  $P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ .

Lemmas 15 and 17 are the main results that provide conditions for the existence of  $P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ . This means that if there exists a  $\theta$  satisfying these conditions, then  $k$  cannot distinguish between  $P$  and  $Q$  where  $P$  is defined as in Eq. (16). Thus, the existence (resp. non-existence) of  $\theta$  results in a non-injective (resp. injective) map  $\Pi$ . It is clear from Lemmas 15 and 17 that the dependence of  $\gamma_{\mathcal{F}}$  on the kernel appears in the form of the support of the kernel spectrum. Therefore, two scenarios exist: (a)  $\text{supp}(\Lambda) = \mathbb{R}^d$  and (b)  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$ . The case of  $\text{supp}(\Lambda) = \mathbb{R}^d$  is addressed by Theorem 7 while that of  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  is addressed by Theorem 9. Using Lemma 17, the following result proves the existence of  $P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$  while using a kernel with  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$ .

**Lemma 19** *Let  $\mathcal{F}$  be a unit ball in an RKHS  $(\mathcal{H}, k)$  defined on  $\mathbb{R}^d$ . Let  $\mathfrak{D}$  be the set of all non-compactly supported probability measures on  $\mathbb{R}^d$  with characteristic functions in  $L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)$ . Suppose  $k$  satisfies Assumption 1 and  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$ . Then  $\exists P \neq Q$ ,  $P, Q \in \mathfrak{D}$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ .*

<sup>13</sup>Choosing  $\mathfrak{D}$  to be the set of all probability measures with characteristic functions in  $L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)$  is the best possible restriction that avoids treating  $\theta$  as a tempered distribution. The classical Fourier transforms on  $\mathbb{R}^d$  are defined for functions in  $L^p(\mathbb{R}^d)$ ,  $1 < p \leq 2$ . For  $p > 2$ , the only reasonable way to define Fourier transforms on  $L^p(\mathbb{R}^d)$  is through distribution theory.

**Proof:** We claim that there exists a non-zero function,  $\theta$  satisfying (i) – (v) in Lemma 17 which therefore proves the result. Consider the following function,  $g_{\beta, \omega_0} \in C^\infty(\mathbb{R}^d)$  supported in  $[\omega_0 - \beta, \omega_0 + \beta]$ ,

$$g_{\beta, \omega_0}(\omega) = \prod_{j=1}^d \mathbb{1}_{[-\beta_j, \beta_j]}(\omega_j - \omega_{0,j}) e^{-\frac{\beta_j^2}{\beta_j^2 - (\omega_j - \omega_{0,j})^2}}, \quad (17)$$

where  $\omega = (\omega_1, \dots, \omega_d)$ ,  $\omega_0 = (\omega_{0,1}, \dots, \omega_{0,d})$  and  $\beta = (\beta_1, \dots, \beta_d)$ . Since  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$ , there exists an open set  $U \subset \mathbb{R}^d$  on which  $\Lambda$  is null. So, there exists  $\beta$  and  $\omega_0 \neq 0$  with  $\omega_0 > \beta$  such that  $[\omega_0 - \beta, \omega_0 + \beta] \subset U$ . Choose  $\theta = \alpha(g_{\beta, \omega_0} + g_{\beta, -\omega_0})$ ,  $\alpha \in \mathbb{R} \setminus \{0\}$ , which implies  $\text{supp}(\theta) = [-\omega_0 - \beta, -\omega_0 + \beta] \cup [\omega_0 - \beta, \omega_0 + \beta]$  is compact. Therefore, by the Paley-Wiener theorem (Theorem 24),  $\check{\theta}$  is a rapidly decaying function, i.e.,  $\check{\theta} \in \mathcal{S}_d$ . Since  $\theta(0) = 0$  (by construction),  $\check{\theta}$  will take negative values. However,  $\check{\theta}$  decays faster than some  $Q \in \mathcal{D}$  of the form  $q(x) \propto \prod_{j=1}^d \frac{1}{1+|x_j|^{l+\epsilon}}$ ,  $\forall l \in \mathbb{N}$ ,  $\epsilon > 0$  where  $x = (x_1, \dots, x_d)$ . It can be verified that  $\theta$  satisfies conditions (i) – (v) in Lemma 17. We conclude, there exists a non-zero  $\theta$  as claimed earlier, which completes the proof. ■

The above result shows that  $k$  with  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  is not characteristic to the class of non-compactly supported probability measures on  $\mathbb{R}^d$  with characteristic functions in either  $L^1(\mathbb{R}^d)$  or  $L^2(\mathbb{R}^d)$ .

## 6.2 Main theorems: Proofs

We are now in a position to prove Theorems 7 and 9.

### 6.2.1 Proof of Theorem 7

( $\Rightarrow$ ) Let  $\text{supp}(\Lambda) = \mathbb{R}^d$ .  $k$  is a characteristic kernel to  $\mathfrak{S}$  if  $\gamma_{\mathcal{F}}(P, Q) = 0 \Leftrightarrow P = Q$  for  $P, Q \in \mathfrak{S}$ . We only need to show the implication  $\gamma_{\mathcal{F}}(P, Q) = 0 \Rightarrow P = Q$  as the other direction is trivial.

Assume that  $\exists P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ . Then by Lemma 15,  $\exists \theta$  satisfying (i) and (ii) given in Lemma 15. By Proposition 16,  $\theta\Lambda = 0$  implies  $\text{supp}(\theta) \subset \mathbb{R}^d \setminus \text{supp}(\Lambda)$ . Since  $\text{supp}(\Lambda) = \mathbb{R}^d$  and  $\theta$  is a uniformly continuous function in  $\mathbb{R}^d$ , we have  $\text{supp}(\theta) = \emptyset$  which means  $\theta = 0$  a.e. Therefore, by (i) of Theorem 15, we have  $P = Q$ , leading to a contradiction. Thus,  $\nexists P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ .

( $\Leftarrow$ ) Suppose  $k$  is characteristic to  $\mathfrak{S}$ . We then need to show that  $\text{supp}(\Lambda) = \mathbb{R}^d$ . This is equivalent to proving that  $k$  is not characteristic to  $\mathfrak{S}$  when  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$ . Let  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$ . Choose  $\mathcal{D} \subsetneq \mathfrak{S}$  as the set of all non-compactly supported probability measures on  $\mathbb{R}^d$  with characteristic functions in  $L^1(\mathbb{R}^d) \cup L^2(\mathbb{R}^d)$ . By Lemma 19,  $\exists P \neq Q, P, Q \in \mathcal{D} \subsetneq \mathfrak{S}$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ . Therefore,  $k$  is not characteristic to  $\mathfrak{S}$ . ■

### 6.2.2 Proof of Theorem 9

Suppose  $\exists P \neq Q, P, Q \in \mathcal{D} \subsetneq \mathfrak{S}$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ . Then by Lemma 15, there exists a  $\theta \in \mathcal{S}'_d$  such that  $\check{\theta} = p - q$  where  $p$  and  $q$  are the distributional derivatives of  $P$  and  $Q$ , respectively. Since  $P, Q \in \mathcal{D}$ , we can apply Lemma 17 and so  $\theta$  is a non-zero function that satisfies conditions (i) – (v) in Lemma 17. The condition  $\theta\Lambda = 0$  implies  $\text{supp}(\theta) \subset$

$\mathbb{R}^d \setminus \text{supp}(\Lambda)$ . Since  $\text{supp}(\Lambda)$  has a non-empty interior, we have  $\text{supp}(\theta) \subsetneq \mathbb{R}^d$ . Thus, there exists an open set,  $U \subset \mathbb{R}^d$  such that  $\theta(x) = 0, \forall x \in U$ . By Lemma 25, this means that  $\check{\theta}$  is not compactly supported in  $\mathbb{R}^d$ . Condition (iv) implies  $\int_{\mathbb{R}^d} \check{\theta}(x) dx = 0$ , which means that  $\check{\theta}$  takes negative values. Since  $q$  is compactly supported in  $\mathbb{R}^d$ ,  $q(x) + \check{\theta}(x) < 0$  for some  $x \in \mathbb{R}^d \setminus \text{supp}(Q)$ , which violates condition (v) in Lemma 17. In other words, there does not exist a non-zero  $\theta$  that satisfies conditions (i) – (v) in Lemma 17, thereby leading to a contradiction. ■

As discussed in §4.1, the condition that  $\text{supp}(\Lambda)$  has a non-empty interior is important for Theorem 9 to hold. This is because if  $\text{supp}(\Lambda)$  has an empty interior, then  $\text{supp}(\theta) = \mathbb{R}^d$ . In principle, one can construct such a  $\theta$  by selecting  $\theta \in \mathcal{S}_d$  so that it satisfies conditions (i) – (iv) of Lemma 17 while satisfying the decay conditions (Eq. (29) and Eq. (30)) given in the Paley-Wiener theorem (see Theorem 24). Therefore, by the Paley-Wiener theorem,  $\check{\theta}$  is a  $C^\infty$  function with compact support. If  $\theta$  is chosen such that  $\text{supp}(\check{\theta}) \subset \text{supp}(Q)$ , then condition (v) of Theorem 17 will be satisfied. Thus, one can construct  $P \neq Q, P, Q \in \mathcal{D}$  ( $\mathcal{D}$  being defined in Theorem 9) such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ . Note that conditions (i) and (ii) of Lemma 17 are automatically satisfied (except for conjugate symmetry) by choosing  $\theta \in \mathcal{S}_d$ . However, choosing  $\theta$  such that it is also an entire function (so that the Paley-Wiener theorem can be applied) is not straightforward. In the following, we provide a simple example to show that  $P \neq Q, P, Q \in \mathcal{D}$  can be constructed such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ , where  $\mathcal{F}$  corresponds to a unit ball in an RKHS  $(\mathcal{H}, k)$  induced by a periodic translation-invariant kernel for which  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  has an empty interior.

**Example 3** Let  $Q$  be a uniform distribution on  $[-\beta, \beta] \subset \mathbb{R}$ , i.e.,  $q(x) = \frac{1}{2\beta} \mathbb{1}_{[-\beta, \beta]}(x)$  with its characteristic function,  $\phi_Q(\omega) = \frac{1}{\beta\sqrt{2\pi}} \frac{\sin(\beta\omega)}{\omega}$  in  $L^2(\mathbb{R})$ . Let  $\psi$  be the Dirichlet kernel with period  $\tau$ , where  $\tau \leq \beta$ , i.e.,  $\psi(x) = \frac{\sin\left(\frac{(2l+1)\pi x}{\tau}\right)}{\sin\left(\frac{\pi x}{\tau}\right)}$  and  $\Psi(\omega) = \sum_{j=-l}^l \delta\left(\omega - \frac{2\pi j}{\tau}\right)$  with  $\text{supp}(\Psi) = \left\{\frac{2\pi j}{\tau}, j \in \{0, \pm 1, \dots, \pm l\}\right\}$ . Clearly,  $\text{supp}(\Psi)$  has an empty interior. Let  $\theta$  be

$$\theta(\omega) = \frac{8\sqrt{2}\alpha}{i\sqrt{\pi}} \sin\left(\frac{\omega\tau}{2}\right) \frac{\sin^2\left(\frac{\omega\tau}{4}\right)}{\tau\omega^2}, \quad (18)$$

with  $\alpha \leq \frac{1}{2\beta}$ . It is easy to verify that  $\theta \in L^1(\mathbb{R}) \cap L^2(\mathbb{R}) \cap C_b(\mathbb{R})$  and so  $\theta$  satisfies (i) in Lemma 17. Since  $\theta(\omega) = 0$  at  $\omega = \frac{2\pi l}{\tau}, l \in \mathbb{Z}$ ,  $\theta$  also satisfies (iii) and (iv) in Lemma 17.  $\check{\theta}$  is given by

$$\check{\theta}(x) = \begin{cases} \frac{2\alpha|x+\frac{\tau}{2}|}{\tau} - \alpha, & -\tau \leq x \leq 0 \\ \alpha - \frac{2\alpha|x-\frac{\tau}{2}|}{\tau}, & 0 \leq x \leq \tau \\ 0, & \text{otherwise,} \end{cases} \quad (19)$$

where  $\check{\theta} \in L^1(\mathbb{R}) \cap L^2(\mathbb{R}) \cap C_b(\mathbb{R})$  satisfies (ii) in Lemma 17. Now, consider  $p = q + \check{\theta}$  which is given as

$$p(x) = \begin{cases} \frac{1}{2\beta}, & x \in [-\beta, -\tau] \cup [\tau, \beta] \\ \frac{2\alpha|x+\frac{\tau}{2}|}{\tau} + \frac{1}{2\beta} - \alpha, & x \in [-\tau, 0] \\ \alpha + \frac{1}{2\beta} - \frac{2\alpha|x-\frac{\tau}{2}|}{\tau}, & x \in [0, \tau] \\ 0, & \text{otherwise.} \end{cases}$$

Clearly,  $p(x) \geq 0$ ,  $\forall x$  and  $\int_{\mathbb{R}} p(x) dx = 1$ .  $\phi_P = \phi_Q + \theta = \phi_Q + i\theta_I$  where  $\theta_I = \text{Im}[\theta]$  and  $\phi_P \in L^2(\mathbb{R})$ . We have therefore constructed  $P \neq Q$  such that  $\gamma_{\mathcal{F}}(P, Q) = 0$ , where  $P$  and  $Q$  are compactly supported in  $\mathbb{R}$  with characteristic functions in  $L^2(\mathbb{R})$ .

The condition of the compact support for probability measures mentioned in Theorem 9 is also critical for the result to hold. If this condition is relaxed, then  $k$  with  $\text{supp}(\Lambda) \subsetneq \mathbb{R}^d$  is no longer characteristic to  $\mathfrak{D}$ , as shown in Lemma 19.

## 7 Concluding Remarks

Previous works have studied the Hilbert space embedding for probability measures using universal kernels, which form a restricted family of positive definite kernels. These works showed that if the kernel is universal, then the embedding function from the space of probability measures to a reproducing kernel Hilbert space is injective. In this paper, we extended this approach to a larger family of kernels which are translation-invariant on  $\mathbb{R}^d$ . We showed that the support of the Fourier spectrum of the kernel determines whether the embedding is injective. In particular, the necessary and sufficient condition for the embedding to be injective is that the Fourier spectrum of the kernel should have the entire domain as its support. Our study in this paper was limited to kernels and probability measures that are defined on  $\mathbb{R}^d$ , and the results have been derived using Fourier analysis in  $\mathbb{R}^d$ . Since Fourier theory is available for more general groups apart from  $\mathbb{R}^d$ , one direction for future work is to extend the analysis to positive definite kernels defined on other groups.

## Appendix A Supplementary Results

We show five supplementary results used to prove the results in §4 and §6. The first two are basic, and deal with the Fourier transform of a measure and the convolution theorem. The remaining three (the Riemann-Lebesgue lemma, the Paley-Wiener theorem, and its corollary) are stated without proof.

**Lemma 20 (Fourier transform of a measure)** *Let  $\mu$  be a finite Borel measure on  $\mathbb{R}^d$ . The Fourier transform of  $\mu$  is a tempered distribution given by*

$$\hat{\mu}(\omega) = \int_{\mathbb{R}^d} e^{-i\omega^T x} d\mu(x), \quad \forall \omega \in \mathbb{R}^d \quad (20)$$

which is a bounded, uniformly continuous function on  $\mathbb{R}^d$ . In addition,  $\hat{\mu}$  satisfies the following properties:

- (i)  $\overline{\hat{\mu}(\omega)} = \hat{\mu}(-\omega)$ ,  $\forall \omega \in \mathbb{R}^d$ ,
- (ii)  $\hat{\mu}(\omega) = \hat{\mu}(-\omega)$ ,  $\forall \omega \in \mathbb{R}^d$  if and only if  $D_{\mu}(\varphi) = D_{\mu}(\tilde{\varphi})$ ,  $\forall \varphi \in \mathcal{S}_d$  where  $D_{\mu}$  is the tempered distribution defined by  $\mu$  and  $\tilde{\varphi}(x) := \varphi(-x)$ ,  $\forall x \in \mathbb{R}^d$ .

**Proof:** Let  $D_{\mu}$  denote a tempered distribution defined by  $\mu$ . For  $\varphi \in \mathcal{S}_d$ , we have  $\widehat{D_{\mu}}(\varphi) = D_{\mu}(\hat{\varphi}) = \int_{\mathbb{R}^d} \hat{\varphi}(\omega) d\mu(\omega) = \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} e^{-i\omega^T x} \varphi(x) dm_d(x) d\mu(\omega)$ . From Fubini's theorem,

$$\widehat{D_{\mu}}(\varphi) = \int_{\mathbb{R}^d} \left[ \int_{\mathbb{R}^d} e^{-ix^T \omega} d\mu(\omega) \right] \varphi(x) dm_d(x), \quad (21)$$

which proves Eq. (20). Clearly  $\hat{\mu}$  is bounded as  $|\hat{\mu}(\omega)| \leq 1$ . By Lebesgue's dominated convergence theorem,  $\hat{\mu}$  is uniformly continuous on  $\mathbb{R}^d$  as  $\lim_{h \rightarrow 0} |\hat{\mu}(\omega + h) - \hat{\mu}(\omega)| \leq \lim_{h \rightarrow 0} \int_{\mathbb{R}^d} |e^{-j h^T x} - 1| d\mu(x) = 0$ , for any  $\omega \in \mathbb{R}^d$ .

$$(i) \overline{\hat{\mu}(\omega)} = \int_{\mathbb{R}^d} e^{i\omega^T x} d\mu(x) = \hat{\mu}(-\omega).$$

(ii) ( $\Rightarrow$ ) For  $\varphi \in \mathcal{S}_d$ ,  $\widehat{D_{\mu}}(\varphi) = D_{\mu}(\hat{\varphi}) = \int_{\mathbb{R}^d} \hat{\varphi}(x) d\mu(x) = \int_{\mathbb{R}^d} \hat{\mu}(x) \varphi(x) dm_d(x)$ . Since  $\hat{\varphi} \in \mathcal{S}_d$  and  $D_{\mu}(\varphi) = D_{\mu}(\tilde{\varphi})$ ,  $\forall \varphi \in \mathcal{S}_d$ , we have  $D_{\mu}(\hat{\varphi}) = D_{\mu}(\tilde{\tilde{\varphi}}) = \int_{\mathbb{R}^d} \tilde{\varphi}(-x) d\mu(x)$ . Substituting for  $\hat{\varphi}(-x)$ , we get

$$D_{\mu}(\hat{\varphi}) = \int_{\mathbb{R}^d} \hat{\mu}(-x) \varphi(x) dm_d(x) = \int_{\mathbb{R}^d} \hat{\mu}(x) \varphi(x) dm_d(x),$$

for every  $\varphi \in \mathcal{S}_d$ , which implies  $\hat{\mu}(x) = \hat{\mu}(-x)$ ,  $\forall x \in \mathbb{R}^d$ .

( $\Leftarrow$ ) For  $\varphi \in \mathcal{S}_d$ , we have  $D_{\mu}(\varphi) = (\widehat{D_{\mu}})^{\vee}(\varphi) = \widehat{D_{\mu}}(\tilde{\varphi}) = \int_{\mathbb{R}^d} \hat{\mu}(x) \tilde{\varphi}(x) dm_d(x) = \int_{\mathbb{R}^d} \hat{\mu}(-x) \tilde{\varphi}(x) dm_d(x)$ . Applying Fubini's theorem after substituting for  $\hat{\mu}(-x)$  and  $\tilde{\varphi}(x)$  gives

$$\begin{aligned} D_{\mu}(\varphi) &= \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \delta(y + \omega) \varphi(y) dm_d(y) d\mu(\omega) \\ &= \int_{\mathbb{R}^d} \varphi(-\omega) d\mu(\omega) = D_{\mu}(\tilde{\varphi}), \end{aligned}$$

for every  $\varphi \in \mathcal{S}_d$ . ■

**Remark 21** (a) *Property (i) in Lemma 20 shows that the Fourier transform of a finite Borel measure on  $\mathbb{R}^d$  is “conjugate symmetric”, which means that  $\text{Re}[\hat{\mu}]$  is an even function and  $\text{Im}[\hat{\mu}]$  is an odd function.*

(b) *Property (ii) shows that real symmetric tempered distributions have real symmetric Fourier transforms. This can be easily understood when  $\mu$  is absolutely continuous w.r.t. the Lebesgue measure. Suppose  $d\mu = \Psi dm_d$ . Then property (ii) implies that  $\hat{\mu}$  is real and symmetric if and only if  $\Psi$  is real and symmetric.*

The following result is popularly known as the *convolution theorem*. Before providing the result, we first define convolution: if  $f$  and  $g$  are complex functions in  $\mathbb{R}^d$ , their convolution  $f * g$  is

$$(f * g)(x) = \int_{\mathbb{R}^d} f(y)g(x - y) dy, \quad (22)$$

provided that the integral exists for almost all  $x \in \mathbb{R}^d$ , in the Lebesgue sense. Let  $\mu$  be a finite Borel measure on  $\mathbb{R}^d$  and  $f$  be a bounded measurable function on  $\mathbb{R}^d$ . The convolution of  $f$  and  $\mu$ ,  $f * \mu$ , which is a bounded measurable function, is defined by

$$(f * \mu)(x) = \int_{\mathbb{R}^d} f(x - y) d\mu(y). \quad (23)$$

**Theorem 22 (Convolution Theorem)** *Let  $\mu$  be a finite Borel measure and  $f$  be a bounded function on  $\mathbb{R}^d$ . Suppose  $f$  is written as*

$$f(x) = \int_{\mathbb{R}^d} e^{ix^T \omega} d\Lambda(\omega), \quad (24)$$

with a finite Borel measure  $\Lambda$  on  $\mathbb{R}^d$ . Then

$$(f * \mu)^\wedge = \hat{\mu}\Lambda, \quad (25)$$

where the right hand side is a finite Borel measure<sup>14</sup> and the equality holds as a tempered distribution.

**Proof:** Since the Fourier and inverse Fourier transform give one-to-one correspondence of  $\mathcal{S}'_d$ , it suffices to show

$$f * \mu = (\hat{\mu}\Lambda)^\vee. \quad (27)$$

For an arbitrary  $\varphi \in \mathcal{S}_d$ ,

$$(\hat{\mu}\Lambda)^\vee(\varphi) = (\hat{\mu}\Lambda)(\check{\varphi}) = \int_{\mathbb{R}^d} \check{\varphi}(x)\hat{\mu}(x) d\Lambda(x). \quad (28)$$

Substituting for  $\hat{\mu}$  in Eq. (28) and applying Fubini's theorem, we have  $(\hat{\mu}\Lambda)^\vee(\varphi) =$

$$\int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \left[ \int_{\mathbb{R}^d} e^{i(\omega-y)^T x} d\Lambda(x) \right] \varphi(\omega) dm_d(\omega) d\mu(y),$$

which reduces to  $\int_{\mathbb{R}^d} [\int_{\mathbb{R}^d} f(\omega - y) d\mu(y)]\varphi(\omega) dm_d(\omega) = (f * \mu)(\varphi)$  and therefore proves Eq. (27). ■

The following result, called the Riemann-Lebesgue lemma, is quoted from [Rud91, Theorem 7.5].

**Lemma 23 (Riemann-Lebesgue)** *If  $f \in L^1(\mathbb{R}^d)$ , then  $\hat{f} \in C_b(\mathbb{R}^d)$ , and  $\|\hat{f}\|_\infty \leq \|f\|_1$ .*

The following theorem is a version of the *Paley-Wiener theorem* for  $C^\infty$  functions, and is proved in [Str03, Theorem 7.2.2].

**Theorem 24 (Paley-Wiener)** *Let  $f$  be a  $C^\infty$  function supported in  $[-\beta, \beta]$ . Then  $\hat{f}(\omega + i\sigma)$  is a entire function of exponential type  $\beta$ , i.e.,  $\exists C$  such that*

$$\left| \hat{f}(\omega + i\sigma) \right| \leq C e^{\beta|\sigma|}, \quad (29)$$

and  $\hat{f}(\omega)$  is rapidly decreasing, i.e.,  $\exists c_n$  such that

$$\left| \hat{f}(\omega) \right| \leq \frac{c_n}{(1 + |\omega|)^n}, \quad \forall n \in \mathbb{N}. \quad (30)$$

Conversely, if  $F(\omega + i\sigma)$  is an entire function of exponential type  $\beta$ , and  $F(\omega)$  is rapidly decaying, then  $F = \hat{f}$  for some such function  $f$ .

The following lemma is a corollary of the Paley-Wiener theorem, and is proved in [Mal98, Theorem 2.6].

**Lemma 25 ([Mal98])** *If  $g \neq 0$  has compact support, then its Fourier transform  $\hat{g}$  cannot be zero on a whole interval. Similarly, if  $\hat{g} \neq 0$  has compact support, then  $g$  cannot be zero on a whole interval.*

<sup>14</sup>Let  $\mu$  be a finite Borel measure and  $f$  be a bounded measurable function on  $\mathbb{R}^d$ . We then define a finite Borel measure  $f\mu$  by

$$(f\mu)(E) = \int_{\mathbb{R}^d} I_E(x)f(x) d\mu(x), \quad (26)$$

where  $E$  is an arbitrary Borel set and  $I_E$  is its indicator function.

## References

- [Aro50] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.
- [BJ02] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.
- [Dud02] R. M. Dudley. *Real Analysis and Probability*. Cambridge University Press, Cambridge, UK, 2002.
- [FGSS08] K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 489–496, Cambridge, MA, 2008. MIT Press.
- [GBR<sup>+</sup>07] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel method for the two sample problem. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press, 2007.
- [GHS<sup>+</sup>05] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, December 2005.
- [GSB<sup>+</sup>04] A. Gretton, A. Smola, O. Bousquet, R. Herbrich, B. Schölkopf, and N. Logothetis. Behaviour and convergence of the constrained covariance. Technical Report 130, MPI for Biological Cybernetics, 2004.
- [GW99] C. Gasquet and P. Witomski. *Fourier Analysis and Applications*. Springer-Verlag, New York, 1999.
- [Mal98] S. G. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.
- [RS72] M. Reed and B. Simon. *Functional Analysis*. Academic Press, New York, 1972.
- [Rud91] W. Rudin. *Functional Analysis*. McGraw-Hill, USA, 1991.
- [SGSS07] A. J. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *Proc. 18th International Conference on Algorithmic Learning Theory*, pages 13–31. Springer-Verlag, Berlin, Germany, 2007.
- [Sho00] G. R. Shorack. *Probability for Statisticians*. Springer-Verlag, New York, 2000.
- [SS02] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- [Ste02] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2002.
- [Str03] R. S. Strichartz. *A Guide to Distribution Theory and Fourier Transforms*. World Scientific Publishing, Singapore, 2003.
- [Wen05] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, Cambridge, UK, 2005.

---

# Almost Tight Upper Bound for Finding Fourier Coefficients of Bounded Pseudo-Boolean Functions

---

**Sung-Soon Choi\***  
Random Graph Research Center  
Yonsei University  
Seoul, 120-749 Korea  
ss.choi@yonsei.ac.kr

**Kyomin Jung**  
Department of Mathematics  
MIT  
Cambridge, MA02139, USA  
kmjung@mit.edu

**Jeong Han Kim†**  
Department of Mathematics and  
Random Graph Research Center  
Yonsei University  
Seoul, 120-749 Korea  
jehkim@yonsei.ac.kr

## Abstract

A pseudo-Boolean function is a real-valued function defined on  $\{0, 1\}^n$ . A  $k$ -bounded function is a pseudo-Boolean function that can be expressed as a sum of subfunctions each of which depends on at most  $k$  input bits. The  $k$ -bounded functions for constant  $k$  play an important role in a number of research areas including molecular biology, biophysics, and evolutionary computation. In this paper, we consider the problem of finding the Fourier coefficients of  $k$ -bounded functions with a series of function evaluations at any input strings. Suppose that a  $k$ -bounded function  $f$  with  $m$  non-zero Fourier coefficients is given. Our main result is to present an adaptive randomized algorithm to find the Fourier coefficients of  $f$  with high probability in  $\mathcal{O}(m \log n)$  function evaluations for constant  $k$ . Up to date, the best known upper bound is  $\mathcal{O}(\alpha(n, m)m \log n)$ , where  $\alpha(n, m)$  is between  $n^{\frac{1}{2}}$  and  $n$  depending on  $m$ . Thus, our bound improves the previous bound by a factor of  $\Omega\left(n^{\frac{1}{2}}\right)$ . Also, it is almost tight with respect to the known lower bound  $\Omega\left(\frac{m \log n}{\log m}\right)$ . To obtain the main result, we first show that the problem of finding the Fourier coefficients of a  $k$ -bounded function is reduced to the problem of finding a  $k$ -bounded hypergraph with a certain type of queries under an oracle with one-sided error. For this, we devise a method to test with one-sided error whether there is a dependency within some set of input bits among a collection of sets of input bits. Then, we give a randomized algorithm for the hypergraph finding problem and obtain the desired bound by analyzing the algorithm based on a large deviation result for a sum of independent random variables.

---

\*This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. R16-2007-075-01000-0).

†This work was partially supported by Yonsei University Research Funds 2006-1-0078 and 2007-1-0025, and by the second stage of the Brain Korea 21 Project in 2007, and by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-312-C00455).

## 1 Introduction

A *pseudo-Boolean* function is a real-valued function defined on the set of binary strings of fixed length. If a pseudo-Boolean function can be expressed as a sum of subfunctions each of which depends on at most  $k$  input bits, it is called *k-bounded*. Given a 2-SAT formula, for example, the number of clauses an assignment satisfies is a 2-bounded pseudo-Boolean function of the assignment. Note that a  $k$ -bounded pseudo-Boolean function is a polynomial of Boolean variables of degree  $k$  or less, and vice versa. In this paper, we consider the problem of finding the Fourier coefficients of  $k$ -bounded pseudo-Boolean functions. In the problem, we assume the oracle that, given any binary string, returns the function value at the string. Our main concern is the query complexity to solve the problem, i.e., the number of function evaluations required to find the Fourier coefficients of  $k$ -bounded pseudo-Boolean functions. (Unless otherwise specified, a  $k$ -bounded function means a  $k$ -bounded pseudo-Boolean function in this paper.)

The  $k$ -bounded functions have played an important role in molecular biology and biophysics. In those areas, a number of mathematical models have been proposed to study the evolution of a population of organisms (or biological objects) [Ewe79, FL70, KL87, Lew74, MP89]. In many of the models including the NK model [Kau89],  $k$ -bounded functions have been used to measure the fitness of an organism in an environment. In the NK model [Kau89], each subfunction represents the contribution of a gene of the organism to the overall fitness, interacting with a fixed number of other genes. Hence, a  $k$ -bounded function may be regarded as a sum of subfunctions each of which depends on at most  $k$  genes. The  $k$ -bounded functions with small  $k$  in the NK model induce the fitness landscapes of reasonable evolvability and complexity, which were used for describing the evolution of living systems [Kau93]. They were also used as a benchmark for comparing the landscapes arising in RNA folding [FSBB<sup>+</sup>93]. In this regard,  $k$ -bounded functions with small  $k$  have been paid attention.

The  $k$ -bounded functions have been also used as testbed problems for comparing the performance of heuristic algorithms in the area of evolutionary computation [CC06, HG97, MM99, MG99, PG00]. The problem of maximizing arbitrary  $k$ -bounded functions is NP-hard even for  $k = 2$  as it is at least as hard as the MAX-2-SAT problem [GJS76]. The larger the value of  $k$  is, the higher is the degree of the depen-

dependency among the input bits in a  $k$ -bounded function. By controlling the degree of the dependency (the value of  $k$ ), in general, we may control the difficulty of the problem of maximizing the  $k$ -bounded functions. There are good heuristic algorithms to approximate the maximum of a  $k$ -bounded function when the dependency among the input bits are known [dBIV97, Gol89, MM99, PGCP00, Str04].

Fourier transform is a formal approach to define the dependency among the input bits of a pseudo-Boolean function. There have been a number of papers addressing the problem of finding the Fourier coefficients of a  $k$ -bounded function  $f : \{0, 1\}^n \rightarrow \mathbb{R}$  with constant  $k$ . Kargupta and Park [KP01] presented a deterministic algorithm using  $\mathcal{O}(n^k)$  function evaluations. Later, Heckendorn and Wright [HW03, HW04] proposed a randomized algorithm for the problem. They analyzed the algorithm to show that, with negligible error probability, it finds the Fourier coefficients in  $\mathcal{O}(n^2 \log n)$  function evaluations on average for the  $k$ -bounded functions with  $\mathcal{O}(n)$  non-zero Fourier coefficients generated from a random model. For the  $k$ -bounded functions with  $m$  non-zero Fourier coefficients, Choi, Jung, and Moon [CJM08] proved that any randomized algorithm requires  $\Omega\left(\frac{m \log n}{\log m}\right)$  function evaluations to find the Fourier coefficients with error probability at most a given constant. By analyzing the algorithm of Heckendorn and Wright, they also proved that  $\mathcal{O}(\alpha(n, m)m \log n)$  function evaluations, where  $\alpha(n, m)$  is between  $n^{\frac{1}{2}}$  and  $n$  depending on  $m$ , are enough to find the Fourier coefficients. Recently, for 2-bounded functions of which non-zero Fourier coefficients are between  $n^{-a}$  and  $n^b$  in absolute value for some positive constants  $a$  and  $b$ , Choi and Kim [CK08] showed that there exists a deterministic algorithm using  $\mathcal{O}\left(\frac{m \log n}{\log m}\right)$  function evaluations, provided that  $m \geq n^\epsilon$  for any constant  $\epsilon > 0$ . This algorithm is non-adaptive while the previous algorithms are adaptive.<sup>1</sup> However, an explicit construction of the algorithm is unknown.

Our main result is

**Theorem 1** *Suppose that  $f$  is a  $k$ -bounded function defined on  $\{0, 1\}^n$  for constant  $k$  and that  $f$  has  $m$  non-zero Fourier coefficients. Then, there exists an adaptive algorithm to find the Fourier coefficients of  $f$  in  $\mathcal{O}(m \log n)$  function evaluations with probability  $1 - \mathcal{O}\left(\frac{1}{n}\right)$ .*

We prove Theorem 1 by showing an explicit construction of the desired algorithm. This result improves the best known upper bound  $\mathcal{O}(\alpha(n, m)m \log n)$  by a factor of  $\Omega\left(n^{\frac{1}{2}}\right)$  and it is almost tight with respect to the lower bound  $\Omega\left(\frac{m \log n}{\log m}\right)$ .

We should note that there have been a number of papers addressing the problem of finding the Fourier coefficients of Boolean functions [BJT04, BT96, Jac97, KM93, Man94]. The KM algorithm [KM93] is one of the most famous algorithms for the problem and most of the subsequent algorithms have been based on the algorithm. These algorithms for Boolean functions can be extended to pseudo-Boolean functions. However, the extensions of the algorithms do not

<sup>1</sup>An algorithm is called *adaptive* if the algorithm uses a sequence of queries in which some queries depend on the previous queries. Otherwise, it is called *non-adaptive*.

give a good bound for  $k$ -bounded pseudo-Boolean functions. One of the main reasons is that their query complexities depend on the values of the target function. For example, for a  $k$ -bounded function  $f$ , (to the best of our knowledge) the most efficient extension [BJT04] among those has the query complexity of  $\Omega\left(r \left(\frac{B}{\theta}\right)^2\right)$ , where  $r$  is the number of input bits on which  $f$  depends,  $B$  is the maximum absolute value of  $f$ , and  $\theta$  is the minimum absolute value of the non-zero Fourier coefficients of  $f$ . Thus, the query complexity may be made arbitrarily large depending on  $B$  and  $\theta$ .<sup>2</sup> The query complexity of our algorithm is independent of the values of the target function.

To prove Theorem 1, we first show that the problem of finding the Fourier coefficients of a  $k$ -bounded function is reduced to the problem of finding a  $k$ -bounded hypergraph<sup>3</sup> (with a certain type of queries under a probabilistic oracle). For a pseudo-Boolean function  $f$  defined on  $\{0, 1\}^n$ , we consider the hypergraph representing the dependency among the input bits as follows. Suppose that  $H$  is a subset of  $[n]$ , where  $[n]$  is the set of the integers from 1 to  $n$ . We say that there is a *linkage* among the input bits in  $H$  if, for any additive expression of  $f$ ,  $f = \sum_i f_i$ , there is  $j$  such that  $H$  is included in the support set of  $f_j$ .<sup>4</sup> The *linkage graph* of  $f$  is a hypergraph  $G_f = ([n], E)$ , where each bit in  $[n]$  represents a vertex and a subset  $H$  of  $[n]$  belongs to the edge set  $E$  if and only if there is a linkage among the bits in  $H$ .

For example, consider the following function:

$$f(x_1, x_2, x_3, x_4, x_5) = 5x_1x_2 - 3x_2x_3x_4.$$

If we let  $f_1(x_1, x_2) = 5x_1x_2$  and  $f_2(x_2, x_3, x_4) = -3x_2x_3x_4$ ,  $f$  can be represented as an additive expression,  $f = f_1 + f_2$ . In this expression, each subfunction of  $f$  has a support set of which size is at most three and so  $f$  is 3-bounded. It can be shown that the support sets of  $f_1$  and  $f_2$ ,  $\{1, 2\}$  and  $\{2, 3, 4\}$ , are hyperedges of  $G_f$ . By definition of linkage, the non-empty subsets of  $\{1, 2\}$  and  $\{2, 3, 4\}$  are also hyperedges of  $G_f$ . Generally, if a set of vertices is a hyperedge of  $G_f$ , then any non-empty subset of the set is also a hyperedge of  $G_f$ . We call this property the *hierarchical property* among hyperedges. The linkage graph  $G_f$  has nine hyperedges:  $\{1\}$ ,  $\{2\}$ ,  $\{3\}$ ,  $\{4\}$ ,  $\{1, 2\}$ ,  $\{2, 3\}$ ,  $\{2, 4\}$ ,  $\{3, 4\}$ , and  $\{2, 3, 4\}$ . There is no hyperedge containing 5 since  $f$  does not depend on  $x_5$ .

It is known that, for a  $k$ -bounded function with constant  $k$ , the problem of finding the Fourier coefficients is asymptotically equivalent to the problem of finding the linkage graph in terms of the number of function evaluations [HW03, HW04].

<sup>2</sup>To see a typical behavior of the complexity, we may consider the NK model [Kau89]. The NK model with parameters  $N = n$  and  $K = k - 1$  generates a class of  $k$ -bounded functions that are expressed as a sum of  $n$  subfunctions. When  $f$  is a function randomly generated from the NK model with parameters  $N = n$  and  $K = k - 1$  for constant  $k$ , it is not difficult to show that  $r = \Theta(n)$ ,  $B = \Omega(n)$ , and  $\theta = \mathcal{O}(1)$  with high probability. Thus, the query complexity of the algorithm [BJT04] for  $f$  is  $\Omega(n^3)$  with high probability while the query complexity of our algorithm for  $f$  is  $\mathcal{O}(n \log n)$ .

<sup>3</sup>A hypergraph is  $k$ -bounded if the order of each hyperedge is at most  $k$ .

<sup>4</sup>The term linkage is from genetics and it means the interaction among the genes.

(The description of the asymptotic equivalence is provided in Section 2.) In a hypergraph, we say that a hyperedge *crosses* among certain disjoint sets of vertices if the number of the sets is equal to the order of the hyperedge and each of the sets contains exactly one vertex in the hyperedge. (By definition, a hyperedge of order one crosses among any set of vertices including the hyperedge.) Our main contribution is to show that, given a collection of disjoint sets of vertices, the existence of a hyperedge of the linkage graph crossing among those sets is testable with one-sided error by using a constant number of function evaluations.

**Theorem 2** *Suppose that  $f$  is a  $k$ -bounded function defined on  $\{0, 1\}^n$  and  $S_1, \dots, S_j$  are  $j$  disjoint subsets of  $[n]$ . Then, we can use  $2^j$  function evaluations of  $f$  to test the existence of a hyperedge in the linkage graph  $G_f$  crossing among  $S_i$ 's, where the test result is correct with probability at least  $\frac{1}{2^{2k}}$  if such a hyperedge exists and it is correct with probability 1 otherwise.*

Theorem 2 is an extension of a previous theorem of Heckendorn and Wright [HW04] (Proposition 1 in Section 2), which holds only for the case when each of  $S_i$ 's is a singleton set of vertices. To prove Theorem 2, we devise a random perturbation method for testing the existence of a hyperedge. It tests the existence of a hyperedge by flipping a randomly generated string at certain bit positions and evaluating the function values at the flipped strings. We obtain the desired result by analyzing the method. The analysis extensively uses the properties of basis functions in the Fourier transform of a  $k$ -bounded function.

Theorem 2 implies that the problem of finding the linkage graph of a  $k$ -bounded function is reduced to the following graph finding problem. Suppose that a hypergraph  $G$  has  $n$  vertices and  $m$  hyperedges and the hyperedges of  $G$  are unknown. A *cross-membership query* asks the existence of a hyperedge crossing among certain disjoint sets of vertices. We assume the *oracle with one-sided error*  $\delta$  as follows. Given a cross-membership query, the oracle correctly answers with probability at least  $1 - \delta$  if the true answer for the query is YES and it correctly answers with probability 1 otherwise. The problem is to find the hyperedges of  $G$  by using as few queries to the oracle as possible.

In fact, it is enough for our purpose to consider the hypergraph finding problem for the  $k$ -bounded hypergraphs with the hierarchical property. Since we think that the problem is of self interest, however, we consider the problem for arbitrary  $k$ -bounded hypergraphs. We present an adaptive randomized algorithm for the problem to show

**Theorem 3** *Suppose that  $G$  is an unknown  $k$ -bounded hypergraph with  $n$  vertices and  $m$  edges for constant  $k$ . Then, for any constant  $0 \leq \delta < 1$ , the hyperedges of  $G$  can be found with probability  $1 - \mathcal{O}\left(\frac{1}{n}\right)$  by using  $\mathcal{O}(m \log n)$  cross-membership queries under the oracle with one-sided error  $\delta$ . (The number of cross-membership queries is  $2^{\mathcal{O}(k)}$  in  $k$ .)*

Our algorithm for Theorem 3 iteratively uses binary search to find the hyperedges. In this sense, it is analogous to the algorithm of Angluin and Chen [AC04, AC05, AC06] for the hypergraph finding problem with edge-detecting queries or to the algorithm of Reyzin and Srivastava [RS07] for the graph

finding problem with edge-counting (or additive) queries.<sup>5</sup> On the other hand, since the answers of the oracle may contain errors in our situation, we need to handle the error bound more carefully, which is the main task in proving Theorem 3. A large deviation result for a sum of independent random variables with geometric distribution is crucially used for the task. (There have been a number of papers addressing the problem of finding a graph or a hypergraph by using various types of queries. For example, see [AA04, AA05, ABK<sup>+</sup>02, ABK<sup>+</sup>04, AC06, BAA<sup>+</sup>01, BGK05, CK08, GK00].)

Theorem 1 is obtained from Theorems 2 and 3 and the equivalence between the problems of finding the Fourier coefficients and the linkage graph.

The remainder of the paper is organized as follows. In Section 2, we review some basic facts and previous results for the problem of finding the Fourier coefficients of  $k$ -bounded functions. In Section 3, we prove Theorem 2, which states the linkage testability of a linkage graph, by proving relevant lemmas. Section 4 deals with the graph finding problem with cross-membership queries under the probabilistic oracle as an independent problem. In the section, we give a randomized algorithm for the problem and analyze it to obtain Theorem 3. In Section 5, some remarks on the query and time complexity of the proposed algorithm are provided along with a factor of improving the complexity. Finally, concluding remarks closes the paper in Section 6.

## 2 Preliminaries

### 2.1 Linkage Test Function

Munetomo and Goldberg [MG99] proposed a perturbation method to test for the existence of linkage in a 2-subset of  $[n]$ . Given a 2-subset  $S$  and a string  $x$ , it checks the non-linearity between the two bits in  $H$  by flipping the two bits of  $x$  individually and simultaneously and adding/subtracting the function values at the flipped strings. Heckendorn and Wright [HW04] generalized the method to detect linkage for subsets of any order. Suppose that  $f$  is a pseudo-Boolean function defined on  $\{0, 1\}^n$ ,  $S$  is a subset of  $[n]$ , and  $x$  is a string in  $\{0, 1\}^n$ . They considered the *linkage test function*  $\mathcal{L}$  depending on  $f$ ,  $S$ , and  $x$  as follows:

$$\mathcal{L}(f, S, x) = \sum_{A \subseteq S} (-1)^{|A|} f(x \oplus 1_A).$$

Here,  $1_A$  represents the string consisting of ones in the bit positions of  $A$  and zeros in the rest. For two strings  $x, y \in \{0, 1\}^n$ ,  $x \oplus y$  means the bitwise addition modulo 2 of  $x$  and  $y$ . The linkage test function  $\mathcal{L}$  performs a series of function evaluations at  $x$  and the strings obtained by flipping  $x$  in order to detect the existence of the linkage among the bits in  $S$ . Heckendorn and Wright [HW04] proved the following theorem, which shows the usefulness of the linkage test function in finding hyperedges of  $G_f$ .

<sup>5</sup>An edge-detecting query asks the existence of an edge (or a hyperedge) in a set of vertices while an edge-counting query asks the number of edges (or hyperedges) in a set of vertices.

**Proposition 1** Suppose that  $f$  is a  $k$ -bounded function defined on  $\{0, 1\}^n$ . Then, the followings hold:

- (a) A subset  $S$  of  $[n]$  is a hyperedge of  $G_f$  if and only if  $\mathcal{L}(f, S, x) \neq 0$  for some string  $x \in \{0, 1\}^n$ .  
(b) For a hyperedge  $S$  of order  $j$  in  $G_f$ , the probability that  $\mathcal{L}(f, S, x) \neq 0$  for a string  $x$  chosen uniformly at random from  $\{0, 1\}^n$  is at least  $\frac{1}{2^{k-j}}$ .

Proposition 1 indicates that the linkage test function determines the existence of a hyperedge with one-sided error. Thus, by repeatedly evaluating the linkage test function for randomly chosen strings, we can make the error arbitrarily small. In particular, when  $k$  is a constant, this implies that a constant number of linkage tests (consequently, a constant number of function evaluations) is enough for determining the existence of a hyperedge with error probability at most a given constant. The hierarchical property among hyperedges implies that, for  $j \geq 2$ , a  $j$ -subset  $H$  can be a hyperedge only if every  $(j-1)$ -subset of  $H$  is a hyperedge. Based on this observation, Heckendorn and Wright [HW04] proposed a randomized algorithm that performs linkage test only for such a hyperedge candidate: The algorithm first detects the hyperedges of order one by investigating all the singleton subsets of  $[n]$ . Then, for  $j$  from 2 to  $k$ , it detects the hyperedges of order  $j$  by performing linkage test for the hyperedge candidates of order  $j$  that have been identified from the information of the hyperedges of lower order. Recently, the performance of the algorithm was fully analyzed by Choi *et al.* [CJM08]. Given a  $k$ -bounded function  $f$  with  $m$  hyperedges and a constant  $\varepsilon > 0$ , they showed that the algorithm finds the linkage graph  $G_f$  in  $\mathcal{O}(\alpha(n, m)m \log n)$  function evaluations with error probability at most  $\varepsilon$ , where  $\alpha(n, m)$  is between  $n^{\frac{1}{2}}$  and  $n$  depending on  $m$ .

## 2.2 A Fourier Transform

Walsh transform is a Fourier transform for the space of pseudo-Boolean functions in which a pseudo-Boolean function is represented as a linear combination of  $2^n$  basis functions called *Walsh functions* [Wal23]. For each subset  $H$  of  $[n]$ , the Walsh function corresponding to  $H$ ,  $\psi_H : \{0, 1\}^n \rightarrow \mathbb{R}$ , is defined as

$$\psi_H(x) = (-1)^{\sum_{i \in H} x[i]},$$

where  $x[i]$  represents the  $i^{\text{th}}$  bit value in  $x$ . If we define an inner product of two pseudo-Boolean functions  $f$  and  $g$  as

$$\langle f, g \rangle = \sum_{x \in \{0, 1\}^n} \frac{f(x) \cdot g(x)}{2^n},$$

the set of Walsh functions,  $\{\psi_H \mid H \subseteq [n]\}$ , becomes an orthonormal basis of the space of pseudo-Boolean functions. Hence, a pseudo-Boolean function  $f$  can be represented as

$$f = \sum_{H \subseteq [n]} \hat{f}(H) \cdot \psi_H,$$

where  $\hat{f}(H) = \langle f, \psi_H \rangle$  is called the *Fourier coefficient* corresponding to  $H$ . Specifically, if  $\hat{f}(H) \neq 0$  and  $\hat{f}(H') = 0$  for any  $H' \supsetneq H$ ,  $\hat{f}(H)$  is called a *maximal non-zero Fourier coefficient* of  $f$ . We refer to [HW99] for surveys of the properties of Walsh functions and Walsh transform in the space of pseudo-Boolean functions.

Heckendorn and Wright [HW04] provided a number of results to show the relation between the linkage test function and the Fourier coefficients. Some of them are summarized in the following proposition.

**Proposition 2** Suppose that  $f$  is a pseudo-Boolean function defined on  $\{0, 1\}^n$ . Then, the followings hold:

- (a) For a subset  $H$  of  $[n]$ ,  $\hat{f}(H)$  is a maximal non-zero Fourier coefficient of  $f$  if and only if  $H$  is a maximal hyperedge of  $G_f$ .  
(b) For a maximal hyperedge  $H \subseteq [n]$ ,

$$\hat{f}(H) = \frac{\mathcal{L}(f, H, 0^n)}{2^{|H|}}.$$

- (c) For a subset  $H$  of  $[n]$ ,

$$\hat{f}(H) = \frac{\mathcal{L}(f, H, 0^n)}{2^{|H|}} - \sum_{H' \subsetneq H} \hat{f}(H').$$

- (d) For subsets  $H$  and  $H'$  of  $[n]$  with  $H \subseteq H'$ ,

$$\mathcal{L}(f, H', 0^n) = \sum_{A \subseteq H' \setminus H} (-1)^{|A|} \mathcal{L}(f, H, 1_A).$$

Proposition 2 (a) says that the subsets of  $[n]$  with maximal non-zero Fourier coefficients of  $f$  are the maximal hyperedges in the linkage graph of  $f$ . Thus, from Proposition 2 (b), the maximal non-zero Fourier coefficients of  $f$  are found by evaluating the linkage test function at the zero string for each maximal hyperedge. Once the maximal non-zero Fourier coefficients are found, the Fourier coefficients corresponding to the subsets of lower orders can be found by successively applying Proposition 2 (c). Proposition 2 (d) implies that no additional function evaluations are required for finding the Fourier coefficients corresponding to the subsets of lower orders. Hence, if  $f$  is  $k$ -bounded for constant  $k$  and  $m$  is the number of hyperedges in  $G_f$ ,  $\mathcal{O}(m)$  additional function evaluations are enough to find the Fourier coefficients of  $f$ . On the other hand,  $\Omega\left(\frac{m \log n}{\log m}\right)$  function evaluations are required for finding the linkage graph of a  $k$ -bounded function for constant  $k$  as shown in [CJM08]. Thus, the problem of finding the Fourier coefficients of a  $k$ -bounded function for constant  $k$  is equivalent to the problem of finding the linkage graph in terms of the number of function evaluations required up to a constant factor.

## 3 Generalized Linkage Test

### 3.1 Generalized Linkage Test Function

Let  $f$  be a pseudo-Boolean function defined on  $\{0, 1\}^n$ ,  $\mathcal{S}$  be a collection of disjoint subsets of  $[n]$ , and  $x$  be a string in  $\{0, 1\}^n$ . We define the *generalized linkage test function*  $\mathcal{L}^*$  depending on  $f$ ,  $\mathcal{S}$ , and  $x$  as follows:

$$\mathcal{L}^*(f, \mathcal{S}, x) = \sum_{S' \subseteq \mathcal{S}} (-1)^{|S'|} f\left(x \oplus \left(\bigoplus_{A \in S'} 1_A\right)\right).$$

If we let  $\mathcal{S}_H = \{\{a\} \mid a \in H\}$  for a subset  $H$  of  $[n]$ , we see that  $\mathcal{L}^*(f, \mathcal{S}_H, x) = \mathcal{L}(f, H, x)$  for any  $x \in \{0, 1\}^n$ .

The following lemmas describes the basic properties of the generalized linkage test function.

**Lemma 4** Suppose that  $\mathcal{S}$  is a collection of disjoint subsets of  $[n]$ . Then, the followings hold:

(a) (Linearity) If  $f_1, \dots, f_\ell$  are pseudo-Boolean functions defined on  $\{0, 1\}^n$  and  $c_1, \dots, c_\ell$  are constants,

$$\mathfrak{L}^* \left( \sum_{i=1}^{\ell} c_i f_i, \mathcal{S}, x \right) = \sum_{i=1}^{\ell} c_i \mathfrak{L}^*(f_i, \mathcal{S}, x)$$

for all  $x \in \{0, 1\}^n$ .

(b) (Recursion) If  $f$  is a pseudo-Boolean function defined on  $\{0, 1\}^n$ ,

$$\mathfrak{L}^*(f, \mathcal{S}, x) = \mathfrak{L}^*(f, \mathcal{S} \setminus \{A\}, x) - \mathfrak{L}^*(f, \mathcal{S} \setminus \{A\}, x \oplus 1_A)$$

for any  $A \in \mathcal{S}$  and any  $x \in \{0, 1\}^n$ .

**Proof:** Omitted.  $\blacksquare$

**Lemma 5** Suppose that  $f$  is a pseudo-Boolean function defined on  $\{0, 1\}^n$  and  $\mathcal{S}$  is a collection of disjoint subsets of  $[n]$ . If the support set of  $f$  is disjoint with some  $A \in \mathcal{S}$ ,  $\mathfrak{L}^*(f, \mathcal{S}, x) = 0$  for all  $x \in \{0, 1\}^n$ .

**Proof:** Omitted.  $\blacksquare$

### 3.2 Linkage Test Theorem

A collection of disjoint subsets of  $[n]$ ,  $\mathcal{R} = \{R_1, \dots, R_j\}$ , is called a *setwise subcollection* of  $\mathcal{S}$  if  $R_i \subseteq S_i$  for all  $1 \leq i \leq j$ . In this case, we denote by  $\mathcal{R} \subseteq \mathcal{S}$ . Note that a setwise subcollection  $\mathcal{R}$  of  $\mathcal{S}$  is allowed to contain multiple empty sets from the definition. We consider a random model  $\Gamma(\mathcal{S})$  that generates a setwise subcollection of  $\mathcal{S}$  as follows: For each  $S_i \in \mathcal{S}$ , we select each element in  $S_i$  independently and with probability  $\frac{1}{2}$  and put it into  $R_i$ . Then, we build a setwise subcollection  $\mathcal{R}$  of  $\mathcal{S}$  by letting  $\mathcal{R} = \{R_i \mid 1 \leq i \leq j\}$ . In the following,  $\text{str}(\mathcal{R})$  denotes the set of the strings,  $x$ 's, such that  $x$  has the same bit value in the bit positions in  $R_i$  for all  $1 \leq i \leq j$ :

$$\text{str}(\mathcal{R}) = \{x \in \{0, 1\}^n \mid x[a] = x[b] \text{ for all } a, b \text{ such that } a, b \in R_i \text{ for some } i \text{ with } 1 \leq i \leq j\}.$$

**Theorem 6** Suppose that  $f$  is a  $k$ -bounded function and  $\mathcal{S}$  is a collection of disjoint subsets of  $[n]$ . Then, the followings hold:

(a) The linkage graph  $G_f$  contains a hyperedge crossing among  $\mathcal{S}$  if and only if there exist  $\mathcal{R} \subseteq \mathcal{S}$  and  $x \in \text{str}(\mathcal{R})$  such that  $\mathfrak{L}^*(f, \mathcal{R}, x) \neq 0$ .

(b) If  $G_f$  contains a hyperedge crossing among  $\mathcal{S}$ , the probability that  $\mathfrak{L}^*(f, \mathcal{R}, x) \neq 0$  for a randomly generated  $\mathcal{R}$  from  $\Gamma(\mathcal{S})$  and a string  $x$  chosen uniformly at random from  $\text{str}(\mathcal{R})$  is at least  $\frac{1}{2^{2k}}$ .

Theorem 6 implies Theorem 2 and provides an efficient method to test for the existence of a hyperedge crossing among a given collection of sets of vertices in the linkage graph.

**Proof:** Since (b) implies the only-if part of (a), we first prove the if part of (a) and then prove (b).

Suppose that  $G_f$  does not contain any hyperedge crossing among  $\mathcal{S}$ . Let  $\mathcal{R}$  be a setwise subcollection of  $\mathcal{S}$  and let  $x$  be a string in  $\{0, 1\}^n$ . Since  $G_f$  does not contain any hyperedge crossing among  $\mathcal{S}$ ,  $G_f$  does not contain any hyperedge

crossing among  $\mathcal{R}$ . This implies that  $\widehat{f}(H) = 0$  for all  $H$  such that  $H \cap A \neq \emptyset$  for all  $A \in \mathcal{R}$ , by definition of  $G_f$  and Proposition 2. Thus, by Lemma 4 (a),

$$\mathfrak{L}^*(f, \mathcal{R}, x) = \sum_H \widehat{f}(H) \mathfrak{L}^*(\psi_H, \mathcal{R}, x),$$

where the summation is over the subsets,  $H$ 's, such that  $H \cap A = \emptyset$  for some  $A \in \mathcal{R}$ . Since the support set of  $\psi_H$  is  $H$  for any  $H \subseteq [n]$ ,  $\mathfrak{L}^*(\psi_H, \mathcal{R}, x) = 0$  for  $H$ 's in the summation by Lemma 5 and so  $\mathfrak{L}^*(f, \mathcal{R}, x) = 0$ .

Now, consider the proof of (b). Let  $\mathcal{S} = \{S_1, \dots, S_j\}$ . For a setwise subcollection  $\mathcal{R}$  of  $\mathcal{S}$ , let  $\mathcal{R} = \{R_1, \dots, R_j\}$ , where  $R_i \subseteq S_i$  for all  $1 \leq i \leq j$ . Let  $r_i = |R_i|$  and  $r = \sum_{i=1}^j r_i$ . For each  $R_i$ , set a distinct bit position  $a_i \in [n - r + j]$  and, for each  $i' \in [n] \setminus (\bigcup_i R_i)$ , set a distinct bit position  $b_{i'} \in [n - r + j]$ . For each  $H \subseteq [n]$ , define  $\varphi_{H, \mathcal{R}} : \{0, 1\}^{n-r+j} \rightarrow \mathbb{R}$  as follows: If  $|H \cap R_i|$  is odd for all  $1 \leq i \leq j$ ,

$$\varphi_{H, \mathcal{R}}(y) = (-1)^{\sum_{i=1}^j y[a_i] + \sum_{i' \in H \setminus (\bigcup_i R_i)} y[b_{i}]}$$

for any  $y \in \{0, 1\}^{n-r+j}$ . Otherwise,  $\varphi_{H, \mathcal{R}}$  is the zero function that assigns zero value to all input strings  $y \in \{0, 1\}^{n-r+j}$ . For each  $x \in \text{str}(\mathcal{R})$ , assign the string  $y_{x, \mathcal{R}} \in \{0, 1\}^{n-r+j}$  such that  $y_{x, \mathcal{R}}[a_i] = x[a]$  for some  $a \in R_i$  for all  $1 \leq i \leq j$  and  $y_{x, \mathcal{R}}[b_{i'}] = x[i']$  for all  $i' \in [n] \setminus (\bigcup_i R_i)$ . Note that  $\{y_{x, \mathcal{R}} \mid x \in \text{str}(\mathcal{R})\} = \{0, 1\}^{n-r+j}$  and the sets  $\text{str}(\mathcal{R})$  and  $\{y_{x, \mathcal{R}} \mid x \in \text{str}(\mathcal{R})\}$  are in one-to-one correspondence. Letting  $\mathcal{S}_{\mathcal{R}} = \{\{a_i\} \mid 1 \leq i \leq j\}$ , we have

**Claim 7** For any  $H \subseteq [n]$ ,

$$\mathfrak{L}^*(\psi_H, \mathcal{R}, x) = \mathfrak{L}^*(\varphi_{H, \mathcal{R}}, \mathcal{S}_{\mathcal{R}}, y_{x, \mathcal{R}})$$

for all  $x \in \text{str}(\mathcal{R})$ .

**Proof:** Suppose that  $|H \cap R_i|$  is odd for all  $1 \leq i \leq j$ . Let  $u_i$  be a bit position in  $H \cap R_i$  for  $1 \leq i \leq j$ . For all  $x \in \text{str}(\mathcal{R})$ ,  $\sum_{u \in H \cap R_i} x[u] = |H \cap R_i| \cdot x[u_i] = x[u_i] \pmod{2}$  for all  $1 \leq i \leq j$  and so

$$\begin{aligned} \psi_H(x) &= (-1)^{\sum_i \sum_{u \in H \cap R_i} x[u] + \sum_{i' \in H \setminus (\bigcup_i R_i)} x[i']} \\ &= (-1)^{\sum_i x[u_i] + \sum_{i' \in H \setminus (\bigcup_i R_i)} x[i']} \\ &= (-1)^{\sum_i y_{x, \mathcal{R}}[a_i] + \sum_{i' \in H \setminus (\bigcup_i R_i)} y_{x, \mathcal{R}}[b_{i'}]} \\ &= \varphi_{H, \mathcal{R}}(y_{x, \mathcal{R}}). \end{aligned}$$

Let  $x_{\mathcal{R}'} = x \oplus (\bigoplus_{A \in \mathcal{R}'} 1_A)$  for  $\mathcal{R}' \subseteq \mathcal{R}$ . If  $x \in \text{str}(\mathcal{R})$ ,  $x_{\mathcal{R}'} \in \text{str}(\mathcal{R})$  and  $y_{x_{\mathcal{R}'}, \mathcal{R}} = y_{x, \mathcal{R}} \oplus (\bigoplus_{i: R_i \in \mathcal{R}'} 1_{\{a_i\}})$ .

Hence, for all  $x \in \text{str}(\mathcal{R})$ ,

$$\begin{aligned}
& \mathfrak{L}^*(\psi_H, \mathcal{R}, x) \\
&= \sum_{\mathcal{R}' \subseteq \mathcal{R}} (-1)^{|\mathcal{R}'|} \psi_H \left( x \oplus \left( \bigoplus_{A \in \mathcal{R}'} 1_A \right) \right) \\
&= \sum_{\mathcal{R}' \subseteq \mathcal{R}} (-1)^{|\mathcal{R}'|} \psi_H(x_{\mathcal{R}'}) \\
&= \sum_{\mathcal{R}' \subseteq \mathcal{R}} (-1)^{|\mathcal{R}'|} \varphi_{H, \mathcal{R}}(y_{x_{\mathcal{R}'}, \mathcal{R}}) \\
&= \sum_{\mathcal{R}' \subseteq \mathcal{R}} (-1)^{|\{a_i | R_i \in \mathcal{R}'\}|} \varphi_{H, \mathcal{R}} \left( y_{x, \mathcal{R}} \oplus \left( \bigoplus_{i: R_i \in \mathcal{R}'} 1_{\{a_i\}} \right) \right) \\
&= \sum_{\mathcal{S}' \subseteq \mathcal{S}_{\mathcal{R}}} (-1)^{|\mathcal{S}'|} \varphi_{H, \mathcal{R}} \left( y_{x, \mathcal{R}} \oplus \left( \bigoplus_{B \in \mathcal{S}'} 1_B \right) \right) \\
&= \mathfrak{L}^*(\varphi_{H, \mathcal{R}}, \mathcal{S}_{\mathcal{R}}, y_{x, \mathcal{R}}).
\end{aligned}$$

Now, suppose that  $|H \cap R_i|$  is even for some  $i$ . For all  $x \in \{0, 1\}^n$ ,  $\mathfrak{L}^*(\psi_H, \mathcal{R} \setminus \{R_i\}, x) = \mathfrak{L}^*(\psi_H, \mathcal{R} \setminus \{R_i\}, x \oplus 1_{R_i})$  and so  $\mathfrak{L}^*(\psi_H, \mathcal{R}, x) = 0$  by Lemma 4 (b). Since  $\varphi_{H, \mathcal{R}}$  is the zero function, on the other hand,  $\mathfrak{L}^*(\varphi_{H, \mathcal{R}}, \mathcal{S}_{\mathcal{R}}, y) = 0$  for all  $y \in \{0, 1\}^{n-r+j}$ . Hence,

$$\mathfrak{L}^*(\psi_H, \mathcal{R}, x) = \mathfrak{L}^*(\varphi_{H, \mathcal{R}}, \mathcal{S}_{\mathcal{R}}, y_{x, \mathcal{R}})$$

for all  $x \in \text{str}(\mathcal{R})$ .  $\blacksquare$

Define the pseudo-Boolean function  $g_{f, \mathcal{R}} : \{0, 1\}^{n-r+j} \rightarrow \mathbb{R}$  by

$$g_{f, \mathcal{R}} = \sum_{H \subseteq [n]} \widehat{f}(H) \cdot \varphi_{H, \mathcal{R}}.$$

**Claim 8** For all  $x \in \text{str}(\mathcal{R})$ ,

$$\mathfrak{L}^*(f, \mathcal{R}, x) = \mathfrak{L}^*(g_{f, \mathcal{R}}, \mathcal{S}_{\mathcal{R}}, y_{x, \mathcal{R}}).$$

**Proof:** By Lemma 4 (a) and Claim 7,

$$\begin{aligned}
\mathfrak{L}^*(f, \mathcal{R}, x) &= \sum_{H \subseteq [n]} \widehat{f}(H) \cdot \mathfrak{L}^*(\psi_H, \mathcal{R}, x) \\
&= \sum_{H \subseteq [n]} \widehat{f}(H) \cdot \mathfrak{L}^*(\varphi_{H, \mathcal{R}}, \mathcal{S}_{\mathcal{R}}, y_{x, \mathcal{R}}) \\
&= \mathfrak{L}^*(g_{f, \mathcal{R}}, \mathcal{S}_{\mathcal{R}}, y_{x, \mathcal{R}}),
\end{aligned}$$

for all  $x \in \text{str}(\mathcal{R})$ .  $\blacksquare$

Suppose that  $G_f$  contains a hyperedge crossing among  $\mathcal{S}$ .

**Claim 9** Suppose that a setwise subcollection  $\mathcal{R}$  is randomly generated from  $\Gamma(\mathcal{S})$ . Then, the probability that the linkage graph of  $g_{f, \mathcal{R}}$  has the hyperedge crossing among  $\mathcal{S}_{\mathcal{R}}$  is at least  $\frac{1}{2^{k+j}}$ .

**Proof:** Since  $G_f$  contains a hyperedge crossing among  $\mathcal{S}$ , there exist subsets  $H$ 's such that  $\widehat{f}(H) \neq 0$  and  $H \cap S_i \neq \emptyset$  for all  $S_i \in \mathcal{S}$ . Among those subsets, we choose a maximal subset  $H^*$  in viewpoint of the size of intersection with  $S_i$ 's: For each  $1 \leq i \leq j$ ,  $|H^* \cap S_i| \geq |H \cap S_i|$  for any  $H$  such that  $\widehat{f}(H) \neq 0$ ,  $H \cap S_i \neq \emptyset$  for all  $S_i \in \mathcal{S}$ , and  $|H \cap S_i| =$

$|H^* \cap S_i|$  for all  $1 \leq l \leq i - 1$ . Let  $A_i$  be a set consisting of an element in  $H^* \cap S_i$  and let  $B_i = (H^* \cap S_i) \setminus A_i$ . Let  $\mathcal{R} = \{R_1, \dots, R_j\}$ , where  $R_i \subseteq S_i$  for all  $1 \leq i \leq j$ . Since  $A_i \cup B_i = H^* \cap S_i$  and  $\sum_i |A_i \cup B_i| = \sum_i |H^* \cap S_i| \leq |H^*| \leq k$ , the probability that  $R_i \supseteq A_i$  and  $R_i \not\supseteq B_i$  for all  $1 \leq i \leq j$  is at least  $\frac{1}{2^k}$ .

Consider the condition that  $R_i \supseteq A_i$  and  $R_i \not\supseteq B_i$  for all  $1 \leq i \leq j$ . Denote

$$\begin{aligned}
\mathcal{H}^* &= \{H \subseteq [n] \mid \widehat{f}(H) \neq 0, \\
&\quad H \supseteq (\cup_i B_i) \cup (H^* \setminus (\cup_i S_i)), \\
&\quad \text{and } |H \cap S_i| = |H^* \cap S_i| \text{ for all } i\}.
\end{aligned}$$

It is clear that  $H^* \in \mathcal{H}^*$ . Given the condition, if  $\varphi_{H, \mathcal{R}} = \varphi_{H^*, \mathcal{R}}$ ,  $H$  should be in  $\mathcal{H}^*$ . Thus, in the Walsh transform of  $g_{f, \mathcal{R}}$ , the Walsh coefficient corresponding to the Walsh function  $\varphi_{H^*, \mathcal{R}}$  is equal to  $\sum_H \widehat{f}(H)$ , where the summation is over  $H$ 's such that  $H \in \mathcal{H}^*$  and  $(H \cap S_i \setminus B_i) \subseteq R_i$  for all  $i$ . Since  $H^*$  was chosen in a maximal sense as mentioned, for any  $H \in \mathcal{H}^*$ ,  $|H \cap S_i \setminus B_i| = 1$  for all  $1 \leq i \leq j$ . Thus, when we choose each element in  $S_i \setminus (A_i \cup B_i)$  independently and with probability  $\frac{1}{2}$  and put it into  $R_i$ , the conditional probability that  $\sum_H \widehat{f}(H) \neq 0$ , where the summation is over  $H$ 's such that  $H \in \mathcal{H}^*$  and  $(H \cap S_i \setminus B_i) \subseteq R_i$  for all  $i$ , is at least  $\frac{1}{2^j}$ . In this case,  $\varphi_{H^*, \mathcal{R}}$  may be expressed as  $\psi_{H'}$  for  $H' \subseteq [n - r + j]$  such that

$H' = \{a_i \mid 1 \leq i \leq j\} \cup \{b_{i'} \mid i' \in (\cup_i B_i) \cup (H^* \setminus (\cup_i S_i))\}$  and the Walsh coefficient corresponding to  $\psi_{H'}$  in the Walsh transform of  $g_{f, \mathcal{R}}$  is non-zero. At this time, the linkage graph of  $g_{f, \mathcal{R}}$  has the  $j$ -hyperedge crossing among  $\mathcal{S}_{\mathcal{R}} = \{\{a_i\} \mid 1 \leq i \leq j\}$ .

Therefore, the probability that the linkage graph of  $g_{f, \mathcal{R}}$  has the hyperedge crossing among  $\mathcal{S}_{\mathcal{R}}$  for a setwise subcollection  $\mathcal{R}$  randomly generated from  $\Gamma(\mathcal{S})$  is at least  $\frac{1}{2^{k+j}}$  and the proof is completed.  $\blacksquare$

Since  $f$  is a  $k$ -bounded function,  $g_{f, \mathcal{R}}$  is also  $k$ -bounded. Thus, when the linkage graph of  $g_{f, \mathcal{R}}$  has the hyperedge crossing among  $\mathcal{S}_{\mathcal{R}}$ , the probability that  $\mathfrak{L}^*(g_{f, \mathcal{R}}, \mathcal{S}_{\mathcal{R}}, y) \neq 0$  for a string  $y$  chosen uniformly at random from  $\{0, 1\}^{n-r+j}$  is at least  $\frac{1}{2^{k-j}}$  by Proposition 1 (b). Hence, by Claim 9, the probability that  $\mathfrak{L}^*(g_{f, \mathcal{R}}, \mathcal{S}_{\mathcal{R}}, y) \neq 0$  for a setwise subcollection  $\mathcal{R}$  randomly generated from  $\Gamma(\mathcal{S})$  and a string  $y$  chosen uniformly at random from  $\{0, 1\}^{n-r+j}$  is at least  $\frac{1}{2^{2k}}$ . Since the sets  $\text{str}(\mathcal{R})$  and  $\{0, 1\}^{n-r+j} = \{y_{x, \mathcal{R}} \mid x \in \text{str}(\mathcal{R})\}$  are in one-to-one correspondence, we have the part (b) of the theorem by Claim 8.  $\blacksquare$

## 4 Finding Graphs with Cross-Membership Queries

In this section, we focus on the problem to find an unknown hypergraph with cross-membership queries under the oracle with one-sided error  $\delta$ . Recall that, given a cross-membership query, the oracle with one-sided error  $\delta$  correctly answers with probability at least  $1 - \delta$  if the true answer for the query is YES and it correctly answers with probability 1 otherwise. Section 4.1 presents a randomized algorithm for the graph finding problem. The algorithm is analyzed in Section 4.2, which induces Theorem 3.

---

```

GRAPHFINDINGALGORITHM( $n, k, \delta$ )
//  $E_j$  : the set of the hyperedges of order  $j$  found so far
//  $Q$  : the set of the vertices in the hyperedges of order  $j$  found so far
//  $W$  : the set of the vertices  $v$  such that all the hyperedges of order  $j$  containing  $v$  have been found by the algorithm
for  $j$  from 1 to  $k$ 
     $Q \leftarrow \emptyset, W \leftarrow \emptyset;$ 
     $E_j \leftarrow \emptyset;$ 
    repeat
         $(S_i)_{i=1}^j \leftarrow \text{CHECKEXISTENCE}(\emptyset, W, j);$ 
        if  $(S_i)_{i=1}^j = \text{NULL}$ , break;
         $v \leftarrow \text{BINARYSEARCH}((S_i)_{i=1}^j, 1);$ 
         $Q \leftarrow Q \cup \{v\};$ 
        while  $Q \setminus W \neq \emptyset$ 
            choose a vertex  $v$  in  $Q \setminus W$ ;
             $E_{v,j} \leftarrow \text{FINDHYPEREDGES}(\{v\}, W, j);$ 
             $E_j \leftarrow E_j \cup E_{v,j};$ 
             $Q \leftarrow Q \cup \left( \bigcup_{H \in E_{v,j}} H \right);$ 
             $W \leftarrow W \cup \{v\};$ 
     $E \leftarrow \bigcup_{j=1}^k E_j;$ 
    return  $E$ ;

```

---

Figure 1: Main procedure of the algorithm GFA (The output of GFA is the set of the hyperedges of the input graph that have been found. For the subprocedures, CHECKEXISTENCE, BINARYSEARCH, and FINDHYPEREDGES, see Figures 2, 3, and 4, respectively.)

#### 4.1 Algorithm for Finding Graphs

In this section, we present the algorithm to find an unknown hypergraph with cross-membership queries under the oracle with one-sided error  $\delta$ , the *Graph Finding Algorithm* (GFA). The algorithm GFA takes three arguments: The number of vertices of the unknown hypergraph  $n$ , the order of the hypergraph  $k$ , and the error bound for the answer of the oracle  $0 \leq \delta < 1$ . It returns the set of the hyperedges of the hypergraph that have found. The algorithm GFA consists of the main procedure GRAPHFINDINGALGORITHM (Figure 1) and the three subprocedures CHECKEXISTENCE (Figure 2), BINARYSEARCH (Figure 3), and FINDHYPEREDGES (Figure 4). In the pseudocode, the values of  $n$ ,  $k$ , and  $\delta$  can be accessed by any procedure. All other variables are local to the given procedure.

Suppose that  $G$  is an unknown hypergraph given to GFA and let  $G_j$  be the induced subgraph of  $G$  consisting of the hyperedges of order  $j$  for  $1 \leq j \leq k$ . The algorithm GFA successively finds the hyperedges of  $G_1$ ,  $G_2$ , and so on. After the algorithm finally finds the hyperedges of  $G_k$ , it returns all the hyperedges found so far. To find the hyperedges of  $G_j$  for  $j = 1, \dots, k$ , the algorithm iteratively checks whether there is a hyperedge of order  $j$  that has not been found and, if such a hyperedge exists, the algorithm finds all the hyperedges in the connected component that the hyperedge belongs to. It continues this process until there is no more hyperedge that can be found.

In the main procedure GRAPHFINDINGALGORITHM, the variable  $Q$  contains the vertices in the hyperedges found so far. The variable  $W$  contains the vertices  $v$  such that all the hyperedges of order  $j$  containing  $v$  have been found by the

algorithm. The variable  $E_j$  contains the hyperedges of order  $j$  found so far. To check the existence of a new connected component of two or more vertices in the subgraph consisting of the hyperedges of order  $j$ , GRAPHFINDINGALGORITHM calls the subprocedure CHECKEXISTENCE.

Given sets of vertices  $U$  and  $W$  and a positive integer  $j$ , the procedure CHECKEXISTENCE performs a randomized test for whether there is a hyperedge of order  $j$  that contains all the vertices in  $U$  and does not contain the vertices in  $W$ . For the purpose, it iteratively generates a collection of disjoint sets of vertices  $(S_i)_{i=1}^j$  for a cross-membership query as follows. Letting  $U = \{v_1, \dots, v_{|U|}\}$ , the set  $S_i$  is fixed with  $S_i = \{v_i\}$  for  $1 \leq i \leq |U|$ . The sets  $S_{|U|+1}, \dots, S_j$  are generated as a uniform random partition of vertices in  $[n] \setminus (U \cup W)$ . If the oracle answers YES for the cross-membership query with some  $(S_i)_{i=1}^j$ , there is a hyperedge of order  $j$  crossing among  $S_i$ 's, which contains the vertices in  $U$  and does not contain the vertices in  $W$ . In this case, CHECKEXISTENCE returns the generated sets  $(S_i)_{i=1}^j$ . If the oracle answers NO for all the generated collections of disjoint sets, CHECKEXISTENCE returns NULL regarding that there is no such a hyperedge.

If CHECKEXISTENCE returns NULL, GRAPHFINDINGALGORITHM regards that there is no hyperedge of order  $j$  and continues to find the hyperedges of order  $j + 1$ . If CHECKEXISTENCE returns a (non-NULL) collection of disjoint sets of vertices, this implies that there is a hyperedge of order  $j$ . To find a vertex in the hyperedge, GRAPHFINDINGALGORITHM calls the subprocedure BINARYSEARCH. Given a collection of disjoint sets of vertices  $(S_i)_{i=1}^j$  and a positive integer  $r$  between 1 and  $j$ , the procedure BINARY-

---

```

CHECKEXISTENCE( $U, W, j$ )
  label the vertices in  $U$  as  $v_1, \dots, v_{|U|}$ ;
  for  $i$  from 1 to  $|U|$ 
     $S_i \leftarrow \{v_i\}$ ;
  for  $i$  from  $|U| + 1$  to  $j$ 
     $S_i \leftarrow \emptyset$ ;
  repeat  $\lceil \frac{e^j \sqrt{j+1}}{1-\delta} \log n \rceil$  times
    for each  $v \in [n] \setminus (U \cup W)$ 
      choose  $i$  uniformly at random from  $\{|U| + 1, \dots, j\}$ ;
       $S_i \leftarrow S_i \cup \{v\}$ ;
    if  $\text{CMQ}(S_1, \dots, S_j) = \text{YES}$ 
      return  $(S_i)_{i=1}^j$ ;
  return NULL;

```

---

Figure 2: Procedure to check the existence of a hyperedge of order  $j$  that contains all the vertices in  $U$  and does not contain the vertices in  $W$  (Here,  $\text{CMQ}((S_i)_{i=1}^j)$  is the answer of the oracle for the cross-membership query  $(S_i)_{i=1}^j$ .)

---

```

BINARYSEARCH( $(S_i)_{i=1}^j, r$ )
  if  $|S_r| = 1$ , return the vertex in  $S_r$ ;
  repeat  $\lceil \frac{6(j+1)}{1-\delta} \log n \rceil$  times
    choose a subset  $S'_r$  of  $S_r$  uniformly at random among the subsets of order  $\lfloor \frac{|S_r|}{2} \rfloor$ ;
    if  $\text{CMQ}(S_1, \dots, S_{r-1}, S'_r, S_{r+1}, \dots, S_j) = \text{YES}$ ,
       $S_r \leftarrow S'_r$ ;
      if  $|S_r| = 1$ , return the vertex in  $S_r$ ;
  return a vertex in  $S_r$ ;

```

---

Figure 3: Procedure to search a vertex in  $S_r$  that is contained in a hyperedge of order  $j$  crossing among  $S_1, \dots, S_j$  (Here,  $\text{CMQ}((S_i)_{i=1}^j)$  is the answer of the oracle for the cross-membership query  $(S_i)_{i=1}^j$ .)

SEARCH returns a vertex that is in  $S_r$  and in one of the hyperedges crossing among  $S_i$ 's. Among the subsets of  $S_r$  of order  $\lfloor \frac{|S_r|}{2} \rfloor$ , it chooses a subset  $S'_r$  uniformly at random. For the sets of vertices  $(S_i)_{i=1}^j$  in which  $S_r$  is replaced with  $S'_r$ , it asks the cross-membership query to check whether there is a hyperedge crossing among the sets. If the answer of the oracle is YES, i.e., if it turns out that there is a hyperedge crossing among the sets, it replaces  $S_r$  with  $S'_r$ . The procedure BINARYSEARCH repeats this process at most a specified number of times until there remains one vertex in  $S_r$ . If there remains one vertex in  $S_r$  before the specified number of iterations, BINARYSEARCH returns the vertex. Otherwise, it fails to exactly search the desired vertex and returns an arbitrary vertex in  $S_r$ .

Once a vertex in the new connected component is found by BINARYSEARCH, GRAPHFINDINGALGORITHM puts the vertex into  $Q$  and repeats the following process while  $Q \setminus W \neq \emptyset$ . It chooses a vertex  $v$  in  $Q \setminus W$  and finds all the hyperedges of order  $j$  containing  $v$  by calling the subprocedure FINDHYPEREDGES. Given two sets of vertices  $U$  and  $W$  and a positive integer  $j$ , FINDHYPEREDGES returns the set of the hyperedges of order  $j$  that contain the vertices in  $U$  and do not contain the vertices in  $W$ . In the procedure FINDHYPEREDGES, the variable  $A$  contains the vertices such that the desired hyperedges of order  $j$  containing the vertices in  $A$

have been found. Initially,  $A$  is set to be empty. If  $|U| = j$ ,  $U$  is the only hyperedge of order  $j$  containing the vertices in  $U$  and FINDHYPEREDGES returns the set consisting of  $U$ . Otherwise, it recursively finds the desired hyperedges of order  $j$  as follows. By calling CHECKEXISTENCE, it first checks whether there is a hyperedge of order  $j$  that contains the vertices in  $U$  and does not contain the vertices in  $W$ . If CHECKEXISTENCE returns NULL, FINDHYPEREDGES regards that there is no such a hyperedge and returns the set of the hyperedges found so far. Otherwise, it chooses a vertex  $v$  in the hyperedge by calling BINARYSEARCH. Then, it finds the hyperedges of order  $j$  that contain the vertices in  $U \cup \{v\}$  and does not contain the vertices in  $W \cup A$  by calling FINDHYPEREDGES recursively. After that, it puts  $v$  into  $A$  and continues to find the desired hyperedges of order  $j$  not containing the vertices in  $A$ .

After all the hyperedges of order  $j$  containing  $v$  are found, they are put into  $E_j$ . The vertices contained in the hyperedges are put into  $Q$  to mark that they are in the connected component being searched. The vertex  $v$  is put into  $W$  to prevent the hyperedges of order  $j$  containing  $v$  from being searched again.

## 4.2 Algorithm Analysis

In this section, we analyze the algorithm GFA to obtain Theorem 3. We first analyze the number of cross-membership

---

```

FINDHYPEREDGES( $U, W, j$ )
  if  $|U| = j$ , return  $\{U\}$ ;
   $E_{U,j} \leftarrow \emptyset, A \leftarrow \emptyset$ ;
  repeat
     $(S_i)_{i=1}^j \leftarrow \text{CHECKEXISTENCE}(U, W \cup A, j)$ ;
    if  $(S_i)_{i=1}^j = \text{NULL}$ , break;
     $v \leftarrow \text{BINARYSEARCH}((S_i)_{i=1}^j, |U| + 1)$ ;
     $E_{U,j} \leftarrow E_{U,j} \cup \text{FINDHYPEREDGES}(U \cup \{v\}, W \cup A, j)$ ;
     $A \leftarrow A \cup \{v\}$ ;
  return  $E_{U,j}$ ;

```

---

Figure 4: Procedure to find the hyperedges of order  $j$  that contain all the vertices in  $U$  and do not contain the vertices in  $W$

queries used in GFA.

**Lemma 10** *Suppose that  $G$  is an unknown  $k$ -bounded hypergraph with  $n$  vertices and  $m$  hyperedges for constant  $k$ . Then, for any constant  $0 \leq \delta < 1$ , GFA uses  $\mathcal{O}(m \log n)$  cross-membership queries for  $G$  under the oracle with one-sided error  $\delta$ .*

**Proof:** Omitted. ■

To analyze the error probability of GFA, we need a large deviation result for a sum of independent random variables following geometric distributions. A random variable  $X$  follows the geometric distribution with parameter  $p$  if, for a coin of which HEAD appears with probability  $p$ ,  $X$  is the number of coin tosses until the first HEAD appears. It is easy to show that the expectation of  $X$  is  $\frac{1}{p}$ . We obtain the desired result by using the Chernoff bound as follows [Che52, MR95].

**Proposition 3** *Suppose that, for some  $0 < p \leq 1$ ,  $X_1, \dots, X_\ell$  are independent random variables such that  $\Pr[X_i = 1] = p$  and  $\Pr[X_i = 0] = 1 - p$  for all  $1 \leq i \leq \ell$ . Let  $X = \sum_{i=1}^{\ell} X_i$ . Then, for any  $0 \leq \alpha < 1$ ,*

$$\Pr[X \leq (1 - \alpha)E[X]] \leq \exp\left(-\frac{E[X]\alpha^2}{2}\right).$$

Now, we present the result for a sum of independent random variables following geometric distributions.

**Lemma 11** *Suppose that, for some  $0 < p \leq 1$ ,  $X_1, \dots, X_\ell$  are independent random variables each of which follows the geometric distribution with parameter  $p$ . Let  $X = \sum_{i=1}^{\ell} X_i$ . Then, for any  $\alpha > 0$ ,*

$$\Pr[X > (1 + \alpha)E[X]] \leq \exp\left(-\frac{\alpha^2 \ell}{2(1 + \alpha)}\right).$$

**Proof:** Omitted. ■

**Lemma 12** *Suppose that  $G$  is an unknown  $k$ -bounded hypergraph with  $n$  vertices and  $m$  hyperedges for constant  $k$ . Then, for any  $0 \leq \delta < 1$ , GFA correctly finds the hyperedges of  $G$  with probability  $1 - \mathcal{O}\left(\frac{1}{n}\right)$  under the oracle with one-sided error  $\delta$ .*

**Proof:** We will show that the probability that GFA does not find all the hyperedges of  $G_j$  is  $\mathcal{O}\left(\frac{1}{n}\right)$  for each  $j$  with  $1 \leq j \leq k$ . Then, the lemma follows by the union bound.

We first consider the probability that CHECKEXISTENCE performs incorrectly for given arguments  $U, W$ , and  $j$ . Suppose that there is no hyperedge of order  $j$  in  $G$  that contains the vertices in  $U$  and does not contain the vertices in  $W$ . In this case, CHECKEXISTENCE returns NULL and the probability of CHECKEXISTENCE being incorrect is zero. Suppose that there is a hyperedge of order  $j$  in  $G$  that contains the vertices in  $U$  and does not contain the vertices in  $W$ . Let  $U = \{v_1, \dots, v_{|U|}\}$  and let the hyperedge of order  $j$  be  $\{v_1, \dots, v_{|U|}, v_{|U|+1}, \dots, v_j\}$ . The probability that  $v_{|U|+1}, \dots, v_j$  are put into different  $S_i$ 's is  $\frac{(j-|U|)!}{(j-|U|)^{j-|U|}}$ . When  $v_{|U|+1}, \dots, v_j$  are put into different  $S_i$ 's, the probability that the oracle answers YES for the cross-membership query  $(S_i)_{i=1}^j$  is at least  $1 - \delta$ . Thus, for each iteration of the repeat loop in CHECKEXISTENCE, the probability that the hyperedge is not detected is at most  $1 - \frac{(j-|U|)!}{(j-|U|)^{j-|U|}}(1 - \delta)$ . Hence, the probability that the hyperedge is not detected for  $\lceil \frac{e^j \sqrt{j+1}}{1-\delta} \log n \rceil$  iterations of the repeat loop is at most

$$\left(1 - \frac{(j-|U|)!}{(j-|U|)^{j-|U|}}(1 - \delta)\right)^{\frac{e^j \sqrt{j+1}}{1-\delta} \log n}.$$

By using the fact that  $1 - x \leq e^{-x}$  for any real  $x$ , this value is at most

$$\exp\left(-\frac{(j-|U|)!e^j \sqrt{j+1}}{(j-|U|)^{j-|U|}} \log n\right).$$

After some calculation using the facts that  $\frac{(j-|U|)!}{(j-|U|)^{j-|U|}} \geq \frac{j!}{j^j}$  and  $j! > \sqrt{2\pi j} \left(\frac{j}{e}\right)^j e^{\frac{1}{12j+1}}$ , we have

$$\begin{aligned} & \exp\left(-\frac{(j-|U|)!e^j \sqrt{j+1}}{(j-|U|)^{j-|U|}} \log n\right) \\ & \leq \exp(-(j+1) \log n) \\ & = \frac{1}{n^{j+1}}. \end{aligned}$$

Thus, the probability of CHECKEXISTENCE being incorrect is at most  $\frac{1}{n^{j+1}}$ .

Now, we bound the probability that BINARYSEARCH performs incorrectly for given arguments  $(S_i)_{i=1}^j$  and  $r$ . To

this end, we consider an imaginary procedure BS' that is the same as BINARYSEARCH except that, in the procedure BS', the repeat loop continues until the size of  $S_r$  becomes one. In the repeat loop of BS',  $S_r$  is iteratively halved and updated. Suppose that the size of  $S_r$  becomes one after  $S_r$  is halved and updated  $t$  times. For  $1 \leq i \leq t$ , let  $X_i$  be the number of iterations of the repeat loop between the  $(i-1)^{\text{th}}$  update and the  $i^{\text{th}}$  update of  $S_r$ . Let  $v$  be a vertex of a hyperedge crossing among  $S_i$ 's that is in the initial  $S_r$ . When  $v$  is in the  $(i-1)$  times updated  $S_r$ , the probability that  $v$  is chosen as an element of  $S_r$  is at least  $\frac{1}{3}$ . (The extreme case is when the order of  $S_r$  is three.) Thus,  $X_i$  follows a geometric distribution with the parameter at least  $\frac{1}{3}(1-\delta)$ . If we let  $X = \sum_{i=1}^t X_i$ , by linearity of expectation,

$$\mathbb{E}[X] \leq \frac{3t}{1-\delta}.$$

Thus,

$$\begin{aligned} \Pr \left[ X > \frac{6(j+1)}{1-\delta} \log n \right] \\ &= \Pr \left[ X > \left( \frac{2(j+1) \log n}{t} \right) \left( \frac{3t}{1-\delta} \right) \right] \\ &\leq \Pr \left[ X > \left( \frac{2(j+1) \log n}{t} \right) \mathbb{E}[X] \right]. \end{aligned}$$

Since  $X_i$ 's are independent, letting  $1 + \alpha = \frac{2(j+1) \log n}{t}$ , we apply Lemma 11 to the above inequality to obtain

$$\begin{aligned} \Pr \left[ X > \frac{6(j+1)}{1-\delta} \log n \right] &\leq \exp \left( -\frac{\alpha^2 t}{2(1+\alpha)} \right) \\ &\leq \exp \left( -(j+1) \log n \right) \\ &= \frac{1}{n^{j+1}}. \end{aligned}$$

Thus, the probability of BINARYSEARCH performing incorrectly is at most  $\frac{1}{n^{j+1}}$  as it is at most the probability of  $X$  being more than  $\lceil \frac{6(j+1)}{1-\delta} \log n \rceil$ .

The number of CHECKEXISTENCE and BINARYSEARCH being called for GFA to find the hyperedges of  $G_j$  are at most  $j^2 m$ , respectively. Thus, in the process of GFA finding the hyperedges of  $G_j$ , the probability that CHECKEXISTENCE or BINARYSEARCH incorrectly perform once or more times is at most  $\frac{2j^2 m}{n^{j+1}} \leq \frac{2j^2 n^j}{n^{j+1}} = \frac{2j^2}{n}$ , which is  $\mathcal{O}(\frac{1}{n})$  since  $j \leq k$  for constant  $k$ . This means that, with probability  $1 - \mathcal{O}(\frac{1}{n})$ , CHECKEXISTENCE and BINARYSEARCH performs correctly throughout the process of GFA finding the hyperedges of  $G_j$ .

Suppose the condition that CHECKEXISTENCE and BINARYSEARCH correctly perform throughout the process of GFA finding the hyperedges of  $G_j$ . We show that, given  $U$ ,  $W$ , and  $j$ , FINDHYPEREDGES correctly return the set of the hyperedges of order  $j$  containing the vertices in  $U$  and not containing the vertices in  $W$ . Suppose that, for any  $u \in A$ , the hyperedges of order  $j$  containing the vertices in  $U \cup \{u\}$  and not containing the vertices in  $W$  have been found by FINDHYPEREDGES. At this time, any hyperedge that has not been found is a hyperedge containing the vertices in  $U \cup \{v\}$  and not containing the vertices in  $W \cup A$  for

some  $v \notin U \cup W \cup A$ . Thus, it must be found by a recursive call of FINDHYPEREDGES later.

Returning to the main procedure GRAPHFINDINGALGORITHM, for each vertex  $v \in [n]$ , the hyperedges of order  $j$  containing  $v$  are found by FINDHYPEREDGES in the while loop and so all the hyperedges of  $G_j$  are found by GFA. It is clear that the set of the hyperedges of order  $j$  returned by GFA is included in the set of the hyperedges of  $G_j$ . Thus, GFA finds the hyperedges of  $G_j$  correctly, given the condition that CHECKEXISTENCE and BINARYSEARCH correctly perform. Therefore, GFA correctly finds the hyperedges of  $G_j$  with probability  $1 - \mathcal{O}(\frac{1}{n})$ . ■

Theorem 3 follows from Lemmas 10 and 12. Here, we mention that it is more straightforward to obtain  $\mathcal{O}(m \log^2 n)$  algorithm for the hypergraph finding problem (and hence  $\mathcal{O}(m \log^2 n)$  algorithm for finding the Fourier coefficients) by querying the oracle  $\Theta(\log n)$  times for each cross-membership query to make the error probability  $\mathcal{O}(1/\text{poly}(n))$ .

For the  $k$ -bounded hypergraph finding problem, it is not difficult to show that any randomized algorithm requires  $\Omega(m \log n)$  cross-membership queries for constant  $k$  to make the error probability at most a given constant, provided that  $m \leq n^{k-\varepsilon}$  for any constant  $\varepsilon > 0$ . (To obtain the lower bound, we may use Yao's minimax principle [Yao77] and the information-theoretic arguments based on the fact that, for a cross-membership query, the oracle returns one of two values.) Thus, GFA is optimal up to a constant factor, provided that  $m \leq n^{k-\varepsilon}$  for any constant  $\varepsilon > 0$ . Note that this does not mean the optimality of the proposed algorithm for the problem of finding Fourier coefficients. While the oracle for the hypergraph finding problem gives binary values, function evaluations for the problem of finding Fourier coefficients give real values that may give more information about the Fourier coefficients.

## 5 Remarks on Query and Time Complexity

Suppose that we are given a  $k$ -bounded function  $f$  defined on  $\{0, 1\}^n$  with  $m$  non-zero Fourier coefficients. To find the Fourier coefficients of  $f$ , we first find the hyperedges of the linkage graph of  $f$ . From Theorem 6, we have the oracle with one-sided error  $\delta = 1 - \frac{1}{2^{2k}}$  that gives the answer for a cross-membership query by using  $2^k$  function evaluations. Since  $f$  has  $m$  non-zero Fourier coefficients, the linkage graph of  $f$  has at most  $2^k m$  hyperedges. Given a  $k$ -bounded hypergraph with  $n$  vertices and at most  $2^k m$  hyperedges, GFA uses  $\mathcal{O}\left(\frac{(2e)^k k^{3.5}}{1-\delta} m \log n\right)$  cross-membership queries as shown in the proof of Lemma 10. Thus, we can find the hyperedges of the linkage graph of  $f$  (with high probability) by using  $\mathcal{O}\left((16e)^k k^{3.5} m \log n\right)$  function evaluations.

Once the linkage graph of  $f$  is obtained, the Fourier coefficients can be found by using  $\mathcal{O}(2^k m)$  additional function evaluations from Proposition 2. Thus, the overall query complexity of finding the Fourier coefficients of  $f$  (with high probability) is  $\mathcal{O}\left((16e)^k k^{3.5} m \log n\right)$ . This is  $\mathcal{O}(m \log n)$  for constant  $k$  and Theorem 1 follows. Another important issue in practical applications is the time complexity of the algorithm. From the pseudocode of the proposed algorithm, we can check that the time complexity of the algorithm is

$\mathcal{O}(nm \log n)$  for constant  $k$ . (It is exponential in  $k$ .)

We should note that GFA does not assume the hierarchical property among the hyperedges. The query complexity of GFA can be improved for the restricted class of the  $k$ -bounded hypergraphs with the hierarchical property. Thus, the query complexity of finding the Fourier coefficients of a  $k$ -bounded function can be improved for general  $k$ . More concretely, to find the hyperedges of order  $j$ , we consider only the subsets of order  $j$  that contain some hyperedge of order  $j - 1$  that have been already found. This reduces it to  $\mathcal{O}\left(\frac{j}{1-\delta} \log n\right)$  the number of iterations of the repeat loop in CHECKEXISTENCE for checking the existence of a hyperedge of order  $j$ . (It also reduces the number of CHECKEXISTENCE and BINARYSEARCH being called to  $\mathcal{O}(km)$ .) By this modification, the query complexity of GFA for finding a  $k$ -bounded hypergraph with  $n$  vertices and at most  $2^k m$  hyperedges is reduced to  $\mathcal{O}\left(\frac{2^k k^2}{1-\delta} m \log n\right)$ . If we use this modified version of GFA, the query complexity of finding the Fourier coefficients is to be  $\mathcal{O}\left((16)^k k^2 m \log n\right)$  for a  $k$ -bounded function defined on  $\{0, 1\}^n$  with  $m$  non-zero Fourier coefficients.

## 6 Conclusion

In this paper, we showed that the Fourier coefficients of a  $k$ -bounded function with  $m$  non-zero Fourier coefficients can be found in  $\mathcal{O}(m \log n)$  function evaluations for constant  $k$ . To this end, we first showed that the problem of finding the Fourier coefficients of a  $k$ -bounded function is reduced to the problem of finding a  $k$ -bounded hypergraph with cross-membership queries under the oracle with one-sided error. Then, we gave a randomized algorithm for the hypergraph finding problem and analyzed it to obtain the desired bound.

As shown in the previous section, the query (and time) complexity of the proposed algorithm is exponential in  $k$ . Although the main concern of this paper is the case when  $k$  is constant, it would be worth trying to find an algorithm with better query (and time) complexity for general  $k$ .

## References

- [AA04] N. Alon and V. Asodi. Learning a hidden subgraph. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP 2004)*, pages 110–121, 2004.
- [AA05] N. Alon and V. Asodi. Learning a hidden subgraph. *SIAM Journal on Discrete Mathematics*, 18(4):697–712, 2005.
- [ABK<sup>+</sup>02] N. Alon, R. Beigel, S. Kasif, S. Rudich, and B. Sudakov. Learning a hidden matching. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2002)*, pages 197–206, 2002.
- [ABK<sup>+</sup>04] N. Alon, R. Beigel, S. Kasif, S. Rudich, and B. Sudakov. Learning a hidden matching. *SIAM Journal on Computing*, 33(2):487–501, 2004.
- [AC04] D. Angluin and J. Chen. Learning a hidden graph using  $\mathcal{O}(\log n)$  queries per edge. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT 2004)*, pages 210–223, 2004.
- [AC05] D. Angluin and J. Chen. Learning a hidden hypergraph. In *Proceedings of the 18th Annual Conference on Learning Theory (COLT 2005)*, pages 561–575, 2005.
- [AC06] D. Angluin and J. Chen. Learning a hidden hypergraph. *Journal of Machine Learning Research*, 7:2215–2236, 2006.
- [BAA<sup>+</sup>01] R. Beigel, N. Alon, M. S. Apaydin, L. Fortnow, and S. Kasif. An optimal procedure for gap closing in whole genome shotgun sequencing. In *Proceedings of the Fifth Annual International Conference on Computational Molecular Biology (RECOMB 2001)*, pages 22–30, 2001.
- [BGK05] M. Bouvel, V. Grebinski, and G. Kucherov. Combinatorial search on graphs motivated by bioinformatics applications: A brief survey. In *the 31st International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2005)*, pages 16–27, 2005.
- [BJT04] N. H. Bshouty, J. C. Jackson, and C. Tamon. More efficient PAC-learning of DNF with membership queries under the uniform distribution. *Journal of Computer and System Sciences*, 68(1):205–234, 2004.
- [BT96] N. H. Bshouty and C. Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996.
- [CC06] D. J. Coffin and C. D. Clack. gLINC: Identifying composability using group perturbation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1133–1140, 2006.
- [Che52] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.
- [CJM08] S. S. Choi, K. Jung, and B. R. Moon. Lower and upper bounds for linkage discovery. *IEEE Trans. on Evolutionary Computation*, 2008. In press.
- [CK08] S. S. Choi and J. H. Kim. Optimal query complexity bounds for finding graphs. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC 2008)*, 2008. To appear.
- [dBIV97] J. S. de Bonet, C. L. Isbell, Jr., and P. Viola. MIMIC: Finding optima by estimating probability densities. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 9, pages 424–430. The MIT Press, 1997.
- [Ewe79] W. Ewens. *Mathematical Population Genetics*. Springer Verlag, 1979.
- [FL70] I. Franklin and R. Lewontin. Is the gene the unit of selection? *Genetics*, 65:707–734, 1970.
- [FSBB<sup>+</sup>93] W. Fontana, P. Stadler, E. Bornberg-Bauer, T. Griesmacher, I. Hofacker, M. Tacker, P. Tarazona, E. Weinberger, and P. Schuster. RNA

- folding and combinatorial landscapes. *Physical Review E*, 47(3):2083–2099, 1993.
- [GJS76] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.
- [GK00] V. Grebinski and G. Kucherov. Optimal reconstruction of graphs under the additive model. *Algorithmica*, 28:104–124, 2000.
- [Gol89] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989.
- [HG97] G. R. Harik and D. E. Goldberg. Learning linkage. In *Foundations of Genetic Algorithms*, volume 4, pages 247–262. Morgan Kaufmann, 1997.
- [HW99] R. B. Heckendorn and D. Whitley. Predicting epistasis directly from mathematical models. *Evolutionary Computation*, 7(1):69–101, 1999.
- [HW03] R. B. Heckendorn and A. H. Wright. Efficient linkage discovery by limited probing. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003)*, pages 1003–1014, 2003.
- [HW04] R. B. Heckendorn and A. H. Wright. Efficient linkage discovery by limited probing. *Evolutionary Computation*, 12(4):517–545, 2004.
- [Jac97] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):42–65, 1997.
- [Kau89] S. A. Kauffman. Adaptation on rugged fitness landscapes. In D. Stein, editor, *Lectures in the Sciences of Complexity*, pages 527–618. Addison Wesley, 1989.
- [Kau93] S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
- [KL87] S. A. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128:11–45, 1987.
- [KM93] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.
- [KP01] H. Kargupta and B. Park. Gene expression and fast construction of distributed evolutionary representation. *Evolutionary Computation*, 9(1):1–32, 2001.
- [Lew74] R. Lewontin. *The Genetic Basis of Evolutionary Change*. Columbia University Press, 1974.
- [Man94] Y. Mansour. Learning Boolean functions via the Fourier transform. In V. Roychowdhury, K. Y. Siu, and A. Orlicsky, editors, *Theoretical Advances in Neural Computation and Learning*, pages 391–424. Kluwer Academic, 1994.
- [MG99] M. Munetomo and D. E. Goldberg. Identifying linkage groups by nonlinearity/non-monotonicity detection. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 433–440, 1999.
- [MM99] H. Mühlenbein and T. Mahnig. FDA – A scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(1):45–68, 1999.
- [MP89] C. A. Macken and A. S. Perelson. Protein evolution on rugged landscapes. In *Proceedings of the National Academic of Science, USA*, volume 86, pages 6191–6195, 1989.
- [MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [PG00] M. Pelikan and D. E. Goldberg. Hierarchical problem solving by the Bayesian optimization algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 267–274, 2000.
- [PGCP00] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. Linkage problem, distribution estimation, and Bayesian networks. *Evolutionary Computation*, 8(3):311–340, 2000.
- [RS07] L. Reyzin and N. Srivastava. Learning and verifying graphs using queries with a focus on edge counting. In *Proceedings of the 18th International Conference on Algorithmic Learning Theory (ALT 2007)*, pages 285–297, 2007.
- [Str04] M. J. Streeter. Upper bounds on the time and space complexity of optimizing additively separable functions. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004)*, pages 186–197, 2004.
- [Wal23] J. L. Walsh. A closed set of orthogonal functions. *American Journal of Mathematics*, 55:5–24, 1923.
- [Yao77] A. C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science*, pages 222–227, 1977.

---

# Teaching Dimensions Based on Cooperative Learning

---

Sandra Zilles<sup>1</sup>, Steffen Lange<sup>2</sup>, Robert Holte<sup>1</sup>, and Martin Zinkevich<sup>3</sup>

<sup>1</sup> University of Alberta, Dept. of Computing Science, Edmonton, AB, Canada, {zilles, holte}@cs.ualberta.ca

<sup>2</sup> Darmstadt University of Applied Sciences, Dept. of Computer Science, Darmstadt, Germany, s.lange@fbi.h-da.de

<sup>3</sup> Yahoo! Research, Mission College, CA, USA, maz@yahoo-inc.com

## Abstract

The problem of how a teacher and a learner can cooperate in the process of learning concepts from examples in order to minimize the required sample size without “coding tricks” has been widely addressed, yet without achieving teaching and learning protocols that meet what seems intuitively an optimal choice for selecting samples in teaching.

We introduce the model of subset teaching sets, based on the idea that both teacher and learner can exploit the assumption that the partner is cooperative. We show how this can reduce the sample size drastically without using coding tricks. For instance, monomials can be taught with only two examples independent of the number of variables.

The corresponding variant of the teaching dimension (STD) turns out to be nonmonotonic with respect to subclasses of concept classes. We discuss why this nonmonotonicity might be inherent in optimal cooperative teaching scenarios. Nevertheless, trying to overcome nonmonotonicity, we introduce a second variant, the recursive teaching dimension (RTD), which is monotonic and yields the same positive results for some concept classes, such as the class of all monomials, yet can be arbitrarily worse than the STD.

## 1 Introduction

### 1.1 Motivation and approach

One major branch of learning theory and machine learning is the theory and practice of learning concepts from examples. Considering a finite instance space and a class of (thus finite) concepts over that space, it is obvious that each concept can be uniquely determined if enough examples are known. Much less obvious is how to minimize the number of examples required to identify a concept, and with this aim in mind models of *cooperative learning* and learning from *good examples* were designed and analyzed. The selection of good examples to be presented to a learner is often modeled using a teaching device (teacher) that is assumed to be benevolent by selecting examples expediting the learning process (see for instance [AK97, JT92, GM96, Mat97]).

Throughout this paper we assume that teaching/learning proceeds stepwise; in each step the teacher presents an example (that is, an instance paired with a label 1 or 0, according to whether or not the instance belongs to the target concept) to the learner and the learner returns a concept it believes to be the target concept. If the learner’s conjecture is right the process ends, otherwise both proceed to the next step. This process will terminate successfully for any concept  $c$  in a given concept class  $C$  if the following three conditions hold: (1) the teacher never presents any example twice, (2) the teacher labels the examples correctly according to the current target concept, and (3) the learner always returns a concept consistent with the examples seen so far. The sample size, i.e., the number of examples the teacher presents to the learner enroute to termination, is the object of optimization; in particular we are concerned with the worst case sample size measured over all concepts in  $C$ . Other than that, computational complexity issues are not the focus of this paper.

A typical question is *How can a teacher and a learner cooperatively minimize the worst case sample size without using coding tricks?*—a coding trick being, e.g., any *a priori* agreement on encoding concepts in examples, depending on the concept class  $C$ . For instance, if teacher and learner agreed on a specific order for the concept representations and the instances and agreed to use the  $j^{\text{th}}$  instance in this ordering to teach the  $j^{\text{th}}$  concept, that would be a coding trick.<sup>1</sup>

A considerable amount of the learning theory literature deals with the teaching dimension of concept classes (and variants thereof, see, e.g., [SM91, GK95, ABCS92]). The teaching dimension of a concept  $c \in C$  is the size of the minimum sample that is consistent with  $c$  but not with any other concept in  $C$ . Obviously teacher and learner can succeed with such a sample without coding tricks.

The teaching dimension however does not always seem to capture the intuitive idea of cooperation in teaching and learning. Consider the following simple example. Let  $C_0$  consist of the empty concept and all singleton concepts over a given instance space  $X = \{x_1, \dots, x_n\}$ . Each singleton concept  $\{x_i\}$  has a teaching dimension of 1, since the single positive example  $(x_i, 1)$  is sufficient for determining

---

<sup>1</sup>There is so far no generally accepted definition of what a coding trick (sometimes also called “collusion”) in general is. The reader is referred to [AK97, OS02, GM96] for a treatment of this question in different learning models.

$\{x_i\}$ . In contrast to that, the empty concept has a teaching dimension of  $n$ —every example has to be presented. However, if the learner assumed the teacher was cooperative—and would therefore present a positive example if the target concept was non-empty—the learner could confidently conjecture the empty concept upon seeing just one negative example.

Let us extend this reasoning to a slightly more complex example, the class of all boolean functions that can be represented as a monomial over  $m$  variables ( $m = 4$  in this example). Imagine yourself in the role of a learner knowing your teacher will present helpful examples. If the teacher sent you the examples

$$(0100, 1), (0111, 1),$$

what would be your conjecture? Presumably most people would conjecture the monomial  $M \equiv \bar{v}_1 \wedge v_2$ , as does for instance the algorithm proposed in [Val84]. Note that this choice is not uniquely determined by the data: the empty monomial and the monomials  $\bar{v}_1$  and  $v_2$  are also consistent with these examples. And yet  $M$  seems the best choice, because we'd think the teacher would not have kept any bit in the two examples constant if it was not in the position of a relevant variable. In this example, the natural conjecture is the most specific concept consistent with the sample, but that does not, in general, capture the intuitive idea of cooperative learning. For example, consider the concept class consisting of just the three concepts  $\{\beta\}, \{\alpha, \beta\}, \{\alpha, \gamma\}$ . If the teacher presented  $(\alpha, 1)$  as an example, there would be two most specific consistent concepts. But a learner that assumed the teacher was cooperative could confidently guess  $\{\alpha, \beta\}$  to be the target concept, because a cooperative teacher would have presented the unambiguous  $(\gamma, 1)$  if  $\{\alpha, \gamma\}$  was the target concept.

Could the learner's reasoning about the teacher's behavior in these examples be implemented without a coding trick? We will show below that no coding trick is necessary to achieve exactly this behavior of teacher and learner; there is a general principle that teachers and learners can independently implement to cooperatively learn any finite concept class. When applied to the class of monomials this principle enables any monomial to be learned from just two examples, regardless of the number  $m$  of variables.

Our approach is to define a new model of cooperation in learning, based on the idea that each partner in the cooperation tries to reduce the sample size by exploiting the assumption that the other partner does so. If this idea is iteratively propagated by both partners, one can refine teaching sets iteratively ending up with a framework for highly efficient teaching and learning without any coding tricks. It is important to note that teacher and learner do not agree on any order of the concept class or any order of the instances. All they know about each others' strategies is a general assumption about how cooperation should work independent of the concept class or its representation.

We show that the resulting variant of the teaching dimension—called the *subset teaching dimension (STD)*—is not only a uniform lower bound of the teaching dimension but can be constant where the original teaching dimension is exponential, even in cases where only one iteration is needed. For example, as illustrated above, the STD of the class of

monomials over  $m$  variables is 2, in contrast to its original teaching dimension of  $2^m$ .

Some examples however will reveal a nonmonotonicity of the subset teaching dimension: some classes possess subclasses with a higher subset teaching dimension, which is at first glance not very intuitive. We will explain below why in a cooperative model such a nonmonotonicity does not have to contradict intuition; additionally we introduce a second model of cooperative teaching and learning, that results in a monotonic dimension, called the *recursive teaching dimension (RTD)*. Comparing our complexity notions in terms of the sample size required for teaching and learning shows that achieving monotonicity here results in a loss in terms of sample efficiency; however, even though the RTD has some deficiencies compared to the STD, it still significantly improves on previously studied variants of the teaching dimension.

## 1.2 Related work

The problem of defining good or helpful examples in learning has been studied in different fields of learning theory. Various learning models that involve one particular teacher can be found in [AK97, JT92, GM96, Mat97]; these mostly focus on learning boolean functions.

The teaching dimension has been analyzed in the context of online learning [BE98, RY95] and in the model of learning from queries, e.g., in [Heg95] and in [Han07], with a focus on active learning in the PAC framework. In contrast to these models, in inductive inference the learning process is not necessarily considered to be finite. Approaches to defining learning infinite concepts from good examples [FKW93, LNW98] do not focus on the size of a finite sample of good examples, but rather on characterizing the cases in which learners can identify concepts from only finitely many examples.

The approach we present in this paper is mainly based on an idea by Balbach [Bal08]. He defined and analyzed a model in which, under the premise that the teacher uses a minimal teaching set as a sample, a learner can reduce the size of a required sample by eliminating concepts which possess a teaching set smaller than the number of examples provided by the teacher so far. Iterating this idea, the size of the teaching sets might be gradually reduced significantly. Though our approach is syntactically quite similar to Balbach's, the underlying idea is a different one (we do not consider elimination by the sample size but elimination by the sample content as compared to all possible teaching sets). The resulting variant of the teaching dimension in general yields a much better performance in terms of sample size than Balbach's model does.

## 2 Preliminaries

Let  $\mathbb{N}$  denote the set of all non-negative integers,  $\emptyset$  denote the empty set, and  $|A|$  denote the cardinality of a finite set  $A$ . Concerning the teaching framework, we will mostly follow the notation used in [Bal08].

In the models of teaching and learning to be defined below, we will always assume that the goal in an interaction between a teacher and a learner is to make the learner identify a (finite) concept  $c$  over a (finite) instance space  $X$ . To formalize this, let  $n > 0$  be a natural number and let  $X =$

$\{x_1, \dots, x_n\}$  be an instance space. A *concept*  $c$  is a subset of  $X$  and a *concept class*  $C$  is a set of concepts. Consequently, concepts and concept classes considered below will always be finite. As a special case we sometimes consider boolean functions over variables  $v_1, \dots, v_m$  as concepts, which just means to represent the instance space  $X$  by  $\{0, 1\}^m$ .

We identify every concept  $c$  with its membership function given by  $c(x_i) = 1$  if  $x_i \in c$ , and  $c(x_i) = 0$  if  $x_i \notin c$ , where  $1 \leq i \leq n$ . Given a *sample*, i.e., a set  $S = \{(y_1, b_1), \dots, (y_j, b_j)\} \subseteq X \times \{0, 1\}$  of labeled *examples*, we say that  $c$  is consistent with  $S$  if  $c(y_i) = b_i$  for all  $i \in \{1, \dots, j\}$ . If  $C$  is a concept class then we define

$$\text{Cons}(S, C) = \{c \in C \mid c \text{ is consistent with } S\}.$$

The sample  $S$  is called a *teaching set* for  $c$  with respect to  $C$  if  $\text{Cons}(S, C) = \{c\}$ . A teaching set allows a learning algorithm to uniquely identify a concept in the concept class  $C$ . Striving for sample efficiency, one is particularly interested in teaching sets of minimal size, called *minimal teaching sets*. The *teaching dimension* of  $c$  in  $C$  is the size of such a minimal teaching set, i.e.,  $TD(c, C) = \min\{|S| \mid \text{Cons}(S, C) = \{c\}\}$ , the worst case of which defines the teaching dimension of  $C$ , i.e.,  $TD(C) = \max\{TD(c, C) \mid c \in C\}$ . To refer to the set of all minimal teaching sets of  $c$  with respect to  $C$ , we use

$$TS(c, C) = \{S \mid \text{Cons}(S, C) = \{c\} \text{ and } |S| = TD(c, C)\}.$$

The reader is referred to [GK95, SM91] for original studies on teaching sets.

Recall our assumptions concerning the learning process: it proceeds stepwise; in each step the teacher presents a single example to the learner and the learner returns a conjecture about the target concept. The process stops when and only when a correct conjecture is made by the learner. Our minimal requirements on cooperative partners here is that teachers never present any example twice and always label the examples correctly according to the target concept, and that every conjecture a learner returns is consistent with the information seen up to that step.

The teaching dimension [GK95] then gives a measure of the worst case sample size needed by a learner if the teacher uses only minimal teaching sets for teaching. The reason is that a teaching set eliminates all but one concept due to inconsistency. However, if the learner knows  $TD(c, C)$  for every  $c \in C$  then sometimes concepts could also be eliminated by the mere number of examples presented to the learner. For instance, assume a learner knows that all but one concept  $c \in C$  have a teaching set of size one and that the teacher will teach using teaching sets. After having seen 2 examples, no matter what they are, the learner could eliminate all concepts but  $c$ . This idea, referred to as elimination by sample size, was introduced in [Bal08]. If a teacher knew that a learner eliminates by consistency and by sample size then the teacher could consequently reduce some teaching sets (e.g. here, if  $TD(c, C) \geq 3$ , a new “teaching set” for  $c$  could be built consisting of only 2 examples).

More than that—this idea is iterated by Balbach [Bal08]: if the learner knew that the teacher uses such reduced “teaching sets” then the learner could adapt his assumption on the size of the samples to be expected for each concept, which

could in turn result in a further reduction of the “teaching sets” by the teacher and so on. The following definition captures this idea formally.

**Definition 1 (Balbach teaching dimension [Bal08])**

Let  $C$  be a concept class,  $c \in C$ , and  $S$  a sample. Let  $BTD^0(c, C) = TD(c, C)$ . We define iterated dimensions for all  $k \in \mathbb{N}$  as follows.

- $\text{Cons}_{\text{size}}(S, C, k)$   
 $= \{c \in \text{Cons}(S, C) \mid BTD^k(c, C) \geq |S|\}$ .
- $BTD^{k+1}(c, C)$   
 $= \min\{|S| \mid \text{Cons}_{\text{size}}(S, C, k) = \{c\}\}$

Let  $z$  be minimal such that  $BTD^{z+1}(c, C) = BTD^z(c, C)$  for all  $c \in C$ . The iterated Balbach teaching dimension of  $c$  in  $C$  is defined by  $BTD(c, C) = BTD^z(c, C)$  and the iterated Balbach teaching dimension of the class  $C$  is  $BTD(C) = \max\{BTD(c, C) \mid c \in C\}$ .<sup>2</sup>

Obviously,  $BTD(C) \leq TD(C)$  for every concept class  $C$ . How much the sample complexity can actually be reduced by a cooperative teacher/learner pair according to this “elimination by sample size” principle, is illustrated by the concept class  $C_0$  consisting of the empty concept and all singleton concepts over  $X$ . The teaching dimension of this class is  $n$ , whereas the  $BTD$  is 2. A more interesting example is the class of monomials, which contains only one concept for which the  $BTD$ -iteration yields an improvement.

**Theorem 2 (Balbach [Bal08])** Let  $m \in \mathbb{N}$  and  $C$  the class of all boolean functions over  $m \geq 2$  variables that can be represented by a monomial. Let  $c_0 = \emptyset$  be the concept represented by a contradictory monomial.

1.  $BTD(c_0, C) = m + 2 < 2^m = TD(c_0, C)$ .
2.  $BTD(c, C) = TD(c, C)$  for all  $c \in C$  with  $c \neq c_0$ .

The intuitive reason why  $BTD(c_0, C) = m + 2$  in Theorem 2 is that samples for  $c_0$  of size  $m + 1$  or smaller are consistent also with monomials different from  $c_0$ . These other monomials hence cannot be eliminated—neither by size nor by inconsistency.

### 3 Teaching and learning using subset teaching sets

#### 3.1 The model

The approach studied by Balbach [Bal08] does not fully meet the intuitive idea of teacher and learner exploiting the knowledge that either partner behaves cooperatively. Consider for instance one more time the class  $C_0$  containing the empty concept and all singletons over  $X = \{x_1, \dots, x_n\}$ . Each concept  $\{x_i\}$  has the unique minimal teaching set  $\{(x_i, 1)\}$  in this class, whereas the empty concept only has a teaching set of size  $n$ , namely  $\{(x_1, 0), \dots, (x_n, 0)\}$ . The idea of elimination by size allows a learner to conjecture the empty

<sup>2</sup> [Bal08] denotes this by *IOTTD*, called iterated optimal teacher teaching dimension; we deviate from this notation for the sake of convenience.

concept as soon as two examples have been provided, due to the fact that all other concepts possess a teaching set of size one. This is why the empty concept has an *BTD* equal to 2 in this example.

However, as we have argued in the introduction, it would also make sense to devise a learner in a way to conjecture the empty concept as soon as a first example for that concept is provided—knowing that the teacher would not use a negative example for any other concept in the class. In terms of teaching sets this means to reduce the teaching sets to their minimal subsets that are not contained in minimal teaching sets for other concepts in the given concept class.

Formally, we define this refinement operator and its iteration as follows.

**Definition 3** Let  $C$  be a concept class,  $c \in C$ , and  $S$  a sample. Let  $STD^0(c, C) = TD(c, C)$ ,  $STS^0(c, C) = TS(c, C)$ . We define iterated sets for all  $k \in \mathbb{N}$  as follows.

- $Cons_{sub}(S, C, k) = \{c \in C \mid S \subseteq S' \text{ for some } S' \in STS^k(c, C)\}$ .
- $STD^{k+1}(c, C) = \min\{|S| \mid Cons_{sub}(S, C, k) = \{c\}\}$
- $STS^{k+1}(c, C) = \{S \mid Cons_{sub}(S, C, k) = \{c\}, |S| = STD^{k+1}(c, C)\}$ .

Let  $z$  be minimal such that  $STS^{z+1}(c, C) = STS^z(c, C)$  for all  $c \in C$ .<sup>3</sup>

A sample  $S$  with  $Cons_{sub}(S, C, z) = \{c\}$  is called a subset teaching set for  $c$  in  $C$ . The subset teaching dimension of  $c$  in  $C$  is defined as  $STD(c, C) = STD^z(c, C)$  and we denote by  $STS(c, C) = STS^z(c, C)$  the set of all minimal subset teaching sets for  $c$  in  $C$ . The subset teaching dimension of  $C$  is  $STD(C) = \max\{STD(c, C) \mid c \in C\}$ .

For illustration, consider again the concept class  $C_0$ , i.e.,  $C_0 = \{c_i \mid 0 \leq i \leq n\}$ , where  $c_0 = \emptyset$  and  $c_i = \{x_i\}$  for all  $i \in \{1, \dots, n\}$ . Obviously, for  $k \geq 1$ ,

$$STS^k(c_i) = \{\{(x_i, 1)\}\} \text{ for all } i \in \{1, \dots, n\}$$

and

$$STD^k(c_0) = \{\{(x_i, 0)\} \mid 1 \leq i \leq n\}.$$

Hence  $STD(C_0) = 1$ .

The definition of  $STS(c, C)$  induces a protocol for teaching and learning: for a target concept  $c$ , a teacher presents the examples in a subset teaching set for  $c$  to the learner. The learner will also be able to pre-compute all subset teaching sets for all concepts and determine the target concept from the sample provided by the teacher.<sup>4</sup>

**Protocol 4** Let  $C$  be a concept class.

0. Teacher and learner both compute  $STS(c, C)$  for all  $c \in C$ .

Let  $c \in C$  be a target concept known to the teacher.

<sup>3</sup>Such a  $z$  exists because  $STD^0(c, C)$  is finite and can hence be reduced only finitely often.

<sup>4</sup>Note that we focus on sample size here, but neglect efficiency issues arising from the pre-computation of all subset teaching sets.

1. The teacher chooses a set  $S \in STS(c, C)$  at random.
2. The teacher presents  $S$  to the learner (stepwise/batch).
3. The learner looks up and identifies the unique concept  $c \in C$  for which  $S \in STS(c, C)$ .

It is important to note at this point that Definition 3 as such is independent of the particular shape or structure of the concept class. It does not presume any special order of the concept representations or of the instances, i.e., teacher and learner do not have to agree on any such order to make use of the teaching and learning protocol. That means, given a special concept class  $C$ , the computation of its subset teaching sets does not involve any special coding trick depending on  $C$ —it just follows a general rule.

### 3.2 Comparison to the Balbach teaching dimension

Obviously, Protocol 4 based on the subset teaching dimension never requires a sample larger than a teaching set; often a smaller sample is sufficient. Similarly, the subset teaching dimension compares to the Balbach teaching dimension as follows.

**Proposition 5** 1.  $STD(C) \leq BTD(C)$  for every concept class  $C$ .

2. There is a concept class  $C$  with  $STD(C) < BTD(C)$ .

*Proof.* Assertion (1) immediately follows from the definitions. Informally, if a (Balbach) teaching set  $S$  in one iteration for a concept  $c$  is going to be reduced according to the *BTD*-rule (see Definition 1), then  $|S| \geq |S'| + 2$  for every (Balbach) teaching set  $S'$  on the current state of iteration for some concept  $c' \neq c$  consistent with  $S$ . In particular, if the Balbach teaching dimension of  $c$  is reduced to some value  $u < |S|$ , then  $S$  has got a subset of size  $u$  (or even smaller) that is not contained in any teaching set for any concept  $c' \neq c$  in  $C$ . The minimal such subset has cardinality at most  $u$  and is at least as big as a minimal subset teaching set for  $c$ .

Assertion (2) is witnessed by the class  $C_0$  containing the empty concept and all singletons over  $X$ . ■

The second assertion of this proposition even holds in a stronger form, see Theorem 6.

**Theorem 6** For each  $u \in \mathbb{N}$  there is a concept class  $C$  such that  $STD(C) = 1$  and  $BTD(C) = u$ .

*Proof.* Let  $n = 2^u + u$  be the number of instances in  $X$ . Define a concept class  $C = C_{0/1}^u$  as follows. For every  $s = (s_1, \dots, s_u) \in \{0, 1\}^u$ ,  $C$  contains the concepts  $c_{s,0} = \{x_i \mid 1 \leq i \leq u \text{ and } s_i = 1\}$  and  $c_{s,1} = c_{s,0} \cup \{x_{u+1+int(s)}\}$ . Here  $int(s) \in \mathbb{N}$  is defined by  $int(s) = \sum_{i=0}^{u-1} s_{i+1} \cdot 2^i$ . We claim that  $STD(C) = 1$  and  $BTD(C) = u$ .

Let  $s = (s_1, \dots, s_u) \in \{0, 1\}^u$ . Then

$$TS(c_{s,0}, C) = \{\{(x_i, s_i) \mid 1 \leq i \leq u\} \cup \{(x_{u+1+int(s)}, 0)\}\}$$

$$TS(c_{s,1}, C) = \{\{(x_{u+1+int(s)}, 1)\}\}$$

Since for each  $c \in C$  the minimal teaching set for  $c$  with respect to  $C$  contains an example that does not occur in the

minimal teaching set for any other concept  $c' \in C$ , one obtains  $STD(C) = 1$  in just one iteration. See Table 1 for the case  $u = 2$ .

In contrast to that, we obtain  $BTD^0(c_{s,0}, C) = u + 1$ ,  $BTD^1(c_{s,0}, C) = u$ , and  $BTD^0(c_{s,1}, C) = 1$  for all  $s \in \{0, 1\}^u$ . Consider any  $s \in \{0, 1\}^u$  and any sample  $S \subseteq \{(x, c_{s,0}(x)) \mid x \in X\}$  with  $|S| = u - 1$ . Clearly there is some  $s' \in \{0, 1\}^u$  with  $s' \neq s$  such that  $c_{s',0} \in Cons(S, C)$ . So  $|Cons(S, C, 1)| > 1$  and in particular  $Cons(S, C, 1) \neq \{c_{s,0}\}$ . Hence  $BTD^2(c_{s,0}, C) = BTD^1(c_{s,0}, C)$ , which finally implies  $BTD(C) = u$ . ■

concept	$STS^0$	$STS^1$
$\emptyset$	$\{(x_1, 0), (x_2, 0), (x_3, 0)\}$	$\{(x_3, 0)\}$
$\{x_3\}$	$\{(x_3, 1)\}$	$\{(x_3, 1)\}$
$\{x_2\}$	$\{(x_1, 0), (x_2, 1), (x_4, 0)\}$	$\{(x_4, 0)\}$
$\{x_2, x_4\}$	$\{(x_4, 1)\}$	$\{(x_4, 1)\}$
$\{x_1\}$	$\{(x_1, 1), (x_2, 0), (x_5, 0)\}$	$\{(x_5, 0)\}$
$\{x_1, x_5\}$	$\{(x_5, 1)\}$	$\{(x_5, 1)\}$
$\{x_1, x_2\}$	$\{(x_1, 1), (x_2, 1), (x_6, 0)\}$	$\{(x_6, 0)\}$
$\{x_1, x_2, x_6\}$	$\{(x_6, 1)\}$	$\{(x_6, 1)\}$

Table 1: Iterated subset teaching sets for the class  $C_{0/1}^u$  with  $u = 2$ , where  $C_{0/1}^u = \{c_{00,0}, c_{00,1}, \dots, c_{11,0}, c_{11,1}\}$  with  $c_{00,0} = \emptyset$ ,  $c_{00,1} = \{x_3\}$ ,  $c_{01,0} = \{x_2\}$ ,  $c_{01,1} = \{x_2, x_4\}$ ,  $c_{10,0} = \{x_1\}$ ,  $c_{10,1} = \{x_1, x_5\}$ ,  $c_{11,0} = \{x_1, x_2\}$ ,  $c_{11,1} = \{x_1, x_2, x_6\}$ .

### 3.3 Teaching monomials

This section provides an analysis of the  $STD$  for a more natural example, the monomials, showing that the very intuitive example given in the introduction is indeed what a cooperative teacher and learner in our model would come up with. The main result is that the  $STD$  of the class of all monomials is 2, independent on the number  $m$  of variables, whereas its teaching dimension is exponential in  $m$  and its  $BTD$  is linear in  $m$ , cf. [Bal08].

**Theorem 7** *Let  $m \in \mathbb{N}$  and  $C$  the class of all boolean functions over  $m$  variables that can be represented by a monomial. Then  $STD(C) = 2$ .*

*Proof.* Let  $m \in \mathbb{N}$  and  $s = (s_1, \dots, s_m)$ ,  $s' = (s'_1, \dots, s'_m)$  elements in  $\{0, 1\}^m$ . Let  $\Delta(s, s')$  denote the Hamming distance of  $s$  and  $s'$ , i.e.,  $\Delta(s, s') = \sum_{1 \leq i \leq m} |s(i) - s'(i)|$ .

We distinguish the following types of monomials  $M$  over  $m$  variables.

Type 1:  $M$  is the empty monomial.

Type 2:  $M$  has got  $m$  variables,  $M \neq v_1 \wedge \bar{v}_1$ .

Type 3:  $M$  has got  $k$  variables,  $1 \leq k < m$ ,  $M \neq v_1 \wedge \bar{v}_1$ .

Type 4:  $M$  is contradictory, i.e.,  $M \equiv v_1 \wedge \bar{v}_1$ .

The following facts state some properties of the corresponding minimal teaching sets.

Fact 1: If  $M$  is of type 1 and  $S \in STS^0(M, C)$ , then  $S$  contains two positive examples of Hamming distance  $m$ .

Fact 2: If  $M$  is of type 2 and  $S \in STS^0(M, C)$ , then  $S$  contains (i) one positive example and (ii)  $m$  negative examples, where the Hamming distance between two negative examples is less than  $m$ .

Fact 3: If  $M$  is of type 3 and  $S \in STS^0(M, C)$ , then  $S$  contains (i) two positive examples of Hamming distance  $m - k$  and (ii)  $k$  negative examples, where the Hamming distance between each two negative examples is less than  $m$ .

Fact 4: If  $M$  is of type 4 and  $S \in STS^0(M, C)$ , then  $S = \{(s, 0) \mid s \in \{0, 1\}^m\}$ .

Fact 5: For every  $s \in \{0, 1\}^m$  there are two different monomials  $M, M'$  of type 3 such that  $(s, 1) \in S \cap S'$  for some  $S \in STS^0(M, C)$  and some  $S' \in STS^0(M', C)$ .

Fact 6: For every  $s \in \{0, 1\}^m$  there are two different monomials  $M, M'$  of type 3 such that  $(s, 0) \in S \cap S'$  for some  $S \in STS^0(M, C)$  and some  $S' \in STS^0(M', C)$ .

Fact 7: For every  $s \in \{0, 1\}^m$  there are two different monomials  $M, M'$  of type 2 such that  $(s, 0) \in S \cap S'$  for some  $S \in STS^0(M, C)$  and some  $S' \in STS^0(M', C)$ .

Fact 8: If  $M$  is of type 2,  $S \in STS^0(M, C)$  and  $S' \subset S$ , then there is a monomial  $M_3$  of type 3 such that  $S' \subseteq S_3$  for some  $S_3 \in STS^0(M_3, C)$ .

After the first iteration we obtain the following facts.

Fact 9: If  $M$  is of type 1 and  $S \in STS^1(M, C)$ , then  $S \in STS^0(M, C)$ .

Fact 10: If  $M$  is of type 2 and  $S \in STS^1(M, C)$ , then  $S \in STS^0(M, C)$ .

Fact 11: If  $M$  is of type 3 and  $S \in STS^1(M, C)$ , then  $S$  contains two positive examples.

Fact 12: If  $M$  is of type 4 and  $S \in STS^1(M, C)$ , then  $S$  contains two negative examples of Hamming distance  $m$ .

After the second iteration we obtain the following facts.

Fact 13: If  $M$  is of type 1 and  $S \in STS^2(M, C)$ , then  $S \in STS^1(M, C)$ .

Fact 14: If  $M$  is of type 2 and  $S \in STS^2(M, C)$ , then  $S$  contains one positive and one negative example. Moreover, for every  $s \in \{0, 1\}^m$ , there is a monomial  $M$  of type 2 such that  $(s, 0) \in S$  for some  $S \in STS^2(M, C)$ .

Fact 15: If  $M$  is of type 3 and  $S \in STS^1(M, C)$ , then  $S \in STS^2(M, C)$ .

Fact 16: If  $M$  is of type 4 and  $S \in STS^2(M, C)$ , then  $S \in STS^1(M, C)$ .

Combining the insights achieved so far, it is easily seen that  $STD^3(M, C) = STD^2(M, C) = 2$  for all  $M \in C$ . ■

For illustration of this proof in case  $m = 2$  see Table 2.

A further simple example showing that the  $STD$  can be constant as compared to an exponential teaching dimension, this time with an  $STD$  of 1, is the following.

Let  $C_{\vee DNF}^m$  contain all boolean functions over  $m$  variables that can be represented by a 2-term DNF of the form  $v_1 \vee M$ , where  $M$  is a monomial that contains, for each  $i$  with  $2 \leq i \leq m$ , either the literal  $v_i$  or the literal  $\bar{v}_i$ . Moreover,  $C_{\vee DNF}^m$  contains the boolean function that can be represented by the monomial  $M' \equiv v_1$ .

**Theorem 8** *Let  $m \in \mathbb{N}$ .*

1.  $TD(C_{\vee DNF}^m) = 2^{m-1}$ .
2.  $STD(C_{\vee DNF}^m) = 1$ .

	$STS^0$	$STS^1$
$v_1$	$\{(10,1),(11,1),(00,0)\}$ $\{(10,1),(11,1),(01,0)\}$	$\{(10,1),(11,1)\}$
$\bar{v}_1$	$\{(00,1),(01,1),(10,0)\}$ $\{(00,1),(01,1),(11,0)\}$	$\{(00,1),(01,1)\}$
$v_2$	$\{(01,1),(11,1),(00,0)\}$ $\{(01,1),(11,1),(10,0)\}$	$\{(01,1),(11,1)\}$
$\bar{v}_2$	$\{(00,1),(10,1),(01,0)\}$ $\{(00,1),(10,1),(11,0)\}$	$\{(00,1),(10,1)\}$
$v_1 \wedge v_2$	$\{(11,1),(01,0),(10,0)\}$	$\{(11,1),(01,0),(10,0)\}$
$v_1 \wedge \bar{v}_2$	$\{(10,1),(00,0),(11,0)\}$	$\{(10,1),(00,0),(11,0)\}$
$\bar{v}_1 \wedge v_2$	$\{(01,1),(00,0),(11,0)\}$	$\{(01,1),(00,0),(11,0)\}$
$\bar{v}_1 \wedge \bar{v}_2$	$\{(00,1),(01,0),(10,0)\}$	$\{(00,1),(01,0),(10,0)\}$
$v_1 \wedge \bar{v}_1$	$\{(00,0),(01,0),(10,0),(11,0)\}$	$\{(00,0),(01,0)\}$ $\{(00,0),(10,0)\}$ $\{(01,0),(11,0)\}$ $\{(10,0),(11,0)\}$
$\lambda$	$\{(00,1),(11,1)\}$ $\{(01,1),(10,1)\}$	$\{(00,1),(11,1)\}$ $\{(01,1),(10,1)\}$

	$STS^2$	$STS^3$
$v_1$	$\{(10,1),(11,1)\}$	$\{(10,1),(11,1)\}$
$\bar{v}_1$	$\{(00,1),(01,1)\}$	$\{(00,1),(01,1)\}$
$v_2$	$\{(01,1),(11,1)\}$	$\{(01,1),(11,1)\}$
$\bar{v}_2$	$\{(00,1),(10,1)\}$	$\{(00,1),(10,1)\}$
$v_1 \wedge v_2$	$\{(11,1),(01,0)\}$ $\{(11,1),(10,0)\}$	$\{(11,1),(01,0)\}$ $\{(11,1),(10,0)\}$
$v_1 \wedge \bar{v}_2$	$\{(10,1),(00,0)\}$ $\{(10,1),(11,0)\}$	$\{(10,1),(00,0)\}$ $\{(10,1),(11,0)\}$
$\bar{v}_1 \wedge v_2$	$\{(01,1),(00,0)\}$ $\{(01,1),(11,0)\}$	$\{(01,1),(00,0)\}$ $\{(01,1),(11,0)\}$
$\bar{v}_1 \wedge \bar{v}_2$	$\{(00,1),(01,0)\}$ $\{(00,1),(10,0)\}$	$\{(00,1),(01,0)\}$ $\{(00,1),(10,0)\}$
$v_1 \wedge \bar{v}_1$	$\{(00,0),(01,0)\}$ $\{(00,0),(10,0)\}$ $\{(01,0),(11,0)\}$ $\{(10,0),(11,0)\}$	$\{(00,0),(01,0)\}$ $\{(00,0),(10,0)\}$ $\{(01,0),(11,0)\}$ $\{(10,0),(11,0)\}$
$\lambda$	$\{(00,1),(11,1)\}$ $\{(01,1),(10,1)\}$	$\{(00,1),(11,1)\}$ $\{(01,1),(10,1)\}$

Table 2: Iterated subset teaching sets for the class of all monomials over  $m = 2$  variables. Here  $\lambda$  denotes the empty monomial.

*Proof.* The straightforward details concerning the proof of Assertion (2) are omitted; Assertion (1) can be verified as follows.

Let  $S$  be a sample that is consistent with  $M'$ . Assume that for some  $s \in \{0, 1\}^m$ , the sample  $S$  does not contain the negative example  $(s, 0)$ . Obviously, there is a 2-term DNF  $D \equiv v_1 \vee M$  such that  $D$  is consistent with  $S \cup \{(s, 1)\}$ . Hence  $S$  is not a teaching set for  $M'$ . Since there are exactly  $2^{m-1}$  2-term DNFs that represent different functions in  $C$ , a teaching set for  $M'$  must contain at least  $2^{m-1}$  examples. ■

## 4 Nonmonotonicity and the recursive teaching dimension

### 4.1 Nonmonotonicity versus redundancy of variables

Interpreting the subset teaching dimension as a measure of complexity of a concept class in terms of cooperative teach-

ing and learning, we observe a fact that is worth discussing, namely the nonmonotonicity of this complexity notion, as stated by the following theorem.

**Theorem 9** *There is a concept class  $C$  with  $STD(C') > STD(C)$  for some subclass  $C' \subset C$ .*

*Sketch of proof.* This is witnessed by the concept classes  $C = C_{0/1}^u$  and their subclasses  $C' = \{c_{s,0} \mid s \in \{0, 1\}^u\}$  used in the proof of Theorem 6 (see Table 1 for  $u = 2$ ). ■

Note that this nonmonotonicity result holds with a fixed number of instances  $n$ . In fact, if  $n$  was not considered fixed then every concept class  $C'$  would have a superset  $C$  (via addition of instances) of lower subset teaching dimension. However, the same even holds for the teaching dimension itself which we yet consider monotonic since it is monotonic given fixed  $n$ . So whenever we speak of monotonicity we assume a fixed instance space  $X$ .

Of course such an instance space  $X$  might contain *redundant* instances the removal of which would not affect the subset teaching dimension and would retain a non-redundant subset of the set of all subset teaching sets. In the following subsection, where we discuss a possible intuition behind the nonmonotonicity of the  $STD$ , redundancy conditions on instances will actually play an important role and show the usefulness of the following technical discussion. However, it is not straightforward to impose a suitable redundancy condition characterizing when an instance can be removed.

We derive such a condition starting with a redundancy condition for the original variant of teaching sets. For that purpose we introduce the notion  $C^{-x}$  for the concept class resulting from  $C$  after removing the instance  $x$  from the instance space  $X$ . Here  $C$  is any concept class over  $X$  and  $x \in X$  is any instance. For example, if  $X = \{x_1, x_2, x_3\}$  and  $C = \{\{x_1\}, \{x_1, x_2\}, \{x_2, x_3\}\}$  then

$$C^{-x_3} = \{\{x_1\}, \{x_1, x_2\}, \{x_2\}\}$$

considered over the instance space  $\{x_1, x_2\}$ .

**Lemma 10** *Let  $C$  be a concept class over  $X$  and  $x \in X$ . For all  $c \in C$  and for all  $S \in TS(c, C)$*

$$(x, c(x)) \in S \Rightarrow$$

$$\exists y \neq x [(S \setminus \{(x, c(x))\}) \cup \{(y, c(y))\} \in TS(c, C)],$$

*then for all  $c \in C$  and for all samples  $S$*

$$S \in TS(c, C^{-x}) \iff [S \in TS(c, C) \wedge (x, c(x)) \notin S].$$

*Proof.* Note that  $|C^{-x}| = |C|$ . Let  $c \in C$  be an arbitrary concept and let  $S$  be any sample over  $X$ .

First assume  $S \in TS(c, C)$  and  $(x, c(x)) \notin S$ . Since obviously  $TD(c, C^{-x}) \geq TD(c, C)$  we immediately obtain  $S \in TS(c, C^{-x})$ .

Second assume  $S \in TS(c, C^{-x})$ . By definition, we have  $(x, c(x)) \notin S$ . Hence it remains to prove that  $S \in TS(c, C)$ . If  $S \notin TS(c, C)$  then there exists some  $T \in TS(c, C)$  with  $|T| < |S|$ . We distinguish two cases.

*Case 1.*  $(x, c(x)) \notin T$ .

Then  $T \in TS(c, C^{-x})$  in contradiction to the facts  $S \in TS(c, C^{-x})$  and  $|S| \neq |T|$ .

Case 2.  $(x, c(x)) \in T$ .

Then by the premise of the lemma there exists a  $y \neq x$  such that

$$A \stackrel{\text{def}}{=} (S \setminus \{(x, c(x))\}) \cup \{(y, c(y))\} \in TS(c, C).$$

Since  $(x, c(x)) \notin A$  we have  $A \in TS(c, C^{-x})$  and  $|A| = |T| \neq |S|$ . This again contradicts  $S \in TS(c, C^{-x})$ .

Since both cases reveal a contradiction, we obtain  $S \in TS(c, C)$ . ■

For illustration see Table 3. In this example the instances  $x_4$  and  $x_5$  meet the redundancy condition. After eliminating  $x_5$ ,  $x_4$  still meets the condition and can be removed as well. The new representation of the concept class then involves only the instances  $x_1, x_2, x_3$ .

concept in $C$	$TS$
$\emptyset$	$\{(x_1, 0), (x_3, 0)\}, \{(x_1, 0), (x_4, 0)\}, \{(x_1, 0), (x_5, 0)\}$
$\{x_1\}$	$\{(x_1, 1), (x_2, 0)\}, \{(x_1, 1), (x_5, 0)\}$
$\{x_3, x_4, x_5\}$	$\{(x_2, 0), (x_3, 1)\}, \{(x_2, 0), (x_4, 1)\}, \{(x_2, 0), (x_5, 1)\}$
$\{x_2, x_3, x_4, x_5\}$	$\{(x_1, 0), (x_2, 1)\}, \{(x_2, 1), (x_4, 1)\}$
$\{x_1, x_2, x_5\}$	$\{(x_2, 1), (x_3, 0)\}, \{(x_3, 0), (x_5, 1)\}$
$\{x_1, x_2, x_3, x_5\}$	$\{(x_1, 1), (x_3, 1)\}, \{(x_3, 1), (x_4, 1)\}$

concept in $(C^{-x_5})^{-x_4}$	$TS$
$\emptyset$	$\{(x_1, 0), (x_3, 0)\}$
$\{x_1\}$	$\{(x_1, 1), (x_2, 0)\}$
$\{x_3\}$	$\{(x_2, 0), (x_3, 1)\}$
$\{x_2, x_3\}$	$\{(x_1, 0), (x_2, 1)\}$
$\{x_1, x_2\}$	$\{(x_2, 1), (x_3, 0)\}$
$\{x_1, x_2, x_3\}$	$\{(x_1, 1), (x_3, 1)\}$

Table 3: Teaching sets for a class  $C$  before and after elimination of two redundant instances.

Lemma 10 provides a condition on an instance  $x$ . If that instance is eliminated from the instance space then the resulting concept class  $C^{-x}$  does not only have the same teaching dimension as  $C$  but, even more, for each of its concepts  $c$  the teaching sets are exactly those that are teaching sets for  $c$  in  $C$  and do not contain an example involving the eliminated instance  $x$ . Note that even though several instances might meet that condition at the same time, only one at a time may be removed. For the remaining instances it has to be checked whether the condition still holds after elimination of the first redundant instance.

So one legitimate redundancy condition for instances—considering teaching sets—is the one given in the premise of Lemma 10.

This condition can be extended to a redundancy condition with respect to subset teaching sets.

**Theorem 11** *Let  $C$  be a concept class over  $X$  and  $x \in X$ . If for all  $k \in \mathbb{N}$ , for all  $c \in C$ , and for all  $S \in STS^k(c, C)$*

$$(x, c(x)) \in S \Rightarrow$$

$$\exists y \neq x [(S \setminus \{(x, c(x))\}) \cup \{(y, c(y))\} \in STS^k(c, C)],$$

*then for all  $k \in \mathbb{N}$ , for all  $c \in C$ , and for all samples  $S$*

$$S \in STS^k(c, C^{-x}) \iff [S \in STS^k(c, C) \wedge (x, c(x)) \notin S].$$

*Proof.* Note that  $|C^{-x}| = |C|$ . We prove the theorem by induction on  $k$ .

For  $k = 0$  this follows immediately from Lemma 10. So assume that the claim is proven for some  $k$  (induction hypothesis). It remains to show that it then also holds for  $k + 1$ .

For that purpose note that

$$\forall c \in C \forall A \in STS^k(c, C) \exists B \in STS^k(c, C^{-x}) [ |A| = |B| \wedge A \setminus \{(x, c(x))\} \subseteq B ] (*)$$

by combination of the induction hypothesis with the premise of the theorem.

Choose an arbitrary  $c \in C$ .

First assume  $S \in STS^{k+1}(c, C)$  and  $(x, c(x)) \notin S$ . By the definition of subset teaching sets, there is an  $S' \in STS^k(c, C)$  with

$$S \subseteq S'. \quad (1)$$

Using (\*) we can assume without loss of generality that

$$S' \in STS^k(c, C^{-x}). \quad (2)$$

Moreover, again by the definition of subset teaching sets, one obtains  $S \not\subseteq S''$  for every  $S'' \in STS^k(c', C)$  with  $c' \neq c$ . The induction hypothesis then implies

$$S \not\subseteq S'' \text{ for every } S'' \in STS^k(c', C^{-x}) \text{ with } c' \neq c. \quad (3)$$

Due to (1), (2), (3) we get either  $S \in STS^{k+1}(c, C^{-x})$  or  $|S| > STD^{k+1}(c, C^{-x})$ . In the latter case there would be a set  $T \in STS^{k+1}(c, C^{-x})$  with  $|T| < |S|$ .  $T$  is a subset of some set in  $STS^k(c, C^{-x})$  and thus also of some set in  $STS^k(c, C)$  by induction hypothesis. If  $T$  was contained in some  $T' \in STS^k(c', C)$  for some  $c' \neq c$  then we could again assume without loss of generality, using (\*) and  $(x, c(x)) \notin T$ , that  $T$  is contained in some set in  $STS^k(c', C^{-x})$ —in contradiction to  $T \in STS^{k+1}(c, C^{-x})$ . Therefore  $T \in STS^{k+1}(c, C)$  and so  $|T| = |S|$ —a contradiction. This implies  $S \in STS^{k+1}(c, C^{-x})$ .

Second assume that  $S \in STS^{k+1}(c, C^{-x})$ . Obviously,  $(x, c(x)) \notin S$ , so that it remains to show  $S \in STS^{k+1}(c, C)$ .

Because of  $S \in STS^{k+1}(c, C^{-x})$  there exists some set  $S' \in STS^k(c, C^{-x})$  such that

$$S \subseteq S'. \quad (4)$$

The induction hypothesis implies

$$S' \in STS^k(c, C). \quad (5)$$

Moreover, by the definition of subset teaching sets, one obtains  $S \not\subseteq S''$  for every  $S'' \in STS^k(c', C^{-x})$  with  $c' \neq c$ . If there was a set  $S'' \in STS^k(c', C)$  with  $c' \neq c$  and  $S \subseteq S''$  then (\*) would imply that without loss of generality  $S'' \in STS^k(c', C^{-x})$ . So we have

$$S \not\subseteq S'' \text{ for every } S'' \in STS^k(c', C) \text{ with } c' \neq c. \quad (6)$$

Combining (4), (5), (6) we get either  $S \in STS^{k+1}(c, C)$  or  $|S| > STD^{k+1}(c, C)$ . In the latter case there would be a set  $T \in STS^{k+1}(c, C)$  with  $|T| < |S|$ .  $T$  is a subset of some set  $T' \in STS^k(c, C)$ . We can assume without loss of generality, using (\*), that  $T' \in STS^k(c, C^{-x})$ . If  $T$  was contained in some set in  $STS^k(c', C^{-x})$  for some  $c' \neq c$  then by induction hypothesis  $T$  would be contained in some set in  $STS^k(c', C)$  for some  $c' \neq c$ . This is a contradiction to  $T \in STS^{k+1}(c, C)$ . So  $T \in STS^{k+1}(c, C^{-x})$  and hence  $|T| = |S|$ —a contradiction. Thus  $S \in STS^{k+1}(c, C)$ . ■

## 4.2 The reason for nonmonotonicity

The idea about why the teaching dimension can decrease when a concept class increases is best illustrated by an example in which the addition of a single concept has this effect. In a simple such example, the instance space consists of three elements  $\alpha, \beta, \gamma$ . First, consider the four distinct concepts that all contain  $\gamma$ ,  $c_{001} = \{\gamma\}$ ,  $c_{011} = \{\beta, \gamma\}$ ,  $c_{101} = \{\alpha, \gamma\}$ ,  $c_{111} = \{\alpha, \beta, \gamma\}$ . When these four concepts are the only ones in the class the teaching sets for them all are necessarily size two—elements  $\alpha$  and  $\beta$  and their respective labels—because  $\gamma$  is a member of all of them, it cannot be part of any teaching set. If one more concept is added to the class the subset teaching sets all become size 1. Table 4 shows the computation when  $c_{000} = \emptyset$  is added.

concept	$STS^0$	$STS^1$
$\emptyset$	$\{(\gamma, 0)\}$	$\{(\gamma, 0)\}$
$\{\gamma\}$	$\{(\alpha, 0), (\beta, 0), (\gamma, 1)\}$	$\{(\gamma, 1)\}$
$\{\beta, \gamma\}$	$\{(\alpha, 0), (\beta, 1)\}$	$\{(\alpha, 0), (\beta, 1)\}$
$\{\alpha, \gamma\}$	$\{(\alpha, 1), (\beta, 0)\}$	$\{(\alpha, 1), (\beta, 0)\}$
$\{\alpha, \beta, \gamma\}$	$\{(\alpha, 1), (\beta, 1)\}$	$\{(\alpha, 1), (\beta, 1)\}$

concept	$STS^2$	$STS^3$
$\emptyset$	$\{(\gamma, 0)\}$	$\{(\gamma, 0)\}$
$\{\gamma\}$	$\{(\gamma, 1)\}$	$\{(\gamma, 1)\}$
$\{\beta, \gamma\}$	$\{(\alpha, 0)\}$	$\{(\alpha, 0)\}$
$\{\alpha, \gamma\}$	$\{(\beta, 0)\}$	$\{(\beta, 0)\}$
$\{\alpha, \beta, \gamma\}$	$\{(\alpha, 1), (\beta, 1)\}$	$\{(\beta, 1)\}$

Table 4: Illustration of the nonmonotonicity of  $STD$ .

From a more general point of view, it is not obvious how to explain why a teaching dimension resulting from a cooperative model should be nonmonotonic.

First of all, this is a counter-intuitive observation when considering  $STD$  as a notion of complexity—intuitively any subclass of  $C$  should be at most as complex for teaching and learning as  $C$ .

However, there is in fact an intuitive explanation for the nonmonotonicity of the complexity in cooperative teaching and learning: when teaching  $c \in C$ , instead of providing examples that eliminate all concepts in  $C \setminus \{c\}$  (as is the idea underlying minimal teaching sets) cooperative teachers would rather pick only those examples that distinguish  $c$  from its “most similar” concepts in  $C$ . Similarity here is measured by the number of instances on which two concepts agree (i.e., dissimilarity is given by the Hamming distance between the concepts, where a concept  $c$  is represented as a

bit vector  $(c(x_1), \dots, c(x_n))$ ). This is reflected in the subset teaching sets in all illustrative examples considered above.

Considering a class  $C = C_{0/1}^u$ , one observes that a subset teaching set for a concept  $c_{s,0}$  contains only the negative example  $(x_{u+1+int(s)}, 0)$  distinguishing it from  $c_{s,1}$  (its nearest neighbor in terms of Hamming distance). A learner will recognize this example as the one that separates only that one pair  $(c_{s,0}, c_{s,1})$  of nearest neighbors. In contrast to that, if we consider only the subclass  $C' = \{c_{s,0} \mid s \in \{0, 1\}^u\}$ , the nearest neighbors of each  $c_{s,0}$  are different ones, and every single example separating one nearest neighbor pair also separates other nearest neighbor pairs. Thus no single example can be recognized by the learner as a separating example for one unique pair of concepts.

This intuitive idea of subset teaching sets being used for distinguishing a concept from its nearest neighbors has to be treated with care though. The reason is that the concept class may contain “redundant” instances, i.e., instances that could be removed from the instance space according to Theorem 11.

Such redundant instances might on the other hand affect Hamming distances and nearest neighbor relations. Only after their elimination the notion of nearest neighbors in terms of Hamming distance becomes well-defined. Consider for instance Table 3. In the concept class  $C$  over 5 instances the only nearest neighbor of  $\emptyset$  is  $\{x_1\}$  and an example distinguishing  $\emptyset$  from  $\{x_1\}$  would be  $(x_1, 0)$ . Moreover, no other concept is distinguished from its nearest neighbors by the instance  $x_1$ . According to the intuition explained here, this would suggest  $\{(x_1, 0)\}$  being a subset teaching set for  $\emptyset$  although the subset teaching sets here equal the teaching sets and are all of cardinality 2.

After instance elimination of  $x_4, x_5$  there is only one subset teaching set for  $\emptyset$ , namely  $\{(x_1, 0), (x_3, 0)\}$ . This is still of cardinality 2 but note that now  $\emptyset$  has two nearest neighbors, namely  $\{x_1\}$  and  $\{x_3\}$ . The two examples in the subset teaching set are those that distinguish  $\emptyset$  from its nearest neighbors. Note that either one of these two examples is not unique as an example used for distinguishing a concept from its nearest neighbors:  $(x_1, 0)$  would be used by  $\{x_2, x_3\}$  for distinguishing itself from its nearest neighbor  $\{x_1, x_2, x_3\}$ ;  $(x_3, 0)$  would be used by  $\{x_1, x_2\}$  for distinguishing itself from its nearest neighbor  $\{x_1, x_2, x_3\}$ . So the subset teaching set for  $\emptyset$  has to contain both examples.

This shows that in general a subclass of a class  $C$  can have a higher complexity than  $C$  if crucial nearest neighbors of some concepts are missing.

To summarize,

- nonmonotonicity has an intuitive reason and is not an indication for an ill-defined version of the teaching dimension,
- nonmonotonicity is in fact required if we want to capture the idea that the existence of specific concepts to distinguish a target concept from is beneficial for teaching and learning.

So, the  $STD$  captures certain intuitions about teaching and learning that monotonic dimensions *cannot* capture; at the same time monotonicity might in other respects itself be

an intuitive property of teaching and learning which then the STD cannot capture.

In particular there are two underlying intuitive properties that seem to not be satisfiable by a single variant of the teaching dimension.

So in contrast one may wish to have a cooperative teaching and learning model going along with a monotonic complexity measure. It is not hard to show that *BTD* in fact is monotonic, see Theorem 12.

**Theorem 12** *If  $C$  is a concept class and  $C' \subseteq C$  a subclass of  $C$ , then  $BTD(C') \leq BTD(C)$ .*

*Proof.* Fix  $C$  and  $C' \subseteq C$ . We will prove by induction on  $k$  that

$$BTD^k(c, C') \leq BTD^k(c, C) \text{ for all } c \in C \quad (7)$$

for all  $k \in \mathbb{N}$ .

$k = 0$ : Property (7) holds because of  $BTD^0(c, C') = TD(c, C') \leq TD(c, C) = BTD^0(c, C)$  for all  $c \in C$ .

Induction hypothesis: assume (7) holds for a fixed  $k$ .

$k \rightsquigarrow k + 1$ : First, observe that

$$\begin{aligned} & \text{Cons}_{\text{size}}(S, C', k) \\ &= \{c \in \text{Cons}(S, C') \mid BTD^k(c, C') \geq |S|\} \\ &\subseteq \{c \in \text{Cons}(S, C') \mid BTD^k(c, C) \geq |S|\} \text{ (ind. hyp.)} \\ &\subseteq \{c \in \text{Cons}(S, C) \mid BTD^k(c, C) \geq |S|\} \\ &= \text{Cons}_{\text{size}}(S, C, k) \end{aligned}$$

Second, for all  $c \in C$  we obtain

$$\begin{aligned} & BTD^{k+1}(c, C') \\ &= \min\{|S| \mid \text{Cons}_{\text{size}}(S, C', k) = \{c\}\} \\ &\leq \min\{|S| \mid \text{Cons}_{\text{size}}(S, C, k) = \{c\}\} \\ &\leq BTD^{k+1}(c, C) \end{aligned}$$

This completes the proof.  $\blacksquare$

So, on the one hand, we have the teaching framework based on the subset teaching dimension which results in a nonmonotonic dimension, and on the other hand we have a monotonic dimension in the *BTD* framework, which unfortunately does not always meet our idea of a best possible cooperative teaching and learning protocol. That raises the question whether nonmonotonicity is necessary to achieve certain positive results. In fact, the nonmonotonicity concerning the class  $C_{0/1}^u$  is not counter-intuitive, but would a dimension that is monotonic also result in a worse sample complexity than the *STD* in general, such as, e.g., for the monomials?

In other words, is there a teaching/learning framework

- resulting in a monotonic variant of a teaching dimension and
- achieving similarly good results as the subset teaching dimension?

At this point of course it is difficult to define what “similarly good” means. However, we would like to have a constant dimension for the class of all monomials, as well as, e.g., a

teaching set of size 1 for the empty concept in our often used concept class  $C_0$ .

We will now via several steps introduce at least a monotonic variant of the teaching dimension and show that for most of the examples studied above, it is as low as the subset teaching dimension. General comparisons will be made in Section 5, in particular in order to show that this new framework is uniformly at least as efficient as the *BTD* framework (or better), while sometimes being less efficient than the *STD* framework. This reflects to a certain extent that monotonicity constraints might affect sample efficiency.

### 4.3 The teaching plan model

We will first define the notion for our variant of teaching dimension and show its monotonicity. The nonmonotonicity of *STD* is caused by considering every  $STS^k$ -set for every concept when computing an  $STS^{k+1}$ -set for a single concept. Hence the idea in the following approach is to impose an order onto the concept class, in terms of the “teaching complexity” of the concepts. This is what the teaching dimension does as well, but our design principle is a recursive one. After selecting a concept which is “easy to teach” because of possessing a small minimal teaching set, we eliminate this concept from our concept class and consider only the remaining concepts. Again we determine the one with the lowest teaching dimension, now however measured with respect to the class of remaining concepts, and so on. The resulting notion of dimension is therefore called the *recursive teaching dimension*.

**Definition 13** *Let  $C$  be a concept class,  $|C| = N$ . A teaching plan for  $C$  is a sequence  $p = ((c_1, S_1), \dots, (c_N, S_N)) \in (C \times 2^{X \times \{0,1\}})^N$  such that*

1.  $C = \{c_1, \dots, c_N\}$ .
2.  $S_j \in TS(c_j, \{c_j, \dots, c_N\})$  for  $1 \leq j \leq N - 1$ .
3.  $S_N = \{(x, 1 - b) \mid (x, b) \in S_{N-1}\}$ .<sup>5</sup>

*The order of  $p$  is given by  $\text{ord}(p) = \max\{|S_j| \mid 1 \leq j \leq N\}$ . The recursive teaching dimension of  $C$  is defined by  $RTD(C) = \min\{\text{ord}(p) \mid p \text{ is a teaching plan for } C\}$ .*

The desired monotonicity property, see Proposition 14, follows immediately from the definition.

**Proposition 14** *If  $C$  is a concept class and  $C' \subseteq C$  is a subclass of  $C$ , then  $RTD(C') \leq RTD(C)$ .*

We can define a set of canonical teaching plans for any finite concept class  $C$ . As it will turn out, their order always equals  $RTD(C)$ .

**Definition 15** *Let  $C$  be a concept class,  $p = ((c_1, S_1), \dots, (c_N, S_N))$  a teaching plan for  $C$ .  $p$  is called a canonical teaching plan for  $C$ , if for any  $i, j \in \{1, \dots, N\}$ :*

$$i < j \Rightarrow TD(c_i, \{c_i, \dots, c_N\}) \leq TD(c_j, \{c_i, \dots, c_N\}).$$

**Theorem 16** *Let  $C$  be a concept class and  $p$  a canonical teaching plan for  $C$ . Then  $\text{ord}(p) = RTD(C)$ .*

<sup>5</sup>Note that the cardinality of both  $S_{N-1}$  and  $S_N$  must be 1.

*Proof.* Let  $C$  and  $p$  as in the theorem be given,  $p = ((c_1, S_1), \dots, (c_N, S_N))$ . Let  $p' = ((c'_1, S'_1), \dots, (c'_N, S'_N))$  be any teaching plan for  $C$ . It remains to prove that  $\text{ord}(p) \leq \text{ord}(p')$ .

For that purpose choose the minimal  $j \in \{1, \dots, N\}$  such that  $|S_j| = \text{ord}(p)$ . By definition of a teaching plan,  $TD(c_j, \{c_j, \dots, c_N\}) = \text{ord}(p)$ . Let  $i \in \{1, \dots, N\}$  be minimal such that  $c'_i \in \{c_j, \dots, c_N\}$ . Let  $k \in \{1, \dots, N\}$  fulfill  $c_k = c'_i$ . By definition of a canonical teaching plan,  $TD(c_k, \{c_j, \dots, c_N\}) \geq TD(c_j, \{c_j, \dots, c_N\}) = \text{ord}(p)$ . This obviously yields  $\text{ord}(p') \geq TD(c'_i, \{c'_i, \dots, c'_N\}) \geq TD(c_k, \{c_j, \dots, c_N\}) \geq \text{ord}(p)$ . ■

To summarize briefly, the recursive teaching dimension is a monotonic complexity notion which in fact has got some of the properties we desired; e.g., it is easily verified that  $RTD(C_0) = 1$  (by any teaching plan in which the empty concept occurs last) and that the  $RTD$  of the class of all monomials equals 2 (see below). Thus the  $RTD$  overcomes some of the weaknesses of  $BTD$ , while at the same time preserving monotonicity.

As it will turn out later, there are some interesting relations between  $BTD$ ,  $STD$ , and  $RTD$ .

A property that might be relevant for establishing these relations is based on the following definition.

**Definition 17** Let  $C$  be a concept class,  $|C| = N$ . A  $TS$ -teaching plan for  $C$  is a sequence

$$p = ((c_1, S_1^1), \dots, (c_N, S_1^N, \dots, S_N^N))$$

such that

1.  $C = \{c_1, \dots, c_N\}$ .
2.  $S_k^j \in TS(c_j, \{c_k, \dots, c_N\})$  for  $1 \leq k \leq j \leq N$ .
3.  $S_k^j \subseteq S_{k-1}^j$  for  $1 < k \leq j \leq N$ .

The order of  $p$  is given by  $\text{ord}(p) = \max\{|S_j^j| \mid 1 \leq j \leq N\}$ . The recursive  $TS$ -teaching dimension of  $C$  is defined by  $RTTD(C) = \min\{\text{ord}(p) \mid p \text{ is a } TS\text{-teaching plan for } C\}$ .

$TS$ -teaching plans differ from original teaching plans in that they require their sets being built up in stages as subsets of those in previous stages, starting from teaching sets.

However, as it turns out, concerning the  $RTD$  it suffices to consider this restricted form of teaching plans.

**Lemma 18** Let  $C$  be a concept class. Then  $RTTD(C) = RTD(C)$ . In particular, there is a  $TS$ -teaching plan  $p = ((c_1, S_1^1), \dots, (c_N, S_1^N, \dots, S_N^N))$  for  $C$  such that  $\text{ord}(p) = RTD(C)$  and  $((c_1, S_1^1), \dots, (c_N, S_N^N))$  is a canonical teaching plan for  $C$ .

The proof is omitted.

#### 4.4 Monomials revisited

In this subsection, we will pick up the two examples from Subsection 3.3 again, this time to determine the recursive teaching dimension.

**Theorem 19** Let  $m \in \mathbb{N}$  and  $C$  the class of all boolean functions over  $m$  variables that can be represented by a monomial. Then  $RTD(C) = 2$ .

*Proof.* Fix  $m$  and  $C$ . For all  $i \in \{0, \dots, m\}$  let  $C^i$  be the subclass of all  $c \in C$  that can be represented by a non-contradictory monomial  $M$  that has got  $i$  variables. There is exactly one concept in  $C$  not belonging to any subclass  $C^i$  of  $C$ , namely the concept  $c^*$  representable by a contradictory monomial.

The proof is based on the following observation.

*Observation.* For any  $i \in \{0, \dots, m\}$  and any  $c \in C^i$ :  $TD(c, C' \cup \{c^*\}) \leq 2$ , where  $C' = \bigcup_{i \leq j \leq m} C^j$ .

Now it is easily seen that  $\text{ord}(p) \leq 2$  for every teaching plan  $p = ((c_1, S_1), \dots, (c_N, S_N))$  for  $C$  that meets the following requirements:

- (a)  $c_1 \in C^0$  and  $c_N = c^*$ .
- (b) For any  $k, k' \in \{0, \dots, N-1\}$ : If  $k < k'$ , then  $c_k \in C^i$  and  $c_{k'} \in C^j$  for some  $i, j \in \{0, \dots, m\}$  with  $i \leq j$ .

Since obviously  $TD(c, C) \geq 2$  for all  $c \in C$ , we obtain  $RTD(C) = 2$ .

For illustration of the case  $m = 2$  see Table 5. ■

		$TS$
$\lambda$	$C^0$	$\{(00,1), (11,1)\}$
$v_1$	$C^1$	$\{(10,1), (11,1)\}$
$\bar{v}_1$	$C^1$	$\{(00,1), (01,1)\}$
$v_2$	$C^1$	$\{(01,1), (11,1)\}$
$\bar{v}_2$	$C^1$	$\{(00,1), (10,1)\}$
$v_1 \wedge v_2$	$C^2$	$\{(11,1)\}$
$v_1 \wedge \bar{v}_2$	$C^2$	$\{(10,1)\}$
$\bar{v}_1 \wedge v_2$	$C^2$	$\{(01,1)\}$
$\bar{v}_1 \wedge \bar{v}_2$	$C^2$	$\{(00,1)\}$
$v_1 \wedge \bar{v}_1$		$\{(00,0)\}$

Table 5: Recursive teaching sets in a teaching plan of order 2 for the class of all monomials over  $m = 2$  variables.  $\lambda$  denotes the empty monomial.

For the sake of completeness, note  $RTD(C_{\sqrt{DNF}}^m) = 1$  where  $C_{\sqrt{DNF}}^m$  is the class of boolean functions over  $m$  variables as defined in Subsection 3.3.

**Theorem 20**  $RTD(C_{\sqrt{DNF}}^m) = 1$  for all  $m \in \mathbb{N}$ .

*Sketch of proof.* This follows straightforwardly from the fact that  $TD(c, C_{\sqrt{DNF}}^m) = 1$  for every concept  $c$  corresponding to a 2-term DNF of form  $v_1 \vee M$ .

For illustration see Table 2. ■

## 5 Comparison of teaching dimension notions

This section provides an analysis of the relationships between  $RTD$ ,  $BTD$ , and  $STD$ .

**Theorem 21** 1. If  $C$  is a concept class then  $RTD(C) \leq BTD(C)$ .

2. There is a concept class  $C$  with  $RTD(C) < BTD(C)$ .

*Proof.* Assertion (2) is witnessed by the concept class  $C_0$  containing the empty concept and all singletons. Obviously,  $RTD(C_0) = 1$  and  $BTD(C_0) = 2$ .

To prove Assertion (1), let  $C$  be a concept class with  $RTD(C) = u$ . By Theorem 16 there is a canonical teaching plan  $p = ((c_1, S_1), \dots, (c_N, S_N))$  for  $C$  with  $ord(p) = u$ . Fix  $j \leq N$  minimal such that  $|S_j| = u$  and define  $C' = \{c_j, \dots, c_N\}$ . Obviously,  $RTD(C') = u$ . Moreover, using Theorem 12,  $BTD(C') \leq BTD(C)$ . Thus it suffices to prove  $u \leq BTD(C')$ .

To achieve this, we will prove by induction on  $k$  that  $u \leq BTD^k(c, C')$  for all  $k \in \mathbb{N}$  for all  $c \in C'$ .

$k = 0$ :  $BTD^0(c, C') = TD(c, C') \geq u$  for all  $c \in C'$ .

Induction hypothesis: assume  $u \leq BTD^k(c, C')$  for all  $c \in C'$  holds for a fixed  $k$ .

$k \rightsquigarrow k + 1$ : Suppose by way of contradiction that there is a concept  $c^* \in C'$  with  $u > BTD^{k+1}(c^*, C')$ . In particular, there exists a sample  $S^*$  such that  $|S^*| < u$  and  $Cons_{size}(S^*, C', k) = \{c^*\}$ .

By induction hypothesis, the set  $Cons_{size}(S^*, C', k)$  defined by  $\{c \in Cons(S^*, C') \mid BTD^k(c, C') \geq |S^*|\}$  is equal to  $Cons(S^*, C')$ . Note that  $TD(c, C') \geq u$  for all  $c \in C'$  implies either  $|Cons(S^*, C')| \geq 2$  or  $Cons(S^*, C') = \emptyset$ . We obtain a contradiction to  $Cons_{size}(S^*, C', k) = \{c^*\}$ .

This completes the proof.  $\blacksquare$

Comparing the  $STD$  to the  $RTD$  turns out to be a bit more complex. We can show that the recursive teaching dimension can be arbitrarily larger than the subset teaching dimension; it can even be larger than the maximal  $STD$  computed over all subsets of the concept class.

**Theorem 22** 1. For each  $u \in \mathbb{N}$  there is a concept class  $C$  such that  $STD(C) = 1$  and  $RTD(C) = u$ .  
2. There is a concept class  $C$  such that  $\max\{STD(C') \mid C' \subseteq C\} < RTD(C)$ .

*Sketch of proof.* Assertion (1) is witnessed by the classes  $C_{0/1}^u$  defined in the proof of Theorem 6.

To verify Assertion (2), consider the concept class  $C = \{c_1, \dots, c_6\}$  given by  $c_1 = \emptyset$ ,  $c_2 = \{x_1\}$ ,  $c_3 = \{x_1, x_2\}$ ,  $c_4 = \{x_2, x_3\}$ ,  $c_5 = \{x_2, x_4\}$ ,  $c_6 = \{x_2, x_3, x_4\}$ . It is not hard to verify that  $TD(c, C) = 2$  for all  $c \in C$  and thus  $ord(p) = 2$  for every teaching plan  $p$  for  $C$ . Therefore  $RTD(C) = 2$ . Moreover  $STD(C') = 1$  for all  $C' \subseteq C$  (the computation of  $STD(C)$  is shown in Table 6; further details are omitted).  $\blacksquare$

concept	$STS^0$	$STS^1$	$STS^2$
$\emptyset$	$\{(x_1, 0), (x_2, 0)\}$	$\{(x_1, 0)\}$	$\{(x_1, 0)\}$
$\{x_1\}$	$\{(x_1, 1), (x_2, 0)\}$	$\{(x_1, 1), (x_2, 0)\}$	$\{(x_1, 1)\}$ $\{(x_2, 0)\}$
$\{x_1, x_2\}$	$\{(x_1, 1), (x_2, 1)\}$	$\{(x_2, 1)\}$	$\{(x_2, 1)\}$
$\{x_2, x_3\}$	$\{(x_3, 1), (x_4, 0)\}$	$\{(x_4, 0)\}$	$\{(x_4, 0)\}$
$\{x_2, x_4\}$	$\{(x_3, 0), (x_4, 1)\}$	$\{(x_3, 0)\}$	$\{(x_3, 0)\}$
$\{x_2, x_3, x_4\}$	$\{(x_3, 1), (x_4, 1)\}$	$\{(x_3, 1), (x_4, 1)\}$	$\{(x_3, 1)\}$ $\{(x_4, 1)\}$

Table 6: Iterated subset teaching sets for the class  $C = \{c_1, \dots, c_6\}$  given by  $c_1 = \emptyset$ ,  $c_2 = \{x_1\}$ ,  $c_3 = \{x_1, x_2\}$ ,  $c_4 = \{x_2, x_3\}$ ,  $c_5 = \{x_2, x_4\}$ ,  $c_6 = \{x_2, x_3, x_4\}$ .

We conjecture moreover that  $STD(C) \leq RTD(C)$  for all concept classes  $C$ , however, we cannot prove that at the

time of writing. However, we can provide a general proof idea that solely relies on a lemma that we conjecture.

**Lemma 23 (Conjecture)** Let  $C$  be a concept class and  $p = ((c_1, S_1), \dots, (c_N, S_N))$  a teaching plan for  $C$ . Let  $j$  fulfill  $ord(p) = |S_j|$  and  $STD(c_j, C) \geq ord(p)$ . Then there is a teaching plan

$$p = ((c_1, S'_1), \dots, (c_N, S'_N))$$

for  $C$  and a sample  $S \in STS(c_j, C)$  such that  $S'_j \subseteq S$ .

The proof of the following theorem, which helps to summarize the relations between our different variants of teaching dimensions, relies on this lemma—hence in fact the theorem is also a conjecture at the time of writing. Note that its correctness, together with Theorem 21 and Lemma 18, would imply

$$STD(C) \leq RTD(C) = RTTD(C) \leq BTD(C)$$

for all concept classes  $C$ . Here all inequalities are necessary since proven to not be equalities.

**Theorem 24 (Based on conjecture Lemma 23)** Let  $C$  be a concept class. Then  $STD(C) \leq RTD(C)$ .

Sketch of proof (relying on Lemma 23). Prove property  $(P_j)$  by induction for all  $j \geq 1$ .

$(P_j)$ :

If  $C$  is a concept class of at least  $j$  concepts and  $p$  is any teaching plan for  $C$  (not necessarily canonical), then  $STD(c_j, C) \leq ord(p)$  where  $c_j$  is the  $j^{\text{th}}$  concept in the teaching plan  $p$ .

For  $j = 1$  this is obvious, because

$$STD(c_1, C) \leq TD(c_1, C) \leq ord(p).$$

The induction hypothesis is that  $(P_i)$  holds for all  $i \leq j$ ,  $j$  fixed.

To prove  $(P_{j+1})$ , choose a concept class  $C$  and a teaching plan  $p = ((c_1, S_1), \dots, (c_N, S_N))$  for  $C$ . Consider the  $j + 1^{\text{st}}$  concept  $c_{j+1}$  in  $p$ .

Case 1.  $|S_{j+1}| < ord(p)$ .

If  $|S_{j+1}| < ord(p)$ , then we swap  $c_j$  and  $c_{j+1}$  and get a new teaching plan

$$p = ((c_1, S_1), \dots, (c_{j-1}, S_{j-1}), (c_{j+1}, T), (c_j, T'), \dots, (c_n, S_n))$$

for  $C$ . Note that  $|T'| \leq |S_j|$ . Now  $c_{j+1}$  is in  $j^{\text{th}}$  position and its corresponding set  $T$ , due to the swap, fulfills  $|T| \leq |S_{j+1}| + 1 \leq ord(p)$ . By induction hypothesis we get  $STD(c_{j+1}, C) \leq ord(p)$ .

Case 2.  $|S_{j+1}| = ord(p)$ .

This is the more difficult case. Using Lemma 18 we can prove that  $S_{j+1}$  is a subset of a teaching set of  $c_{j+1}$  with respect to any of the classes  $\{c_i, \dots, c_N\}$  where  $i \leq j + 1$ .

But in fact we would need Lemma 23 to tell us that  $S_{j+1}$  is a subset of a subset teaching set of  $c_{j+1}$  with respect to  $C$ .

Assume that  $STD(c_{j+1}, C) > ord(p)$ . This implies that  $S_{j+1}$  is a subset of some subset teaching set for  $c_{j+1}$  without being contained in any other subset teaching set for any

other concept. Then  $S_{j+1}$  would itself be a subset teaching set for  $c_{j+1}$  in contradiction to its size being smaller than  $STD(c_{j+1}, C)$ .

To see why  $S_{j+1}$  couldn't be contained in any subset teaching set for any  $c \neq c_{j+1}$ ,  $c \in C$ , note that  $c_{j+2}, \dots, c_N$  are not consistent with  $S_{j+1}$  and the concepts  $c_1, \dots, c_j$  by induction hypothesis have a too low subset teaching dimension in  $C$ . ■

## 6 Conclusions and open problems

We have introduced a new model of teaching and learning, based on what we call subset teaching sets. This model captures the idea of a teacher and a learner cooperating in order to learn concepts in finite classes from small samples.

This model avoids coding tricks and provides a generally applicable procedure for a uniform protocol of cooperative learning. It achieves results that are, for a specific concept class, such as the monomials, no less efficient than known algorithms that are designed especially for that one concept class (and perform inefficiently in terms of sample size on others).

The resulting subset teaching dimension turns out to be nonmonotonic—a fact that is illustrated and explained by the nature of the underlying definition.

In order to compare this subset teaching dimension to monotonic variants of teaching dimensions related to cooperation in learning, we introduced two equivalent notions of “recursive teaching dimensions”, being monotonic by definition. They turn out to be very helpful in providing bounds for previous notions (they are significantly better than the original teaching dimension and variants thereof). However, even though they behave so well, the nonmonotonic subset teaching dimension in general seems to be better.

Examples have shown that even the recursive teaching dimensions cannot always compete with the subset teaching dimension, though our conjecture that the recursive teaching dimension can never be lower than the subset teaching dimension is still open.

We plan to close this gap in our proof, to find characterizations for these teaching dimensions, and to provide evidence to another conjecture, namely that, for reasonable definitions of the term “coding trick”, there is no teaching and learning model that avoids coding tricks and is better than the model based on the subset teaching dimension.

## Acknowledgments

We gratefully acknowledge the support of Laura Zilles and Michael Geilke who developed and provided a software tool for computing subset teaching sets and teaching plans.

Many thanks are due to the anonymous referees for their helpful comments.

This work was partly funded by the Alberta Ingenuity Centre for Machine Learning.

## References

[ABCS92] M. Anthony, G. Brightwell, D.A. Cohen, and J. Shawe-Taylor. On exact specification by examples. In *Proc. of 5th Annual Workshop*

- on *Computational Learning Theory (COLT'92)*, pages 311–318. ACM, New York, 1992.
- [AK97] D. Angluin and M. Krikis. Teachers, learners and black boxes. In *Proc. of the 10th Annual Conference on Computational Learning Theory (COLT'97)*, pages 285–297. ACM, New York, 1997.
- [Bal08] F. Balbach. Measuring teachability using variants of the teaching dimension. *Theoret. Comput. Sci.*, 397(1-3):94–113, 2008.
- [BE98] S. Ben-David and N. Eiron. Self-directed learning and its relation to the VC-dimension and to teacher-directed learning. *Machine Learning*, 33(1):87–104, 1998.
- [FKW93] R. Freivalds, E.B. Kinber, and R. Wiehagen. On the power of inductive inference from good examples. *Theoret. Comput. Sci.*, 110(1):131–144, 1993.
- [GK95] S.A. Goldman and M.J. Kearns. On the complexity of teaching. *J. Comput. Syst. Sci.*, 50(1):20–31, 1995.
- [GM96] S.A. Goldman and H.D. Mathias. Teaching a smarter learner. *J. Comput. Syst. Sci.*, 52(2):255–267, 1996.
- [Han07] S. Hanneke. Teaching dimension and the complexity of active learning. In *Proc. of the 20th Annual Conference on Learning Theory (COLT 2007)*, pages 66–81. LNCS 4539, Springer, Berlin, 2007.
- [Heg95] T. Hegedüs. Generalized teaching dimensions and the query complexity of learning. In *Proc. of the 8th Annual Conference on Computational Learning Theory (COLT'95)*, pages 108–117. ACM, New York, 1995.
- [JT92] J. Jackson and A. Tomkins. A computational model of teaching. In *Proc. of 5th Annual Workshop on Computational Learning Theory (COLT'92)*, pages 319–326. ACM, New York, 1992.
- [LNW98] S. Lange, J. Nessel, and R. Wiehagen. Learning recursive languages from good examples. *Ann. Math. Artif. Intell.*, 23(1-2):27–52, 1998.
- [Mat97] H.D. Mathias. A model of interactive teaching. *J. Comput. Syst. Sci.*, 54(3):487–501, 1997.
- [OS02] Matthias Ott and Frank Stephan. Avoiding coding tricks by hyperrobust learning. *Theoret. Comput. Sci.*, 284(1):161–180, 2002.
- [RY95] R.L. Rivest and Y.L. Yin. Being taught can be faster than asking questions. In *Proc. of the 8th Annual Conference on Computational Learning Theory (COLT'95)*, pages 144–151. ACM, New York, 1995.
- [SM91] A. Shinohara and S. Miyano. Teachability in computational learning. *New Generation Comput.*, 8(4):337–348, 1991.
- [Val84] L.G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

---

# On The Power of Membership Queries in Agnostic Learning

---

Vitaly Feldman\*

IBM Almaden Research Center  
650 Harry rd.  
San Jose, CA 95120  
vitaly@post.harvard.edu

## Abstract

We study the properties of the agnostic learning framework of Haussler [Hau92] and Kearns, Schapire and Sellie [KSS94]. In particular, we address the question: is there any situation in which membership queries are useful in agnostic learning?

Our results show that the answer is negative for distribution-independent agnostic learning and positive for agnostic learning with respect to a specific marginal distribution. Namely, we give a simple proof that any concept class learnable agnostically by a distribution-independent algorithm with access to membership queries is also learnable agnostically without membership queries. This resolves an open problem posed by Kearns *et al.* [KSS94]. For agnostic learning with respect to the uniform distribution over  $\{0, 1\}^n$  we show a concept class that is learnable with membership queries but computationally hard to learn from random examples alone (assuming that one-way functions exist).

## 1 Introduction

The agnostic framework [Hau92, KSS94] is a natural generalization of Valiant's PAC learning model [Val84]. In this model no assumptions are made on the labels of the examples given to the learning algorithm, in other words, the learning algorithm has no prior beliefs about the target concept (and hence the name of the model). The goal of the agnostic learning algorithm for a concept class  $\mathcal{C}$  is to produce a hypothesis  $h$  whose error on the target concept is close to the best possible by a concept from  $\mathcal{C}$ . This model reflects a common empirical approach to learning, where few or no assumptions are made on the process that generates the examples and a limited space of candidate hypothesis functions is searched in an attempt to find the best approximation to the given data.

Designing algorithms that learn efficiently in this model is notoriously hard and very few positive results are known

---

\*Part of the work done while the author was at Harvard University supported by grants from the National Science Foundation NSF-CCF-04-32037 and NSF-CCF-04-27129.

[KSS94, LBW95, GKS01, KKMS05, GKK08, KMV08]. Furthermore, strong computational hardness results are known for agnostic learning of even the simplest classes of functions such as parities, monomials and halfspaces [Hås01, Fel06, FGKP06, GR06] (albeit only for *proper* learning). Reductions from long-standing open problems for PAC learning to agnostic learning of simple classes of functions provide another indication of the hardness of agnostic learning [KSS94, KKMS05, FGKP06].

A membership oracle allows a learning algorithm to obtain the value of the unknown target function  $f$  on any point in the domain. It can be thought of as modeling the access to an expert or ability to conduct experiments. Learning with membership queries in both PAC and Angluin's exact models [Ang88] was studied in numerous works. For example monotone DNF formulas, finite automata and decision trees are only known to be learnable with membership queries [Val84, Ang88, Bsh95]. It is well-known and easy to prove that the PAC model with membership queries is strictly stronger than the PAC model without membership queries (if one-way functions exist).

Membership queries are also used in several agnostic learning algorithms. The first one is the famous algorithm of Goldreich and Levin introduced in a cryptographic context (even before the definition of the agnostic learning model) [GL89]. Their algorithm learns parities agnostically with respect to the uniform distribution using membership queries. Kushilevitz and Mansour used this algorithm to PAC learn decision trees [KM93] and it has since found numerous other significant applications. More efficient versions of this algorithm were also given by Levin [Lev93], Bshouty, Jackson and Tamon [BJT99] and Feldman [Fel07]. Recently, Gopalan, Kalai and Klivans gave an elegant algorithm that learns decision trees agnostically over the uniform distribution and uses membership queries [GKK08].

### 1.1 Our Contribution

In this work we study the power of membership queries in the agnostic learning model. This question was posed by Kearns *et al.* [KSS94] and, to the best of our knowledge, has not been addressed prior to our work. In this work we present two results on this question. In the first result we prove that every concept class learnable agnostically without membership queries is also learnable agnostically without membership queries (see Theorem 6 for a formal statement). This proves the conjecture of Kearns *et al.* [KSS94]. The

reduction we give modifies the distribution of examples and therefore is only valid for distribution-independent learning, that is, when a single learning algorithm is used for every distribution over the examples. The simple proof of this result explains why the known distribution-independent agnostic learning algorithms do not use membership queries [KSS94, KKMS05, KMV08].

The proof of this result also shows equivalence of two standard agnostic models: the one in which examples are labeled by an unrestricted function and the one in which examples come from a joint distribution over the domain and the labels.

Our second result is a proof that there exists a concept class that is agnostically learnable with membership queries over the uniform distribution on  $\{0, 1\}^n$  but hard to learn in the same setting without membership queries. This result is based on the most basic cryptographic assumption, namely the existence of one-way functions. Note that an unconditional separation of these two models would imply  $\text{NP} \neq \text{P}$ . Cryptographic assumptions are essential for numerous other hardness results in learning theory (cf. [KV94, Kha95]). Our construction is based on the use of pseudorandom function families, list-decodable codes and a variant of an idea from the work of Elbaz, Lee, Servedio and Wan [ELSW07]. Sections 4.1 and 4.2 describe the technique and its relation to prior work in more detail.

This result is, perhaps, unsurprising since agnostic learning of parities with respect to the uniform distribution from random examples only is commonly considered hard and is known to be equivalent to decoding of random linear codes, a long-standing open problem in coding theory. The best known algorithm for this problem runs in time  $O(2^{n/\log n})$  [FGKP06]. It is therefore natural to expect that membership queries are provably helpful for uniform distribution agnostic learning. The proof of this result however is substantially less straightforward than one might expect (and than the analogous separation for PAC learning). Here the main obstacle is the same as in proving positive results for agnostic learning: the requirements of the model impose severe limits on concept classes for which the agnostic guarantees can be provably satisfied.

## 1.2 Organization

Following the preliminaries, our first result is described in Section 3. The second result appears in Section 4.

## 2 Preliminaries

Let  $X$  denote the domain or the *input space* of a learning problem. The domain of the problems that we study is  $\{0, 1\}^n$ , or the  $n$ -dimensional *Boolean hypercube*. A *concept* over  $X$  is a  $\{-1, 1\}$  function over the domain and a *concept class*  $\mathcal{C}$  is a set of concepts over  $X$ . The unknown function  $f \in \mathcal{C}$  that a learning algorithm is trying to learn is referred to as the *target concept*.

A parity function is a function equal to the *XOR* of some subset of variables. For a Boolean vector  $a \in \{0, 1\}^n$  we define the parity function  $\chi_a(x)$  as  $\chi_a(x) = (-1)^{a \cdot x} = (-1)^{\oplus_{i \leq n} a_i x_i}$ . We denote the concept class of parity functions  $\{\chi_a \mid a \in \{0, 1\}^n\}$  by  $\text{PAR}$ . A *k-junta* is a function that depends only on  $k$  variables.

A *representation class* is a concept class defined by providing a specific way to represent each function in the concept class. All of the above concept classes are in fact representation classes. For a representation class  $\mathcal{F}$  we say that an algorithm outputs  $f \in \mathcal{F}$  if the algorithm outputs  $f$  in the representation associated with  $\mathcal{F}$ .

### 2.1 PAC Learning Model

The learning models discussed in this work are based on Valiant's well-known PAC model [Val84]. In this model, for a concept  $f$  and distribution  $D$  over  $X$ , an *example oracle*  $\text{EX}(D, f)$  is the oracle that, upon request, returns an example  $(x, f(x))$  where  $x$  is chosen randomly with respect to  $D$ . For  $\epsilon \geq 0$  we say that function  $g$   $\epsilon$ -approximates a function  $f$  with respect to distribution  $D$  if  $\Pr_D[f(x) = g(x)] \geq 1 - \epsilon$ . In the PAC learning model the learner is given access to  $\text{EX}(D, f)$  where  $f$  is assumed to belong to a fixed concept class  $\mathcal{C}$ .

**Definition 1** For a concept class  $\mathcal{C}$ , we say that an algorithm  $\text{Alg}$  PAC learns  $\mathcal{C}$ , if for every  $\epsilon > 0$ ,  $\delta > 0$ ,  $f \in \mathcal{C}$ , and distribution  $D$  over  $X$ ,  $\text{Alg}$ , given access to  $\text{EX}(D, f)$ , outputs, with probability at least  $1 - \delta$ , a hypothesis  $h$  that  $\epsilon$ -approximates  $f$ .

The learning algorithm is *efficient* if its running time and the time to compute  $h$  are polynomial in  $1/\epsilon, 1/\delta$  and the *size*  $\sigma$  of the learning problem. Here by the size we refer to the maximum description length of an element in  $X$  (e.g.  $n$  when  $X = \{0, 1\}^n$ ) plus a bound on the length of the description of a concept in  $\mathcal{C}$  in the representation associated with  $\mathcal{C}$ .

An algorithm is said to *weakly learn*  $\mathcal{C}$  if it produces a hypothesis  $h$  that  $(\frac{1}{2} - \frac{1}{p(\sigma)})$ -approximates  $f$  for some polynomial  $p$ .

### 2.2 Agnostic Learning Model

The *agnostic learning* model was introduced by Haussler [Hau92] and Kearns *et al.* [KSS94] in order to model situations in which the assumption that examples are labeled by some  $f \in \mathcal{C}$  does not hold. In its least restricted version the examples are generated from some unknown distribution  $A$  over  $X \times \{-1, 1\}$ . The goal of an agnostic learning algorithm for a concept class  $\mathcal{C}$  is to produce a hypothesis whose error on examples generated from  $A$  is close to the best possible by a concept from  $\mathcal{C}$ . Class  $\mathcal{C}$  is referred to as the *touchstone* class in this setting. More generally, the model allows specification of the assumptions made by a learning algorithm by describing a set  $\mathcal{A}$  of distributions over  $X \times \{-1, 1\}$  that restricts the distributions over  $X \times \{-1, 1\}$  seen by a learning algorithm. Such  $\mathcal{A}$  is referred to as the *assumption class*. Any distribution  $A$  over  $X \times \{-1, 1\}$  can be described uniquely by its marginal distribution  $D$  over  $X$  and the expectation of  $b$  given  $x$ . That is, we refer to a distribution  $A$  over  $X \times \{-1, 1\}$  by a pair  $(D_A, \phi_A)$  where  $D_A(z) = \Pr_{(x,b) \sim A}[x = z]$  and

$$\phi_A(z) = \mathbf{E}_{(x,b) \sim A}[b \mid z = x].$$

Formally, for a Boolean function  $h$  and a distribution  $A = (D, \phi)$  over  $X \times \{-1, 1\}$ , we define

$$\Delta(A, h) = \Pr_{(x,b) \sim A}[h(x) \neq b] = \mathbf{E}_D[|\phi(x) - h(x)|/2].$$

Similarly, for a concept class  $\mathcal{C}$ , define

$$\Delta(A, \mathcal{C}) = \inf_{h \in \mathcal{C}} \{\Delta(A, h)\}.$$

Kearns *et al.* define agnostic learning as follows [KSS94].

**Definition 2** An algorithm  $\text{Alg}$  agnostically learns a concept class  $\mathcal{C}$  by a representation class  $\mathcal{H}$  assuming  $\mathcal{A}$  if for every  $\epsilon > 0, \delta > 0, A \in \mathcal{A}$ ,  $\text{Alg}$  given access to examples drawn randomly from  $A$ , outputs, with probability at least  $1 - \delta$ , a hypothesis  $h \in \mathcal{H}$  such that  $\Delta(A, h) \leq \Delta(A, \mathcal{C}) + \epsilon$ .

The learning algorithm is *efficient* if it runs in time polynomial  $1/\epsilon, \log(1/\delta)$  and  $\sigma$  (the size of the learning problem). If  $\mathcal{H} = \mathcal{C}$  then, by analogy with the PAC model, the learning is referred to as *proper*. We drop the reference to  $\mathcal{H}$  to indicate that  $\mathcal{C}$  is learnable for some  $\mathcal{H}$ .

A number of versions of the agnostic model are commonly considered (and often referred to as *the* agnostic learning model). In fully agnostic learning  $\mathcal{A}$  is the set of all distributions over  $X \times \{-1, 1\}$ . Another version assumes that examples are labeled by an unrestricted function. That is, the set  $\mathcal{A}$  contains distribution  $A = (D, f)$  for every Boolean function  $f$  and distribution  $D$ . Note that access to random examples from  $A = (D, f)$  is equivalent to access to  $\text{EX}(D, f)$ . Following Kearns *et al.*, we refer to this version as *agnostic PAC learning* [KSS94] (they also require that  $\mathcal{H} = \mathcal{C}$  but this constraint is unrelated and is now generally referred to as *properness*). Theorem 6 implies that these versions are essentially equivalent. In *distribution-specific* versions of this model for every  $(D, \phi) \in \mathcal{A}$ ,  $D$  equals to some fixed distribution known in advance.

We also note that the agnostic PAC learning model can also be thought of as a model of adversarial classification noise. By definition, a Boolean function  $g$  differs from some function  $f \in \mathcal{C}$  on  $\Delta(g, \mathcal{C})$  fraction of the domain. Therefore  $g$  can be thought of as  $f$  corrupted by noise of rate  $\Delta_D(f, \mathcal{C})$ . Unlike in the random classification noise model the points on which a concept can be corrupted are unrestricted and therefore we refer to it as adversarial noise.

### Uniform Convergence

A natural approach to agnostic learning is to first draw a sample of fixed size and then choose a hypothesis that best fits the observed labels. The conditions in which this approach is successful were studied in works of Dudley [Dud78], Pollard [Pol84], Haussler [Hau92], Vapnik [Vap98] and others. They give a number of conditions on the hypothesis class  $\mathcal{H}$  that guarantee *uniform convergence* of empirical error to the true error. That is, existence of a function  $m_{\mathcal{H}}(\epsilon, \delta)$  such that for every distribution  $A$  over examples, every  $h \in \mathcal{H}$ ,  $\epsilon > 0, \delta > 0$ , the empirical error of  $h$  on sample of  $m_{\mathcal{H}}(\epsilon, \delta)$  examples randomly chosen from  $A$  is, with probability at least  $1 - \delta$ , within  $\epsilon$  of  $\Delta(A, h)$ . We denote the empirical error of  $h$  on sample  $S$  by  $\Delta(S, h)$ . In the Boolean case, the following result of Vapnik and Chervonenkis will be sufficient for our purposes [VC71].

**Theorem 3** Let  $\mathcal{H}$  be a concept class over  $X$  of VC dimension  $d$ . Then for every distribution  $A$  over  $X \times \{-1, 1\}$ ,

every  $h \in \mathcal{H}$ ,  $\epsilon > 0, \delta > 0$ , and sample  $S$  of size  $m = O(d/\epsilon^2 \cdot \log(1/\delta))$  randomly drawn with respect to  $A$ ,

$$\Pr[|\Delta(A, h) - \Delta(S, h)| \geq \epsilon] \leq \delta.$$

In fact a simple uniform convergence result based on the cardinality of the function class follows easily from Chernoff bounds (cf. [Hau92]). That is Theorem 3 holds for  $m = O(\log |\mathcal{H}|/\epsilon^2 \cdot \log(1/\delta))$ . This result would also be sufficient for our purposes but might give somewhat weaker bounds.

### 2.3 Membership Queries

A membership oracle for a function  $f$  is the oracle that, given any point  $z \in \{0, 1\}^n$ , returns the value  $f(z)$  [Val84]. We denote it by  $\text{MEM}(f)$ . We refer to agnostic PAC learning with access to  $\text{MEM}(f)$  where  $f$  is the unknown function that labels the examples as *agnostic PAC+MQ* learning. Similarly, one can extend the definition of a membership oracle to fully agnostic learning. For a distribution  $A$  over  $X \times \{-1, 1\}$ , let  $\text{MEM}(A)$  be the oracle that, upon query  $z$ , returns  $b \in \{-1, 1\}$  with probability  $\Pr_A[(x, b) \mid x = z]$ . We say that  $\text{MEM}(A)$  is *persistent* if given the same query the oracle responds with the same label.

### 2.4 Fourier Transform

Our separation result uses Fourier-analytic techniques introduced to learning theory by Linial, Mansour and Nisan [LMN93]. It is used primarily in the context of learning with respect to the uniform distribution and therefore in the discussion below all probabilities and expectations are taken with respect to the uniform distribution  $U$  unless specifically stated otherwise.

Define an inner product of two real-valued functions over  $\{0, 1\}^n$  to be  $\langle f, g \rangle = \mathbf{E}_x[f(x)g(x)]$ . The technique is based on the fact that the set of all parity functions  $\{\chi_a(x)\}_{a \in \{0, 1\}^n}$  forms an orthonormal basis of the linear space of real-valued functions over  $\{0, 1\}^n$  with the above inner product. This fact implies that any real-valued function  $f$  over  $\{0, 1\}^n$  can be uniquely represented as a linear combination of parities, that is  $f(x) = \sum_{a \in \{0, 1\}^n} \hat{f}(a)\chi_a(x)$ . The coefficient  $\hat{f}(a)$  is called Fourier coefficient of  $f$  on  $a$  and equals  $\mathbf{E}_x[f(x)\chi_a(x)]$ ;  $a$  is called the *index* of  $\hat{f}(a)$ . We say that a Fourier coefficient  $\hat{f}(a)$  is  $\theta$ -heavy if  $|\hat{f}(a)| \geq \theta$ . Let  $L_2(f) = \mathbf{E}_x[(f(x))^2]^{1/2}$ . Parseval's identity states that

$$(L_2(f))^2 = \mathbf{E}_x[(f(x))^2] = \sum_a \hat{f}^2(a).$$

Let  $A = (U, \phi)$  be a distribution over  $\{0, 1\}^n \times \{-1, 1\}$  with uniform marginal distribution over  $\{0, 1\}^n$ . Fourier coefficient  $\hat{\phi}(a)$  can be easily related to the error of  $\chi_a(x)$  on  $A$ . That is,

$$\Pr_A[b \neq \chi_a(x)] = (1 - \hat{\phi}(a))/2. \quad (1)$$

Therefore, agnostic learning of parities amounts to finding the largest (within  $\epsilon$ ) Fourier coefficient of  $\phi(x)$ . The first algorithm for this task was given by Goldreich and Levin [GL89]. Given access to membership oracle, for every  $\epsilon > 0$  their algorithm can efficiently find all  $\epsilon$ -heavy Fourier coefficients.

**Theorem 4 ([GL89])** *There exists an algorithm  $\text{GL}$  that for every distribution  $A = (U, \phi)$  and every  $\epsilon, \delta > 0$ , given access to  $\text{MEM}(A)$ ,  $\text{GL}(\epsilon, \delta)$  returns, with probability at least  $1 - \delta$ , a set of indices  $T \subseteq \{0, 1\}^n$  that contains all  $a$  such that  $|\hat{\phi}(a)| \geq \epsilon$  and for all  $a \in T$ ,  $|\hat{\phi}(a)| \geq \epsilon/2$ . Furthermore, the algorithm runs in time polynomial in  $n, 1/\epsilon$  and  $\log(1/\delta)$ .*

Note that by Parseval's identity, the condition  $|\hat{\phi}(a)| \geq \epsilon/2$  implies that there are at most  $4/\epsilon^2$  elements in  $T$ .

## 2.5 Pseudo-random Function Families

A key part of our construction in Section 4 will be based on the use of pseudorandom functions families defined by Goldreich, Goldwasser and Micali [GGM86].

**Definition 5** *A function family  $\mathcal{F} = \{F\}_{n=1}^\infty$  where  $F_n = \{\pi_z\}_{z \in \{0,1\}^n}$  is a pseudorandom function family if*

- For every  $n$  and  $z \in \{0, 1\}^n$ ,  $\pi_z$  is an efficiently evaluable Boolean function on  $\{0, 1\}^n$ .
- Any adversary  $M$  whose resources are bounded by a polynomial in  $n$  can distinguish between a function  $\pi_z$  (where  $z \in \{0, 1\}^n$  is chosen randomly and kept secret) and a totally random function from  $\{0, 1\}^n$  to  $\{-1, 1\}$  only with negligible probability. That is, for every probabilistic polynomial time  $M$  with an oracle access to a function from  $\{0, 1\}^n$  to  $\{-1, 1\}$  and a negligible function  $\nu(n)$ ,

$$|\Pr[M^{\pi_z}(1^n) = 1] - \Pr[M^\rho(1^n) = 1]| \leq \nu(n),$$

where  $\pi_z$  is a function randomly and uniformly chosen from  $F_n$  and  $\rho$  is a randomly chosen function from  $\{0, 1\}^n$  to  $\{-1, 1\}$ . The probability is taken over the random choice of  $\pi_z$  or  $\rho$  and the coin flips of  $M$ .

Håstad *et al.* give a construction of pseudorandom function families based on the existence of one-way functions [HILL99].

## 3 Distribution-Independent Agnostic Learning

In this section we show that in distribution-independent agnostic learning membership queries do not help. In addition, we prove that fully agnostic learning is equivalent to agnostic PAC learning. Our proof is based on two simple observations about agnostic learning via empirical error minimization. Values of the unknown function on points outside of the sample can be set to any value without changing the best fit by a function from the touchstone class. Therefore membership queries do not make empirical error minimization easier. In addition, points with contradicting labels do not influence the complexity of empirical error minimization since any function has the same error on pairs of contradicting labels. We will now provide the formal statement of this result.

**Theorem 6** *Let  $\text{Alg}$  be an algorithm that agnostically PAC+MQ learns a concept class  $\mathcal{C}$  by a representation class  $\mathcal{H}$  in time  $T(\sigma, \epsilon, \delta)$  and outputs a hypothesis from a class  $\mathcal{H}$  of VC dimension  $d(\sigma, \epsilon)$ . Then  $\mathcal{C}$  is (fully) agnostically learnable by  $\mathcal{H}$  in time  $T(\sigma, \epsilon/2, \delta/2) + O(d(\sigma, \epsilon/2) \cdot \epsilon^{-2} \log(1/\delta))$ .*

**Proof:** Let  $A = (D, \phi)$  be a distribution over  $X \times \{-1, 1\}$ . Our reduction works as follows. Start by drawing  $m$  examples from  $A$  for  $m$  to be defined later. Denote this sample by  $S$ . Let  $S'$  be  $S$  with all contradicting pairs of examples removed, that is for each example  $(x, 1)$  we remove it together with one example  $(x, -1)$ . Every function has the same error rate of  $1/2$  with examples in  $S \setminus S'$ . Therefore for every function  $h$ ,

$$\begin{aligned} \Delta(S, h) &= \frac{\Delta(S', h)|S'| + |S \setminus S'|/2}{|S|} \\ &= \Delta(S', h) \frac{|S'|}{m} + \frac{m - |S'|}{2m} \end{aligned} \quad (2)$$

and hence

$$\Delta(S, \mathcal{C}) = \Delta(S', \mathcal{C}) \frac{|S'|}{m} + \frac{m - |S'|}{2m} \quad (3)$$

Let  $f(x)$  denote the function equal to  $b$  if  $(x, b) \in S'$  and equal to 1 otherwise. Let  $D_{S'}$  denote the uniform distribution over  $S'$ . Given the sample  $S'$  we can easily simulate the example oracle  $\text{EX}(D_{S'}, f)$  and  $\text{MEM}(f)$ . We run  $\text{Alg}(\epsilon/2, \delta/2)$  with these oracles and denote its output by  $h$ . Note, that this simulates  $\mathcal{A}$  in the agnostic PAC+MQ setting over distribution  $(D_{S'}, f)$ .

By the definition of  $D_{S'}$ , for any Boolean function  $g(x)$ ,

$$\begin{aligned} \Pr_{D_{S'}}[f(x) \neq g(x)] &= \frac{1}{|S'|} |\{x \in S' \mid f(x) \neq g(x)\}| \\ &= \Delta(S', g). \end{aligned}$$

That is, the error of any function  $g$  on  $D_{S'}$  is exactly the empirical error of  $g$  on sample  $S'$ . Thus  $\Delta((D_{S'}, f), h) = \Delta(S', h)$  and  $\Delta((D_{S'}, f), \mathcal{C}) = \Delta(S', \mathcal{C})$ . By the correctness of  $\text{Alg}$ , with probability at least  $1 - \delta/2$ ,  $\Delta(S', h) \leq \Delta(S', \mathcal{C}) + \epsilon/2$ . By equations (2) and (3) we thus obtain that

$$\begin{aligned} \Delta(S, h) &= \Delta(S', h) \frac{|S'|}{m} + \frac{m - |S'|}{2m} \\ &\leq (\Delta(S', \mathcal{C}) + \frac{\epsilon}{2}) \frac{|S'|}{m} + \frac{m - |S'|}{2m} = \Delta(S, \mathcal{C}) + \frac{\epsilon}{2} \frac{|S'|}{m} \end{aligned}$$

Therefore  $\Delta(S, h) \leq \Delta(S, \mathcal{C}) + \epsilon/2$ . We can apply the VC dimension-based uniform convergence results for  $\mathcal{H}$  [VC71] (Theorem 3) to conclude that for

$$m(\epsilon/4, \delta/4) = O\left(\frac{d(\sigma, \epsilon/2) \log(1/\delta)}{\epsilon^2}\right),$$

with probability at least  $1 - \delta/2$ ,  $\Delta(A, h) \leq \Delta(S, h) + \frac{\epsilon}{4}$  and  $\Delta(S, \mathcal{C}) + \frac{\epsilon}{4} \leq \Delta(A, \mathcal{C})$  (we can always assume that  $\mathcal{C} \subseteq \mathcal{H}$ ). Finally, we obtain that with probability at least  $1 - \delta$ ,

$$\Delta(A, h) \leq \Delta(S, h) + \frac{\epsilon}{4} \leq \Delta(S, \mathcal{C}) + \frac{3\epsilon}{4} \leq \Delta(A, \mathcal{C}) + \epsilon.$$

It is easy to verify that the running time and hypothesis space of this algorithm are as claimed. ■

Note that if  $\text{Alg}$  is efficient then  $d(\sigma, \epsilon/2)$  is polynomial in  $\sigma$  and  $1/\epsilon$  and, in particular, the obtained algorithm is efficient. In addition, in place of VC-dim one can use the uniform convergence result based on the cardinality of the hypothesis space. The description length of a hypothesis output by  $\text{Alg}$  is polynomial in  $\sigma$  and  $1/\epsilon$  and hence in this case a polynomial number of samples will be required to simulate  $\text{Alg}$ .

**Remark 7** We note that while this proof is given for the strongest version of agnostic learning in which the error of an agnostic algorithm is bounded by  $\Delta(A, \mathcal{C}) + \epsilon$ , it can be easily extended to weaker forms of agnostic learning, such as algorithms that only guarantee error bounded by  $\alpha \cdot \Delta(A, \mathcal{C}) + \beta + \epsilon$  for some  $\alpha \geq 1$  and  $\beta \geq 0$ . This is true since the reduction adds at most  $\epsilon/2$  to the error of the original algorithm (and the additional time required is polynomial in  $1/\epsilon$ ).

## 4 Learning with Respect to the Uniform Distribution

In this section we show that when learning with respect to the uniform distribution over  $\{0, 1\}^n$ , membership queries are helpful. Specifically, we show that if one-way functions exist, then there exists a concept class  $\mathcal{C}$  that is not agnostically PAC learnable (even weakly) with respect to the uniform distribution but is agnostically learnable over the uniform distribution given membership queries. Our agnostic learning algorithm is successful only when  $\epsilon \geq 1/p(n)$  for a polynomial  $p$  fixed in advance (the definition of  $\mathcal{C}$  depends on  $p$ ). While this is slightly weaker than required by the definition of the model it still exhibits the gap between agnostic learning with and without membership queries. We remark that a number of known PAC and agnostic learning algorithms are efficient only for restricted values of  $\epsilon$  (cf. [KKMS05, OS06, GKK08]).

### 4.1 Background

We first show why some of the known separation results will not work in the agnostic setting. It is well-known that the PAC model with membership queries is strictly stronger than the PAC model without membership queries (under the same cryptographic assumption). The separation result is obtained by using a concept class  $\mathcal{C}$  that is not PAC learnable and augmenting each concept  $f \in \mathcal{C}$  with the encoding of  $f$  in a fixed part of the domain. This encoding is readable using membership queries and therefore an MQ algorithm can “learn” the augmented  $\mathcal{C}$  by querying the points that contain the encoding. On the other hand, with overwhelming probability this encoding will not be observed in random examples and therefore does not help learning from random examples. This simple approach would fail in the agnostic setting. The unknown function might be random on the part of the domain that contains the encoding and equal to a concept from  $\mathcal{C}$  elsewhere. The agreement of the unknown function with a concept from  $\mathcal{C}$  is almost 1 but membership queries on the points of encoding will not yield any useful information.

A similar problem arises with encoding schemes used in the separation results of Elbaz *et al.* [ELSW07] and Feldman, Shah and Wadhwa [FSW07]. There too the secret encoding can be rendered unusable by a function that agrees with a concept in  $\mathcal{C}$  on a significant fraction of the domain.

### 4.2 Outline

We start by presenting some of the intuition behind our construction. As in most other separation results our goal is to create a concept class that is not learnable from uniform examples but includes an encoding of the unknown function that is readable using membership queries. We first note that

in order for this approach to work in the agnostic setting the secret encoding has to be “spread” over at least  $1 - 2\epsilon$  fraction of  $\{0, 1\}^n$ . To see this let  $f$  be a concept and let  $S \subseteq \{0, 1\}^n$  be the subset of the domain where the encoding of  $f$  is contained. Assume, for simplicity, that without the encoding the learning algorithm cannot predict  $f$  on  $\bar{S} = \{0, 1\}^n \setminus S$  with any significant advantage over random guessing. Let  $f'$  be a function equal to  $f$  on  $S$  and truly random on  $\bar{S}$ . Then

$$\Pr[f = f'] = (|\bar{S}| + |S|/2)/2^n = 1/2 + \frac{|\bar{S}|}{2^{n+1}}.$$

On the other hand,  $f'$  does not contain any information about the encoding of  $f$  and therefore, by our assumption, no efficient algorithm can produce a hypothesis with advantage significantly higher than  $1/2$  on both  $S$  and  $\bar{S}$ . This means that the error of any efficient algorithm will be higher by at least  $|\bar{S}|/2^{n+1}$  than the best possible. To ensure that this difference is at most  $\epsilon$ , we need  $|S| \geq (1 - 2\epsilon)2^n$ .

Another requirement that the construction has to satisfy is that the encoding of the secret has to be resilient to almost any amount of noise. In particular, since the encoding is a part of the function, we also need to be able to reconstruct an encoding that is close to the best possible. An encoding with this property is in essence a list-decodable binary code. In order to achieve the strongest separation result we will use the code of Guruswami and Sudan that is the concatenation of Reed-Solomon code with the binary Hadamard code [GS00]. However, to simplify the presentation, we will use the more familiar binary Hadamard code in our construction. In Section 4.6 we provide the details on the use of the Guruswami-Sudan code in place of the Hadamard code.

The Hadamard code is equivalent to encoding a vector  $a \in \{0, 1\}^n$  as the values of the parity function  $\chi_a$  on all points in  $\{0, 1\}^n$ . That is,  $n$  bit vector  $a$  is encoded into  $2^n$  bits given by  $\chi_a(x)$  for every  $x \in \{0, 1\}^n$ . This might appear quite inefficient since a learning algorithm will not be able to read all the bits of the encoding. However the Goldreich-Levin algorithm provides an efficient way to recover the indices of all the parities that agree with a given function with probability significantly higher than  $1/2$  [GL89]. Therefore the Hadamard code can be decoded by reading the code in only a polynomial number of (randomly-chosen) locations.

The next problem that arises is that the encoding should not be readable from random examples. As we have observed earlier, we cannot simply “hide” it on a negligible fraction of the domain. Specifically, we need to make sure that our Hadamard encoding is not recoverable from random examples. While it is not known how to learn parities with noise from random examples alone and this problem is conjectured to be very hard, for all we know, it is possible that one-way functions exist whereas learning of parities with noise is tractable. It is known however that if learning of parities with noise is hard then one-way functions exist [BFKL93]. Our solution to this problem is to use a pseudo-random function to make values on random examples indistinguishable from random coin flips. Specifically, let  $a \in \{0, 1\}^n$  be the vector we want to encode and let  $b : \{0, 1\}^n \rightarrow \{-1, 1\}$  be a pseudo-random function. We define a function  $g : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{-1, 1\}$  as

$$g(z, x) = b(z) \oplus \chi_a(x).$$

( $\oplus$  is simply the product in  $\{-1, 1\}$ ). The label of a random example  $(z, x) \in \{0, 1\}^{2n}$  is a XOR of a pseudorandom bit with an independent bit and therefore is pseudorandom. Values of a pseudorandom function  $b$  on any polynomial set of distinct points are pseudorandom and therefore random examples will have pseudorandom labels as long as their  $z$  parts are distinct. In a sample of polynomial in  $n$  size of random and uniform points from  $\{0, 1\}^{2n}$  this happens with overwhelming probability and therefore  $g(z, x)$  is not learnable from random examples. On the other hand, for a fixed  $z$ ,  $b(z) \oplus \chi_a(x)$  gives a Hadamard encoding of  $a$  or its negation. Hence it is possible to find  $a$  using membership queries with the same prefix. A construction based on a similar idea was used by Elbaz *et al.* in their separation result [ELSW07].

Finally, the problem with the construction we have so far is that while a membership query learning algorithm can find the secret, it cannot predict the encoding of the secret  $g(z, x)$  without knowing  $b(z)$ . This means that we also need to provide a description of  $b(z)$  to the learning algorithm. It is tempting to use the Hadamard code to encode the description of  $b(z)$  together with  $a$ . However, a bit of the encoding of  $b$  is no longer independent of  $b(z)$ , and therefore the previous argument does not hold. We refer to the vector that describes  $b(z)$  by  $d(b)$ . We are unaware of any constructions of pseudorandom functions that would remain pseudorandom when the value of the function is “mixed” with the description of the function. An identical problem also arises in the construction of Elbaz *et al.* [ELSW07]. They used another pseudorandom function  $b_1$  to encode  $d(b)$ , then used another pseudorandom function  $b_2$  to encode  $d(b_1)$  and so on. The fraction of the domain used up for the encoding of  $d(b_i)$  is becoming progressively smaller as  $i$  grows. In their construction a PAC learning algorithm can recover as many of the encodings as is required to reach accuracy  $\epsilon$ . This method would not be effective in our case. First, in the agnostic setting all the encodings but the one using the largest fraction of the domain can be corrupted. This makes the largest encoding unrecoverable and implies that the best  $\epsilon$  achievable is at most half of the fraction of the domain used by the largest encoding. In addition, in the agnostic setting the encoding of  $d(b_i)$  for every odd  $i$  can be completely corrupted making all the other encodings unrecoverable. To solve this problem in our construction we use a pseudorandom function  $b_i$  to encode  $d(b_j)$  for all  $j < i$ . We also use encodings of the same size. In this construction at most one of the encodings that are not completely corrupted cannot be recovered. It is the encoding with  $b_i(z)$  such that the encodings with  $b_j(z)$  are completely corrupted for all  $j > i$  (since those are the ones that contain the encoding of  $d(b_i)$ ). Therefore by making the number of encodings larger than  $1/\epsilon$ , we can make sure that there exists an efficient algorithm that finds a hypothesis with the error within  $\epsilon$  of the optimum.

### 4.3 The Construction

We will now describe the construction formally and give a brief proof of its correctness. Let  $p = p(n)$  be a polynomial, let  $\ell = \log p(n)$  (we assume for simplicity that  $p(n)$  is a power of 2) and let  $m = \ell + n \cdot p$ . We refer to an element of  $\{0, 1\}^m$  by triple  $(k, z, \bar{x})$  where  $k \in [p]$ ,  $z \in \{0, 1\}^n$ , and

$$\bar{x} = (x^1, x^2, \dots, x^{p-1}) \in \{0, 1\}^{n \times (p-1)}.$$

Here  $k$  indexes the encodings,  $z$  is the input to the  $k$ -th pseudorandom function and  $\bar{x}$  is the input to a parity function on  $n(p-1)$  variables that encodes the secret keys for all pseudorandom functions used for encodings 1 through  $k-1$ . Formally, let

$$\bar{d} = (d^1, d^2, \dots, d^{p-1})$$

be a vector in  $\{0, 1\}^{n \times (p-1)}$  (where each  $d^i \in \{0, 1\}^n$ ) and for  $k \in [p]$  let

$$\bar{d}(k) = (d^1, d^2, \dots, d^{k-1}, 0^n, \dots, 0^n).$$

Let  $\mathcal{F} = \{\pi_y\}_{y \in \{0, 1\}^*}$  be a pseudorandom function family (Definition 5). We define  $g_{\bar{d}} : \{0, 1\}^m \rightarrow \{-1, 1\}$  as follows:

$$g_{\bar{d}}(k, z, \bar{x}) = \pi_{d^k}(z) \oplus \chi_{\bar{d}(k)}(\bar{x}) \quad (4)$$

Denote

$$\mathcal{C}_n^p = \left\{ g_{\bar{d}} \mid \bar{d} \in \{0, 1\}^{n \times (p-1)} \right\}.$$

### 4.4 Hardness of Learning $\mathcal{C}_n^p$ From Random Examples

We start by showing that  $\mathcal{C}_n^p$  is not agnostically learnable from random and uniform examples only. In fact, we will show that it is not even weakly PAC learnable. Our proof is analogous to the proof by Elbaz *et al.* who show that the same holds for the concept class they define [ELSW07].

**Theorem 8** *There exists no efficient algorithm that weakly PAC learns  $\mathcal{C}_n^p$  with respect to the uniform distribution over  $\{0, 1\}^m$ .*

**Proof:** In order to prove the claim we show that a weak PAC learning algorithm for  $\mathcal{C}_n^p$  can be used to distinguish a pseudorandom function family from a truly random function. A weak learning algorithm for  $\mathcal{C}_n^p$  implies that every function in  $\mathcal{C}_n^p$  can be distinguished from a truly random function on  $\{0, 1\}^m$ . If, on the other hand, in the computation of  $g_{\bar{d}}(k, z, \bar{x})$  we used a truly random function in place of each  $\pi_{d^k}(z)$  then the resulting labels would be truly random and, in particular, unpredictable.

Formally, let  $\text{Alg}$  be a weak learning algorithm for  $\mathcal{C}_n^p$  that, with probability at least  $1/2$ , produces a hypothesis with error of at most  $1/2 - 1/q(m)$  and runs in time polynomial in  $t(m)$  for some polynomials  $t$  and  $q$ . Our concept class  $\mathcal{C}_n^p$  uses numerous pseudorandom functions from  $F_n$  and therefore we use a so-called “hybrid” argument to show that one can replace a single  $\pi_{d^k}(z)$  with a truly random function to cause  $\text{Alg}$  to fail.

For  $0 \leq i \leq p$ , let  $\mathbb{O}(i)$  denote an oracle randomly chosen according to the following procedure. First choose randomly and uniformly  $\pi_{d^1}, \pi_{d^2}, \dots, \pi_{d^i} \in F_n$  and then choose randomly and uniformly  $\rho_{i+1}, \rho_{i+2}, \dots, \rho_k$  from the set of all Boolean functions over  $\{0, 1\}^n$ . Upon request such an oracle returns an example  $((k, z, \bar{x}), b)$  where  $(k, z, \bar{x})$  is chosen randomly and uniformly from  $\{0, 1\}^m$  and

$$b = \begin{cases} \pi_{d^k}(z) \oplus \chi_{\bar{d}(k)}(\bar{x}) & k \leq i \\ \rho_k(z) & k > i \end{cases}$$

We note that in order to simulate such an oracle it is not needed to explicitly choose  $\rho_{i+1}, \rho_{i+2}, \dots, \rho_k$ . Instead their values can be generated upon request by flipping a fair coin.

This means that for every  $i$ ,  $\mathbb{O}(i)$  can be chosen and then simulated in time polynomial in  $m$  and the number of examples requested. We denote by  $\delta_i$  the probability of the following event:  $\text{Alg}$  with oracle  $\mathbb{O}(i)$  outputs a hypothesis that has error of at most  $1/2 - 2/(3q(m))$  relative to  $\mathbb{O}(i)$ . We refer to this condition as success. The error is obtained by estimating it on new random examples from  $\mathbb{O}(i)$  to within  $1/(3q(m))$  and with probability at least  $7/8$ . The probability is taken over the random choice and simulation of  $\mathbb{O}(i)$  and the coin flips of  $\text{Alg}$ . The bounds on the running time of  $\text{Alg}$  and Chernoff bounds imply that this test can be performed in time polynomial in  $m$ .

**Claim 9**  $\delta_p - \delta_0 \geq 1/4$ .

**Proof:** To see this we first observe that  $\mathbb{O}(0)$  is a truly random oracle and therefore the error of the hypothesis produced by  $\text{Alg}$  is at least  $1/2 - \nu(m)$  for some negligible  $\nu$ . This means that the error estimate can be lower than  $1/2 - 2/(3q(m))$  only if the estimation fails. By the definition of our error estimation procedure this implies that  $\delta_0 \leq 1/8$ . On the other hand,  $\mathbb{O}(p)$  is equivalent to  $\text{EX}(U, g_{\bar{d}})$  for a randomly chosen  $\bar{d}$ . This implies that with probability at least  $1/2$ ,  $\text{Alg}$  outputs a hypothesis with error of at most  $1/2 - 1/q(m)$ . With probability at least  $7/8$  the estimate of the error is correct and therefore  $\delta_p \geq 3/8$ .  $\blacksquare$ (C1.9)

We now describe our distinguisher  $M$ . Let  $\pi(x)$  denote the function given to  $M$  as an oracle. Our distinguisher chooses a random  $i \in [p]$  and a random oracle  $\mathbb{O}(i)$  as described above but using the oracle  $\pi$  in place of  $\pi_{d^i}$ . That is it generates examples  $((k, z, \bar{x}), b)$  where  $(k, z, \bar{x})$  is chosen randomly and uniformly from  $\{0, 1\}^m$  and

$$b = \begin{cases} \pi_{d^k}(z) \oplus \chi_{\bar{d}(k)}(\bar{x}) & k < i \\ \pi(z) \oplus \chi_{\bar{d}(k)}(\bar{x}) & k = i \\ \rho_k(z) & k > i \end{cases}$$

Denote this oracle by  $\mathbb{O}^\pi(i)$ . The distinguisher simulates  $\text{Alg}$  with examples from  $\mathbb{O}^\pi(i)$  and outputs 1 whenever the test of the output of  $\text{Alg}$  is successful.

We first observe that if  $\pi$  is chosen randomly from  $F_n$  then choosing and simulating a random  $\mathbb{O}^\pi(i)$  is equivalent to choosing and simulating a random  $\mathbb{O}(i)$ . Therefore  $M$  will output 1 with probability

$$\frac{1}{p(n)} \sum_{i \in [p]} \delta_i.$$

On the other hand, if  $\pi$  is a truly random function then  $\mathbb{O}^\pi(i)$  is equivalent to  $\mathbb{O}(i-1)$  and hence the simulator will output 1 with probability

$$\frac{1}{p(n)} \sum_{i \in [p]} \delta_{i-1}.$$

Therefore, by Claim 9 this implies that  $M$  distinguishes  $F_n$  from a truly random function with probability at least

$$\frac{1}{p(n)} \left( \sum_{i \in [p]} \delta_i - \delta_{i-1} \right) \geq \frac{1}{p(n)} (\delta_p - \delta_0) \geq 1/4p(n).$$

The efficiency of  $M$  follows readily from the efficiency of the test we demonstrated above and gives us the contradiction to the properties of  $\mathcal{F}$ .  $\blacksquare$ (Th.8)

## 4.5 Agnostic Learning of $C_n^p$ with Membership Queries

We now describe a (fully) agnostic learning algorithm for  $C_n^p$  that uses membership queries and is successful for any  $\epsilon \geq 1/p(n)$ .

**Theorem 10** *There exists a randomized algorithm  $\text{AgnLearn}$  that for every distribution  $A = (U, \phi)$  over  $\{0, 1\}^m$  and every  $\epsilon \geq 1/p(n)$ ,  $\delta > 0$ , given access to  $\text{MEM}(A)$ , with probability at least  $1 - \delta$ , finds  $h$  such that  $\Delta(A, h) \leq \Delta(A, C_n^p) + \epsilon$ . The probability is taken over the coin flips of  $\text{MEM}(A)$  and  $\text{AgnLearn}$ .  $\text{AgnLearn}$  runs in time polynomial in  $m$  and  $\log(1/\delta)$ .*

**Proof:** Let  $g_{\bar{e}}$  for  $\bar{e} = (e^1, e^2, \dots, e^{p-1}) \in \{0, 1\}^{(p-1) \times n}$  be the function for which  $\Delta(A, g_{\bar{e}}) = \Delta(A, C_n^p)$ . The goal of our algorithm is to find the largest  $j$  such that on random examples from the  $j$ -th encoding  $A$  agrees with the encoding of  $\bar{e}(j) = (e^1, e^2, \dots, e^{j-1}, 0^n, \dots, 0^n)$  with probability at least  $1/2 + \epsilon/4$ . Such  $j$  can be used to find  $\bar{e}(j)$  and therefore allows us to reconstruct  $g_{\bar{e}}$  on all points  $(k, z, \bar{x})$  for  $k < j$ . For points with  $k \geq j$  our hypothesis is either constant 1 or constant -1, whichever has the higher agreement with  $A$ . This guarantees that the error on this part is at most  $1/2$ . By the definition of  $j$ ,  $g_{\bar{e}}$  has error of at least

$$1/2 - \epsilon/4 - 1/(2p) \geq 1/2 - \epsilon$$

on this part of the domain and therefore the hypothesis has error close to that of  $g_{\bar{e}}$ .

We now describe  $\text{AgnLearn}$  formally. For every  $i \in [p]$ ,  $\text{AgnLearn}$  chooses  $y \in \{0, 1\}^n$  randomly and uniformly. Then  $\text{AgnLearn}$  runs Goldreich-Levin algorithm over  $\{0, 1\}^{(p-1) \times n}$  using  $\text{MEM}(A_{i,y})$ . When queried on a point  $\bar{x} \in \{0, 1\}^{(p-1) \times n}$   $\text{MEM}(A_{i,y})$  returns the value of  $\text{MEM}(A)$  on query  $(i, y, \bar{x})$ . That is  $\text{MEM}(A_{i,y})$  is a restriction of  $A$  to points in  $\{0, 1\}^m$  with prefix  $i, y$ . Let  $T$  denote the set of indices of heavy Fourier coefficients returned by  $\text{GL}(\epsilon/4, 1/2)$ . For each vector  $\bar{d} \in T$  and  $b \in \{-1, 1\}$ , let  $h_{\bar{d}, i, b}$  be defined as

$$h_{\bar{d}, i, b}(k, z, \bar{x}) = \begin{cases} \pi_{d^k}(z) \oplus \chi_{\bar{d}(k)}(\bar{x}) & k < i \\ b & k \geq i \end{cases}$$

(Here  $\pi_{d^k}$  is an element of the pseudorandom function family  $\mathcal{F}$  used in the construction.) Next  $\text{AgnLearn}$  approximates  $\Delta(A, h_{\bar{d}, i, b})$  to within accuracy  $\epsilon/8$  with confidence  $1 - \delta/t$  using random samples from  $A$  (for  $t$  to be defined later). We denote the estimate obtained by  $\tilde{\Delta}_{\bar{d}, i, b}$ .  $\text{AgnLearn}$  repeats this  $r$  times (generating new  $y$  each time) and returns  $h_{\bar{d}, i, b}$  for which  $\tilde{\Delta}_{\bar{d}, i, b}$  is the smallest. For  $i = 1$  and any  $\bar{d}$ ,  $h_{\bar{d}, 1, b} \equiv b$ . Therefore for  $i = 1$  instead of the above procedure  $\text{AgnLearn}$  tests two constant hypotheses  $h_1 \equiv 1$  and  $h_{-1} \equiv -1$ .

**Claim 11** *For  $t = O(p \cdot \log(1/\delta)/\epsilon^3)$  and  $r = O(\log(1/\delta)/\epsilon)$ , with probability at least  $1 - \delta$ ,  $\text{AgnLearn}$  returns  $h$  such that  $\Delta(A, h) \leq \Delta(A, C_n^p) + \epsilon$ .*

**Proof:** We show that among the hypotheses considered by  $\text{AgnLearn}$  there will be a hypothesis  $h'$  such that  $\Delta(A, h') \leq \Delta(A, g_{\bar{e}}) + 3\epsilon/4$  (with sufficiently high probability). The estimates of the error of each hypothesis are within  $\epsilon/8$  of the

true error and therefore the hypothesis  $h$  with the smallest estimated error will satisfy

$$\Delta(A, h) \leq \Delta(A, h') + \epsilon/4 \leq \Delta(A, g_{\bar{e}}) + \epsilon.$$

For  $i \in [p]$ , denote

$$\Delta_i = \Pr_{((k,z,\bar{x}),b) \sim A} [b \neq g_{\bar{e}}(k, z, \bar{x}) \mid k = i].$$

By the definition,

$$\frac{1}{p} \sum_{i \in [p]} \Delta_i = \Delta(A, g_{\bar{e}}).$$

Let  $j$  be the largest  $i$  such that  $\Delta_{i'} \leq 1/2 - \epsilon/4$  and for all  $i' > i$ ,  $\Delta_{i'} > 1/2 - \epsilon/4$ . If such  $j$  does not exist then  $\Delta(A, g_{\bar{e}}) > 1/2 - \epsilon/4$ . Either  $h_1$  or  $h_{-1}$  has error of at most  $1/2$  on  $A$  and therefore for  $i = 1$  `AgnLearn` will find a hypothesis  $h'$  such that  $\Delta(A, h') \leq \Delta(A, g_{\bar{e}}) + 3\epsilon/4$ .

We can now assume that  $j$  as above exists. Denote

$$\Delta_{i,y} = \Pr_{((k,z,\bar{x}),b) \sim A} [b \neq g_{\bar{e}}(k, z, \bar{x}) \mid k = i, z = y].$$

By the definition,

$$\mathbf{E}_{y \in \{0,1\}^n} \Delta_{i,y} = \Delta_i.$$

This implies that for a randomly and uniformly chosen  $y$ , with probability at least  $\epsilon/4$ ,  $\Delta_{j,y} \leq 1/2 - \epsilon/8$ . This is true since otherwise

$$\Delta_j \geq (1 - \frac{\epsilon}{4})(\frac{1}{2} - \frac{\epsilon}{8}) > \frac{1}{2} - \frac{\epsilon}{4},$$

contradicting the choice of  $j$ . We now note that by the definition of  $A_{i,y}$ ,

$$\Delta_{i,y} = \Pr_{(\bar{x},b) \sim A_{i,y}} [b \neq g_{\bar{e}}(i, y, \bar{x})].$$

The function  $g_{\bar{e}}(i, y, \bar{x})$  equals  $\pi_{dj}(y) \oplus \chi_{\bar{e}(j)}(\bar{x})$ , and therefore if  $\Delta_{i,y} \leq 1/2 - \epsilon/8$  then by equation (1),

$$|\widehat{A_{i,y}}(\bar{e}(j))| \geq \epsilon/4.$$

This implies that `GL`( $\epsilon/4, 1/2$ ) with `MEM`( $A_{i,y}$ ) will return  $\bar{e}(j)$  (possibly, among other vectors). Let

$$b_j = \text{sign}(\mathbf{E}_{((k,z,\bar{x}),b) \sim A} [b \mid k \geq j])$$

be the constant with the lowest error on examples from  $A$  for which  $k \geq j$ . Clearly, this error is at most  $1/2$ . The hypothesis  $h_{\bar{e}(j),j,b_j}$  equals  $g_{\bar{e}}$  on points for which  $k < j$  and equals  $b_j$  on the rest of the points. Therefore

$$\Delta(A, h_{\bar{e}(j),j,b_j}) \leq \frac{1}{p} \left( \sum_{i < j} \Delta_i + \frac{p-j+1}{2} \right).$$

On the other hand, by the properties of  $j$ , for all  $i > j$ ,  $\Delta_i \geq 1/2 - \epsilon/4$  and thus

$$\begin{aligned} \Delta(A, g_{\bar{e}}) &= \frac{1}{p} \left( \sum_{i \in [p]} \Delta_i \right) \\ &\geq \frac{1}{p} \left( \sum_{i < j} \Delta_i + (p-j) \left( \frac{1}{2} - \frac{\epsilon}{4} \right) \right). \end{aligned}$$

By combining these equations we obtain that

$$\Delta(A, h_{\bar{e}(j),j,b_j}) - \Delta(A, g_{\bar{e}}) \leq \frac{1}{2p} + \frac{\epsilon}{4} \leq \frac{3\epsilon}{4}.$$

All that is left to show now are the choices of  $r$  and  $t$  for which the desired  $h$  will be found with probability at least  $1 - \delta$ . As we have observed, for a randomly and uniformly chosen  $y$ , with probability at least  $\epsilon/4$ ,  $\Delta_{j,y} \leq 1/2 - \epsilon/8$  and in this case `GL`( $\epsilon/4, 1/2$ ) will find  $\bar{e}(j)$  with probability at least  $1/2$ . By repeating this procedure  $O(\log(1/\delta)/\epsilon)$  times we can ensure that  $\bar{e}(j)$  is found with probability at least  $1 - \delta/2$ . By Parseval's identity there are  $O(1/\epsilon^2)$  elements in each set of vectors returned by `GL`. Hence the number of error estimations performed by `AgnLearn` is  $O(p \cdot r/\epsilon^2)$ . This means that for  $t = O(p \cdot \log(1/\delta)/\epsilon^3)$  all estimations will be within  $\epsilon/8$  with probability  $1 - \delta/2$ .  $\blacksquare$ (Cl.11)

Given Claim 11, we only need to check that the running time of `AgnLearn` is polynomial in  $m$  and  $\log(1/\delta)$ . This follows easily from the polynomial bound on the running time of `GL` and computation of each  $\pi \in F_n$ , and polynomial number of samples required to estimate the errors of the candidate hypotheses.  $\blacksquare$ (Th.10)

#### 4.6 Bounds on $\epsilon$

Theorem 10 shows that  $\mathcal{C}_n^p$  is defined over  $\{0,1\}^m$  for  $m = n \cdot p(n) + \log p(n)$  and is learnable agnostically for any  $\epsilon \geq 1/p(n)$ . This means that this construction cannot achieve dependence on  $\epsilon$  beyond  $1/m$ . To improve this dependence we can use a more efficient encoding scheme in place of Hadamard code. Let  $C : \{0,1\}^k \rightarrow \{0,1\}^v$  be a binary code of message length  $k$  and block length  $v$ . The following properties of the code are required by our construction:

- Efficient encoding algorithm. For any  $z \in \{0,1\}^k$  and  $j \leq v$ ,  $C(z)_j$  (the  $j^{\text{th}}$  bit of  $C(z)$ ) is computable in time polynomial in  $k$  and  $\log v$ .
- Efficient local list decoding from  $(1/2 - \gamma)v$  errors in time polynomial in  $k$  and  $1/\gamma$  for any  $\gamma \geq \epsilon/8$ . That is, an algorithm that given oracle access to the bits of string  $y \in \{0,1\}^v$  produces the list of all messages  $z$  such that  $\Pr_{j \in [v]} [C(z)_j \neq y_j] \leq 1/2 - \gamma$  (in time polynomial in  $k$  and  $1/\gamma$ ).

Guruswami and Sudan gave a list decoding algorithm for Reed-Solomon code concatenated with Hadamard code that has the desired properties for  $v = O(k^2/\epsilon^4)$  [GS00] (see also [Tre05, Lecture 14] for a simplified presentation). Note that this is exponentially more efficient than Hadamard code for which  $v = 2^k$ . In fact for this code we can afford to read the whole codeword in polynomial time. This means that we can assume that the output of the list-decoding algorithm is exact (and not approximate as in the case of list decoding using Goldreich-Levin algorithm).

In our construction  $k = n(p(n) - 1)$ . To apply the above code we index a position in the code using  $\log v = O(\log(n/\epsilon))$  bits. Further we can use pseudorandom functions over  $\{0,1\}^{n/2}$  instead of  $\{0,1\}^n$  in the definition of  $\mathcal{C}_n^p$ . We would then obtain that the dimension of  $\mathcal{C}_n^p$  is  $m = n/2 + \log v + \log p(n) \leq n$  for any polynomial  $p(n)$  and

$\epsilon \geq 1/p(n)$ . This implies that our learning algorithm is successful for every  $\epsilon \geq 1/p(n) \geq 1/p(m)$ . It is easy to verify that Theorems 8 and 10 still hold for this variant of the construction.

## 5 Discussion

Our results clarify the role of membership queries in agnostic learning. They imply that in order to extract any meaningful information from membership queries the learner needs to have significant prior knowledge about the distribution of examples. Specifically, either the set of possible classification functions has to be restricted (as in the PAC model) or the set of possible marginal distributions (as in distribution-specific agnostic learning).

A interesting result in this direction would be a demonstration that membership queries are useful for distribution-specific agnostic learning of a natural concept class such as halfspaces.

## Acknowledgments

We thank Parikshit Gopalan, Salil Vadhan and David Woodruff for valuable discussions and comments on this research.

## References

- [Ang88] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.
- [BFKL93] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Proceedings of International Cryptology Conference on Advances in Cryptology (CRYPTO)*, pages 278–291, 1993.
- [BJT99] N. Bshouty, J. Jackson, and C. Tamon. More efficient PAC learning of DNF with membership queries under the uniform distribution. In *Proceedings of COLT*, pages 286–295, 1999.
- [Bsh95] N. Bshouty. Exact learning via the monotone theory. *Information and Computation*, 123(1):146–153, 1995.
- [Dud78] R. Dudley. Central limit theorems for empirical measures. *Annals of Probability*, 6(6):899–929, 1978.
- [ELSW07] A. Elbaz, H. Lee, R. Servedio, and A. Wan. Separating models of learning from correlated and uncorrelated data. *Journal of Machine Learning Research*, 8:277–290, 2007.
- [Fel06] V. Feldman. Optimal hardness results for maximizing agreements with monomials. In *Proceedings of Conference on Computational Complexity (CCC)*, pages 226–236, 2006.
- [Fel07] V. Feldman. Attribute efficient and non-adaptive learning of parities and DNF expressions. *Journal of Machine Learning Research*, (8):1431–1460, 2007.
- [FGKP06] V. Feldman, P. Gopalan, S. Khot, and A. Ponnuswami. New results for learning noisy parities and halfspaces. In *Proceedings of FOCS*, pages 563–574, 2006.
- [FSW07] V. Feldman, S. Shah, and N. Wadhwa. Separating models of learning with faulty teachers. In *Proceedings of ALT*, pages 94–106, 2007.
- [GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.
- [GKK08] P. Gopalan, A. Kalai, and A. Klivans. Agnostically learning decision trees. To appear in *Proceedings of STOC*, 2008.
- [GKS01] S. A. Goldman, S. Kwek, and S. D. Scott. Agnostic learning of geometric patterns. *Journal of Computer and System Sciences*, 62(1):123–151, 2001.
- [GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of STOC*, pages 25–32, 1989.
- [GR06] V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *Proceedings of FOCS*, pages 543–552, 2006.
- [GS00] G. Guruswami and M. Sudan. List decoding algorithms for certain concatenated codes. In *Proceedings of STOC*, pages 181–190, 2000.
- [Hås01] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.
- [Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.
- [HILL99] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
- [Kha95] M. Kharitonov. Cryptographic lower bounds for learnability of boolean functions on the uniform distribution. *Journal of Computer and System Sciences*, 50:600–610, 1995.
- [KKMS05] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of FOCS*, pages 11–20, 2005.
- [KM93] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.
- [KMV08] A. Kalai, Y. Mansour, and E. Verbin. Agnostic boosting and parity learning. To appear in *Proceedings of STOC*, 2008.
- [KSS94] M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.
- [KV94] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.
- [LBW95] W. S. Lee, P. L. Bartlett, and R. C. Williamson. On efficient agnostic learning of linear combinations of basis functions. In *Proceedings of COLT*, pages 369–376, 1995.
- [Lev93] L. Levin. Randomness and non-determinism. *Journal of Symbolic Logic*, 58(3):1102–1103, 1993.

- [LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.
- [OS06] R. O’Donnell and R. Servedio. Learning monotone decision trees in polynomial time. In *Proceedings of IEEE Conference on Computational Complexity*, pages 213–225, 2006.
- [Pol84] D. Pollard. *Convergence of Stochastic Processes*. Springer-Verlag, 1984.
- [Tre05] L. Trevisan. Pseudorandomness and combinatorial constructions (lecture notes). Available at <http://www.cs.berkeley.edu/~luca/pacc/>, 2005.
- [Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
- [Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.
- [VC71] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probab. and its Applications*, 16(2):264–280, 1971.

---

# Dimension and Margin Bounds for Reflection-invariant Kernels \*

---

Thorsten Doliwa, Michael Kallweit, and Hans Ulrich Simon  
Fakultät für Mathematik, Ruhr-Universität Bochum, D-44780 Bochum, Germany  
{thorsten.doliwa,michael.kallweit,hans.simon}@rub.de

## Abstract

A kernel over the Boolean domain is said to be reflection-invariant, if its value does not change when we flip the same bit in both arguments. (Many popular kernels have this property.) We study the geometric margins that can be achieved when we represent a specific Boolean function  $f$  by a classifier that employs a reflection-invariant kernel. It turns out  $\|\hat{f}\|_\infty$  is an upper bound on the average margin. Furthermore,  $\|\hat{f}\|_\infty^{-1}$  is a lower bound on the smallest dimension of a feature space associated with a reflection-invariant kernel that allows for a correct representation of  $f$ . This is, to the best of our knowledge, the first paper that exhibits margin and dimension bounds for *specific functions* (as opposed to *function families*). Several generalizations are considered as well. The main mathematical results are presented in a setting with arbitrary finite domains and a quite general notion of invariance.

## 1 Introduction

There has been much interest in margin and dimension bounds during the last decade. The simplest way to cast (most of) the existing results in this direction is offered by the notion of margin and dimension complexity associated with a given sign matrix  $A \in \{-1, 1\}^{m \times n}$ . A linear arrangement, given by unit vectors  $u_1, \dots, u_m; v_1, \dots, v_n$  (taken from an inner product space), is said to represent  $A$  if, for all  $i = 1, \dots, m$  and  $j = 1, \dots, n$ ,  $A_{i,j} = \text{sign}(\langle u_i, v_j \rangle)$ . The dimension complexity of

$A$  is the smallest dimension of an inner product space that allows for such a representation. The margin complexity is obtained similarly by looking for the linear arrangement that leads to the maximum average margin (or, alternatively, to the maximum margin that can be guaranteed for all choices of  $i$  and  $j$ ). Applying counting arguments, Ben-David, Eiron, and Simon [1] have shown that, loosely speaking, an overwhelming majority of sign matrices of small VC-dimension do not allow for a linear arrangement whose margin or dimension is significantly better than what can be guaranteed in a trivial fashion. Starting with Forster's celebrated exponential lower bound on the dimension complexity of the Walsh-Hadamard matrix [4], there has been a series of papers [5, 6, 10, 7, 13, 15] presenting (increasingly powerful) techniques for deriving upper margin bounds or lower dimension bounds on the complexity of sign matrices.

Note that a sign matrix represents a *family* of Boolean functions, one Boolean function per column say. The lack of non-trivial margin or dimension bounds for a *specific* Boolean function has a simple explanation: a specific function  $f(x)$  can always trivially be represented in a 1-dimensional space with geometric margin 1 by mapping an instance  $x \in \{-1, 1\}^n$  to  $f(x) \in \{-1, 1\}$ . The corresponding kernel would map a pair  $(x, x')$  of instances to 1 if  $f(x) = f(x')$ , and to  $-1$  otherwise. Clearly, the 1-dimensional "linear arrangement" for  $f$  does not say much about the ability of kernel-based large margin classifier systems to "learn"  $f$  because we would need to know  $f$  perfectly prior to the choice of the kernel. (If we had this knowledge, there would be nothing to learn anymore.) Nevertheless, this discussion shows that one cannot expect non-trivial margin or dimension bounds for *specific functions* that hold *uniformly for all kernels*.

In this paper, we introduce the concept of distributed functions that are invariant under a group  $\mathcal{G}$  of transformations. We present the mathematical results about invariant distributed functions in a quite general setting (because it does not make

---

\*This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views. This work was furthermore supported by the Deutsche Forschungsgemeinschaft Grant SI 498/8-1.

sense to impose unnecessary restrictions). In particular, we derive non-trivial margin and dimension bounds for specific Boolean functions that are valid for all linear arrangements resulting from  $\mathcal{G}$ -invariant kernels. If the domain of the distributed function can be cast as a finite Abelian group, the margin and dimension bounds for a function  $f$  can be nicely expressed in terms of  $f$ 's Fourier-spectrum. As always,  $\|\hat{f}\|_\infty$  denotes the largest absolute value found in the spectrum of  $f$ 's Fourier-coefficients. We show that  $\|\hat{f}\|_\infty$  is an upper bound on the largest possible average margin, and  $\|\hat{f}\|_\infty^{-1}$  is a lower bound on the smallest possible dimension. Our general results easily apply to a special case of high learning-theoretic relevance, namely the reflection-invariant kernels. Their relevance comes from the fact that, as demonstrated in the paper, many popular kernels actually happen to be reflection-invariant.

The remainder of the paper is structured as follows. In Section 2, we fix some notation and recall some facts about Fourier-expansions over finite Abelian groups and kernel-based classification. In Section 3, we present our results for arbitrary finite domains and a quite general notion of invariance. In Section 4, we introduce the concept of rotation-invariance and mention some connections between the Fourier-expansion over an arbitrary finite Abelian group and the spectral decomposition of such functions. In Section 5, we consider distributed functions over the Boolean domain and the concept of reflection-invariance, which is simply rotation-invariance over a Boolean domain. Section 6 presents the margin and dimension bounds that are valid for reflection-invariant kernels. Section 7 offers a possible interpretation of our results, and mentions a connection to a recent paper by Haasdonk and Burkhardt [8] along with some open problems.

## 2 Definitions and Notations

We assume familiarity with basics in matrix and learning theory. For example, notions like

- singular values, eigenvalues, spectral norm
- kernels, feature map, Reproducing Kernel Hilbert Space

are assumed as known (although we shall occasionally refresh the readers memory). Some central definitions and facts concerning

- linear arrangements representing a given sign matrix,
- margin and dimension associated with such a linear arrangement,

will be given later in the paper at the place where it is required. In the following we fix some notation

and recall the Fourier-expansion over finite Abelian groups as well as the notion of margin in kernel-based classification.

### 2.1 Preliminaries

Throughout the paper,  $\delta$  denotes the Kronecker-symbol, i.e.,  $\delta(a, b) = 1$  if  $a = b$  and  $\delta(a, b) = 0$  otherwise. For two  $n$ -dimensional vectors  $x, y$ , we define  $x \circ y$  to be the vector obtained by multiplying  $x$  and  $y$  componentwise, i.e.,  $(x \circ y)_i := x_i y_i$  for  $i = 1, \dots, n$ . The  $n$ -dimensional ‘‘all-ones vector’’ is given by

$$\vec{e} = (1, \dots, 1) .$$

The vector with 1 in component  $k$  and zeros elsewhere is denoted as  $\vec{e}_k$ . We consider functions over a finite domain  $D$  with values in  $\mathbb{R}$  (or in  $\mathbb{C}$ , resp.). These functions form a  $|D|$ -dimensional vector space. A *distributed function over  $D$*  is a function over the domain  $D \times D$ . We will occasionally identify a distributed function  $f$  over  $D$  with the  $(D \times D)$ -matrix  $F$  given by  $F_{x,y} = f(x, y)$ .

### 2.2 Fourier-expansions over Finite Abelian Groups

Let  $(D, +)$  be a finite Abelian group of size  $d = |D|$ . A function  $\chi : D \rightarrow \mathbb{C}$  is called a *character* over  $D$  if, for every  $x, y \in D$ ,

$$\chi(x + y) = \chi(x) \cdot \chi(y) .$$

It is well-known that there are exactly  $d$  characters, and they form an orthonormal basis of the vector space  $\mathbb{C}^D$  with respect to the inner product

$$\langle f, g \rangle := \frac{1}{d} \cdot \sum_{x \in D} f(x) \cdot \overline{g(x)} . \quad (1)$$

We may fix a bijection between  $D$  and the set of characters and write  $\chi_z$  for the character that corresponds to  $z \in D$ . Every function  $f : D \rightarrow \mathbb{C}$  can be written in the form

$$f(x) = \sum_{z \in D} \hat{f}(z) \cdot \chi_z(x) \quad (2)$$

where

$$\hat{f}(z) := \langle f, \chi_z \rangle = \frac{1}{d} \cdot \sum_{y \in D} f(y) \cdot \overline{\chi_z(y)} .$$

Equation (2) is referred to as the *Fourier expansion* of  $f$ , and  $\hat{f}(z)$  is called the *Fourier-coefficient* of  $f$  at  $z$ .

According to the ‘‘Fundamental Theorem for Finitely Generated Abelian Groups’’, every finite Abelian group is, up to isomorphism, of the form

$$D = \mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_n} \quad (3)$$

for some sequence  $q_1, \dots, q_n$  of prime powers. Equation (3) is assumed henceforth so that

$$d = |D| = \prod_{k=1}^n q_k .$$

It is well-known that the characters over  $\mathbb{Z}_m$  are given by

$$\chi_k^{(m)}(j) = \omega_m^{jk} ,$$

where

$$\omega_m = \exp\left(\frac{2\pi i}{m}\right)$$

is a primitive root of unity of order  $m$ . The characters over  $D$  are then given by

$$\chi_z(x) = \prod_{k=1}^n \chi_{z_k}^{(q_k)}(x_k) .$$

Consider now the matrix  $H = (H_{x,z})_{x,z \in D}$  given by

$$H_{x,z} = \chi_z(x) . \quad (4)$$

It is obvious that  $H$  is symmetric. By the orthonormality of the characters with respect to the inner product in (1), it follows that

$$H^* \cdot H = H \cdot H^* = d \cdot I ,$$

where  $I$  denotes the identity matrix.

### 2.3 Kernel-based Classification

Let  $K : D \times D \rightarrow \mathbb{R}$  be a valid kernel over a finite domain  $D$ . In other words,  $K(x, y)$  is a real-valued distributed function over  $D$  which, considered as matrix, is symmetric and positive semidefinite. Let  $\Phi_K$  be the feature map and  $\langle \cdot, \cdot \rangle_K$  the inner product that represent  $K$  in the Reproducing Kernel Hilbert Space, and let  $\| \cdot \|_K$  be the norm induced by  $\langle \cdot, \cdot \rangle_K$ .<sup>1</sup> Then  $\Phi$  satisfies

$$\forall x, y \in D : K(x, y) = \langle \Phi(x), \Phi(y) \rangle .$$

With every “dual vector”  $\alpha : D \rightarrow \mathbb{R}$ , we associate the “weight vector”

$$w(\alpha) := \sum_{x \in D} \alpha(x) \Phi(x) . \quad (5)$$

In the context of “large margin classification”,  $\alpha$  is considered as a classifier that assigns the label  $\text{sign}(\langle w(\alpha), \Phi(x) \rangle)$  to input  $x$ . Consider a target function  $f : D \rightarrow \{-1, 1\}$  for a binary classification task. Then, a negative sign of  $f(x) \cdot \langle w(\alpha), \Phi(x) \rangle$  indicates a “classification error” on  $x$ . So this expression should be *positive* and it is intuitively even better when it leads to a *large* positive value. Thus, the following number, called the *(geometric) margin achieved by  $\alpha$  on  $x$  w.r.t. target function  $f$  and kernel  $K$* , is of interest:

$$\mu_K(f|\alpha, x) := \frac{f(x) \cdot \langle w(\alpha), \Phi(x) \rangle}{\|w(\alpha)\| \cdot \|\Phi(x)\|} \quad (6)$$

By averaging over all  $x \in D$ , we obtain the function

$$\bar{\mu}_K(f|\alpha) := 2^{-n} \sum_{x \in D} \mu_K(f|\alpha, x) .$$

<sup>1</sup>In the sequel, we drop index  $K$  unless we would like to stress the dependence on  $K$ .

Focusing on the margin that is guaranteed for every  $x \in D$ , we should consider the function

$$\mu_K(f|\alpha) := \min_{x \in D} \mu_K(f|\alpha, x) .$$

By taking the supremum over all  $\alpha : D \rightarrow \mathbb{R}$ , we get the respective parameters of a large margin classifier employing kernel function  $K$ :

$$\begin{aligned} \bar{\mu}_K(f) &:= \sup_{\alpha: D \rightarrow \mathbb{R}} \bar{\mu}_K(f|\alpha) \\ \mu_K(f) &:= \sup_{\alpha: D \rightarrow \mathbb{R}} \mu_K(f|\alpha) \end{aligned}$$

Finally, taking the supremum ranging over all  $K$  from a given kernel class  $\mathcal{C}$ , we get the respective parameters of a best possible large margin classifier among those that employ a kernel from  $\mathcal{C}$ :

$$\begin{aligned} \bar{\mu}_{\mathcal{C}}(f) &:= \sup_{K \in \mathcal{C}} \bar{\mu}_K(f) \\ \mu_{\mathcal{C}}(f) &:= \sup_{K \in \mathcal{C}} \mu_K(f) \end{aligned}$$

We briefly note that, obviously, the guaranteed margin is upper bounded by the average margin:

$$\begin{aligned} \mu_K(f|\alpha) &\leq \bar{\mu}_K(f|\alpha) \\ \mu_K(f) &\leq \bar{\mu}_K(f) \\ \mu_{\mathcal{C}}(f) &\leq \bar{\mu}_{\mathcal{C}}(f) \end{aligned}$$

## 3 A General Notion of Invariance

Throughout this section,  $D$  denotes an arbitrary finite domain,  $\mathcal{S}(D)$  is the group of permutations over  $D$ , and  $\mathcal{G} \leq \mathcal{S}(D)$  is an arbitrary but fixed subgroup. A distributed function over  $D$  with values in  $V \subseteq \mathbb{C}$  is said to be  $\mathcal{G}$ -invariant if, for all  $x, y \in D$  and every  $\sigma \in \mathcal{G}$ , the following holds:

$$f(\sigma(x), \sigma(y)) = f(x, y)$$

We clearly have the

**Pointwise Closure Property:** The pointwise limit of  $\mathcal{G}$ -invariant functions is a  $\mathcal{G}$ -invariant function. Furthermore, if  $f_1, \dots, f_d$  are  $\mathcal{G}$ -invariant functions and  $g : V^d \rightarrow W$  is an arbitrary function with values in  $W \subseteq \mathbb{C}$ , then

$$g(f_1(x, y), \dots, f_d(x, y))$$

is  $\mathcal{G}$ -invariant too.

More interesting is the the following result:

**Lemma 1**  *$\mathcal{G}$ -invariant distributed functions over a finite domain  $D$  are closed under the usual matrix product and under the tensor-product of matrices. More precisely, let  $F(x, y)$  and  $G(x, y)$  be two  $\mathcal{G}$ -invariant distributed functions (here viewed as matrices). Then, the functions  $(F \cdot G)(x, y)$  is  $\mathcal{G}$ -invariant and the function  $(F \otimes G)[(u, x), (v, y)]$  is invariant over  $\mathcal{G} \times \mathcal{G}$  (as subgroup of  $\mathcal{S}(D) \times \mathcal{S}(D)$ ).*

**Proof:** Consider first the function  $(F \cdot G)(x, y)$ . Let  $x, y \in D$  and  $\sigma \in \mathcal{G}$  be arbitrary but fixed. The following calculation shows that it is  $\mathcal{G}$ -invariant:

$$\begin{aligned} (F \cdot G)_{\sigma(x), \sigma(y)} &= \sum_{z \in D} F_{\sigma(x), z} \cdot G_{z, \sigma(y)} \\ &= \sum_{z \in D} F_{x, \sigma^{-1}(z)} \cdot G_{\sigma^{-1}(z), y} \\ &= \sum_{z \in D} F_{x, z} \cdot G_{z, y} \\ &= (F \cdot G)_{x, y} \end{aligned}$$

Now consider the tensor-product  $(F \otimes G)[(u, x), (v, y)]$ , which is a distributed function over  $D \times D$ , i.e., a function over domain  $(D \times D) \times (D \times D)$ . The following calculation shows that it is  $(\mathcal{G} \times \mathcal{G})$ -invariant:

$$\begin{aligned} (F \otimes G)[(\sigma(u), \tau(x)), (\sigma(v), \tau(y))] &= \\ F(\sigma(u), \sigma(v)) \cdot G(\tau(x), \tau(y)) &= \\ F(u, v) \cdot G(x, y) &= \\ (F \otimes G)[(u, x), (v, y)] & \blacksquare \end{aligned}$$

In this section, we shall show the following. If  $f : D \rightarrow \{-1, 1\}$  is a function on domain  $D$  and  $\mathcal{G}$  is a subgroup of  $\mathcal{S}(D)$ , then the largest average (or largest guaranteed, resp.) margin that can be obtained when  $f$  is represented by a  $\mathcal{G}$ -invariant kernel is upper-bounded by the largest average (or largest guaranteed, resp.) margin that can be obtained for the family

$$\mathcal{G}_f := \{f_\sigma : \sigma \in \mathcal{G}\}$$

where

$$f_\sigma(x) := f(\sigma(x)) .$$

Since there are classical margin bounds that apply to the family  $\mathcal{G}_f$ , we obtain corresponding bounds that apply to the single function  $f$ . An analogous remark holds for dimension bounds. Details follow.

Assume that  $K(x, y)$  is a  $\mathcal{G}$ -invariant kernel and consider the feature map  $\Phi = \Phi_K$  that represents  $K$  in the Reproducing Kernel Hilbert Space. Then, for all  $x, y \in D$  and every  $\sigma \in \mathcal{G}$ ,  $\Phi$  satisfies

$$\langle \Phi(\sigma(x)), \Phi(\sigma(y)) \rangle = \langle \Phi(x), \Phi(y) \rangle . \quad (7)$$

**Lemma 2** *If kernel  $K$  is  $\mathcal{G}$ -invariant, then the following holds for every  $x \in D$  and every  $\sigma \in \mathcal{G}$ :*

$$\begin{aligned} \|\Phi_K(\sigma(x))\|_K &= \|\Phi_K(x)\|_K \\ \|w(\alpha)\|_K &= \|w(\alpha_\sigma)\|_K \end{aligned}$$

*In other words, the norm  $\|\cdot\|_K$  is constant on feature vectors of instances taken from the same orbit*

$$x^\mathcal{G} := \{\sigma(x) : \sigma \in \mathcal{G}\}$$

*and it assigns the same value to all dual vectors from the set*

$$\{w(\alpha_\sigma) : \sigma \in \mathcal{G}\} . \quad \blacksquare$$

**Proof:** Let  $\Phi = \Phi_K$ ,  $\|\cdot\| = \|\cdot\|_K$ , and  $\langle \cdot, \cdot \rangle = \langle \cdot, \cdot \rangle_K$ . Clearly,  $\|\Phi(\sigma(x))\| = \|\Phi(x)\|$  because of

$$\begin{aligned} \|\Phi(\sigma(x))\|^2 &= \langle \Phi(\sigma(x)), \Phi(\sigma(x)) \rangle \\ &\stackrel{(7)}{=} \langle \Phi(x), \Phi(x) \rangle \\ &= \|\Phi(x)\|^2 . \end{aligned}$$

As for the second statement, see the following calculation:

$$\begin{aligned} \|w(\alpha_\sigma)\|^2 &= \langle w(\alpha_\sigma), w(\alpha_\sigma) \rangle \\ &\stackrel{(5)}{=} \left\langle \sum_{x \in D} \alpha_\sigma(x) \Phi(x), \sum_{y \in D} \alpha_\sigma(y) \Phi(y) \right\rangle \\ &= \sum_{x, y \in D} \alpha(\sigma(x)) \alpha(\sigma(y)) \langle \Phi(x), \Phi(y) \rangle \\ &= \sum_{x, y \in D} \alpha(x) \alpha(y) \langle \Phi(\sigma^{-1}(x)), \Phi(\sigma^{-1}(y)) \rangle \\ &\stackrel{(7)}{=} \sum_{x, y \in D} \alpha(x) \alpha(y) \langle \Phi(x), \Phi(y) \rangle \\ &= \|w(\alpha)\|^2 \quad \blacksquare \end{aligned}$$

**Lemma 3** *For every  $\mathcal{G}$ -invariant kernel  $K$ , and every choice of  $f : D \rightarrow \{-1, 1\}$ ,  $x \in D$ ,  $\sigma \in \mathcal{G}$ , and  $\alpha : D \rightarrow \mathbb{R}$ , the following holds:*

$$\mu_K(f_\sigma | \alpha_\sigma, x) = \mu_K(f | \alpha, \sigma(x))$$

**Proof:** The proof starts as follows:

$$\begin{aligned} f_\sigma(x) \cdot \langle w(\alpha_\sigma), \Phi(x) \rangle &\stackrel{(5)}{=} \\ f_\sigma(x) \left\langle \sum_{y \in D} \alpha_\sigma(y) \Phi(y), \Phi(x) \right\rangle &= \\ f(\sigma(x)) \sum_{y \in D} \alpha(\sigma(y)) \langle \Phi(y), \Phi(x) \rangle &\stackrel{(7)}{=} \\ f(\sigma(x)) \sum_{y \in D} \alpha(\sigma(y)) \langle \Phi(\sigma(y)), \Phi(\sigma(x)) \rangle &= \\ f(\sigma(x)) \left\langle \sum_{y \in D} \alpha(\sigma(y)) \Phi(\sigma(y)), \Phi(\sigma(x)) \right\rangle &= \\ f(\sigma(x)) \left\langle \sum_{y \in D} \alpha(y) \Phi(y), \Phi(\sigma(x)) \right\rangle &= \\ f(\sigma(x)) \langle w(\alpha), \Phi(\sigma(x)) \rangle & \end{aligned}$$

Using this calculation in combination with Lemma 2, the proof is easy to accomplish:

$$\begin{aligned} \mu_K(f_\sigma | \alpha_\sigma, x) &\stackrel{(6)}{=} \frac{f_\sigma(x) \cdot \langle w(\alpha_\sigma), \Phi(x) \rangle}{\|w(\alpha_\sigma)\| \cdot \|\Phi(x)\|} \\ &= \frac{f(\sigma(x)) \cdot \langle w(\alpha), \Phi(\sigma(x)) \rangle}{\|w(\alpha)\| \cdot \|\Phi(\sigma(x))\|} \\ &\stackrel{(6)}{=} \mu_K(f | \alpha, \sigma(x)) \quad \blacksquare \end{aligned}$$

**Corollary 4** For every  $\mathcal{G}$ -invariant kernel  $K$ , and every choice of  $f : D \rightarrow \{-1, 1\}$ ,  $\sigma \in \mathcal{G}$ , and  $\alpha : D \rightarrow \mathbb{R}$ , the following holds:

$$\begin{aligned}\bar{\mu}_K(f_\sigma|\alpha_\sigma) &= \bar{\mu}_K(f|\alpha) \\ \mu_K(f_\sigma|\alpha_\sigma) &= \mu_K(f|\alpha) \\ \bar{\mu}_K(f_\sigma) &= \bar{\mu}_K(f) \\ \mu_K(f_\sigma) &= \mu_K(f) \\ \bar{\mu}_G(f_\sigma) &= \bar{\mu}_G(f) \\ \mu_G(f_\sigma) &= \mu_G(f)\end{aligned}$$

Note that the last two equations in Corollary 4 basically say that the largest (average or guaranteed) margin that can be achieved for a function  $f$  by a large margin classifier is invariant under  $\mathcal{G}$  (provided that the underlying kernel is  $\mathcal{G}$ -invariant).

Let  $M \in \{-1, 1\}^{r \times s}$  be a sign matrix. Consider a linear arrangement  $\mathcal{A}$  given by unit vectors  $u_1, \dots, u_r; v_1, \dots, v_s \in \mathbb{R}^d$ . The *average margin achieved by this arrangement for sign matrix  $M$*  is defined as follows:

$$\bar{\mu}(M|\mathcal{A}) := \frac{1}{rs} \cdot \sum_{i=1}^r \sum_{j=1}^s M_{i,j} \langle u_i, v_j \rangle$$

The largest average margin that can be achieved for sign matrix  $M$  by any linear arrangement is then given by

$$\bar{\mu}(M) := \sup_{\mathcal{A}} \bar{\mu}(M|\mathcal{A}) ,$$

where the supremum ranges over all linear arrangements  $\mathcal{A}$  for  $M$ . Forster and Simon [7] have shown that, for every  $M \in \mathbb{R}^{r \times s}$ , every  $d \geq 1$ , and every choice of unit vectors  $u_1, \dots, u_r; v_1, \dots, v_s$  in a real inner-product space, the following holds:

$$\sum_{i=1}^r \sum_{j=1}^s M_{i,j} \langle u_i, v_j \rangle \leq \sqrt{rs} \|M\| .$$

From that, we conclude that

$$\bar{\mu}(M) \leq \frac{\|M\|}{\sqrt{rs}} .$$

Consider the sign matrix  $M^{f,\mathcal{G}}$  given by

$$M_{x,\sigma}^{f,\mathcal{G}} := f_\sigma(x) . \quad (8)$$

In combination with Corollary 4, we arrive at the following

**Theorem 5** Let  $D$  be a finite domain, and let  $\mathcal{G}$  be a subgroup of  $\mathcal{S}(D)$ . Then, every function  $f : D \rightarrow \{-1, 1\}$  satisfies

$$\bar{\mu}_G(f) \leq \frac{\|M^{f,\mathcal{G}}\|}{\sqrt{|D| \cdot |\mathcal{G}|}} .$$

In other words, no large margin classifier that employs a  $\mathcal{G}$ -invariant kernel can achieve an average margin for  $f$  which exceeds  $\frac{\|M^{f,\mathcal{G}}\|}{\sqrt{|D| \cdot |\mathcal{G}|}}$ .

As our input space  $D$  is finite, we can assume without loss of generality that the Reproducing Kernel Hilbert Space for a kernel  $K$  on  $D$  coincides with  $\mathbb{R}^{d(K)}$  for some suitable  $1 \leq d(K) \leq |D|$ . We say that  $\alpha : D \rightarrow \mathbb{R}$  represents target function  $f$  correctly w.r.t. kernel  $K$  if

$$\forall x \in D : \mu_K(f|\alpha, x) > 0 .$$

**Corollary 6** Let  $d_G(f)$  denote the smallest dimension of a feature space associated with a  $\mathcal{G}$ -invariant kernel  $K$  that allows for a correct representation of  $f$ . Then,

$$d_G(f) \geq \frac{\sqrt{|D| \cdot |\mathcal{G}|}}{\|M^{f,\mathcal{G}}\|} .$$

**Proof:** According to Lemma 3, a kernel that allows for a correct representation of  $f$  allows also for a correct representation of all  $f_\sigma$ . According to a result by Forster [4], the corresponding feature space must have dimension at least  $\sqrt{|D| \cdot |\mathcal{G}|} / \|M^{f,\mathcal{G}}\|$ . ■

Corollary 6 can be strengthened slightly:

**Corollary 7** Let  $\sigma_i$  denote the  $i$ -th singular value of  $M^{f,\mathcal{G}}$ , where  $\sigma_1, \sigma_2, \dots$  are in decreasing order. Then,  $d_G(f)$  satisfies the following lower bound:

$$d_G(f) \cdot \sum_{i=1}^{d_G(f)} \sigma_i^2 \geq 1 \quad (9)$$

**Proof:** Let  $A \in \{-1, 1\}^{r \times s}$  be a matrix whose columns are viewed as binary functions  $f_1, \dots, f_s$ . It has been shown by Forster and Simon [7] that the dimension  $d$  of a feature space which allows for a correct representation of  $f_1, \dots, f_s$  satisfies

$$d \cdot \sum_{i=1}^d \sigma_i^2(A) \geq rs .$$

This trivially implies (9). ■

## 4 Rotation-invariant Functions

In Section 4.1 we will derive some facts about distributed functions over a finite Abelian group via the Fourier-expansion. Section 4.2 ties everything together and presents the resulting margin and dimension bounds obtained in this restricted setting.

### 4.1 Distributed Functions over Finite Abelian Groups

We apply the results of the preceding section to the case where  $D$  is a Abelian group of finite size  $d$ , and  $\mathcal{G}_{rot}$  is the subgroup of  $\mathcal{S}(D)$  consisting of all permutations of the form  $x \mapsto x + a$ . Note that  $d = |D| = |\mathcal{G}_{rot}|$ .

We are interested in distributed functions  $f : D \times D \rightarrow \mathbb{C}$  and arrange the  $d^2$  Fourier-coefficients of such a function as a matrix as follows:

$$\widehat{F}_{a,b} = \widehat{f}(a, -b) \quad (10)$$

$$= d^{-2} \sum_{(x,y) \in D \times D} f(x,y) \overline{\chi_{(a,-b)}(x,y)} \quad (11)$$

$$= d^{-2} \cdot \sum_{x \in D} \sum_{y \in D} f(x,y) \overline{\chi_a(x)} \chi_b(y) \quad (12)$$

In matrix notation, this reads as

$$\widehat{F} = d^{-2} \cdot H^* \cdot F \cdot H, \quad (13)$$

where  $H$  is the matrix from (4).

A distributed function  $f(x,y)$  over  $D$  is said to be *rotation-invariant* if, for all  $x, y, a \in D$ , the following holds:

$$f(x+a, y+a) = f(x,y)$$

In the sense of the previous section,  $f$  is meant to be  $\mathcal{G}_{rot}$ -invariant.

Here are some examples for rotation-invariant functions:

- A distributed function of the form  $f(x,y) = g(x-y)$  is obviously rotation-invariant. Conversely, any rotation-invariant function  $f(x,y)$  can be written in this form by setting  $g(x) := f(x,0)$  because rotation-invariance implies that

$$f(x,y) = f(x-y,0) = g(x-y).$$

- Because of the obvious identity

$$\chi_z(x-y) = \chi_z(x) \cdot \overline{\chi_z(y)},$$

the distributed function  $\chi_z(x) \cdot \overline{\chi_z(y)}$  is rotation-invariant too.

The fact that  $f(x,y) = g(x-y)$  is a rotation-invariant function can be restated as follows: any function  $f(x,y)$  that can be cast as a function in  $x_1 - y_1 \bmod q_1, \dots, x_n - y_n \bmod q_n$  is rotation-invariant.

In terms of the matrix of Fourier-coefficients,  $\widehat{F}$ , rotation-invariant functions over  $D$  can be characterized as follows:

**Lemma 8** *A distributed function  $f(x,y)$  over  $D$  is rotation-invariant iff  $\widehat{F}$  is a diagonal matrix.*

**Proof:** Assume first that  $f(x,y)$  is rotation-invariant. Consider a Fourier-coefficient in  $\widehat{F}$  outside the main diagonal, say  $\widehat{F}_{a,b}$  so that  $a_k \neq b_k$ . Every pair  $(x,y)$  can be put into the equivalence class

$$\{(x + j\vec{e}_k, y + j\vec{e}_k) : j = 0, \dots, q_k - 1\}.$$

We show that every equivalence class contributes 0 to (12):

$$\begin{aligned} & \sum_{j=0}^{q_k-1} f(x + j\vec{e}_k, y + j\vec{e}_k) \overline{\chi_a(x + j\vec{e}_k)} \cdot \chi_b(y + j\vec{e}_k) = \\ & f(x,y) \overline{\chi_a(x)} \cdot \chi_b(y) \sum_{j=0}^{q_k-1} \overline{\chi_{a_k}^{(q_k)}(j)} \chi_{b_k}^{(q_k)}(j) \end{aligned}$$

The latter sum vanishes because it equals

$$\sum_{j=0}^{q_k-1} \omega_{q_k}^{(b_k - a_k)j}.$$

Recall that  $\delta$  denotes the Kronecker symbol and it is well-known that

$$\sum_{j=0}^{m-1} \omega_m^{(l'-l)j} = m \cdot \delta_{l,l'}.$$

This shows that  $\widehat{F}_{a,b} = 0$ .

Now assume that  $\widehat{F}$  is a diagonal matrix. We conclude from (13) that

$$F = H \cdot \widehat{F} \cdot H^*, \quad (14)$$

which implies that

$$F_{x,y} = \sum_{z \in D} \widehat{F}_{z,z} \cdot \chi_x(z) \cdot \overline{\chi_y(z)}.$$

Rotation-invariance is now easily obtained:

$$\begin{aligned} f(x+a, y+a) &= \sum_{z \in D} \widehat{F}_{z,z} \cdot \chi_{x+a}(z) \cdot \overline{\chi_{y+a}(z)} \\ &= \sum_{z \in D} \widehat{F}_{z,z} \cdot \chi_z(x+a) \cdot \overline{\chi_z(y+a)} \\ &= \sum_{z \in D} \widehat{F}_{z,z} \cdot \chi_z(x) \cdot \overline{\chi_z(y)} \\ &= f(x,y) \end{aligned}$$

In the second-last equation, we used the rotation-invariance of  $\chi_z(x) \cdot \overline{\chi_z(y)}$ . ■

**Corollary 9** *Assume that  $f(x,y)$  is a rotation-invariant distributed function over  $D$  and let  $F_{x,y} = f(x,y)$  denote the corresponding matrix. Then the (complex) eigenvalues of  $d^{-1} \cdot F$  are found on the main diagonal of  $\widehat{F}$ .*

**Proof:** Rewrite (14) as

$$d^{-1}F = (d^{-1/2}H) \cdot \widehat{F} \cdot (d^{-1/2}H^*)$$

and observe that this is nothing but the spectral decomposition of  $d^{-1}F$  (since  $\widehat{F}$  is a diagonal matrix and  $d^{-1/2}H$  is unitary). ■

We briefly note the following result:

**Lemma 10** Let  $\hat{F}$  be the (diagonal) matrix that contains the Fourier-coefficients of the (rotation-invariant) distributed function  $f(x - y)$ . Then, for every  $z \in D$ ,  $\hat{f}(z) = \hat{F}_{z,z}$ .

**Proof:** Consider the function  $f_y(x) := f(x - y)$ . We shall show below that the Fourier coefficients of  $f$  and  $f_y$  are related as follows:

$$\hat{f}_y(z) = \hat{f}(z) \cdot \overline{\chi_y(z)} . \quad (15)$$

The proof is now obtained by the following calculation:

$$\begin{aligned} \hat{F}_{z,z} &= d^{-2} \cdot \sum_{x,y \in D} f(x - y) \cdot \overline{\chi_x(x)} \cdot \chi_z(y) \\ &= d^{-1} \cdot \sum_{y \in D} \left( d^{-1} \cdot \sum_{x \in D} f_y(x) \overline{\chi_x(x)} \right) \chi_z(y) \\ &= d^{-1} \cdot \sum_{y \in D} \hat{f}_y(z) \cdot \chi_z(y) \\ &\stackrel{(15)}{=} \hat{f}(z) \cdot d^{-1} \cdot \sum_{y \in D} \underbrace{\overline{\chi_y(z)} \chi_z(y)}_{=1} \\ &= \hat{f}(z) \end{aligned}$$

The following calculation verifies (15):

$$\begin{aligned} \hat{f}_y(z) &= d^{-1} \cdot \sum_{x \in D} f(x - y) \cdot \overline{\chi_x(x)} \\ &= d^{-1} \cdot \sum_{x \in D} \sum_{w \in D} \hat{f}(w) \cdot \chi_w(x - y) \cdot \overline{\chi_x(x)} \\ &= d^{-1} \cdot \sum_{x \in D} \sum_{w \in D} \hat{f}(w) \cdot \chi_w(x) \cdot \overline{\chi_w(y)} \cdot \overline{\chi_z(x)} \\ &= d^{-1} \cdot \sum_{w \in D} \underbrace{\left( \sum_{x \in D} \chi_w(x) \cdot \overline{\chi_z(x)} \right)}_{=d \cdot \delta_{w,z}} \hat{f}(w) \cdot \overline{\chi_w(y)} \\ &= \hat{f}(z) \cdot \overline{\chi_z(y)} \end{aligned}$$

Corollary 9 and Lemma 10 yield the following.<sup>2</sup>

**Corollary 11** Let  $F$  denote the matrix with entries  $F_{x,y} = f(x - y)$ . Then the spectrum of (complex) eigenvalues of  $d^{-1} \cdot F$  coincides with the spectrum of (complex) Fourier-coefficients of  $f$ .

Consider the sign matrix  $M^{f, \mathcal{G}_{rot}}$ . From (8) and the definition of  $\mathcal{G}_{rot}$ , we conclude that

$$M_{x,y}^{f, \mathcal{G}_{rot}} = f(x + y) .$$

It follows that  $M^{f, \mathcal{G}_{rot}}$  is a symmetric matrix. If  $f$  is real-valued, then  $M^{f, \mathcal{G}_{rot}}$  has real eigenvalues. Note

<sup>2</sup>This result might be known, but we are not aware of an appropriate pointer to the literature.

that  $M^{f, \mathcal{G}_{rot}}$  coincides with matrix  $F_{x,y} = f(x - y)$  up to a permutation of columns (where the column indexed  $y$  is exchanged with the column indexed  $-y$ ). Since the spectrum of eigenvalues (or singular values, resp.) of a matrix is left invariant under a permutation of columns, we obtain the following

**Corollary 12** Let  $f(x - y)$  be real-valued, and let  $F$  be the matrix with entries  $F_{x,y} = f(x - y)$ . Then, the following holds:

1.  $F$  coincides with the symmetric matrix  $M^{f, \mathcal{G}_{rot}}$  up to a permutation of columns.
2. The spectrum of eigenvalues of  $d^{-1} \cdot F$  coincides with the spectrum of (real) eigenvalues of  $d^{-1} \cdot M^{f, \mathcal{G}_{rot}}$  and with the spectrum of Fourier-coefficients of  $f$ .

## 4.2 Margin and Dimension Bounds for Rotation-invariant Kernels

For every function  $f : D \rightarrow \{-1, 1\}$ ,

$$\bar{\mu}_{rot}(f) := \bar{\mu}_{\mathcal{G}_{rot}}(f)$$

denotes the largest possible average margin that can be achieved by a linear arrangement for  $f$  resulting from a rotation-invariant kernel. As for the smallest possible dimension, parameter  $d_{rot}(f)$  is understood analogously.

**Corollary 13** Let  $D$  be a finite Abelian group of size  $d$ . Every function  $f : D \rightarrow \{-1, 1\}$  satisfies

$$\bar{\mu}_{rot}(f) \leq \|\hat{f}\|_{\infty} . \quad (16)$$

In other words, no large margin classifier that employs a rotation-invariant kernel can achieve an average margin for  $f$  which exceeds  $\|\hat{f}\|_{\infty}$ .

**Proof:** According to Theorem 5,

$$\bar{\mu}_{rot}(f) \leq \frac{\|M^{f, \mathcal{G}_{rot}}\|}{\sqrt{|D| \cdot |\mathcal{G}_{rot}|}} = \frac{\|M^{f, \mathcal{G}_{rot}}\|}{d} .$$

We conclude from Corollary 12 that

$$\|M^{f, \mathcal{G}_{rot}}\| = \|F\| = d \cdot \|\hat{f}\|_{\infty} ,$$

which leads us to inequality (16). ■

Corollary 6 and 7 combined with Corollary 11 lead us to the following results:

**Corollary 14** Let  $d_{rot}(f)$  denote the smallest dimension of a feature space associated with a rotation-invariant kernel  $K$  that allows for a correct representation of  $f$ . Then,  $d_{rot}(f) \geq \|\hat{f}\|_{\infty}^{-1}$ .

**Proof:** According to Corollary 6, the corresponding feature space for the kernel must have dimension at least  $\sqrt{|D| \cdot |\mathcal{G}_{rot}|} / \|M^{f, \mathcal{G}_{rot}}\| = d / \|M^{f, \mathcal{G}_{rot}}\|$ . According to Corollary 12, the latter expression evaluates to  $\|\hat{f}\|_{\infty}^{-1}$ . ■

**Corollary 15** Let  $\widehat{f}_i$  denote the  $i$ -th Fourier-coefficient of  $f$ , where  $|\widehat{f}_1|, \dots, |\widehat{f}_d|$  are in decreasing order. Then,

$$d_{rot}(f) \cdot \sum_{i=1}^{d_{rot}(f)} |\widehat{f}_i|^2 \geq 1$$

**Proof:** From (9), we obtain

$$d_{rot}(f) \cdot \sum_{i=1}^{d_{rot}(f)} \sigma_i^2 \geq 1$$

where  $\sigma_i$  denotes the  $i$ -th largest singular value of  $M^{f, \mathcal{G}_{rot}}$ . We conclude from Corollary 12, that  $\sigma_i$  coincides with  $|\widehat{f}_i|$ . ■

## 5 Reflection-invariant Functions

In this section, we consider real-valued functions only. A distributed function  $f(x, y)$  over  $\{-1, 1\}^n$  is said to be *reflection-invariant* if, for all  $x, y, a \in \{-1, 1\}^n$ , the following holds:

$$f(x \circ a, y \circ a) = f(x, y) \quad (17)$$

Note that reflection-invariance corresponds to rotation-invariance with  $(\mathbb{Z}_2^2, +)$  as the underlying (additive) Abelian group is or, equivalently, with  $(\{-1, 1\}^n, \cdot)$  as the underlying (multiplicative) Abelian group. This is because the subgroup  $\mathcal{G}_{rot}$  of  $\mathcal{S}(D)$  that we have used for rotation-invariant distributed functions collapses for  $D = \{-1, 1\}^n$  (with a multiplicative group structure) to the following subgroup of  $\mathcal{S}(\{-1, 1\}^n)$ :

$$\mathcal{G}_{ref} = \{x \mapsto x \circ a : a \in \{-1, 1\}^n\}$$

Thus, reflection-invariant functions inherit all closure properties that hold, in general, for  $\mathcal{G}$ -invariant distributed functions (see the Pointwise Closure Property and Lemma 1 in Section 3):

**Corollary 16** 1. *The pointwise limit of reflection-invariant functions is a reflection-invariant function. Furthermore, if  $f_1, \dots, f_d$  are reflection-invariant functions and  $g : \mathbb{R}^d \rightarrow \mathbb{R}$  is an arbitrary function, then*

$$g(f_1(x, y), \dots, f_d(x, y))$$

*is reflection-invariant too.*

2. *Reflection-invariant distributed functions over  $\{-1, 1\}^n$  are closed under the usual matrix product and under the tensor-product of matrices.*

Furthermore, reflection-invariant functions inherit all properties that hold, in general, for distributed functions over a finite Abelian group:

- A reflection-invariant function  $f(x, y)$  can be decomposed according to (2). Since  $D = \{-1, 1\}^n$ , the character  $\chi_z$  coincides with the parity function induced by  $z$ , i.e.,  $\chi_z(x) = \prod_{z_i=-1} x_i$ .

- The matrix  $\widehat{F}$  whose entries are the Fourier coefficients of  $f$  satisfies (13) where  $H$  is the matrix from (4). Since  $D = \{-1, 1\}^n$ ,  $H$  equals the well-known  $(2^n \times 2^n)$ -Walsh-Hadamard matrix.

Distributed functions  $f(x, y)$  over  $\mathbb{R}^n$  that satisfy (17) for all  $x, y \in \mathbb{R}^n$  and every  $a \in \{-1, 1\}^n$  are said to be *reflection-invariant in the Euclidean space*. Here are some examples (with some overlap to our exemplification of rotation-invariant functions in Section 4):

- A distributed function of the form  $f(x, y) = g(x \circ y)$  is reflection-invariant (in the Euclidean space provided that the domain is  $\mathbb{R}^n$ ):

$$g((x \circ a) \circ (y \circ a)) = g(x \circ y \circ (a \circ a)) = g(x \circ y)$$

Conversely, any reflection-invariant function  $f(x, y)$  (over domain  $\{-1, 1\}^n$ ) can be written in this form by setting  $g(x) := f(x, \vec{e})$  because reflection-invariance implies that

$$f(x, y) = f(x \circ y, y \circ y) = f(x \circ y, \vec{e}) = g(x \circ y) .$$

- Because of the obvious identity

$$\chi_z(x \circ y) = \chi_z(x) \cdot \chi_z(y) ,$$

the distributed function  $\chi_z(x) \cdot \chi_z(y)$  is reflection-invariant too.

- The metric

$$L_p(x - y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

induced by the  $L_p$ -norm is clearly reflection-invariant in the Euclidean space.

In Section 6, we shall see that many popular kernel functions happen to be reflection-invariant.

The fact that  $f(x, y) = g(x \circ y)$  is a reflection-invariant function can be restated as follows: any function  $f(x, y)$  that can be cast as a function in  $x_1 \cdot y_1, \dots, x_n \cdot y_n$  is reflection-invariant. Similarly, any function  $f(x, y)$  that can be cast as a function in  $L_p(x - y)$  (or, more generally, in  $|x_1 - y_1|, \dots, |x_n - y_n|$ ) is reflection-invariant.

## 6 Reflection-invariant Kernels

In this section, we consider kernel functions  $K(x, y)$  over the Boolean or over the Euclidean domain. In other words,  $K(x, y)$  is a distributed function over  $\{-1, 1\}^n$  or over  $\mathbb{R}^n$  with the additional property that every finite principal sub-matrix of  $K$  is symmetric and positive semidefinite. In Section 6.1, we demonstrate that the family of reflection-invariant kernels is quite rich and contains many popular kernels. In Section 6.2, we derive margin and dimension bounds for reflection-invariant kernels.

### 6.1 Examples and Closure Properties

Let us start with some examples. The following (quite popular) kernels (over  $\mathbb{R}^n$  except for the DNF-Kernel that has a Boolean domain) can be cast as functions in  $x_1 \cdot y_1, \dots, x_n \cdot y_n$  or as functions in  $\|x - y\|_2$  and are therefore reflection-invariant:

**Polynomial Kernels:**  $K(x, y) = p(x^\top y)$  for an arbitrary polynomial  $p$  with positive coefficients.

**All-subsets Kernel:**  $K(x, y) = \prod_{i=1}^n (1 + x_i y_i)$ .

**ANOVA Kernel:** Let  $1 \leq s \leq n$  and define

$$K_s(x, y) = \sum_{1 \leq i_1 < \dots < i_s \leq n} \prod_{j=1}^s x_{i_j} y_{i_j} .$$

**DNF-Kernel:**  $K(x, y) = -1 + 2^{-n} \prod_{i=1}^n (x_i y_i + 3)$ .

**Exponential Kernels:**  $K(x, y) = e^{p(x^\top y)}$  for an arbitrary polynomial  $p$  with positive coefficients.

**Gaussian Kernel:**  $K(x, y) = e^{-\|x-y\|_2^2 / \sigma^2}$  for an arbitrary  $\sigma > 0$ .

These kernels have the usual nice properties like being efficiently evaluable although the number of (implicitly represented) features is exponentially large (or even infinite). Polynomial, Exponential, and Gaussian Kernels (first used in [2]) are found in almost any basic text-book that is relevant to the subject (e.g. [3]). The All-subsets Kernel is found in [18], and the ANOVA Kernel is found in [19]. As for the latter two kernels, see also [17]. The DNF-Kernel has been proposed in [16].<sup>3</sup> The reader interested in more information about these (and other) kernels may consult the relevant literature. Here, we simply point to the fact that all kernels mentioned above are reflection-invariant.

We move on and consider the possibility of making new reflection-invariant kernels from kernels that are already known to be reflection-invariant. To this end, we briefly call into mind some basic closure properties of kernels:

**Lemma 17** *Let  $K, K_1, K_2$  be kernels, and let  $c > 0$  be a positive constant. Then, the distributed functions*

$$\begin{aligned} K_1(x, y) + K_2(x, y) & , \quad c \cdot K(x, y) \\ K_1(x, y) \cdot K_2(x, y) & , \quad (K_1 \otimes K_2)[(u, x), (v, y)] \end{aligned}$$

*are kernels too. Moreover, the pointwise limit of kernels yields a kernel.*

<sup>3</sup>In [16], the kernel is defined over the Boolean domain  $\{0, 1\}^n$ . Our formula above is obtained from the formula in [16] by plugging in the affine transformation that identifies 1 with  $-1$  and 0 with 1. A similar remark applies to the Monotone DNF-Kernel discussed at the end of this section.

The proof of Lemma 17 can be looked-up in [3], for example.

**Corollary 18** *If  $K_1, \dots, K_d$  are kernels and  $P : \mathbb{R}^d \rightarrow \mathbb{R}$  is a polynomial (or a converging power series) with positive coefficients, then*

$$P(K_1(x, y), \dots, K_d(x, y))$$

*is a kernel too.*

Note that closure properties of reflection-invariant functions (see Corollary 16) are comparably strong so that Lemma 17 and Corollary 18 remain valid (mutatis mutandis) for reflection-invariant kernels.

The following kernels (proposed in [11] and [9], respectively) define a new kernel-matrix  $K$  in terms of a given symmetric matrix  $B$  (called “similarity matrix” in this context):

**Exponential Diffusion Kernel:** For  $\lambda \in \mathbb{R}$ , define

$$K = e^{\lambda \cdot B} = \sum_{k \geq 0} \frac{\lambda^k}{k!} \cdot B^k .$$

**von Neumann Diffusion Kernel:** For  $0 \leq \lambda < \|B\|^{-1}$ , define

$$K = (I - \lambda \cdot B)^{-1} = \sum_{k \geq 0} \lambda^k \cdot B^k .$$

It follows from the closure properties of reflection-invariant functions that both diffusion kernels would inherit reflection-invariance from the underlying similarity matrix  $B$ .

The family of reflection-invariant kernels is quite rich. But here are two kernels (the first-one from [16], and the second-one from [12]) which are counterexamples:

**Monotone DNF-Kernel:**

$$K(x, y) = -1 + 2^{-2n} \prod_{i=1}^n (x_i y_i - x_i - y_i + 5) .$$

**Spectrum Kernel:** Here,  $x, y \in \{-1, 1\}^n$  are considered as binary strings. For  $1 \leq p \leq n$  and for every substring  $u \in \{-1, 1\}^p$ ,

$$\Phi_v^p(x) = |\{(u, w) : x = uw\}|$$

counts how often  $v$  occurs as a substring of  $x$ . The  $p$ -Spectrum Kernel is then given by

$$K(x, y) = \sum_{v \in \{-1, 1\}^p} \Phi_v^p(x) \cdot \Phi_v^p(y) .$$

It is easy to see that both kernels are *not* reflection-invariant. More generally, string kernels (measuring similarity between strings) often violate reflection-invariance.

## 6.2 Margin and Dimension Bounds for Reflection-invariant Kernels

For every function  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ ,

$$\bar{\mu}_{ref}(f) := \bar{\mu}_{G_{ref}}(f)$$

denotes the largest possible average margin that can be achieved by a linear arrangement for  $f$  resulting from a reflection-invariant kernel. Because reflection-invariance is a special case of rotation-invariance, the following result immediately follows from Corollaries 13, 14, and 15:

**Corollary 19** 1. Every Boolean function  $f$  satisfies

$$\bar{\mu}_{ref}(f) \leq \|\hat{f}\|_\infty.$$

In other words, no large margin classifier that employs a reflection-invariant kernel can achieve an average margin for  $f$  which exceeds  $\|\hat{f}\|_\infty$ .

2. Let  $d_{ref}(f)$  denote the smallest dimension of a feature space associated with a reflection-invariant kernel  $K$  that allows for a correct representation of  $f$ . Then,  $d_{ref}(f) \geq \|\hat{f}\|_\infty^{-1}$ .
3. Let  $\hat{f}_i$  denote the  $i$ -th Fourier-coefficient of  $f$ , where  $|\hat{f}_1|, \dots, |\hat{f}_{2^n}|$  are in decreasing order. Then,  $d_{ref}(f)$  satisfies the following lower bound:

$$d_{ref}(f) \cdot \sum_{i=1}^{d_{ref}(f)} |\hat{f}_i|^2 \geq 1$$

## 7 Conclusions and Open Problems

We start with some remarks which offer a possible interpretation of our results. Finally, some open problems are mentioned.

### 7.1 Discussion of our Results

Ideally the invariance-properties of a kernel reflect symmetries in the data. For example, assume that there exists a set of transformations, say  $T$ , so that, for every instance  $x \in D$  and every transformation  $t \in T$ , the label assigned to  $x$  by target function  $f$  equals the label assigned to  $t(x)$  by  $f$ . Then, it looks desirable to apply a kernel that is invariant under the transformations from  $T$ . It would be surprising if our results implied that such kernels (that sort of perfectly model the symmetries in the data) would inherently lead to small margins or high-dimensional feature spaces. It is, however, easy to argue that (as expected) the contrary is true and our margin and dimension bounds trivialize whenever the invariance of the kernel perfectly matches with symmetries in the data. To see this, consider again (compare with the introduction) the “super-kernel”

$$K(x, y) = \begin{cases} +1 & \text{if } f(x) = f(y) \\ -1 & \text{otherwise} \end{cases}$$

that allows for a 1-dimensional halfspace representation of  $f$  with margin 1, and note that  $K$  actually is invariant under all transformations from  $T$ . Thus, no upper margin bound that holds uniformly for all  $T$ -invariant kernels can be smaller than 1. Similarly, no lower dimension bound can be larger than 1. Note that this is no contradiction to the main results in this paper because the family  $\{f_t : t \in T\}$  of functions  $f_t(x) = f(t(x))$  collapses to the singleton  $\{f\}$ . Thus Forster’s margin and dimension bounds applied to this family do not lead to non-trivial values.

Viewed from this perspective, our results can be interpreted as follows: one should *not* use a kernel that is invariant under a set  $T$  of transformations if  $T$  does *not* reflect symmetries in the data. The kernel becomes very poor especially when the family  $\{f_t : t \in T\}$  contains much “orthogonality” (which is sort of the opposite of collapsing to a singleton or to a family of highly correlated functions) because Forster’s bounds, applied to pairwise (almost) orthogonal functions, are extremely strong.

This interpretation makes clear that our results are not particularly surprising but, on the other hand, quantify (in terms of small margin and large dimension bounds) in a meaningful and rigorous fashion an existing mismatch between a kernel and the (missing or existing) symmetries in the data.

### 7.2 Open Problems

Haasdonk and Burkhardt [8] consider two notions of invariance: “simultaneous invariance” and “total invariance”. Simultaneous invariance very much corresponds to the notion of invariance that we discussed in Section 3 so that our margin and dimension bounds apply. Total invariance is a stronger notion so that our bounds apply more than ever. But the obvious challenge is to find *stronger* margin and dimension bounds for *totally* invariant kernels.

The basic idea behind our paper is roughly as follows. For a family of kernels (e.g., polynomial kernels), we argue that the existence a “good representation” for a particular target function implies the existence of a “good representation” for a whole family of target functions (so that classical margin and dimension bounds can be brought into play). We think that invariance under a group operation (the notion considered in this paper) is just the first obvious thing one should consider. We would like to develop more versatile techniques that, while following the same basic idea, lead to strong margin and dimension bounds for a wider class of kernels.

## References

- [1] Shai Ben-David, Nadav Eiron, and Hans Ulrich Simon. Limitations of learning via embeddings in euclidean half-spaces. *Journal of Machine Learning Research*, 3:441–461, 2002.

- [2] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.
- [3] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
- [4] Jürgen Forster. A linear lower bound on the unbounded error communication complexity. *Journal of Computer and System Sciences*, 65(4):612–625, 2002.
- [5] Jürgen Forster, Matthias Krause, Satyanarayana V. Lokam, Rustam Mubarakzjanov, Niels Schmitt, and Hans Ulrich Simon. Relations between communication complexity, linear arrangements, and computational complexity. In *Proceedings of the 21st Annual Conference on the Foundations of Software Technology and Theoretical Computer Science*, pages 171–182, 2001.
- [6] Jürgen Forster, Niels Schmitt, Hans Ulrich Simon, and Thorsten Suttrop. Estimating the optimal margins of embeddings in euclidean half spaces. *Machine Learning*, 51(3):263–281, 2003.
- [7] Jürgen Forster and Hans Ulrich Simon. On the smallest possible dimension and the largest possible margin of linear arrangements representing given concept classes. *Theoretical Computer Science*, 350(1):40–48, 2006.
- [8] Bernard Haasdonk and Hans Burkhardt. Invariant kernel functions for pattern analysis and machine learning. *Machine Learning*, 68(1):35–61, 2007.
- [9] Jaz S. Kandola, John Shawe-Taylor, and Nello Cristianini. Learning semantic similarity. In *Advances in Neural Information Processing Systems 15*, pages 657–664. MIT Press, 2003.
- [10] Eike Kiltz and Hans Ulrich Simon. Threshold circuit lower bounds on cryptographic functions. *Journal of Computer and System Sciences*, 71(2):185–212, 2005.
- [11] Risi I. Kondor and John D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the 19th International Conference on Machine Learning*, pages 315–322, 2002.
- [12] Christina Leslie, Eleazar Eskin, and William S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Pacific Symposium on Biocomputing*, pages 564–575, 2002.
- [13] Nathan Linial, Shahar Mendelson, Gideon Schechtman, and Adi Shraibman. Complexity measures of sign matrices. *Combinatorica*. To appear.
- [14] Nati Linial and Adi Shraibman. Lower bounds in communication complexity based on factorization norms. In *Proceedings of the 39th Annual Symposium on Theory of Computing*, pages 699–708, 2007.
- [15] Alexander A. Razborov and Alexander A. Sherstov. The sign-rank of  $AC^0$ . Personal Communication.
- [16] Ken Sadohara. Learning of boolean functions using support vector machines. In *Proceedings of the 12th International Conference on Algorithmic Learning Theory*, pages 106–118, 2001.
- [17] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- [18] Eiji Takimoto and Manfred K. Warmuth. Pathe kernels and multiplicative updates. *Journal of Machine Learning Research*, 4:773–818, 2003.
- [19] Vladimir Vapnik, Christopher J. C. Burges, Bernhard Schoelkopf, and R. Lyons. A new method for constructing artificial neural networks. Interim ARPA Technical Report, AT&T Bell Laboratories, 1995.



---

# Learning Acyclic Probabilistic Circuits Using Test Paths

---

Dana Angluin<sup>1</sup> and James Aspnes<sup>1,\*</sup> and Jiang Chen<sup>2,†</sup> and David Eisenstat<sup>3</sup> and Lev Reyzin<sup>1,‡</sup>

<sup>1</sup> Computer Science Department, Yale University  
{angluin, aspnes}@cs.yale.edu, lev.reyzin@yale.edu  
<sup>2</sup> Yahoo! Inc., 701 First Avenue, Sunnyvale, CA 94086  
criver@gmail.com  
<sup>3</sup> eisenstatdavid@gmail.com

## Abstract

We define a model of learning probabilistic acyclic circuits using value injection queries, in which an arbitrary subset of wires is set to fixed values, and the value on the single output wire is observed. We adapt the approach of using test paths from the Circuit Builder algorithm [AACW06] to show that there is a polynomial time algorithm that uses value injection queries to learn Boolean probabilistic circuits of constant fan-in and log depth. In the process, we discover that test paths fail utterly for circuits over alphabets of size greater than two and establish upper and lower bounds on the attenuation factor for general and transitively reduced Boolean probabilistic circuits of test paths versus general experiments. To overcome the limitations of test paths for non-Boolean alphabets, we introduce function injection queries, which allow the symbols on a wire to be mapped to other symbols rather than just to themselves or constants.

## 1 Introduction

Probabilistic networks are used as models in a variety of domains, for example, gene interaction networks, social networks and causal reasoning. In a binary model of gene interaction, the state of each gene is either active or inactive, and the state of each gene is determined as a function of the states of some number of other genes, its inputs. In a probabilistic variant of the model, the activation function specifies, for each possible combination of the states of the inputs, the probability that the gene will be active. In the independent cascade model of social networks, the state of each agent is active or inactive and for each pair  $(u, v)$  of agents, there is a probability that the activation of  $u$  will cause  $v$  to become active. Kempe, Kleinberg and Tardos study the problem of maximizing influence in this and related models of social networks [KKET03, KKT05]. In a Bayesian network there

is an acyclic directed graph and a joint probability distribution over the node values such that the joint distribution is simply the product of each of the marginal distributions for each node given the values of the parents (in-neighbors) of the node.

A fundamental question is how much we can infer about the properties and structure of such networks from observing and experimenting with their behaviors. Prior research suggests that there is no polynomial time algorithm to learn Boolean functions represented by acyclic circuits of constant fan-in and depth  $O(\log n)$  when we can set only the inputs of the circuit and observe only the output [AK95]. In this paper we consider a different setting, **value injection queries**, in which we can fix the values on any subset of wires in the target circuit, but still only observe the output of the circuit.

The idea of value injection queries was inspired by models of gene suppression and gene overexpression in the study of gene interaction networks [AKMM98, ITK00] and was proposed in [AACW06]. They show that with value injection queries, acyclic deterministic circuits with constant-size alphabets, constant fan-in and depth  $O(\log n)$  are learnable up to behavioral equivalence in polynomial time. To extend these results to analog circuits, Angluin et al. [AACR07] consider circuits with polynomial-size alphabets. Larger alphabets make the learning problem significantly harder, necessitating structural restrictions on the graphs of the circuits to achieve polynomial time learnability. They show that with value injection queries, acyclic deterministic circuits that are transitively reduced (or in general, have constant shortcut width) and have polynomial-size alphabets, constant fan-in and unbounded depth are learnable up to behavioral equivalence in polynomial time.

In this paper we investigate how well the above positive results can be extended to the case of acyclic probabilistic circuits. The key technique in the previous work has been the idea of a **test path** for an arbitrary wire  $w$  in the circuit. Informally speaking, a test path is a directed path of wires from  $w$  to the output wire in which each wire is an input of the next wire on the path, and the other (non-path) inputs of wires on the path are fixed to constant values, thus isolating the wires along the path from the rest of the circuit. Ideally, the choice of constant values is made in such a way as to maximize the effect on the output of the circuit of changing  $w$  from one value to another. A test path thus functions as a kind of “microscope” for viewing the effects of different values on the wire  $w$ . The primary focus of this paper is to un-

---

\*Supported in part by NSF grant CNS-0435201.

†Supported in part by a research contract from Consolidated Edison.

‡This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

derstand the properties of test paths in probabilistic circuits, and the extent to which they can be used to give polynomial time algorithms for learning probabilistic acyclic circuits.

In Section 2 we formally define our model of acyclic probabilistic circuits, value injection queries and distribution injection queries, behavioral equivalence, and the learning problem that we consider. In Section 3 we establish some basic results about probabilistic circuits and value and distribution injection experiments. In Section 4 we review the test path lemma used in previous work and show that it fails utterly in probabilistic circuits with alphabet size greater than two. However, for Boolean probabilistic circuits, we show that the test path lemma holds with an attenuation factor that depends on the structure of the circuit. (Lemma 10 treats general acyclic circuits and Corollary 11 specializes the bound to transitively reduced circuits.) In Section 5 we apply the test path lemma in the Boolean case to adapt the Circuit Builder algorithm [AACW06] to find using value injection queries, with high probability, in time polynomial in  $n$  and  $1/\varepsilon$ , a circuit that is  $\varepsilon$ -behaviorally equivalent to a target acyclic Boolean probabilistic circuit of size  $n$  with constant fan-in and depth bounded by a constant times  $\log n$ . In Section 6, we consider lower bounds on the attenuation of paths; Lemma 15 shows that our bound is tight for transitively reduced circuits and Lemma 17 gives a lower bound for the case of general acyclic circuits. In Section 7 we introduce a stronger kind of query, a **function injection query**, and show that test paths with function injections overcome the limitations of test paths for circuits with alphabets of size greater than two.

## 2 Model

### 2.1 Probabilistic Circuits

We extend the circuit learning model studied in [AACR07, AACW06]. to probabilistic gates. An unusual feature of this model is that circuits do not have distinguished inputs—since the learning algorithm seeks to predict the output behavior of value injection experiments that override the values on an arbitrary subset of wires, each wire is a potential input. Probabilistic circuits are closely related to Bayesian networks as well; we have chosen, however, to retain the conventions of the previous works.

A **probabilistic circuit**  $C$  of **size**  $n \geq 1$  has  $n$  **wires**, of which one is the distinguished **output wire**. We call the set of  $C$ 's wires  $W$ , and these wires take values in a finite **alphabet**  $\Sigma$  with  $|\Sigma| \geq 2$ . If  $\Sigma = \{0, 1\}$ , then  $C$  is **Boolean**. The value on a wire is ordinarily determined by the output of an associated probabilistic gate, whose distribution is a function of the values on other wires.

Formally, an **value distribution**  $D$  is a probability distribution over  $\Sigma$ , that is, a map from  $\Sigma$  to the real interval  $[0, 1]$  such that  $\sum_{\sigma \in \Sigma} D(\sigma) = 1$ . The **support** of  $D$  is the set of values  $\sigma \in \Sigma$  such that  $D(\sigma) > 0$ , and when this set is a singleton  $\{\sigma\}$  for some  $\sigma \in \Sigma$ , we say  $D$  is **deterministic**. For nonempty sets of values  $S \subseteq \Sigma$ , the **uniform distribution**  $U(S)$  is the distribution such that  $U(S)(\sigma) = |\sigma \in S|/|S|$ .

A  $k$ -ary **probabilistic gate function**  $f$  maps each  $k$ -tuple  $(\sigma_1, \dots, \sigma_k) \in \Sigma^k$  of values to a value distribution. When  $C$  is Boolean, we can specify  $f$  by a truth table giving the

expected value for each Boolean vector of inputs. A probabilistic gate function is **deterministic** if it maps  $k$ -tuples to deterministic value distributions only.

A **probabilistic gate**  $g$  of **fan-in**  $k$  pairs a  $k$ -ary probabilistic gate function  $f$  with a  $k$ -tuple  $(w_1, \dots, w_k) \in W^k$  of **input wires**.  $g$  is **deterministic** if  $f$  is deterministic. When  $k = 0$ , the gate  $g$  has no inputs, and we can regard it as a value distribution, or, when  $C$  is Boolean, a biased coin flip.

A **probabilistic circuit**  $C$  maps wires to probabilistic gates.  $C$  is **deterministic** if all of its gates are deterministic. The **fan-in** of  $C$  is the maximum fan-in over  $C$ 's gates. The **circuit graph** of  $C$  has nodes  $W$  and a directed edge  $(w, u)$  if  $w$  is one of the input wires of the gate associated with  $u$ . It is important to distinguish between wires in the circuit and edges in the circuit graph. For example, if wire  $w$  is an input of wires  $u$  and  $v$ , then there will be two directed edges,  $(w, u)$  and  $(w, v)$ , in the circuit graph.

Wire  $u$  is **reachable** from wire  $w$  if there is a directed path from  $w$  to  $u$  in the circuit graph. A wire is **relevant** if the output wire is reachable from it. The **depth** of a wire  $w$  is the number of edges in the longest simple path from  $w$  to the output wire in the circuit graph. The **depth** of the circuit is maximum depth of any relevant wire. The circuit is **acyclic** if the circuit graph contains no directed cycles. The circuit is **transitively reduced** if its circuit graph is transitively reduced, that is, if it contains no edge  $(w, u)$  such that there is a directed path of length at least two from  $w$  to  $u$ . In this paper we assume all circuits are acyclic.

### 2.2 Experiments

In an experiment some wires are constrained to be particular symbols or value distributions and the other wires are left free. The behavior of a circuit consists of its responses to all possible experiments. For probabilistic circuits we consider both value injection experiments and distribution injection experiments.

A **distribution injection experiment**  $e$  is a function with domain  $W$  that maps each wire  $w$  to a special symbol  $*$  or to a value distribution. A **value injection experiment**  $e$  is a distribution injection experiment for which every value distribution assigned is deterministic – that is, always generates the same symbol. To simplify notation, we think of a value injection experiment as a mapping from  $W$  to  $(\Sigma \cup \{*\})$ . If  $e$  is either kind of experiment, we say that  $e$  leaves  $w$  **free** if  $e(w) = *$ ; otherwise we say that  $e$  **constrains**  $w$  to  $e(w)$ . If  $e(w)$  is a single symbol, then we say  $e$  **fixes**  $w$  to  $e(w)$ .

We define a partial ordering  $\leq$  on the set containing  $*$  and all value distributions  $D$  as follows:  $D \leq *$  for every value distribution  $D$ , and for two value distributions,  $D_1 \leq D_2$  if the support of  $D_1$  is a subset of the support of  $D_2$ . This ordering is extended to experiments on the same set of wires  $W$  as follows:  $e_1 \leq e_2$  if for every  $w \in W$ ,  $e_1(w) \leq e_2(w)$ . The intuitive meaning of  $e_1 \leq e_2$  is that  $e_1$  is at least as constraining as  $e_2$  for every wire.

If  $e$  is any experiment,  $w$  is a wire, and  $a$  is  $*$  or an element of  $\Sigma$  or a value distribution, then the experiment  $e|_{w=a}$  is defined to be the experiment  $e'$  such that  $e'(w) = a$  and  $e'(u) = e(u)$  for all  $u \in W$  such that  $u \neq w$ .

### 2.3 Behavior

Let  $C$  be a probabilistic circuit. Then a distribution injection experiment  $e$  determines a joint distribution over assignments of elements of  $\Sigma$  to all of the wires of the circuit, as follows. If wire  $w$  is constrained then  $w$  is randomly and independently assigned a value in  $\Sigma$  drawn according to the value distribution  $e(w)$ ; in the case of a value injection experiment, this just assigns a fixed element of  $\sigma$  to  $w$ . If wire  $w$  is free, has probabilistic gate function  $f$  and its inputs  $u_1, \dots, u_k$  have been assigned the values  $\sigma_1, \dots, \sigma_k$ , then  $w$  is randomly and independently assigned a value from  $\Sigma$  according to the value distribution  $f(\sigma_1, \dots, \sigma_k)$ .

Constrained gates and gates of fan-in zero give the base cases for the above recursive definition, which assigns an element of  $\Sigma$  to every wire because the circuit is acyclic. Let  $C(e, w)$  denote the (marginal) value distribution of the assignments of values to  $w$  for the above process. The **output distribution** of the circuit, denoted  $C(e)$ , is the distribution  $C(e, z)$ , where  $z$  is the output wire of the circuit. The **behavior** of a circuit  $C$  is the function that maps value injection experiments  $e$  to output distributions  $C(e)$ .

### 2.4 Example: $C_1$

We give an example of a simple Boolean probabilistic circuit, which we will also refer to later. The 2-input **averaging gate**  $A(b_1, b_2)$  outputs 1 with probability  $(b_1 + b_2)/2$ . We define a circuit  $C_1$  of 4 wires as follows:  $w_4 = A(w_2, w_3)$ ,  $w_3 = w_1$ ,  $w_2 = w_1$ , and  $w_1 = U(\{0, 1\})$ . The output wire is  $w_4$ .  $C_1$  is depicted in Figure 1.

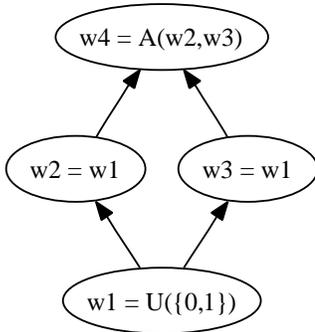


Figure 1: The circuit  $C_1$ ;  $w_4$  is the output wire.

To illustrate the behavior of this circuit, we consider two value injection experiments. Define the experiment  $e_1$  to leave every wire in  $C_1$  free, that is,  $e_1(w_i) = *$  for  $1 \leq i \leq 4$ . Given  $e_1$ , we construct one random outcome as follows. The wire  $w_1$  is assigned a value as the result of an unbiased coin flip – say it is assigned 0. Then the values assigned to  $w_2$  and  $w_3$  are determined because they are each the output of an identity gate with  $w_1$  as input: both are 0. Finally, because both its input wires have been assigned values,  $w_4$  can be assigned a value according to  $A(0, 0)$ , which is deterministically 0. It is easy to see that this is one of two possible outcomes for experiment  $e_1$ ; either all wires are assigned 0 or all wires are assigned 1, and these each occur with probability  $1/2$ . The output distribution  $C_1(e_1)$  is just an unbiased coin flip.

Now consider experiment  $e_2 = e_1|_{w_2=1}$  that fixes  $w_2$  to 1 and leaves the other wires free. Once again, the value of  $w_1$  is determined by a coin flip – say it is assigned 0. Since  $w_2$  is fixed to 1, that is its assignment. Wire  $w_3$  is free, and is therefore assigned the value of  $w_1$ , that is 0. Now the inputs of  $w_4$  have been assigned values, so we consider  $A(1, 0)$ , which randomly and equiprobably selects 0 or 1. If, instead, the coin flip for  $w_1$  had returned 1, all wires would be assigned 1. There are three possible assignments to  $(w_1, w_2, w_3, w_4)$  for experiment  $e_2$ :  $(1, 1, 1, 1)$  with probability  $1/2$ ,  $(0, 1, 0, 0)$  with probability  $1/4$  and  $(0, 1, 0, 1)$  with probability  $1/4$ . The output distribution  $C_1(e_2)$  is a biased coin flip that is 1 with probability  $3/4$ .

### 2.5 Behavioral Equivalence

Two circuits  $C$  and  $C'$  are **behaviorally equivalent** if they have the same set of wires, the same output wire and the same behavior, that is, for every value injection experiment  $e$ ,  $C(e) = C'(e)$ . We also need a concept of approximate equivalence. The **(statistical) distance** between value distributions  $D$  and  $D'$  is  $d(D, D') = (1/2) \sum_{\sigma} |D(\sigma) - D'(\sigma)|$ , which takes values in  $[0, 1]$ . Note that when  $D$  and  $D'$  are deterministic,  $d(D, D')$  is 0 if  $D = D'$  and 1 otherwise. For  $\varepsilon \geq 0$ ,  $C$  is  $\varepsilon$ -**behaviorally equivalent** to  $C'$  if they contain the same wires and the same output wire, and for every value injection experiment  $e$ ,  $d(C(e), C'(e)) \leq \varepsilon$ , where  $d$  is the distance between value distributions defined above.

In Lemma 2 we show that the behavioral equivalence of  $C$  and  $C'$  implies  $C(e) = C'(e)$  for all distribution injection experiments as well. Note that even when all the gates are Boolean, deterministic and relevant, the circuit graph of the target circuit may not be uniquely determined by its behavior [AACW06].

### 2.6 Queries

The learning algorithm gets information about the target circuit by specifying a value injection experiment  $e$  and observing the element of  $\Sigma$  assigned to the output wire. Such an action is termed a **value injection query**, abbreviated VIQ. A value injection query does not return complete information about the value distribution  $C(e)$ , but instead returns an element of  $\Sigma$  selected according to the distribution  $C(e)$ . Thus, in order to approximate the distribution  $C(e)$ , the learner must repeatedly make value injection queries with experiment  $e$ . In this case, the goal of learning is approximate behavioral equivalence.

### 2.7 The Learning Problem

The learning problem is  $\varepsilon$ -**approximate learning**: by making value injection queries to a target circuit  $C$  drawn from a known class of probabilistic circuits, find a circuit  $C'$  that is  $\varepsilon$ -behaviorally equivalent to  $C$ . The inputs to the learning algorithm are the names of the wires in  $C$ , the name of the output wire and positive numbers  $\varepsilon$  and  $\delta$ , where the learning algorithm is required to succeed with probability at least  $(1 - \delta)$ .

## 3 Preliminary Results

In this section we establish some basic results about probabilistic circuits. We first note that if  $C$  is a probabilistic cir-

cuit,  $e$  is a distribution injection experiment and either  $e(w)$  is a value distribution or  $e$  deterministically fixes all the input wires of  $w$ , then there is a value distribution  $D$  such that the value of  $w$  in  $C(e)$  is determined by a random choice according to  $D$ , independent of the values chosen for any other wires. We make systematic use of this observation to reduce the number of experiments under consideration.

**Lemma 1** *Let  $C_1$  and  $C_2$  be probabilistic circuits on wires  $W$  with the same output wire, let  $w \in W$  be a wire, let  $D$  be a value distribution, and let  $e_1$  and  $e_2$  be distribution injection experiments such that  $e_1(w) = e_2(w) = D$ . Then there exists a value  $\sigma \in \text{support}(D)$  such that*

$$d(C_1(e_1|_{w=\sigma}), C_2(e_2|_{w=\sigma})) \geq d(C_1(e_1), C_2(e_2)).$$

**Proof:** We have

$$\begin{aligned} & d(C_1(e_1), C_2(e_2)) \\ &= \frac{1}{2} \sum_{\tau \in \Sigma} \left| C_1(e_1)(\tau) - C_2(e_2)(\tau) \right| \\ &= \frac{1}{2} \sum_{\tau \in \Sigma} \left| \sum_{\rho \in \Sigma} C_1(e_1|_{w=\rho})(\tau) D(\rho) \right. \\ &\quad \left. - \sum_{\rho \in \Sigma} C_2(e_2|_{w=\rho})(\tau) D(\rho) \right| \\ &\leq \frac{1}{2} \sum_{\rho \in \Sigma} D(\rho) \sum_{\tau \in \Sigma} \left| C_1(e_1|_{w=\rho})(\tau) \right. \\ &\quad \left. - C_2(e_2|_{w=\rho})(\tau) \right| \\ &= \sum_{\rho \in \Sigma} D(\rho) d(C(e_1|_{w=\rho}), C(e_2|_{w=\rho})) \end{aligned}$$

by the triangle inequality. Let

$$\sigma = \arg \max_{\rho \in \text{support}(D)} d(C(e_1|_{w=\rho}), C(e_2|_{w=\rho})),$$

so that

$$d(C(e_1|_{w=\sigma}), C(e_2|_{w=\sigma})) \geq d(C(e_1), C(e_2))$$

by an averaging argument. ■

**Lemma 2** *Let  $C_1$  and  $C_2$  be probabilistic circuits on wires  $W$  with the same output wire and let  $e$  be a distribution injection experiment. Then there exists a value injection experiment  $e' \leq e$  such that*

$$d(C_1(e'), C_2(e')) \geq d(C_1(e), C_2(e)).$$

**Proof:** By induction on  $|V|$ , where  $V \subseteq W$  is the set of wires that  $e$  constrains to nonconstant distributions. If  $|V| > 0$ , then let  $w \in V$ . By Lemma 1, there exists a value  $\sigma \in \Sigma$  such that

$$d(C_1(e|_{w=\sigma}), C_2(e|_{w=\sigma})) \geq d(C_1(e), C_2(e)).$$

Since  $e|_{w=\sigma}$  constrains one fewer wire to a nonconstant distribution, the existence of  $e'$  follows from the inductive hypothesis. ■

**Corollary 3** *If circuits  $C_1$  and  $C_2$  are  $\varepsilon$ -behaviorally equivalent with respect to value injection experiments, then  $C_1$  and  $C_2$  are  $\varepsilon$ -behaviorally equivalent with respect to distribution injection experiments.*

Suppose that  $C$  is a probabilistic circuit and  $e_1$  and  $e_2$  are distribution injection experiments. For each wire  $w$ , we say that  $e_1$  and  $e_2$  **agree** on  $w$  if either

- $e_1$  and  $e_2$  constrain  $w$  to the same distribution, or
- $w$  is free in  $e_1$  and  $e_2$ , and  $e_1$  and  $e_2$  agree on all of  $w$ 's inputs.

If  $e_1$  and  $e_2$  agree on a wire  $w$ , then the marginal distributions of  $w$  in  $e_1$  and  $e_2$  are identical, that is,  $C(e_1, w) = C(e_2, w)$ .

**Lemma 4** *Let  $C$  be a probabilistic circuit on wires  $W$  and let  $e_1$  and  $e_2$  be distribution injection experiments that agree on wires  $V \subseteq W$ . Then there exist distribution injection experiments  $e'_1 \leq e_1$  and  $e'_2 \leq e_2$  such that for each wire  $w \in V$ , there exists a value  $\sigma \in \Sigma$  such that  $e'_1(w) = e'_2(w) = \sigma$ , and*

$$d(C(e'_1), C(e'_2)) \geq d(C(e_1), C(e_2)).$$

**Proof:** By induction on the number of unfixed wires  $w \in V$ . If there is such a wire, choose  $v$  to be one that is not reachable from the others. If  $e_1(v) = e_2(v) = *$ , then  $e_1$  and  $e_2$  agree on all of  $v$ 's inputs, and by the choice of  $v$ , all of  $v$ 's inputs are fixed. As such, we may assume without loss of generality that  $e_1$  and  $e_2$  in fact constrain  $v$  to the distribution  $D = C(e_1, v) = C(e_2, v)$ . By Lemma 1, there exists a value  $\sigma \in \text{support}(D)$  such that

$$d(C(e_1|_{v=\sigma}), C(e_2|_{v=\sigma})) \geq d(C(e_1), C(e_2)).$$

The existence of  $e'_1$  and  $e'_2$  follows from the inductive hypothesis. ■

**Lemma 5** *Let  $C$  be a probabilistic circuit on wires  $W$ , let  $e$  be a distribution injection experiment, let  $w \in W$  be a wire free in  $e$ , and let  $D$  be a value distribution. Then  $e$  and  $e|_{w=D}$  agree on all wires  $u \in W$  to which there is no path on free wires from  $w$ .*

**Proof:** If  $u$  is constrained, then the conclusion follows. Otherwise, since  $u$  is free and has no free path from  $w$ , none of  $u$ 's inputs have free paths from  $w$ . We proceed by induction on the length of the longest path to  $u$ . If this length is zero, then  $u$  does not have any inputs. Otherwise, the inductive hypothesis applies to all of  $u$ 's inputs, on which  $e$  and  $e|_{w=D}$  then must agree. It follows that they also agree on  $u$ . ■

**Lemma 6** *Let  $C$  be a probabilistic circuit on wires  $W$ , let  $w \in W$  be a wire, and let  $D_1, D_2$  be value distributions. There exist value distributions  $D'_1, D'_2$  with  $\text{support}(D'_1) \cap \text{support}(D'_2) = \emptyset$  such that for all experiments  $e$ ,*

$$\begin{aligned} & d(C(e|_{w=D_1}), C(e|_{w=D_2})) \\ &= d(D_1, D_2) d(C(e|_{w=D'_1}), C(e|_{w=D'_2})). \end{aligned}$$

**Proof:** We have

$$\begin{aligned} d(C(e|_{w=D_1}), C(e|_{w=D_2})) &= \frac{1}{2} \sum_{\sigma \in \Sigma} \left| C(e|_{w=D_1})(\sigma) - C(e|_{w=D_2})(\sigma) \right| \\ &= \frac{1}{2} \sum_{\sigma \in \Sigma} \left| \sum_{\tau \in \Sigma} C(e|_{w=\tau})(\sigma) (D_1(\tau) - D_2(\tau)) \right|. \end{aligned}$$

If we let

$$\begin{aligned} \widehat{D}_1(\tau) &= D_1(\tau) - \min(D_1(\tau), D_2(\tau)) \\ \widehat{D}_2(\tau) &= D_2(\tau) - \min(D_1(\tau), D_2(\tau)), \end{aligned}$$

then

$$\begin{aligned} d(C(e|_{w=D_1}), C(e|_{w=D_2})) &= \frac{1}{2} \sum_{\sigma \in \Sigma} \left| \sum_{\tau \in \Sigma} C(e|_{w=\tau})(\sigma) (\widehat{D}_1(\tau) - \widehat{D}_2(\tau)) \right|. \end{aligned}$$

Since  $\sum_{\tau \in \Sigma} \widehat{D}_1(\tau) = 1 - \sum_{\tau \in \Sigma} \min(D_1(\tau), D_2(\tau))$  and likewise for  $D_2$ ,

$$\begin{aligned} d(D_1, D_2) &= \frac{1}{2} \sum_{\tau \in \Sigma} \left| D_1(\tau) - D_2(\tau) \right| \\ &= \frac{1}{2} \sum_{\tau \in \Sigma} \left| \widehat{D}_1(\tau) - \widehat{D}_2(\tau) \right| \\ &= \sum_{\tau \in \Sigma} \widehat{D}_1(\tau) = \sum_{\tau \in \Sigma} \widehat{D}_2(\tau). \end{aligned}$$

If  $d(D_1, D_2) > 0$ , then the distributions  $D'_1$  and  $D'_2$  where

$$\begin{aligned} D'_1(\tau) &= \widehat{D}_1(\tau) / d(D_1, D_2) \\ D'_2(\tau) &= \widehat{D}_2(\tau) / d(D_1, D_2) \end{aligned}$$

satisfy the requisite properties. Otherwise, any two distributions with disjoint support will do. ■

## 4 Test Paths

The concept of a test path has been central in previous work on learning deterministic circuits by means of value injection queries [AACR07, AACW06]. A **test path** for a wire  $w$  is a value injection experiment in which the free gates form a directed path in the circuit graph from  $w$  to the output wire. All the other wires in the circuit are fixed; this includes the inputs of  $w$ . A **side wire** with respect to a test path  $p$  is a wire fixed by  $p$  that is input to a free wire in  $p$ . A test path may help the learning algorithm determine the effects of assigning different values to the wire  $w$ . The test-path lemmas from [AACR07, AACW06] may be re-stated as follows.

**Lemma 7** *Let  $C$  be a deterministic circuit. If for some value injection experiment  $e$ , wire  $w$  and alphabet symbols  $\sigma$  and  $\tau$  it is the case that*

$$C(p|_{w=\sigma}) = C(p|_{w=\tau})$$

for every test path  $p \leq e$  then also

$$C(e|_{w=\sigma}) = C(e|_{w=\tau}).$$

Nontrivial complications arise in attempting to carry over this test path lemma to general probabilistic circuits, as we now show. The following lemma shows that for alphabets of size at least four, there are transitively reduced probabilistic circuits for which the test-path lemma fails completely. (A less intuitive version of this construction shows that this phenomenon occurs also at alphabet size three.)

**Lemma 8** *If  $|\Sigma| = 4$ , there exists a probabilistic circuit  $C$ , value injection experiment  $e$ , wire  $w$  and alphabet symbols  $\sigma$  and  $\tau$  such that although for every test path  $p \leq e$  for  $w$ ,  $d(C(p|_{w=\sigma}), C(p|_{w=\tau})) = 0$ , it is nevertheless the case that  $d(C(e|_{w=\sigma}), C(e|_{w=\tau})) = 1$ .*

**Proof:** Assume  $\Sigma = \{00, 01, 10, 11\}$ , and define probabilistic gate functions  $T, L, R$ , and  $X$  as follows.

$$\begin{aligned} T(00) = T(11) &= U(\{00, 11\}), \\ T(01) = T(10) &= U(\{01, 10\}), \\ L(00) = L(01) &= 00, \\ L(10) = L(11) &= 01, \\ R(00) = R(10) &= 00, \\ R(01) = R(11) &= 01, \end{aligned}$$

and  $X(ab, cd) = 0(b \oplus d)$ , where  $\oplus$  is sum modulo 2.

The circuit  $C$  has 5 wires, connected as in Figure 2. The output wire is  $w_5$ ; note that  $C$  is transitively reduced.

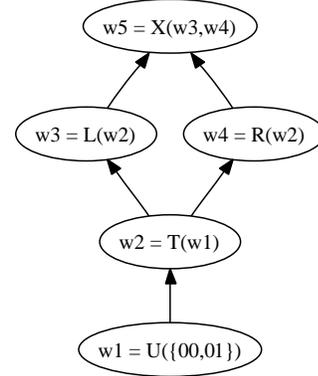


Figure 2: The circuit  $C$ ;  $w_5$  is the output wire.

Consider the experiment  $e$  that leaves all the wires free. We have  $C(e|_{w_1=00}) = 00$  and  $C(e|_{w_1=01}) = 01$ , and thus  $d(C(e|_{w_1=00}), C(e|_{w_1=01})) = 1$ . However, the only test paths for  $w_1$  fix  $w_3$  and leave all other wires free or fix  $w_4$  and leave all other wires free. Calculation verifies that fixing  $w_3$  or  $w_4$  to any value and leaving the other wires free yields the output distribution  $U(\{00, 01\})$  regardless of whether  $w_1$  is fixed to 00 or 01. Thus, for every test path  $p$  for  $w_1$ , we have  $d(C(p|_{w_1=00}), C(p|_{w_1=01})) = 0$ . ■

### 4.1 A Bound for Boolean Probabilistic Circuits

Surprisingly, for Boolean probabilistic circuits there is a useful quantitative relationship between the differences exposed by test paths and the differences exposed by arbitrary experiments.

Let  $e$  be an experiment and  $w$  a wire. Define  $\Pi(e, w)$  to be the set of all directed paths from  $w$  to the output wire on free wires in  $e$ . Let  $S(e)$  be the set of wires that originate a free shortcut, that is, the set of free wires  $w$  such that there exists a path  $p \in \Pi(e, w)$  with two free wires to which  $w$  is an input. Define

$$\kappa(e, w) = \sum_{p \in \Pi(e, w)} 2^{|p \cap S(e)|}.$$

**Lemma 9** *Let  $C$  be a probabilistic circuit,  $e$  be a distribution injection experiment,  $w$  and  $u$  be free wires where  $w$  is an input to  $u$ , and  $D_0$  be a value distribution. Let  $\beta = 2$  if  $w \in S(e)$  and  $\beta = 1$  otherwise. Then*

$$\kappa(e, w) = \kappa(e|_{u=D_0}, w) + \kappa(e|_{w=1}, u) \cdot \beta.$$

**Proof:** The first term of the sum counts paths that don't contain  $u$ , and the second counts paths that do. Let  $e' = e|_{u=D_0}$  and  $e'' = e|_{w=1}$ . We have

$$\begin{aligned} \kappa(e, w) &= \sum_{p \in \Pi(e, w)} 2^{|p \cap S(e)|} \\ &= \sum_{\substack{p \in \Pi(e, w) \\ u \notin p}} 2^{|p \cap S(e)|} + \sum_{\substack{p \in \Pi(e, w) \\ u \in p}} 2^{|p \cap S(e)|} \\ &= \sum_{p \in \Pi(e', w)} 2^{|p \cap S(e')|} + \sum_{p \in \Pi(e'', u)} 2^{|p \cap S(e'')|} \beta \\ &= \kappa(e', w) + \kappa(e'', u) \cdot \beta, \end{aligned}$$

since each path  $p \ni u$  from  $w$  corresponds to the path  $p \setminus \{w\}$  from  $u$ . ■

**Lemma 10** *Let  $C$  be a Boolean probabilistic circuit,  $e$  be a distribution injection experiment,  $w$  be a wire, and  $D_1, D_2$  be value distributions. If there exists  $\varepsilon \geq 0$  such that for all  $w$ -test paths  $p \leq e$ ,*

$$d(C(p|_{w=D_1}), C(p|_{w=D_2})) \leq \varepsilon,$$

then

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) \leq \kappa(e, w) \cdot \varepsilon.$$

**Proof:** By induction on  $\phi(e)$ , the number of free wires in  $e$ . By Lemma 6, assume that  $\text{support}(D_1) \cap \text{support}(D_2) = \emptyset$ . The critical feature of the Boolean case is that it follows that  $D_1 = 0$  and  $D_2 = 1$  without loss of generality—it is important to the following proof that  $D_1$  and  $D_2$  be deterministic.

If  $\phi(e) = 1$ , then either

$$d(C(e|_{w=0}), C(e|_{w=1})) = 0,$$

or  $w$  is the output,  $e$  is a  $w$ -test path, and  $\kappa(e, w) = 1$ . Otherwise, the inductive hypothesis is that the lemma holds for all experiments  $e'$  with  $\phi(e') < \phi(e)$ .

Except for  $w$ , the experiments  $e|_{w=0}$  and  $e|_{w=1}$  agree on all constrained wires, so by Lemmas 4 and 5, assume without loss of generality that every wire with no free path from  $w$  is in fact fixed. Since  $C$  is acyclic, there exists a free wire  $u \neq w$  whose only unfixed input is  $w$ . Let  $g$  be the gate assigned

by  $C$  to  $u$  and let  $B_0 = g(e|_{w=0})$  and  $B_1 = g(e|_{w=1})$ , so that

$$\begin{aligned} C(e|_{w=0}) &= C(e|_{w=0, u=B_0}) \\ C(e|_{w=1}) &= C(e|_{w=1, u=B_1}). \end{aligned}$$

By the triangle inequality,

$$\begin{aligned} d(C(e|_{w=0}), C(e|_{w=1})) &\leq d(C(e|_{w=0, u=B_0}), C(e|_{w=1, u=B_0})) \\ &\quad + d(C(e|_{w=1, u=B_0}), C(e|_{w=1, u=B_1})). \end{aligned}$$

The inductive hypothesis bounds the first term of the sum by  $\kappa(e', w) \cdot \varepsilon$ , where  $e' = e|_{u=B_0}$ . We now derive a bound on  $u$ -test paths so that the inductive hypothesis applies to the second term as well. Let  $\beta = 2$  if  $w \in S(e)$  and  $\beta = 1$  otherwise. Let  $e'' = e|_{w=1}$  and suppose  $p \leq e''$  is a  $u$ -test path. Then

$$\begin{aligned} d(C(p|_{u=B_0}), C(p|_{u=B_1})) &\leq d(C(p|_{w=1, u=B_0}), C(p|_{w=0, u=B_0})) \\ &\quad + d(C(p|_{w=0, u=B_0}), C(p|_{w=1, u=B_1})) \\ &= d(C(p|_{w=0, u=B_0}), C(p|_{w=1, u=B_0})) \\ &\quad + d(C(p|_{w=0, u=*}), C(p|_{w=1, u=*})) \\ &\leq \beta \varepsilon, \end{aligned}$$

since both terms of the sum are bounded by  $\varepsilon$ , and the first is nonzero only if  $w$  is an input to some free wire in  $p$  other than  $u$ . Thus

$$d(C(e''|_{u=0}), C(e''|_{u=1})) \leq \kappa(e'', u) \cdot \beta \varepsilon,$$

and,

$$\begin{aligned} d(C(e|_{w=0}), C(e|_{w=1})) &\leq \kappa(e', w) \cdot \varepsilon + \kappa(e'', u) \cdot \beta \varepsilon \\ &= \kappa(e, w) \cdot \varepsilon, \end{aligned}$$

by Lemma 9. ■

In the case of transitively reduced circuits,  $S(e) = \emptyset$ , and  $\kappa(e, w) = \pi(e, w)$ , where  $\pi(e, w) = |\Pi(e, w)|$ , the number of directed paths on free wires in  $e$  from  $w$  to the output wire.

**Corollary 11** *Let  $C$  be a transitively reduced Boolean probabilistic circuit,  $e$  be a distribution injection experiment, and  $w$  be a wire. If there exists  $\varepsilon \geq 0$  such that for all  $w$ -test paths  $p \leq e$ ,*

$$d(C(p|_{w=0}), C(p|_{w=1})) \leq \varepsilon,$$

then

$$d(C(e|_{w=0}), C(e|_{w=1})) \leq \pi(e, w) \cdot \varepsilon.$$

## 5 Learning Boolean Probabilistic Circuits

The amount of attenuation given by Lemma 10 allows us to adapt the CircuitBuilder algorithm [AACW06] to learn Boolean probabilistic circuits with constant fan-in and log depth in polynomial time.

**Theorem 12** *Given constants  $c$  and  $k$  there is a nonadaptive learning algorithm that with probability at least  $(1 - \delta)$  successfully  $\varepsilon$ -approximately learns any Boolean probabilistic circuit with  $n$  wires, gates of fan-in at most  $k$  and depth at most  $c \log n$  using value injection queries in time bounded by a polynomial in  $n$ ,  $1/\varepsilon$  and  $\log(1/\delta)$ .*

We adapt the Circuit Builder algorithm from [AACW06] to prove Theorem 12 and call the resulting algorithm Probabilistic Circuit Builder (PCB). The algorithm constructs a set  $U$  of experiments such that every test path is equivalent to some experiment in  $U$ , obtains a sufficiently good estimate of the output distribution for each experiment in  $U$ , and then builds a circuit approximately behaviorally equivalent to the target circuit by repeatedly adding sufficiently accurate gates all of whose inputs are in the partially constructed circuit.

Let the target circuit be  $C$  and let positive constants  $\delta$ ,  $\varepsilon$ ,  $k$  and  $c$  be given such that the fan-in of  $C$  is bounded by  $k$  and the depth of  $C$  is bounded by  $c \log n$ . For such a circuit,  $\pi(e, w)$  is bounded above by  $k^{c \log n}$ , so the quantity  $\kappa(e, w)$  is bounded above by

$$\kappa(n) = k^{c \log n} \cdot 2^{c \log n} = n^{c(\log k + 1)} = n^{O(1)}.$$

The PCB algorithm is nonadaptive: it computes a set  $U$  of value injection experiments, repeats each value injection query for  $e \in U$  sufficiently many times to estimate the expected value of  $C(e)$  with enough accuracy, and then uses the results of the queries to build a circuit  $C'$  that is  $\varepsilon$ -behaviorally equivalent to  $C$ .

In choosing the experiments  $U$ , the goal is that for every potential test path,  $U$  includes an equivalent experiment. The structure of the circuit, however, is not known *a priori*, a difficulty that we overcome by the same method as [AACW06]. Let  $U_*$  be a universal set of value injection experiments such that for every set of  $kc \log n$  wires and every assignment of symbols from  $\Sigma \cup \{*\}$  to those wires, some experiment  $e \in U_*$  agrees with the values assigned to those wires. As in [AACW06], it is possible to construct such a set  $U$  of size

$$2^{O(kc \log n)} \log n = n^{O(kc)}$$

in time polynomial in its size.

For every wire  $w$  and test path  $p$  for  $w$ , there is an experiment in  $U_*$  that leaves the path wires of  $p$  free and fixes the side wires of  $p$  to their values in  $p$ . Consequently,  $p$  and this experiment agree on the output wire. Although it is tempting now to set  $U = U_*$ , there is no easy way to determine which experiment a test path corresponds to, making it difficult for PCB to perform comparisons where  $w$  is fixed to different values. For  $b = 0, 1$ , then, let  $U_b$  contain every experiment  $e|_{w=b}$  such that  $e \in U_*$  and  $w$  is free in  $e$ . Now we can take  $U = U_* \cup U_0 \cup U_1$ .

For each  $e \in U$ , PCB repeatedly makes a value injection query with  $e$  to estimate the distribution of  $C(e)$ . By Hoeffding's bound, we have that

$$m = O((n\kappa(n)/\varepsilon)^2 \log(|U|/\delta))$$

trials per experiment  $e$  suffice to guarantee that with probability at least  $1 - \delta$ , for all  $e \in U$ ,

$$d(C(e), \widehat{C}(e)) \leq \varepsilon/(5n\kappa(n)). \quad (1)$$

If (1) holds, then we can compute good estimates for a class of distribution experiments. Let  $e \in U_*$  be a value injection experiment,  $w$  be a wire that  $e$  leaves free, and  $D$  be a value distribution. Then let

$$\widehat{C}(e|_{w=D}) = \sum_{\sigma \in \Sigma} D(\sigma) \widehat{C}(e|_{w=\sigma}).$$

We have

$$\begin{aligned} d(C(e|_{w=D}), \widehat{C}(e|_{w=D})) & \\ & \leq \sum_{\sigma \in \Sigma} D(\sigma) d(C(e|_{w=\sigma}), \widehat{C}(e|_{w=\sigma})) \\ & \leq \varepsilon/(5n\kappa(n)). \end{aligned}$$

From this point on, we assume that the estimates are correct and show that PCB successfully builds a circuit  $C'$  that is  $\varepsilon$ -behaviorally equivalent to  $C$ .

PCB builds the circuit  $C'$  one gate at a time. Initially  $C'$  has no gates assigned to wires. The algorithm tries repeatedly to find a wire  $w$  and a gate  $g$  such that  $g$  is  $\varepsilon/n$ -correct for  $w$  in  $C$  and all of  $g$ 's inputs are in  $C'$ . When this is no longer possible, PCB outputs  $C'$  and halts.

To prove the correctness of PCB, we first establish two lemmas connecting gates, paths and experiments. Given a Boolean probabilistic circuit  $C$  and a probabilistic gate  $g$ ,  $g$  is  $\eta$ -correct for wire  $w$  with respect to  $C$  if for every value injection experiment  $e$  that fixes the input wires for  $g$  we have  $d(C(e), C(e|_{w=g(e)})) \leq \eta$ , where  $g(e)$  denotes the coin flip determined by  $g$  when its inputs are fixed as in  $e$ . Recall that  $\phi(e)$  denotes the number of free wires in experiment  $e$ .

**Lemma 13** *Let  $C$  and  $C'$  be probabilistic circuits on wires  $W$ , and let  $e$  be a distribution injection experiment. If for every wire  $w$ , the gate  $g$  for  $w$  in  $C'$  is  $\eta$ -correct for  $w$  with respect to  $C$ , then*

$$d(C(e), C'(e)) \leq \phi(e) \cdot \eta.$$

**Proof:** By induction on  $\phi(e)$ , the number of free wires in  $e$ . If  $\phi(e) = 0$ , then  $e$  constrains the output wire, and trivially,  $d(C(e), C'(e)) = 0$ . Otherwise, the inductive hypothesis is that  $C$  and  $C'$  are  $\eta$ -behaviorally equivalent with respect to all experiments with fewer free gates.

By Lemma 2, assume that  $e$  is in fact a value injection experiment. Since  $C'$  is acyclic, there exists a free wire  $w$  in  $e$  such that the inputs to  $w$  in  $C'$  are fixed in  $e$  to some  $k$ -tuple  $(\sigma_1, \dots, \sigma_k) \in \Sigma^k$ . Letting  $f$  be the probabilistic gate function for  $w$  in  $C'$ , we have  $C'(e) = C'(e|_{w=f(\sigma_1, \dots, \sigma_k)})$ , and

$$\begin{aligned} d(C(e), C'(e)) & \\ & \leq d(C(e), C(e|_{w=f(\sigma_1, \dots, \sigma_k)})) \\ & \quad + d(C(e|_{w=f(\sigma_1, \dots, \sigma_k)}), C'(e|_{w=f(\sigma_1, \dots, \sigma_k)})) \\ & \leq \eta + (\phi(e) - 1) \cdot \eta = \phi(e) \cdot \eta \end{aligned}$$

by the fact that  $f$  is  $\eta$ -correct and the inductive hypothesis. ■

Next we show that test paths are sufficient to determine whether a gate is  $\eta$ -correct for a wire in  $C$ .

**Lemma 14** *Let  $C$  be a Boolean probabilistic circuit,  $w$  a wire and  $g'$  a probabilistic gate. If for every test path  $p$  for  $w$  that fixes all the inputs of  $g'$ ,  $d(C(p), C(p|_{w=g'(p)})) \leq \eta/\kappa(C)$ , where  $\kappa(C)$  is the maximum value of  $\kappa_{C'}(e, w)$  over all circuits  $C'$  with the same set of wires, all experiments  $e$ , and all wires  $w$ , then  $g'$  is  $\eta$ -correct for  $w$  with respect to  $C$ .*

**Proof:** Let  $g$  be the actual gate that  $C$  assigns to  $w$ . Let  $e$  be a value injection experiment that fixes every input of  $g'$ .  $e$  may not fix all of  $g$ 's inputs, but since  $C$  is acyclic,  $g$ 's inputs are not reachable from  $w$ . By Lemmas 4 and 5, there exists an experiment  $e' \leq e$  that fixes  $g$ 's inputs, with

$$d(C(e'), C(e'|_{w=g'(e')})) \geq d(C(e), C(e|_{w=g'(e)})).$$

Since  $e'$  fixes all of  $g$ 's inputs,  $C(e') = C(e'|_{w=g'(e')})$ . It is given that for all test paths  $p$  that fix all inputs of  $g$  and  $g'$  that

$$d(C(p|_{w=g(p)}), C(p|_{w=g'(p)})) \leq \eta/\kappa(C),$$

so it follows by Lemma 10 that

$$\begin{aligned} d(C(e'|_{w=g(e')}), C(e'|_{w=g'(e')})) \\ \leq \kappa(e', w) \cdot \eta/\kappa(C) \\ \leq \eta, \end{aligned}$$

and  $g'$  is  $\eta$ -correct for  $w$ .  $\blacksquare$

To prove the correctness of PCB, we argue as follows. Let  $V$  be the set of wires to which  $C'$  does not assign a gate. Then since  $C$  is acyclic, there is some wire  $w \in V$  such that none of  $w$ 's inputs in  $C$  belong to  $V$ . PCB looks for a gate  $g'$  such that for each experiment  $e \in U_*$  that leaves  $w$  free and fixes all inputs of  $g'$ ,

$$d(\widehat{C}(e), \widehat{C}(e|_{w=g'(e)})) \leq 3\varepsilon/(5n\kappa(n)). \quad (2)$$

Then

$$d(C(e), \widehat{C}(e)) \leq \varepsilon/(5n\kappa(n))$$

$$d(\widehat{C}(e|_{w=g'(e)}), C(e|_{w=g'(e)})) \leq \varepsilon/(5n\kappa(n)),$$

and

$$d(C(e|_{w=g'(e)}), C(e|_{w=g(e)})) \leq \varepsilon/(n\kappa(n))$$

by (1) and the triangle inequality. It follows by Lemma 14 that  $g'$  is  $\varepsilon/n$ -correct for  $w$  in  $C$ . Let  $g$  be the gate that  $C$  assigns to  $w$  and suppose that  $d(g(e), g'(e)) \leq \varepsilon/(5n\kappa(n))$  for all experiments  $e$  that fix  $g$ 's inputs. Then

$$d(\widehat{C}(e), C(e)) \leq \varepsilon/(5n\kappa(n))$$

$$d(C(e), C(e|_{w=g(e)})) = 0$$

$$d(C(e|_{w=g(e)}), C(e|_{w=g'(e)})) \leq \varepsilon/(5n\kappa(n))$$

$$d(C(e|_{w=g'(e)}), \widehat{C}(e|_{w=g'(e)})) \leq \varepsilon/(5n\kappa(n))$$

and  $g'$  satisfies (2). Therefore, PCB will continue to make progress.

To bound the running time of PCB we argue as follows. The set  $U$  of experiments is of cardinality  $n^{O(kc)}$  and can be constructed in time polynomial in its size. Each experiment in  $U$  is repeated

$$O((n\kappa(n)/\varepsilon)^2 \log(|U|/\delta))$$

times; recall that  $\kappa(n) = O(n^{c(\log k+1)})$ . PCB chooses a gate for a wire  $n$  times. Each gate it tests must be subjected to a polynomial number of experiments; in order to be assured of a sufficiently good approximation, it must iterate over  $O(n^k)$  sets of inputs times  $|\Sigma|^k$  entries times a polynomial number of points in  $[0, 1]^\Sigma$  to be assured of finding a sufficiently good approximation to a true gate. Thus the running time of PCB is polynomial in  $n$ ,  $1/\varepsilon$  and  $1/\delta$ .

## 6 Lower Bounds

We consider lower bounds on the path attenuation factors for Boolean probabilistic circuits. The following lemma shows that the bound of  $\pi(e, w)$  for transitively reduced Boolean probabilistic circuits in Corollary 11 is tight infinitely often.

**Lemma 15** *There is an infinite set of transitively reduced probabilistic Boolean circuits such that for each circuit  $C$  in the family, there exists a value injection experiment  $e$  and a wire  $w$  such that*

$$d(C(e|_{w=0}), C(e_{w=1})) = 1$$

and for every test path  $p$  for  $w$  we have

$$d(C(p|_{w=0}), C(p|_{w=1})) = 1/\pi(e, w).$$

**Proof:** For each positive integer  $\ell$ , define the circuit  $C_\ell$  to be a chain of  $\ell$  copies of the circuit  $C_1$  in Figure 1 with wire  $w_4$  of one copy identified with wire  $w_1$  of the next copy. More formally, the  $3d + 1$  wires are  $w_{0,4}$  and  $w_{i,j}$  for  $i = 1, \dots, d$  and  $j = 2, 3, 4$ . The output wire is  $w_{d,4}$ . The wire  $w_{0,4}$  has no inputs and is determined by an unbiased coin flip, that is,  $U(\{0, 1\})$ . The wires  $w_{i,2}$  and  $w_{i,3}$  are the outputs of deterministic identity gates with input  $w_{i-1,4}$ . The wire  $w_{i,4} = A(w_{i,2}, w_{i,3})$  is the result of applying the two-input averaging gate  $A$  to the wires  $w_{i,2}$  and  $w_{i,3}$ .

The experiment  $e$  leaves all of the wires free. Let  $w$  denote the wire  $w_{0,4}$ . Clearly there are  $2^\ell$  paths on free gates in  $e$  from  $w$  to the output gate, that is,  $\pi(w, e) = 2^\ell$ . For experiment  $e$  we have  $C(e|_{w=0}) = 0$  and  $C(e|_{w=1}) = 1$ , so  $d(C(e|_{w=0}), C(e|_{w=1})) = 1$ . However, any test path  $p$  for  $w$  must fix one of the wires  $w_{i,2}$  or  $w_{i,3}$  for each  $i = 1, \dots, d$ . As the signal proceeds through each level, it is attenuated by  $1/2$ , so the final result for any test path  $p$  for  $w$  is  $d(C(p|_{w=0}), C(p|_{w=1})) = 1/2^\ell = 1/\pi(e, w)$ .  $\blacksquare$

A generalization of this construction shows that for any transitively reduced circuit graph, there is an assignment of Boolean probabilistic functions that matches the attenuation factor of  $\pi(e, w)$ .

**Lemma 16** *Let  $G$  be a transitively reduced directed graph with a designated output node in which there is a path from every node to the output node. There is a Boolean probabilistic circuit  $C$  whose circuit graph is  $G$  such that for every value injection experiment  $e$  and for every test path  $p \leq e$  and every wire  $w$ ,*

$$\begin{aligned} d(C(e|_{w=1}), C(e|_{w=0})) \\ \geq \pi(e, w) \cdot d(C(p|_{w=1}), C(p|_{w=0})). \end{aligned}$$

**Proof:** (Proof omitted in this abstract.)  $\blacksquare$

Can the general bound in Lemma 10 be improved to the bound for transitively reduced circuits in Corollary 11? The following example shows that the better bound is in general not attainable if the circuit is not transitively reduced. It gives a family of circuits of depth  $2d$  for which the worst-case ratio of the differences shown for  $w$  by an experiment  $e$  and the best path for  $w$  is  $(5/4)^d \pi(e, w)$ .

**Lemma 17** *There exists an infinite set of Boolean probabilistic circuits  $D_1, D_2, \dots$  such that for each  $\ell$  there exists a value injection experiment  $e$  and a wire  $w$  such that  $\pi(e, w) = 4^\ell$  and*

$$d(D_\ell(e|_{w=0}), D_\ell(e|_{w=1})) = (5/7)^\ell,$$

but for any test path  $p$  for  $w$ ,

$$d(D_\ell(p|_{w=0}), D_\ell(p|_{w=1})) = (1/7)^\ell.$$

**Proof:** We first define a Boolean probabilistic circuit  $D_1$  and then connect  $\ell$  copies of it in series to get  $D_\ell$ . The wires of  $D_1$  are  $w_1, \dots, w_5$ . They are connected as in Figure 3; the output wire is  $w_5$ . Note that the edge  $(w_1, w_5)$  means that the circuit graph is not transitively reduced. The gate function  $G$

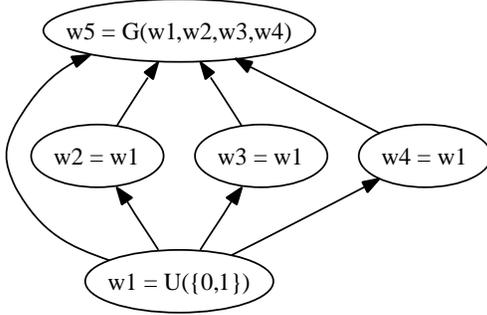


Figure 3: The circuit  $D_1$ ;  $w_5$  is the output wire.

is defined by giving its expected value as a function of its inputs:

$$E[G(w_1, w_2, w_3, w_4)] = ((1 - w_1) + 2w_2 + 2w_3 + 2w_4)/7.$$

Let  $e$  be the experiment that leaves all five wires free. It is clear that

$$d(D_1(e|_{w=0}), D_1(e|_{w=1})) = 5/7.$$

We now show that for any test path  $p$  for  $w_1$ ,

$$d(D_1(p|_{w=0}), D_1(p|_{w=1})) = 1/7.$$

The possible test paths  $p$  for  $w_1$  either fix all of  $w_2, w_3, w_4$  or all but one of them. Thus, as we change from  $w_1 = 0$  to  $w_1 = 1$  in such a test path, the assignments to wires  $(w_1, w_2, w_3, w_4)$  change in one of four possible ways:

- $(0, b_2, b_3, b_4)$  to  $(1, b_2, b_3, b_4)$
- $(0, 0, b_3, b_4)$  to  $(1, 1, b_3, b_4)$
- $(0, b_2, 0, b_4)$  to  $(1, b_2, 1, b_4)$
- $(0, b_2, b_3, 0)$  to  $(1, b_2, b_3, 1)$

Checking each of these possible changes against the definition of  $G$ , we see that each change produces a difference of  $1/7$ , as claimed. (This example can be modified to give a difference of  $1$  versus  $1/5$ ; details are omitted in this abstract.) Thus,  $D_1$  gives the base case of the claim in the lemma.

To construct  $D_\ell$ , we take  $\ell$  copies of  $D_1$  and identify wire  $w_5$  in one copy with wire  $w_1$  in the next copy, making the wire  $w_5$  of the final copy the output wire of the whole

circuit. Let  $w$  denote the wire  $w_1$  in the first such copy. Then  $\pi(e, w) = 4^\ell$  and

$$d(D_\ell(e|_{w=0}), D_\ell(e|_{w=1})) = (5/7)^\ell.$$

For any test path  $p$ , the signal is attenuated by a factor of  $1/7$  for each level, and we have

$$d(D_\ell(p|_{w=0}), D_\ell(p|_{w=1})) = 1/7^\ell. \quad \blacksquare$$

The construction can be generalized to  $k+1$  wires for any odd  $k+1$ , which increases the attenuation. In the base circuit there are  $k$  paths and an attenuation factor of  $1/(2k-3)$ , and the worst-case ratio of differences for an experiment and its test paths in  $D_\ell$  approaches  $2^\ell \pi(e, w)$  as  $k$  goes to infinity.

## 7 Non-Boolean Circuits Revisited

The sharp contrast in results for transitively reduced circuits with alphabet size at least three, for which test paths may show no difference (Lemma 8) and those with alphabet size two, for which test paths must show a significant difference (Lemma 10) motivate us to consider a generalization of the kinds of experiments we consider, to function injection experiments. This generalization allows us to extend the results of Lemma 10 to non-Boolean alphabets.

In a value injection experiment, each wire is either fixed to a constant value or left free. In a function injection experiment, these possibilities are expanded to permit a transformation of the value that the wire would take if it were left free. As an example, consider a transformation in which the values are linearly ordered and all values below a certain threshold are mapped to the minimum value and all other values are mapped to the maximum value. It is conceivable that this kind of transformation could be feasible in some domains; in any case, the theoretical consequences are quite interesting. We first give a general definition of function injection, but in the results below we are primarily concerned with 2-partitions, that is, transformations that are like the above example in that they partition the values into two blocks and map each block to a fixed element of the block.

An **alphabet transformation** is a function  $f$  that maps symbols to distributions over symbols. An alphabet transformation is **deterministic** if it assigns only deterministic distributions, in which case we think of it as a map from symbols to symbols. A deterministic alphabet transformation  $f$  is a  **$k$ -partition** if there exists a partition of  $\Sigma$  into at most  $k$  disjoint nonempty sets  $\Sigma_i$  such that for each  $i$  there exists  $\sigma_i \in \Sigma_i$  such that  $f(\Sigma_i) = \{\sigma_i\}$ . We use 2-partitions to reduce the case of larger alphabets to the binary case. Note that the 2-partitions of a binary alphabet include the identity and the two constant functions, but not the negation function.

If  $D$  is a value distribution and  $f$  is an alphabet transformation, then  $f(D)$  is the value distribution in which

$$(f(D))(\sigma) = \sum_{\tau \in \Sigma} D(\tau)(f(\tau))(\sigma).$$

A **function injection experiment** is a mapping  $e$  with domain  $W$  that assigns to each wire the symbol  $*$  or a symbol from  $\Sigma$  or an alphabet transformation  $f$ . Then  $e$  leaves  $w$  **free** if  $e(w) = *$ , **fixes**  $w$  if  $e(w) \in \Sigma$ , and **transforms**

$w$  if  $e(w)$  is an alphabet transformation  $f$ . We extend the ordering  $\leq$  on experiments by stipulating that each alphabet transformation  $f \leq *$ . A **2-partition experiment** is a function injection experiment in which every alphabet transformation is a 2-partition.

We now define the joint probability distribution on assignments of symbols from  $\Sigma$  to wires determined by a function injection experiment  $e$ . If  $e$  fixes  $w$ , then  $w$  is just assigned  $e(w)$ . Otherwise, if the inputs of  $w$  have been assigned the values  $\sigma_1, \dots, \sigma_k$  and  $f$  is the gate function for  $w$ , we randomly and independently choose a symbol  $\sigma$  according to the value distribution  $f(\sigma_1, \dots, \sigma_k)$ . If  $w$  is free in  $e$ , then  $\sigma$  is the symbol assigned to  $w$ ; however, if  $e(w)$  is an alphabet transformation, then a symbol  $\tau$  is chosen randomly and independently according to the value distribution  $e(\sigma)$  and assigned to  $w$ . That is, when  $e(w)$  is an alphabet transformation, we generate the symbol for  $w$  as though it were free, and then use the distribution  $e(w)$  to transform that symbol. Because  $C$  is acyclic, this process assigns a symbol to every wire of  $C$ .

In a **function injection query** (FIQ), the learning algorithm gives a function injection experiment  $e$  and receives a symbol  $\sigma$  assigned to the output wire of  $C$  by the probability distribution defined above. A **functional test path** for a wire  $w$  is a function injection experiment in which the free and transformed wires are a directed path in the circuit graph from  $w$  to the output wire, and all other wires are fixed.

As an example of how functional test paths help in learning non-Boolean probabilistic circuits, consider the circuit in the proof of Lemma 8. We specify a functional test path  $p$  by  $p(w_1) = p(w_3) = p(w_5) = *$ ,  $p(w_4) = 00$  and  $p(w_2)$  is the alphabet transformation  $00 \rightarrow 00$ ,  $01 \rightarrow 01$ ,  $10 \rightarrow 01$ , and  $11 \rightarrow 00$ . Note that the alphabet transformation is a 2-partition. Then  $C(p|_{w_1=00}) = 00$  but  $C(p|_{w_1=01}) = 01$  deterministically, so this functional test path witnesses a difference of 1, as large as the experiment that leaves all the wires free. Test paths with functions allow us to carry over the results of Lemma 10 to non-Boolean alphabets.

**Lemma 18** *Let  $C$  be a probabilistic circuit,  $e$  be a function injection experiment,  $w$  be a wire, and  $D_1, D_2$  be value distributions. If there exists  $\varepsilon \geq 0$  such that for all functional  $w$ -test paths  $p \leq e$ ,*

$$d(C(p|_{w=D_1}), C(p|_{w=D_2})) \leq \varepsilon,$$

then

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) \leq \kappa(e, w) \cdot \varepsilon.$$

**Proof:** The obstacle in Lemma 10 is that when the alphabet is non-Boolean, we may assume only that  $D_1$  and  $D_2$  have disjoint support, not that they are deterministic. This obstacle can be overcome by injecting a 2-partition at  $w$ . Let  $\Sigma_1 = \text{support}(D_1)$  and  $\Sigma_2 = \text{support}(D_2)$  and assume  $\Sigma_1 \cap \Sigma_2 = \emptyset$ . Then

$$\begin{aligned} & d(C(e|_{w=D_1}), C(e|_{w=D_2})) \\ & \leq \sum_{\substack{\rho_1 \in \Sigma_1 \\ \rho_2 \in \Sigma_2}} D_1(\rho_1) D_2(\rho_2) d(C(e|_{w=\rho_1}), C(e|_{w=\rho_2})) \end{aligned}$$

by the triangle inequality. Let

$$(\sigma, \tau) = \arg \max_{\substack{\rho_1 \in \Sigma_1 \\ \rho_2 \in \Sigma_2}} d(C(e|_{w=\rho_1}), C(e|_{w=\rho_2}))$$

so that

$$\begin{aligned} & d(C(e|_{w=D_1}), C(e|_{w=D_2})) \\ & \leq d(D_1, D_2) d(C(e|_{w=\sigma}), C(e|_{w=\tau})). \end{aligned}$$

Let  $f$  be an alphabet transformation that maps  $\Sigma_1$  to  $\sigma$  and  $\Sigma_2$  to  $\tau$  and all other symbols to either  $\sigma$  or  $\tau$ . Then  $f$  is a 2-partition, and

$$\begin{aligned} & d(C(e|_{w=D_1}), C(e|_{w=D_2})) \\ & \leq d(C(e|_{w=f(D_1)}), C(e|_{w=f(D_2)})). \end{aligned}$$

Since  $f(D_1) = \sigma$  and  $f(D_2) = \tau$ , the rest of the proof goes through. ■

**Corollary 19** *Let  $C$  be a transitively reduced probabilistic circuit,  $e$  be a function injection experiment,  $w$  be a wire, and  $D_1, D_2$  be value distributions. If there exists  $\varepsilon \geq 0$  such that for all functional  $w$ -test paths  $p \leq e$ ,*

$$d(C(p|_{w=D_1}), C(p|_{w=D_2})) \leq \varepsilon,$$

then

$$d(C(e|_{w=D_1}), C(e|_{w=D_2})) \leq \pi(e, w) \cdot \varepsilon.$$

Certain natural questions arise in response to the introduction of function injection experiments. We can define circuits  $C$  and  $C'$  to be **strongly behaviorally equivalent** if  $C(e) = C'(e)$  for every function injection query  $e$ . Does behavioral equivalence imply strong behavioral equivalence? Once again, alphabet size determines the answer: no for alphabet size greater than two, yes for alphabet size two.

**Lemma 20** *For  $\Sigma = \{0, 1, 2\}$ , there exist deterministic circuits  $C_1$  and  $C_2$  that are behaviorally equivalent but not strongly behaviorally equivalent.*

**Proof:** In both  $C_1$  and  $C_2$  there are two wires  $w_1$  and  $w_2$ , where  $w_2$  is the output wire. In both circuits the gate for  $w_2$  has input  $w_1$  and deterministically maps 0 to 0 and maps 1 and 2 to 1. In  $C_1$ ,  $w_1$  is the constant 1 and  $C_2$  it is the constant 2.

Then if  $e$  is the value injection experiment that leaves both wires free,  $C_1(e) = 1 = C_2(e)$ . If  $e$  fixes either  $w_1$  or  $w_2$ , then also  $C_1(e) = C_2(e)$ . Thus  $C_1$  is behaviorally equivalent to  $C_2$ .

However, the 2-partition function injection experiment  $e$  that leaves  $w_2$  free and maps the output of  $w_1$  according to the transformation  $0 \rightarrow 0$ ,  $1 \rightarrow 0$ ,  $2 \rightarrow 2$  yields  $C_1(e) = 0$  and  $C_2(e) = 1$ . Thus  $C_1$  is not strongly behaviorally equivalent to  $C_2$ . ■

However, 2-partition function experiments suffice to establish strong behavioral equivalence.

**Lemma 21** *Let  $C$  and  $C'$  be probabilistic circuits with the same alphabet  $\Sigma$ , the same set of wires and the same output wire. If  $C(e) = C'(e)$  for every 2-partition function experiment  $e$  then  $C$  and  $C'$  are strongly behaviorally equivalent.*

**Proof:** By another modification of the proof of Lemma 10. ■

Because in the Boolean case every 2-partition function injection query is a value injection query, we have the following.

**Corollary 22** *For Boolean probabilistic circuits  $C$  and  $C'$ , if  $C$  is behaviorally equivalent to  $C$  then  $C'$  is strongly behaviorally equivalent to  $C'$ .*

## 8 Discussion and Open Problems

These results concern general probabilistic acyclic circuits, with no restriction other than fan-in on the kinds of probabilistic gates considered. Particular domains may warrant specific assumptions about the gates, which may make the learning problems more tractable. For example, for the problem of learning the structure of an independent cascade social network using exact value injection queries, a query-optimal algorithm is presented in [AAR]. Note that the networks in this domain may contain cycles, which complicates their analysis.

Initial work suggests that Corollary 11 allows us to adapt the Distinguishing Paths algorithm [AACR07] to learn transitively reduced Boolean probabilistic circuits, given a bound on the number of paths in the circuit graph. We would like to adapt Circuit Builder to use functional test paths to learn non-Boolean circuits; in this case the universal set must map wires to the set containing all alphabet symbols from  $\Sigma$  and all 2-partitions of  $\Sigma$ , of which there are fewer than  $|\Sigma|^{2^2|\Sigma|}$ . Thus, the universal set will still be of size  $n^{O(1)}$ , suggesting that a polynomial time algorithm may be attainable in this case. An open question is whether not-injection reduces the maximum path attenuation to just the number of paths for general Boolean probabilistic circuits. A very interesting direction of future work is whether there are computationally feasible approaches to learning probabilistic circuits that use experiments more general than paths and thereby avoid the problem of path attenuation.

## 9 Acknowledgments

This work was done while Jiang Chen was a member of the Center for Computational Learning Systems, Columbia University. The authors thank the reviewers of the present paper for their thoughtful comments.

## References

- [AACR07] Dana Angluin, James Aspnes, Jiang Chen, and Lev Reyzin. Learning large-alphabet and analog circuits with value injection queries. In *the 20th Annual Conference on Learning Theory*, pages 51–65, 2007.
- [AACW06] Dana Angluin, James Aspnes, Jiang Chen, and Yinghua Wu. Learning a circuit by injecting values. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, pages 584–593, New York, NY, USA, 2006. ACM Press.
- [AAR] Dana Angluin, James Aspnes, and Lev Reyzin. Optimally learning social networks with activations and suppressions. Submitted to COLT 2008.
- [AK95] Dana Angluin and Michael Kharitonov. When won't membership queries help? *J. Comput. Syst. Sci.*, 50(2):336–355, 1995.
- [AKMM98] Tatsuya Akutsu, Satoru Kuhara, Osamu Maruyama, and Satoru Miyano. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. In *SODA '98: Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 695–702, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.
- [ITK00] T. Ideker, V. Thorsson, and R Karp. Discovery of regulatory interactions through perturbation: Inference and experimental design. In *Pacific Symposium on Biocomputing 5*, pages 302–313, 2000.
- [KKET03] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, New York, NY, USA, 2003. ACM.
- [KKT05] David Kempe, Jon M. Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *ICALP*, pages 1127–1138, 2005.



---

# Learning Random Monotone DNF Under the Uniform Distribution

---

Linda Sellie\*

University of Chicago, Chicago IL  
lmsellie@uchicago.edu

## Abstract

We show that randomly generated monotone  $c \log(n)$ -DNF formula can be learned exactly in probabilistic polynomial time. Our notion of randomly generated is with respect to a uniform distribution. To prove this we identify the class of *well behaved* monotone  $c \log(n)$ -DNF formulae, and show that almost every monotone DNF formula is well-behaved, and that there exists a probabilistic Turing machine that exactly learns all well behaved monotone  $c \log(n)$ -DNF formula.

## 1 Introduction

Intuitively, a monotone  $c \log(n)$ -DNF,  $f$ , is well behaved if it satisfies three smoothness criteria—the “small”, “medium,” and “large”  $z$  properties—that collectively rule out having an unexpectedly large number of terms having a common subset of the variables. Thus by removing terms we maintain our well-behaved criteria and we have:

**Theorem** (Subset property of the set of well-behaved functions). *If  $f$  is well-behaved and  $f'$  contains a subset of the terms of  $f$  then  $f'$  is also well-behaved.*

The question of what is meant by “a randomly generated monotone  $c \log(n)$ -DNF formula” is somewhat application specific, but because of the subset property of the set of well-behaved functions, our learning algorithm and proof of correctness is quite robust. We imagine a process that randomly selects  $m$  terms of size  $c \log(n)$ ; we show that such a function will be well behaved with high probability as long as  $m \leq 2 \log \log(n) n^c$  (where roughly  $\frac{1}{\log(n)}$  of the examples will be false when  $m = 2 \log \log(n) n^c$ .) This subsumes standard notions of randomness that are intended to generate formula which are expected to be true with fixed probability less than one. For functions with the small and medium smoothness properties and for a set of variables,  $s$ , of bounded size, we can efficiently determine by sampling whether or not there exists a term  $t \in f$  such that  $s \subset t$  with high probability. Our algorithm considers all subsets of variables,  $s$ , of a given, fixed size. To extend  $s$  we make multiple trials of random extension of  $s$ , through

$|s| = \beta(n) = \log \log \sqrt[3]{n}$ . The medium and small subset properties guarantee that with high probability, if  $s' \supset s$  has size at most  $\beta(n)$  and there exists a term  $t \in f$  such that  $s' \subset t$ , then  $s'$  is generated by this process. At this point, the large smoothness property comes into play and guarantees that the previous  $t$  is unique, and therefore can be efficiently found. In this way, we find all terms  $t$  of  $f$  in polynomial time.

### 1.1 Motivation and Past Work

Mentioning DNF, Valiant [12] states:

The possible importance of disjunctions of conjunctions as a knowledge representation stems from the observations that on the one hand humans appear to like using it, and, on the other, that there is circumstantial evidence that significantly larger classes may not be learnable in polynomial time.

Many learning theorist have considered learning monotone DNF formula. Angluin [2] completely solved this problem for the case of exact learning using membership queries — all monotone DNF are learnable in polynomial time in this model for all distributions. This problem has proven more difficult if the learner is restricted to sampling, i.e. learning by example. The obstacle seems to be “cluster structure” within the formula, specifically a relatively large set of variables common to a relatively large number of clauses. Existing results in the literature tackle this obstacle in two different ways. (1) allow the running time of the learner to explode in the face of such clusters, e.g. Verbeurgt [13] learns any poly( $n$ )-size DNF in time  $n^{O(\log(n))}$  from uniform examples. Or (2) consider classes of formula that do not contain such clusters, specifically by random generation and limited number of terms, e.g. Servedio [9] learns any  $2^{\sqrt{\log(n)}}$ -term DNF in polynomial time from a product distribution. Other researchers have used similar approaches to other problems, [10], [8], and [6].

The results of this paper belong to group (2). Our result is distinguished from Servedio [9] in that our definition of *well behaved* represents an initial attempt to formalize the obstacle, and to obtain the best possible result based on that formalization. From this, we obtain conditions of greater generality.

Despite the difficulty of learning monotone DNF with random examples drawn from the uniform distribution, the

---

\*Computer Science Department, University of Chicago.

naturalness of the class suggests in some restricted form, it must be possible to learn. In their 1994 paper, Aizenstein and Pitt proposed learning most DNF instead of all DNF. They defined “most” as the DNF generated randomly with certain parameters set, one parameter is choosing the variables in a term with probability  $\frac{1}{2}$ . They left as an open question a more natural setting of those parameters. Jackson and Servedio in 2006 started answering the open question of Aizenstein and Pitt in their paper [7]. They learned “most” monotone DNF where the number of terms is bounded by  $O(n^{2-\gamma})$  with fixed term size,  $\log m$ , where  $m$  is the number of terms. We continue this work left open by Aizenstein and Pitt, and Jackson and Servedio.

We expand the approach used by Jackson and Servedio in their paper [7]. To learn random monotone DNF with  $O(n^{2-\gamma})$  number of terms, they use a clustering algorithm after using an inclusion/exclusion pair finding algorithm. In our paper, we learn  $O(n^c)$  number of terms in polynomial time for any constant  $c$ , and fixed term size,  $c \log(n)$ .

Similar results are independently obtained by Jackson, Lee, Servedio and Wan [5] but are slightly weaker. They use a similar algorithm but significantly different underlying proofs.

**Theorem 1.** *Given a random monotone DNF,  $f$ , Algorithm Learn Random Monotone DNF finds  $f$  in polynomial time with high probability.*

## 1.2 Our Model and Random Functions

Continuing the work of Aizenstein and Pitt [1] and Jackson and Servedio [7], we explore learning a function chosen randomly from a large class of functions. Jackson and Servedio learn a monotone DNF formula chosen randomly from a subclass of monotone DNF; we do the same except we choose a larger subclass of monotone DNF. As in Jackson and Servedio, we randomly choose the terms for our function from  $\binom{n}{k}$  possible terms of size  $k$ . We differ from Jackson and Servedio’s choice of a class of functions in two ways. The most important is that we learn functions with  $n^c$  terms for any  $c$ , while they learn only for  $c \leq 2 - \gamma$  for  $\gamma > 0$ . The second way we differ is by loosening Jackson and Servedio’s restriction which bounds the function away from 0 and 1 by a constant; we restrict our attention to functions that are bounded away from one by a slow growing function in  $n$ , and without restriction on how close the function is to zero. Even in the case of  $c \leq 2$ , for large  $n$ , the set of functions they learn is a subset of the functions we learn. They allow the number of terms,  $m$ , to be  $\alpha 2^k \leq m \leq 2^{k+1} \ln \frac{2}{\alpha}$  for a constant  $\alpha$ , ( $0 < \alpha < 0.09$ ). Instead, we restrict the number of terms,  $m$ , to be  $m \leq 2^{k+1} c \log(n)$ .

As Jackson and Servedio in [7]; we learn in the uniform distribution model; where each example is chosen uniformly at random and labeled according to the unknown function.

Our goal is stronger than theirs, in that we exactly learn with probability  $1 - \delta$ . (They learn a function which is  $\epsilon$  close with probability  $1 - \delta$ .) We run in time polynomial in the probability of an example satisfying a term, (i.e. time polynomial in  $2^k$ .)

The model for our class of random monotone DNF formulas is as follows, let  $\mathcal{F}^{n,k,m}$  be the set of monotone DNF

over  $n$  variables, with terms of size  $k$ , and  $m$  terms. Or interest is when with  $m \leq 2^{k+1} c \log \log(n)$  where  $c = \frac{k}{\log(n)}$ . Each term is selected independently and uniformly from the set of all  $k$ -variable terms.

## 2 Notation and Definitions

Our function will be defined on  $n$  variables; we let  $X = \{x_1, x_2, \dots, x_n\}$  be the set of variables. For  $s \subset X$  we define  $X \setminus s = \{x \in X \mid x \notin s\}$ . Let  $t \subset X$  be a term, and  $k = |t| = c \log(n)$  be the size of a term. Let  $m$  be the number of terms; where  $m \leq m_{\max} = 2^{k+1} c \log \log(n) = 2n^c \log \log(n)$ . Let  $f = \bigcup_{i=1 \dots m} t_i$ . We define  $f \setminus t = \{t' \in f \mid t' \neq t\}$ .

Let  $E = \{0, 1\}^n$  be the set of all examples, and  $E^+ = \{e \in E \mid f(e)\}$  be the set of all positive examples. Let  $s \subset X$ , and let  $a_s$  be a partial assignment of the variables in  $s$ . For  $x \in s$ , and  $a_s$  a partial assignment, then by an abuse of notation, we define  $x(a_s) = 1$  iff the assignment to  $x$  is 1 and 0 otherwise. Let  $X_{a_s} = \{x \in s \mid x(a_s)\}$  be the set of variables in  $s$  that  $a_s$  satisfies.

We use  $a_s$  to partition the set of examples,  $E^+$ , and the set terms in  $f$ . We then explore the relationship between these sets in the paper with an inclusion/exclusion algorithm that allows us to find subsets of terms in  $f$ . Let  $E_{a_s} = \{e \in E \mid \forall x \in s, x(a_s) = x(e)\}$  be the set of assignments who agree with  $a_s$  on all the variables  $s$ . Let  $E_{a_s}^+ = E_{a_s} \cap E^+$  be the set of positive assignments who agree with  $a_s$  on all the variables  $s$ . Let  $T_{a_s} = \{t \in f \mid t \cap s = X_{a_s}\}$ .

Let  $T_e = \{t \in f \mid t(e)\}$  be the set of terms satisfying an example  $e$ . Let  $\#_0(a_s) = |\{x \in s \mid x(a_s) = 0\}|$ , e.g.  $\#_0(10110) = 2$ .

Let  $\beta(n) = \log \log(\sqrt[3]{n})$ . We use  $\beta(n)$  throughout the paper as a bound that is not constant – and not too large. For technical ease we chose this value of  $\beta(n)$ , we could have chosen other values for  $\beta(n)$ , such as  $\beta(n) = \log \log(n)$ .

For ease of notation, we define  $i^j = \frac{i!}{(i-j)!}$ .

For  $e$  and  $a_s$ , we define a transformation,  $e' = e_{s \leftarrow a_s}$  to be  $\forall x \notin s, x(e') = x(e)$  and  $\forall x \in s, x(e') = x(a_s)$ . For  $e$  and  $x'$  we define  $e' = e_{\text{flip}(x')}$  to be  $\forall x \in (X \setminus \{x'\}), x(e') = x(e)$  and  $x'(e') = 1 - x'(e)$ .

Our algorithm discovers  $f$  by finding subsets of terms in  $f$ . We use the knowledge that a set of variables,  $s$ , is a subset of a term iff there is a positive example,  $e$ , which becomes false for  $e_{\text{flip}(x)}$  for any  $x \in s$ .

**Definition 2.**  $e$  is  $s$ -minimal for  $f$  iff

- $e \in E^+$ , and
- $\forall x \in s, e_{x \leftarrow 0} \notin E^+$ .

We define the set of  $s$ -minimal examples:

**Definition 3.** Let  $\Upsilon_s = \{e \in E^+ \mid e \text{ is } s\text{-minimal}\}$ .

Thus, given any  $e \in \Upsilon_s$  and  $t \in f$ , if  $t(e) = 1$  then  $s \subset t$ .

The idea behind our proof is that we can determine if  $\Upsilon_s$  is non-empty for any  $s$  of cardinality greater than  $c + 1$  and less than or equal to  $\beta(n) + 1$ , thus finding a subset of a term.

### Function Distinguishing Subsets

- $S = \left\{ s \subset X \mid |s| = c + 2, \text{ and } I_s > 2^n \cdot \frac{1}{n^{c+\frac{1}{5}}} \right\}$
- For  $i = (c + 3)$  to  $\beta(n)$ 
  - $S' = \emptyset$
  - For  $s \in S$  and  $x \in X$ 
    - \* If  $I_{s \cup \{x\}} > 2^n \cdot \frac{1}{n^{c+\frac{1}{5}}}$  then add  $(s \cup \{x\})$  to  $S'$
  - $S = S'$
- Return  $S$

Figure 1: Function Distinguishing Subsets

From this knowledge we build the rest of the term. Our goal is to exactly learn  $f$  with probability  $1 - \delta$ .

Unfortunately we don't know how to compute the size of  $\Upsilon_s$ . Instead we estimate  $|\Upsilon_s|$ .

**Definition 4.** Let  $I_s = \sum_{a_s} (-1)^{\#_0(a_s)} |E_{a_s}^+|$ .

Our paper focuses on proving that most monotone  $k$ -DNF are well behaved, that for well behaved functions  $I_s$  approximates  $|\Upsilon_s|$  for  $c + 2 \leq |s| \leq \beta(n) + 1$ , and if  $s \subset t \in f$  then  $\Upsilon_s$  is sufficiently large.

The organization of our paper is as follows: in Section 3 we present a simple algorithm that exploits the knowledge that we can find a subset of at term. In Subsection 4.1 we partition the set of positive examples and use this partition to define how  $I_s$  miscounts the size of  $\Upsilon_s$ . In Subsection 4.2 we prove most  $f \in \mathcal{F}^{n,k,m}$  are well behaved. In Subsection 4.3 we bound by how much  $I_s$  misclassifies  $|\Upsilon_s|$  for well behaved functions. In Subsection 4.4 we prove that for well-behaved  $f \in_{\mathcal{R}} \mathcal{F}^{n,k,m}$ , if  $s \subset t \in f$  then  $\Upsilon_s$  is sufficiently large enabling us to discover if  $s \subset t$ .

We put some of the technical details into the appendix. In Appendix 1A, we provide some observations and simplifications of algebraic expressions used in our proofs. In Appendix 1B, we prove that most  $f \in_{\mathcal{R}} \mathcal{F}^{n,k,m}$  are well behaved. In Appendix 1C, we bound  $\Upsilon_t$  for well behaved DNF. In Appendix 2A, we use standard sampling techniques to prove we can approximate  $|E_{a_s}^+|$  sufficiently. In Appendix 2B, for the sake of completeness, we provide the details that show our algorithm finds the unknown monotone DNF in polynomial time with high probability.

### 3 The Algorithm for Finding $f$ Using $I_s$

Using  $I_s$  as our estimate for  $|\Upsilon_s|$ , our algorithm builds terms in three stages. First our algorithm tests all subsets of size  $c + 2$ , selecting those that are a subset of a term in  $f$ . Next, it builds upon these subsets, variable by variable, till it has found all subsets of terms of  $f$  of size  $\beta(n)$ . Finally, having a subset unique to a single term in  $f$  (we prove the uniqueness of terms of size  $\beta(n)$  later in the paper in Corollary 46,) we find the rest of variables for this term. The steps of the first two stages are in Figure 1 and the steps for the third stage are in Figure 2.

### Algorithm Learn Random Monotone DNF

- $S =$  **Distinguishing Subsets**
- $f = \emptyset$
- For  $s \in S$ 
  - $t = \emptyset$
  - For  $x \in X$ 
    - \* If  $I_{s \cup \{x\}} > 2^n \cdot \frac{1}{n^{c+\frac{1}{5}}}$  then add  $x$  to  $t$
  - add  $t$  to  $f$
- Return  $f$

Figure 2: Algorithm Learn Random Monotone DNF

## 4 Approximating $\Upsilon_s$ by $I_s$

In this section, we show that with high probability  $I_s$  approximates  $|\Upsilon_s|$  for  $c + 2 \leq |s| \leq \beta(n) + 1$  to within  $2^n \cdot \frac{4k \log^6(n)n^{2/3}}{n^{c+1}}$  (i.e.  $|I_s - |\Upsilon_s|| < 2^n \cdot \frac{4k \log^6(n)n^{2/3}}{n^{c+1}}$ .) We use Subsections 4.1 through 4.3 to prove this main theorem. In Subsection 4.4, we prove that  $|\Upsilon_s| \geq 2^n \cdot \frac{1}{8 \log^{4c}(n)} \frac{1}{n^c}$  if and only if  $s \subset t \in f$ .

### 4.1 Observations about $I_s$

To explore how  $I_s$  relates to the size of  $\Upsilon_s$ , we partition the set  $E^+$  of positive examples. We partition  $E^+$  by grouping examples that “map” under  $s$  to the same example in  $E_{1_s}^+$ . Observing the behavior of a partition during the calculation of  $I_s$ , we bound how  $I_s$  misjudges the size of  $\Upsilon_s$ . (We bound the size of the miscalculation in Subsection 4.3.)

**Definition 5.** For  $s \subset X$ , and  $e \in E_{1_s}^+$ , we define a set of partial assignments,  $A_{e,s} = \{a_s \mid e_{s \leftarrow a_s} \in E^+\}$ , which map  $e$  to another positive example under  $s$ .

Next, using this partition of the set of positive examples, we define a criteria for  $e \in E_{1_s}^+$  to be correctly counted.

**Definition 6.** For  $e \in E_{1_s}^+$  we define

$$\mathcal{I}_e = \sum_{a_s \in A_{e,s}} (-1)^{\#_0(a_s)}.$$

**Observation 7.**  $I_s = \sum_{e \in E_{1_s}^+} \mathcal{I}_e$ .

**Definition 8.** An  $e \in E_{1_s}^+$  is correctly counted iff  $e \in E_{1_s}^+ \setminus \Upsilon_s$  then  $\mathcal{I}_e = 0$ , and if  $e \in \Upsilon_s$  then  $\mathcal{I}_e = 1$ .

Note, if all examples in  $E_{1_s}^+$  are counted correctly then  $I_s = |\Upsilon_s|$ . We observe that  $e \in \Upsilon_s$  is correctly counted.

**Lemma 9.** For all  $e \in \Upsilon_s$  then  $\mathcal{I}_e = 1$ .

**Proof:**  $A_{e,s} = \{1_s\}$ . □

By characterizing examples which are correctly counted, we restrict the number of examples that could be incorrectly counted. We describe two ways examples are correctly counted.

**Lemma 10.** *An example,  $e \in E_{1_s}^+$ , is correctly counted if  $\exists x \in s$  such that  $\forall a_s \in A_{e,s}, (a_s)_{\text{flip}(x)} \in A_{e,s}$ .*

**Proof:** Let  $x$  be such that  $\forall a_s \in A_{e,s}$  and  $(a_s)_{\text{flip}(x)} \in A_{e,s}$  then  $(-1)^{\#_0(a_s)}$  and  $(-1)^{\#_0((a_s)_{\text{flip}(x)})}$  are included in the sum, where the parity of  $\#_0(a_s)$  and  $\#_0((a_s)_{\text{flip}(x)})$  are opposite. Thus  $\mathcal{I}_e = 0$  and  $e$  is counted correctly.  $\square$

**Corollary 11.** *An example,  $e \in E_{1_s}^+$ , is correctly counted if  $\exists t \in f$  such that  $t(e)$  and  $t \cap s = \emptyset$ .*

**Proof:** The Corollary follows from Lemma 10 and the definition of  $\mathcal{I}_e$  since all partial assignments are contained in  $A_{e,s}$ .  $\square$

If  $e$  is not known to be correctly counted by Lemma 10 and Corollary 11, it may or may not be correctly counted, but our proof will not need to consider this option.

**Definition 12.** *Let  $s \subset X$ , we define the set of miscounted examples by*

$$M_s = \{e \in E_{1_s}^+ \setminus \Upsilon_s \mid \mathcal{I}_e \neq 0\}.$$

Partitioning  $M_s$  based on sets of partial assignments, we simplify bounding the number of miscounted examples.

**Definition 13.** *Let  $s \subset X$ , and  $A \subset \{0, 1\}^{|s|}$ ; we define  $M_{s,A} = \{e \in M_s \mid A_{e,s} = A\}$ .*

We define a partial order by  $a'_s \prec a''_s$  iff  $\forall x \in s$  then  $x(a'_s) \leq x(a''_s)$  and  $a'_s \neq a''_s$ . The smallest partial assignments are very important to our proof; they determine if an example  $e \in E_{1_s}^+$  is miscounted.

**Definition 14.** *Let  $A \subseteq \{0, 1\}^{|s|}$ , we define*

$$L(A) = \{a_s \in A \mid \forall a'_s \in A, a'_s \not\prec a_s\}.$$

**Lemma 15.** *Let  $e \in M_{s,A}$ , then  $\forall a_s \in L(A), \exists t \in T_{a_s}$  where  $t(e)$ .*

**Proof:** By definition 5, given any  $e \in M_{s,A}$  and  $\forall a_s \in L(A), \exists e' \in E_{a_s}^+$  such that  $e'$  maps under  $s$  to  $e$  (i.e.  $e = e'_{s \leftarrow 1_s}$ ) which implies  $e' = e_{s \leftarrow a_s}$ . Because  $e'$  is  $X_{a_s}$ -minimal, we know  $\exists t \in T_{a_s}$  such that  $t(e')$  which implies  $t(e)$  since  $f$  is monotone.  $\square$

We have now proved in Lemmas 10 and 15 that every miscounted example is satisfied by a set of terms whose union contains  $s$ . We will use this fact in Lemma 24 where we bound the number of miscounted examples in  $M_{s,A}$ .

Knowing  $A$  is a subset of the partial assignments to  $s$ , we calculate by how much an example has been miscounted.

**Observation 16.** *Let  $e \in M_{s,A}$  then  $|\mathcal{I}_e| < |A| \leq 2^{|s|}$ .*

**Definition 17.** *Let  $\mathbf{A}_s = \{A \mid A = A_{e,s} \text{ for an } e \in M_s\}$ .*

**Observation 18.**

$$I_s - |\Upsilon_s| = \sum_{e \in M_s} \mathcal{I}_e = \sum_{A \in \mathbf{A}_s} \sum_{e \in M_{s,A}} \mathcal{I}_e.$$

## 4.2 Properties of Well Behaved Functions

In this subsection, we describe the properties a function needs for our proof to hold; our algorithm works for functions that are not ‘‘clustered’’ together. We prove that with high probability these properties hold for  $f \in \mathcal{R} \mathcal{F}^{n,k,m}$ . We will call DNF formulas that have this property ‘‘well behaved.’’

**Definition 19.** *A monotone DNF function,  $f \in \mathcal{F}^{n,k,m}$  is well behaved iff for all  $s \subset t \in f$  where  $|s| \leq \beta(n) + 1$ , and  $\forall a_s$  where  $z = \#_1(a_s)$  then*

- Small  $z$  property:  
if  $0 < z \leq c$  then  $|T_{a_s}| < 3m_{\max} k^z / n^z$ ,
- Medium  $z$  property:  
if  $c < z < \beta(n)$  then  $|T_{a_s}| < \beta(n)$ , and
- Large  $z$  property:  
if  $z \geq \beta(n)$  then  $|T_{a_s}| \leq 1$ .

Using Chernoff bounds we prove random monotone DNF are well behaved with high probability.

**Theorem 20.** *For a fixed  $c$  and sufficiently large  $n$ , if  $f \in \mathcal{R} \mathcal{F}^{n,k,m}$  for  $m \leq 2^{k+1} c \log \log(n)$  then  $f$  is well behaved with probability at least  $1 - n^{2c} \log(n) \left(\frac{1}{n}\right)^{\beta(n)}$ .*

**Proof:** This follows from Corollaries 42, 44, and 46 (found in the appendix,) and noting that the probability of small, medium and large  $z$  properties of being well behaved are not satisfied with probability at most  $\frac{1}{3} n^{2c} \log(n) \left(\frac{1}{n}\right)^{\beta(n)-1} + \frac{1}{3} n^{2c} \log(n) / 3 \left(\frac{1}{n}\right)^{\beta(n)-1} + \frac{1}{3} n^{2c} \log(n) / 3 \left(\frac{1}{n}\right)^{\beta(n)-1}$ . Consequently  $f \in \mathcal{R} \mathcal{F}^{n,k,m}$  is well behaved with probability at least  $1 - n^{2c} \log(n) \left(\frac{1}{n}\right)^{\beta(n)}$ .  $\square$

## 4.3 Observations about well behaved Monotone DNF Formulas

In this subsection, we derive some properties of well behaved functions. First, we bound the number of variables that occur in more than one term from a set of terms,  $T \subset f$  for  $f \in \mathcal{R} \mathcal{F}^{n,k,m}$ . Next we bound the probability an example satisfies every term in  $T$ . Third, we bound the size of  $M_{s,A}$ , using the probability an example satisfies a term in  $T_{a_s}$  for every  $a_s \in L(A)$ . At the end of this subsection we bound  $|M_s|$  and  $I_s - |M_s|$ .

**Corollary 21.** *Let  $f$  be a well behaved monotone  $k$ -DNF formula and  $T \subset f$ , then  $|\{x \mid x \in (t \cap t') \text{ for some } t, t' \in T\}| < |T|^2 \beta(n)$ .*

**Proof:** For  $f$ , a well behaved monotone  $k$ -DNF, we know that a pair of terms  $t, t' \in f$  have in common at most  $\beta(n)$  variables. Since the number of pairs is  $\binom{|T|}{2}$ , we bound the total number of variables used by more than one term by  $\binom{|T|}{2} \beta(n)$ . Note that what we’ve proved is stronger than what we’ve claimed. The form of our claim is for our subsequent technical convenience.  $\square$

Knowing an upper bound on the number of variables occurring in a set of terms, we bound the probability an example satisfies every term in this set of terms.

**Lemma 22.** Let  $f$  be a well behaved monotone  $k$ -DNF, and  $T \subset f$  a subset of terms then

$$|\{e \in E \mid \forall t \in T, t(e)\}| \leq 2^n \cdot \frac{1}{2^{(|T|k - |T|^2\beta(n))}}.$$

**Proof:** The  $T$  terms share at most  $|T|^2\beta(n)$  variables out  $|T|k$  variables by Corollary 21. Thus the number of variables that need to be satisfied is at least  $|T|k - |T|^2\beta(n)$ .  $\square$

We note that if we restrict our examples to have the bits in  $s$  set to one, we get the following corollary.

**Corollary 23.** For  $s \subset X$  and  $|\{e \in E \mid \forall t \in T, t(e)\}| \leq 2^n \cdot \frac{1}{2^{(|T|k - |T|^2\beta(n))}}$  then

$$|\{e \in E_{1_s} \mid \forall t \in T, t(e)\}| \leq 2^{n-|s|} \cdot \frac{1}{2^{(|T|k - |T|^2\beta(n))}}.$$

**Proof:** The size of the set  $E_{1_s}$  is  $2^{n-|s|}$ . Given that  $|\{e \in E \mid \forall t \in T, t(e)\}| \leq 2^n \cdot \frac{1}{2^{(|T|k - |T|^2\beta(n))}}$ , the restriction of the variables to be from the set  $E_{1_s}$  reduces the number of variables that must be satisfied to at least  $(|T|k - |T|^2\beta(n) - |s|)$ . (i.e. at most  $|s|$  bits were forced to one.) Thus  $|\{e \in E_{1_s} \mid \forall t \in T, t(e)\}| \leq 2^{n-|s|} \cdot \frac{1}{2^{(|T|k - |T|^2\beta(n) - |s|)}}$   
 $= 2^n \cdot \frac{1}{2^{(|T|k - |T|^2\beta(n))}}.$   $\square$

We now bound the number of examples in  $M_{s,A}$ .

**Lemma 24.** For fixed  $c$  and sufficiently large  $n$ , let  $f$  be a well behaved monotone  $k$ -DNF,  $s \subset X$  where  $c + 2 \leq |s| \leq \beta(n) + 1$ , and  $A \in \mathbf{A}_s$  then  $|M_{s,A}| < 2^n \cdot \frac{k \log^5(n)}{n^{c+1}}$ .

**Proof:** Let  $v = |L(A)|$ .

As noted in Lemma 15,  $e \in M_{s,A}$  are satisfied by at least one term from every  $T_{a_s}$  for every  $a_s \in L(A)$ . From Corollary 23, we know that the probability an example satisfies a set of  $v$  terms in  $E_{1_s}$  is at most  $2^n \cdot \frac{1}{2^{vk - v^2\beta(n)}}$

Therefore we bound  $|M_{s,A}|$  by bounding the number of  $e \in M_{s,A}$  which is satisfied by at least one term from every  $T_{a_s}$  for every  $a_s \in L(A)$ . We create this bound by using a Bonferroni type argument.

$$\begin{aligned} & |M_{s,A}| \\ &= |\{e \in M_s \mid \forall a_s \in L(A), \exists t \in T_{a_s}, t(e)\}| \quad (\text{Def. 13.}) \\ &\leq 2^n \cdot \frac{1}{2^{vk - v^2\beta(n)}} \prod_{a_s \in L(A)} |T_{a_s}| \quad (\text{Lemma 15.}) \end{aligned}$$

In counting the number of possible ways an example  $e \in M_{s,A}$  could be satisfied by one term from every  $T_{a_s}$ , for every  $a_s \in L(A)$ , we consider two cases.

In the first case, we assume that for all  $a_s \in L(A)$  that  $\#_1(a_s) \leq c$ . Using the assumption that  $f$  is well defined, we know that  $|T_{a_s}| < 3m_{\max} \left( \frac{k^{\#_1(a_s)}}{n^{\#_1(a_s)}} \right)$ , we compute the probability as follows.

$$|M_{s,A}| \leq 2^n \cdot \frac{1}{2^{vk - v^2\beta(n)}} \prod_{a_s \in L(A)} 3m_{\max} \left( \frac{k^{\#_1(a_s)}}{n^{\#_1(a_s)}} \right).$$

By Lemma 10 and Corollary 11,  $s \subseteq \left( \bigcup_{t \in T_e} t \right)$  and  $\forall a_s \in A_{e,s}, \#_1(a_s) \geq 1$ . Let  $w = \sum_{a_s \in L(A)} \#_1(a_s) \geq$

$\max\{|L(A)|, |s|\}$  (and since  $v = |L(A)|$ .) This implies that

$$\begin{aligned} |M_{s,A}| &\leq 2^n \cdot \frac{3^v m_{\max}^v k^w}{2^{vk - v^2\beta(n)} n^w} \\ &\leq 2^n \cdot \frac{2^{v^2\beta(n)} (6c \log \log(n))^{v n^{cv}} k^w}{2^{vk} n^w} \\ &= 2^n \cdot 2^{v^2\beta(n)} (6c \log \log(n))^v \frac{k^w}{n^w} \quad (n^{cv} = 2^{kv}) \\ &\leq 2^n \cdot \sqrt[3]{n} (6c \log \log(n))^{c+2} \frac{k^{c+2}}{n^{c+2}} \\ &\quad (\text{From Obs. 35, } w \geq v, \text{ and } w \geq |s| \geq c + 2.) \\ &\leq 2^n \cdot \frac{1}{n^{c+1}} \quad (\text{From Observation 33.}) \end{aligned}$$

In the second case, there exists an  $a'_s \in L(A)$  such that  $\#_1(a'_s) > c$ ; by  $f$  being well behaved we know that  $|T_{a'_s}| < \beta(n)$ . Let  $v' = |\{a'_s \in L(A) \mid \#_1(a'_s) > c\}|$ . If  $a_s \in L(A)$  where  $\#_1(a_s) \leq c$  then by  $f$  being well behaved we know that  $|T_{a_s}| < 3m_{\max} \left( \frac{k^{\#_1(a_s)}}{n^{\#_1(a_s)}} \right)$ . Using these bounds, we compute an upper bound by again noting that  $e' \in M_{s,A}$  is satisfied by one from each  $T_{a_s}$  for all  $a_s \in L(A)$ .

$$|M_{s,A}| \leq 2^n \cdot \frac{(\beta(n))^{v'}}{2^{vk - v^2\beta(n)}} \prod_{a_s \in L(A), \#_1(a_s) \leq c} 3m_{\max} \cdot \frac{k^{\#_1(a_s)}}{n^{\#_1(a_s)}}.$$

By Lemma 11, we know  $\#_1(a_s) \geq 1$  for all  $a_s \in L(A)$ , and  $\left( \frac{k^{\#_1(a_s)}}{n^{\#_1(a_s)}} \right) \leq \left( \frac{k}{n} \right)$ . we reduce the formula so that

$$\begin{aligned} & |M_{s,A}| \\ &\leq 2^n \cdot \frac{(\beta(n))^{v'}}{2^{vk - v^2\beta(n)}} (3m_{\max})^{v-v'} \left( \frac{k}{n} \right)^{(v-v')} \\ &\leq 2^n \cdot \frac{(\beta(n))^{v'}}{2^{vk - v^2\beta(n)}} (6c \log \log(n) 2^k)^{(v-v')} \left( \frac{k}{n} \right)^{(v-v')} \\ &\quad (\text{since } n^{c(v-v')} = 2^{k(v-v')}.) \\ &\leq 2^n \cdot \frac{(\beta(n))^{v'} 2^{v^2\beta(n)}}{2^{v'k}} (6c \log \log(n))^{(v-v')} \left( \frac{k}{n} \right)^{(v-v')} \end{aligned}$$

We now break the calculations down into two sub-cases. If  $v = 2$  then the equation is largest if  $v' = 1$ . In this case we bound  $|M_{s,A}|$  by  $2^n \cdot \frac{\beta(n) 2^{4\beta(n)}}{2^k} (6c \log \log(n)) \left( \frac{k}{n} \right) \leq 2^n \cdot \frac{\beta(n) \log^4(\sqrt[3]{n})}{2^k} (6c \log \log(n)) \left( \frac{k}{n} \right) < 2^n \cdot \frac{\log^5(n)k}{n^{2k}}$ .

If  $v \geq 3$ , we note <sup>1</sup> this equation is again largest if  $v' = 1$ , and using Observation 35, we reduce the formula to:

$$2^n \cdot \frac{\beta(n) \sqrt[3]{n}}{2^k} (6c \log \log(n))^{(v-1)} \left( \frac{k}{n} \right)^{(v-1)} \leq 2^n \cdot \frac{1}{n^{2k}}.$$

Therefore  $|M_{s,A}| \leq 2^n \cdot \frac{k \log^5(n)}{n^{c+1}}$ .  $\square$

Having computed an upper bound on the number of miscounted examples in  $M_{s,A}$ , we now bound  $|M_s|$ .

<sup>1</sup>Argument here passes over a minor potential difficulty. i.e. if  $v'$  is large, Corollary 23 does not come into play — but the crucial fact is the nevertheless true as we show in Observation 32.

**Corollary 25.** *Let  $f$  be a well behaved monotone  $k$ -DNF, and let  $s \subset X$  where  $c + 2 \leq |s| \leq \beta(n) + 1$  then  $|M_s| < 2^n \cdot \frac{4k \log^5(n)n^{2/3}}{n^{c+1}}$ .*

**Proof:** This follows from  $|M_s| = \sum_{A \in \mathbf{A}_s} |M_{s,A}| < |\mathbf{A}_s| \left( 2^n \cdot \frac{k \log^5(n)}{n^{c+1}} \right)$ . We note that  $|\mathbf{A}_s|$  is bounded by the number of subsets of the subsets of  $s$ , i.e.  $2^{2^{|s|}} \leq 2^{2^{\beta(n)+1}} = 2^{2^{\log \log(\sqrt[3]{n})+1}} \leq 4n^{2/3}$ .

Thus  $|M_s| < 2^n \cdot \frac{4k \log^5(n)n^{2/3}}{n^{c+1}}$ .  $\square$

Knowing  $|M_s|$ , we now compute the difference between  $I_s$  and  $|\Upsilon_s|$ . This bound is computed by multiplying  $|M_s|$  and a bound of how large the misclassification is for an example.

**Theorem 26.** *Let  $f$  be a well behaved monotone  $k$ -DNF formula, and  $s \subset X$  where  $c + 2 \leq |s| \leq \beta(n) + 1$  then  $|I_s - |\Upsilon_s|| < 2^n \cdot \frac{4k \log^6(n)n^{2/3}}{n^{c+1}}$ .*

**Proof:** As noted earlier,  $I_s - |\Upsilon_s| = \sum_{e \in M_s} \mathcal{I}_e$ .

Using Corollary 25, we know  $|M_s| < 2^n \cdot \frac{4k \log^5(n)n^{2/3}}{n^{c+1}}$ . From Observation 16, we know that for all  $e$ ,  $|\mathcal{I}_e| \leq \log(\sqrt[3]{n})$ .

Consequently,

$$|I_s - |\Upsilon_s|| \leq |M_s| \log(\sqrt[3]{n}) < 2^n \cdot \frac{4k \log^6(n)n^{2/3}}{n^{c+1}}.$$

$\square$

#### 4.4 Bounding $|\Upsilon_s|$

**Definition 27.** *Let  $E_{f \setminus t} = \{e \in E \mid \exists t' \in f \setminus t, t'(e)\}$ .*

Next we prove that every term has a high probability of being uniquely satisfied. Jackson and Servedio have a similar lemma, Lemma (3.6).

**Lemma 28.** *Let  $f \in \mathcal{F}^{n,k,m}$  be a well behaved monotone  $k$ -DNF function,  $t \in f$  then  $|E_{1_t}^+ - E_{f \setminus \{t\}}| \geq 2^n \cdot \frac{1}{8 \log^{4c}(n)} \frac{1}{n^c}$*

The proof of this lemma is found in Appendix 1 in Subsection C.

We note that if  $f$  is a monotone DNF and  $e \in (E_t - E_{f \setminus \{t\}})$ , then  $e \in \Upsilon_t$ .

**Corollary 29.** *Let  $f \in \mathcal{F}^{n,k,m}$  be a well behaved monotone  $k$ -DNF, and  $s \subset t \in f$  then  $|\Upsilon_s| > 2^n \cdot \frac{1}{8 \log^{4c}(n)} \frac{1}{n^c}$ .*

The following theorem is crucial; it is the key computation we use in our algorithm **Learn Random Monotone DNF**.

**Theorem 30.** *Let  $f \in \mathcal{F}^{n,k,m}$  be well behaved, and let  $c + 2 \leq |s| \leq \beta(n) + 1$ :*

- if  $s \subset t \in f$  then  $I_s \geq 2^n \cdot \frac{1}{8 \log^{4c}(n)} \frac{1}{n^c} - 2^n \cdot \frac{4k \log^6(n)n^{2/3}}{n^{c+1}}$ ,
- if  $s \not\subset t \in f$  then  $I_s \leq 2^n \cdot \frac{4k \log^6(n)n^{2/3}}{n^{c+1}}$ .

The previous theorem shows there exists a large gap that reliably determines if  $s \subset t \in f$  for  $c+2 \leq |s| \leq \beta(n)+1$  by computing  $I_s$ . This means that given a small set  $s \subset t \in f$  and  $x \in X \setminus s$  we can determine whether or not  $s \cup x \subset t \in f$ , and this is the key to our algorithm.

In this section, we proved we could determine if a set,  $s$ , is a subset of a term if  $c + 2 \leq |s| \leq \beta(n) + 1$  by computing  $I_s$ . Unfortunately, we cannot efficiently compute  $I_s$  since we cannot compute  $|E_{a_s}^+|$  in polynomial time. Instead we approximate  $I_s$  using standard sampling techniques. We estimate this value by sampling  $g_s = n^{2c+3} 2^{k+|s|}$  uniformly chosen labeled examples from  $E$ . Thus we can effectively estimate  $I_s$  with high probability. Details can be seen in Appendix 2 in Subsection A. Our fairly straightforward algorithm is easily adapted to use our sampled values of  $I_s$ , and thus runs in polynomial time in  $n$  and  $2^k$ . Details can be found in Appendix 2 in Subsection B.

## 5 Future Work

Extensions of the ideas presented here can also handle the non-monotone case. We are currently writing up this case and checking the proofs. We are also working on relaxing the requirement that  $k$  is fixed.

## Acknowledgement

I would like to thank Stuart Kurtz for many conversations and help with the presentation, and Carsten Lund for help with polishing the paper.

## References

- [1] H. Aizenstein and L. Pitt. *On the Learnability of Disjunctive Normal Form Formulas*. Machine Learning, 19:183, 1995.
- [2] D. Angluin. *Queries and concept learning*. Machine Learning, 2(4):319–342, 1988.
- [3] Canny. *Lecture 10 CS 174* [www.cs.berkeley.edu/jfc/cs174/lects/lec10/lec10.pdf](http://www.cs.berkeley.edu/jfc/cs174/lects/lec10/lec10.pdf).
- [4] J. Jackson. *An efficient membership-query algorithm for learning DNF with respect to the uniform distribution*. Journal of Computer and System Sciences, 55(3):414–440, 1997.
- [5] J. Jackson, H. Lee, R. Servedio and A. Wan. *Learning random monotone DNF*. Electronic Colloquium on Computational Complexity, Report No. 129, 2007.
- [6] J. Jackson and R. Servedio. *Learning Random Log-Depth Decision Trees under Uniform Distribution*. SIAM J. on Computing, 34(5), 2005.
- [7] J. Jackson and R. Servedio. *On Learning Random DNF Formulas Under the Uniform Distribution*. Theory of Computing, 2(8):147–172, 2006.
- [8] E. Mossel, R. O’Donnell, R. Servedio. *Learning juntas*. STOC 206–212, 2003.
- [9] R. Servedio. *On learning monotone DNF under product distributions*. Information and Computation, 193(1):57–74, 2004.
- [10] S. Smale. *On the average number of steps of the simplex method of linear programming*. Math. Programming 27: 241–267, 1983.

- [11] Valiant, L.G. (1984). *A theory of the learnable*. Communications of the ACM, 27(11):1134–1142.
- [12] Valiant, L.G. *Learning disjunctions of conjunctions*. In Proceedings of the 9th n International Joint Conference on Artificial Intelligence, Vol. 1, pages 560–566, 1985.
- [13] K. Verbeurgt. *Learning DNF under the uniform distribution in quasi-polynomial time*. In Proceedings of the Third Annual Workshop on Computational Learning Theory, pp. 314326, 1990. [ACM:92571.92657]. 1.1, 2.2

## APPENDIX 1

### A Useful Observations

We use the following observations and simplifications of algebraic expressions in our proofs.

**Observation 31.** For  $f$  a well behaved monotone  $k$ -DNF,  $T \subset f$  where  $|T| \geq 2 \log \log(n)$  then  $|\{e \in E_{1_s} \mid t(e) \forall t \in T\}| < 2^n \cdot \frac{1}{n^{\log \log(n)}}$  for sufficiently large  $n$ .

**Proof:** First, we observe that if  $T' \subset T$  then  $\{e \in E_{1_s} \mid t(e), \forall t \in T'\} \supset \{e \in E_{1_s} \mid t(e), \forall t \in T\}$ .

Therefore, using Corollary 23 we know given any  $T' \subset T$  where  $|T'| = 2 \log \log(n)$ , then  $|\{e \in E_{1_s} \mid t(e), \forall t \in T'\}| \leq 2^n \cdot \frac{1}{2^{2 \log \log(n)k - (2 \log \log(n))^2 \beta(n)}} < 2^n \cdot \frac{1}{n^{\log \log(n)}}$ .  $\square$

**Observation 32.** For a well-behaved monotone  $k$ -DNF, given  $A \subset \{0, 1\}^{|s|}$  where  $|\{a_s \in A \mid \#_1(a_s) > c\}| \geq 2 \log \log(n)$  then  $|M_{s,A}| < \frac{\beta(n)^{2 \log \log(n)}}{n^{\log \log(n)}}$ .

**Proof:** Let  $A' = \{a_s \in A \mid \#_1(a_s) > c\}$ . Using Observation 31, if  $|A'| \geq 2 \log \log(n)$  then  $|M_{s,A}| \leq |\{e \in E_{1_s} \mid \forall a_s \in A', \exists t \in T_{a_s} \text{ such that } t(e)\}| \leq 2^n \cdot \frac{1}{n^{\log \log(n)}} \prod_{a_s \in A'} |T_{a_s}| \leq 2^n \cdot \frac{\beta(n)^{2 \log \log(n)}}{n^{\log \log(n)}}$ . The last inequality follows from noting that  $\forall \#_1(a_s) > c, |T_{a_s}| \leq \beta(n)$  by the large  $z$  property, and from noting that the product is maximized for  $|A'| = 2 \log \log(n)$ .  $\square$

**Observation 33.** For  $c$  a constant, then  $n^{\frac{c+1}{2}} = n(n-1) \cdots (n-c) > n^{\frac{c+1}{2}} - \left(\frac{c(c+1)}{2}\right) n^c$  and  $n^{\frac{c+1}{2}} < n^{\frac{c+1}{2}} - \left(\frac{c(c+1)}{2}\right) n^c + \left(\frac{c(c+1)}{2}\right)^2 n^{c-1}$ .

**Observation 34.** Let  $s \subset X$ ,  $e \in E^+$ , and  $a_s = e|_s$  if  $\#_1(a_s) \leq c, \forall a_s \in L(A_{e,s})$  then  $|L(A_{e,s})| < |s|^{\frac{(c+1)c}{2}}$ .

**Proof:** This follows from observing that  $\binom{|s|}{c} \leq |s|^c$ .  $\square$

**Observation 35.** Let  $s \subset X$  where  $|s| \leq \beta(n) + 1$ , and  $e \in E^+$  then if  $\#_1(a_s) \leq c, \forall a_s \in L(A_{e,s})$  then  $2^{|L(A_{e,s})|} \beta(n) < \sqrt[3]{n}$  for large enough  $n$ .

**Proof:** Let  $s \subset X$  where  $|s| \leq \beta(n) + 1$ , and  $A_{e,s}$  be such that  $\#_1(a_s) \leq c, \forall a_s \in L(A_{e,s})$ . Then using Observation 34 we know  $|L(A_{e,s})| < |s|^{\frac{(c+1)c}{2}}$ .

$$\begin{aligned} 2^{|L(A_{e,s})|} \beta(n) &\leq 2^{\left((\beta(n)+1)^{\frac{(c+1)c}{2}}\right) \beta(n)} \\ &= 2^{(\beta(n)+1)^{(c+1)c} \beta(n)} \\ &< 2^{(\log \log \sqrt[3]{n}+1)^{(c+1)c} \log \log \sqrt[3]{n}} \\ &< (\log \sqrt[3]{n})^{(\log \log \sqrt[3]{n}+1)^{(c+1)c}} \\ &< \sqrt[3]{n}. \end{aligned}$$

**Lemma 36.** For a positive integer  $a \geq \beta(n)$  and  $c \leq \log(n)/(3 \log \log(n))$  then

$$\binom{m}{a} \left(\frac{k^{c+1}}{n^{c+1}}\right)^a > \binom{m}{a+1} \left(\frac{k^{c+1}}{n^{c+1}}\right)^{a+1}.$$

**Proof:** The proof follows from expanding the formulas:

$$\begin{aligned} \binom{m}{a} \left(\frac{k^{c+1}}{n^{c+1}}\right)^a &> \binom{m}{a+1} \left(\frac{k^{c+1}}{n^{c+1}}\right)^{a+1} \Leftrightarrow \\ \frac{m^a}{a!} \left(\frac{1}{(n^{c+1})^a}\right) &> \frac{m^{a+1}}{(a+1)!} \left(\frac{k^{c+1}}{(n^{c+1})^{a+1}}\right) \Leftrightarrow \\ 1 &> \frac{m-a}{a+1} \left(\frac{k^{c+1}}{n^{c+1}}\right). \end{aligned}$$

$\square$

**Observation 37.** For positive integer  $k$ ,  $(1 - \frac{1}{2^k})^{2^k} \geq \frac{1}{4}$

**Proof:** The proof follows from expanding the formula and noting that

$$\binom{2^k}{2i} \left(-\frac{1}{2^k}\right)^{2i} + \binom{2^k}{2i+1} \left(-\frac{1}{2^k}\right)^{2i+1} \geq 0,$$

and

$$\binom{2^k}{2} \left(-\frac{1}{2^k}\right)^2 + \binom{2^k}{3} \left(-\frac{1}{2^k}\right)^3 \geq \frac{1}{4}.$$

Thus

$$\begin{aligned} \left(1 - \frac{1}{2^k}\right)^{2^k} &= \sum_{i=0 \dots 2^k} \binom{2^k}{i} \left(-\frac{1}{2^k}\right)^i \\ &\geq \frac{1}{4}. \end{aligned}$$

$\square$

**Observation 38.** For constant  $c$  and sufficiently large  $n$ ,  $\log(n) \binom{m}{\beta(n)} \left(\frac{k^{c+1}}{n^{c+1}}\right)^{\beta(n)} + m \binom{m}{\log(n)} \left(\frac{k^{c+1}}{n^{c+1}}\right)^{\log(n)} < \frac{1}{n^{\beta(n)-1}}$ .

**Proof:** The proof follows from the following calculations:

$$\begin{aligned} &\log(n) \binom{m}{\beta(n)} \left(\frac{k^{c+1}}{n^{c+1}}\right)^{\beta(n)} \\ &+ m \binom{m}{\log(n)} \left(\frac{k^{c+1}}{n^{c+1}}\right)^{\log(n)} \\ &< \frac{\log(n) (2c \log \log(n))^{\beta(n)} n^{c\beta(n)} k^{(c+1)\beta(n)}}{(n^{c+1} - \frac{(c+1)(c+2)}{2} n^c)^{\beta(n)}} \\ &+ \frac{(2c \log \log(n))^{\log(n)+1} n^{c(\log(n)+1)} k^{(c+1)\log(n)}}{\left(n^{c+1} - \frac{(c+1)(c+2)}{2} n^c\right)^{\log(n)}} \\ &\leq \frac{\log(n) (2c \log \log(n))^{\beta(n)} k^{(c+1)\beta(n)}}{\left(n - \frac{(c+1)(c+2)}{2}\right)^{\beta(n)}} \\ &+ \frac{(2c \log \log(n))^{\log(n)+1} n^c k^{(c+1)\log(n)}}{\left(n - \frac{(c+1)(c+2)}{2}\right)^{\log(n)}} \\ &< \frac{1}{n^{\beta(n)-1}}. \end{aligned}$$

The first inequality follows from Observation 33 and substitution using  $m \leq 2c \log \log(n)n^c$  and  $\binom{m}{i} \leq m^i$ . The second inequality can be seen by multiplying the first summand by  $\frac{1/n^{c\beta(n)}}{1/n^{c\beta(n)}}$  and the second summand by  $\frac{1/n^{c \log(n)}}{1/n^{c \log(n)}}$ . The last inequality can be seen by:  $\left(n - \frac{(c+1)(c+2)}{2}\right)^{\beta(n)} > n^{\beta(n)} - \frac{\beta(n)(c+1)(c+2)}{2} n^{\beta(n)-1}$  and  $\left(n - \frac{(c+1)(c+2)}{2}\right)^{\log(n)} > n^{\log(n)} - \frac{\log(n)(c+1)(c+2)}{2} n^{\log(n)-1}$  and  $\frac{\log(n)(2c \log \log(n))^{\beta(n)} k^{(c+1)\beta(n)}}{\left(1 - \frac{\beta(n)(c+1)(c+2)}{2n}\right)} + \frac{(2c \log \log(n))^{\log(n)+1} n^c k^{(c+1) \log(n)}}{\left(n^{\log(n)-\beta(n)} - \frac{\beta(n)(c+1)(c+2)n^{\log(n)-\beta(n)-1}}{2}\right)} < n$ .  $\square$

## B Proving Random Monotone Functions are Well Behaved with High Probability

In this sections we prove that most monotone DNF in  $\mathcal{F}^{n,k,m}$  are well behaved.

For the small  $z$  property of being well behaved, bounding  $|T_{a_s}|$  for  $\#_1(a_s) \leq c$ , we first find the expected value of  $|T_{a_s}|$ . Next we use Chernoff bounds to give an upper bound on how far  $|T_{a_s}|$  is away from the expected value with high probability.

**Observation 39.** For  $s \subset X$ ,  $a_s$  where  $z = \#_1(a_s)$ , and  $|s| \leq k \leq \log^2(n)$  then

$$m \frac{k^z}{2n^z} < \mathbf{E}_{f \in \mathcal{R} \mathcal{F}^{n,k,m}} \{|T_{a_s}|\} < m \frac{k^z}{n^z}.$$

**Proof:** We first observe that

$$\mathbf{E}_{f \in \mathcal{R} \mathcal{F}^{n,k,m}} \{|T_{a_s}|\} = m \frac{\binom{n-|s|}{k-z}}{\binom{n}{k}} = m \frac{k^z (n-k)^{|s|-z}}{n^{|s|}}.$$

The upper bound follows by observing that

$$\frac{(n-k)^{|s|-z}}{n^{|s|}} = \frac{(n-k)^{|s|-z}}{n^z (n-z)^{|s|-z}} \leq \frac{1}{n^z}$$

(remember  $z \leq k$ .) and thus

$$\mathbf{E}_{f \in \mathcal{R} \mathcal{F}^{n,k,m}} \{|T_{a_s}|\} \leq m \frac{k^z}{n^z}.$$

The lower bound follows from

$$\begin{aligned} \frac{(n-k)^{|s|-z}}{n^{|s|}} &= \frac{1}{n^z} \prod_{i=0 \dots |s|-z-1} \frac{n-k-i}{n-z-i} \\ &= \frac{1}{n^z} \prod_{i=0 \dots |s|-z-1} \left(1 - \frac{k-z}{n-z-i}\right) \\ &\geq \frac{1}{n^z} \left(1 - \frac{k-z}{n-|s|}\right)^{|s|} \\ &\quad \text{By } \left(1 - \frac{k-z}{n-|s|}\right)^{|s|} > 1 - |s| \frac{k-z}{n-|s|}. \\ &> 1/2 \frac{1}{n^z}, \end{aligned}$$

and thus  $m \frac{k^z}{2n^z} < m \frac{k^z (n-k)^{|s|-z}}{n^{|s|}}$ .  $\square$

Next, we state the simplified Chernoff upper bound from Canny's lecture notes [3]; we use this Chernoff bound to bound the expected value.

**Lemma 40 (Chernoff).** Let  $\delta < 2e - 1$ ,  $\mu$  be the expected value, and  $\chi$  be a series of independent Poisson trials, then  $\Pr \{\chi > (1 + \delta)\mu\} < e^{(-\mu\delta^2/4)}$ .

We now bound the number of terms in  $T_{a_s}$  for  $\#_1(a_s) \leq c$ , with high probability.

**Lemma 41.** Fix  $s \subset X$ ,  $a_s$  where  $z = \#_1(a_s) \leq c$ , and  $k \leq \log^2(n)$ , then

$$\Pr_{f \in \mathcal{R} \mathcal{F}^{n,k,m}} \left\{ |T_{a_s}| > 3m_{\max} \frac{k^z}{n^z} \right\} < e^{-c \log \log(n) k^c}.$$

**Proof:** We note that  $z \leq c < c \log(n) = k$ , thus we know the bounds of Observation 39 hold. Let  $f'$  be a random extension of  $f$  to  $m_{\max}$  terms.

$$\begin{aligned} &\Pr_{f \in \mathcal{R} \mathcal{F}^{n,k,m}} \{|T_{a_s}| \geq 3m_{\max} k^z / n^z\} \\ &\leq \Pr_{f' \in \mathcal{R} \mathcal{F}^{n,k,m_{\max}}} \{|T'_{a_s}| \geq 3m_{\max} k^z / n^z\} \\ &\leq \Pr_{f' \in \mathcal{R} \mathcal{F}^{n,k,m_{\max}}} \{|T'_{a_s}| \geq (1 + \delta) \mathbf{E}\{T'_{a_s}\}\} \\ &\leq e^{-u' \delta^2 / 4}, \end{aligned}$$

where  $\mu' = \mathbf{E}\{T'_{a_s}\}$  by Chernoff.

We can obtain an upper bound for this expression from a lower bound for its unnegated exponent. Let  $\delta = 2$ .

$$\begin{aligned} \mu' \delta^2 / 4 &= \mu' \\ &> m_{\max} k^z / 2n^z \\ &> 2n^c c \cdot \log \log(n) k^z / 2n^z \\ &> c \cdot \log \log(n) k^c \quad \text{Since } k^z / 2n^z \geq k^c / 2n^c. \end{aligned}$$

Therefore

$$\Pr_{f \in \mathcal{R} \mathcal{F}^{n,k,m}} \left\{ |T_{a_s}| > 3m_{\max} \frac{k^z}{n^z} \right\} \leq e^{-c \log \log(n) k^c}.$$

$\square$

**Corollary 42 (The Small  $z$  Property Holds with High Probability).** Therefore for  $s \subset t \in f$  where  $|s| \leq \beta(n) + 1$  and  $z = \#_1(a_s)$  then  $\Pr_{f \in \mathcal{R} \mathcal{F}^{n,k,m}} \{\exists a_s, 0 < \#_1(a_s) \leq c, |T_{a_s}| > 3m_{\max} \frac{k^z}{n^z}\} < n^{2c} \log(n) \left(\frac{1}{n}\right)^{\beta(n)-1}$ .

**Proof:** We assume  $c \geq 1$ ; if  $c < 1$  then there does not exist a  $z$  since  $0 < z \leq c < 1$  and  $z$  is an integer.

If  $c \geq 1$  then the number of  $s \subset t$  and  $a_s$  where  $z = \#_1(a_s) \leq c$  is bounded by

$$\begin{aligned} &m \sum_{|s|=1, \dots, \beta(n)+1} \binom{k}{|s|} \sum_{z=1, \dots, c} \binom{|s|}{z} \\ &< m_{\max} \cdot \beta(n) k^{\beta(n)+1} \cdot 2^{\beta(n)} \\ &< \frac{1}{3} n^{c+1}. \end{aligned}$$

(i.e. For a given term, the number of different sets of size  $s$  is  $\binom{k}{|s|}$ . The number of terms is  $m$ . For a given set,  $s$ , the number of ways to choose  $z$  items from the set is  $\binom{|s|}{z}$ .)

Thus, by Lemma 41:

$$\begin{aligned} \Pr_{f \in \mathcal{R}} \mathcal{F} \{ \exists a_s, \#_1(a_s) \leq c, |T_{a_s}| > 3m_{\max} \frac{k^z}{n^z} \} \\ < \frac{1}{3} n^{c+1} e^{-c \log \log(n) k^c} \\ < \frac{1}{3} n^{2c} \log(n) \left( \frac{1}{n} \right)^{\beta(n)-1} \end{aligned}$$

Therefore, with high probability, the small  $z$  property for  $f \in \mathcal{R} \mathcal{F}^{n,k,m}$  is proved.  $\square$

Next we prove the medium  $z$  property: that for  $\#_1(a_s)$  where  $c < \#_1(a_s) < \beta(n)$  then  $|T_{a_s}| < \beta(n)$  with high probability.

**Lemma 43.** *For fixed  $c$ , and sufficiently large  $n$ , let  $s \subset X$ , and  $a_s$  with  $\#_1(a_s) > c$  then*

$$\Pr_{f \in \mathcal{R}} \mathcal{F}^{n,k,m} \{ |T_{a_s}| \geq \beta(n) \} < \left( \frac{1}{n} \right)^{\beta(n)-1}$$

**Proof:** Let  $z = \#_1(a_s)$ .

If  $z > k$  then  $|T_{a_s}| = 0$ , since there does not exist a term with more than  $k$  variables.

If  $z \leq k$  then the probability a random term  $t \in T_{a_s}$  is  $\frac{\binom{n-|s|}{k-z}}{\binom{n}{k}} < \frac{k^z}{n^z}$ . Consequently,

$$\begin{aligned} \Pr_{f \in \mathcal{R}} \mathcal{F}^{n,k,m} \{ |T_{a_s}| \geq \beta(n) \} \\ < \sum_{j=\beta(n) \dots m} \binom{m}{j} \left( \frac{k^z}{n^z} \right)^j \left( 1 - \frac{k^z}{n^z} \right)^{m-j} \\ < \sum_{j=\beta(n) \dots \log(n)-1} \binom{m}{j} \left( \frac{k^z}{n^z} \right)^j \left( 1 - \frac{k^z}{n^z} \right)^{m-j} \\ + \sum_{j=\log(n) \dots m} \binom{m}{j} \left( \frac{k^z}{n^z} \right)^j \left( 1 - \frac{k^z}{n^z} \right)^{m-j} \\ < \log(n) \binom{m}{\beta(n)} \left( \frac{k^{c+1}}{n^{c+1}} \right)^{\beta(n)} \\ + m \binom{m}{\log(n)} \left( \frac{k^{c+1}}{n^{c+1}} \right)^{\log(n)} \\ < \left( \frac{1}{n} \right)^{\beta(n)-1} \end{aligned}$$

The third inequality follows from the observation that the sum is maximized for  $z = c + 1$ , and from Lemma 36. The fourth inequality follows from Observation 38.  $\square$

**Corollary 44** (The Medium  $z$  Property Holds with High Probability). *Therefore for  $s \subset t \in f$  where  $|s| \leq \beta(n) + 1$ ,  $\Pr_{f \in \mathcal{R}} \mathcal{F}^{n,k,m} \{ \exists a_s, c < \#_1(a_s) < \beta(n), |T_{a_s}| \geq \beta(n) \} < \frac{1}{3} n^{2c} \log(n) \left( \frac{1}{n} \right)^{\beta(n)-1}$  for sufficiently large  $n$ .*

**Proof:** The number of  $s \subset t$  and  $a_s$  where  $c < \#_1(a_s) <$

$\beta(n)$  is bounded by

$$\begin{aligned} m \sum_{|s|=[c], \dots, \beta(n)+1} \binom{k}{|s|} \sum_{z=c+1, \dots, [\beta(n)]} \binom{|s|}{z} \\ < m_{\max} \cdot \beta(n) k^{\beta(n)+1} \cdot 2^{\beta(n)} \\ \leq n^{c+1} \\ < \frac{1}{3} n^{2c} \log(n). \end{aligned}$$

(i.e.  $\sum_{|s|=[c], \dots, \beta(n)+1} \binom{k}{|s|}$  is the number of ways to find a subset of  $t \in f$  of size greater than  $c$  and less than or equal to  $\beta(n) + 1$ . The sum  $\sum_{z=c+1, \dots, [\beta(n)]} \binom{|s|}{z}$  is the number of ways to choose a set of size  $z$  from  $|s|$  elements.)

Therefore using Lemma 43, we know

$$\Pr_{f \in \mathcal{R}} \mathcal{F}^{n,k,m} \{ \exists a_s, c < \#_1(a_s) < \beta(n), |T_{a_s}| \geq \beta(n) \} \leq \frac{1}{3} n^{2c} \log(n) \left( \frac{1}{n} \right)^{\beta(n)-1}$$

Consequently, the medium  $z$  property is also satisfied by a random  $f \in \mathcal{R} \mathcal{F}^{n,k,m}$  with high probability.  $\square$

The large  $z$  property is that two terms in  $f$  overlap by at most  $\beta(n)$ ; we prove this with a counting argument. Jackson and Servedio's paper [7] has a similar lemma, Lemma (3.5).

**Lemma 45.** *Let  $s, s' \subseteq X$  be sets of  $k \leq \sqrt[3]{n}$  variables chosen independently at random, then the  $\Pr \{ |s \cap s'| \geq \beta(n) \} < \left( \frac{1}{n} \right)^{\beta(n)-1}$ .*

**Proof**

$$\begin{aligned} \Pr \{ |s \cap s'| \geq \beta(n) \} \\ = \sum_{j=\beta(n)}^k \frac{\binom{n}{j} \binom{n-j}{k-j} \binom{n-k}{k-j}}{\binom{n}{k}^2} \\ = \sum_{j=\beta(n)}^k \frac{(k^j)^2 (n-k)^{k-j}}{j! n^k} \\ \text{(The sum is maximized for } j = \beta(n). \text{)} \\ < k \frac{(k^{\beta(n)})^2}{\beta(n)! n^{\beta(n)}} \\ < \frac{1}{n^{\beta(n)-1}} \end{aligned}$$

$\square$

**Corollary 46** (The Large  $z$  Property Holds with High Probability). *Therefore,  $\Pr_{f \in \mathcal{R}} \mathcal{F}^{n,k,m} \{ \exists t, t' \in f, |t \cap t'| \geq \beta(n) \} < \frac{1}{3} n^{2c} \log(n) \left( \frac{1}{n} \right)^{\beta(n)-1}$ .*

**Proof:** The proof follows from noting that

$$\begin{aligned} \Pr_{f \in \mathcal{R}} \mathcal{F}^{n,k,m} \{ \exists t, t' \in f, |t \cap t'| \geq \beta(n) \} &\leq \binom{m}{2} \frac{1}{n^{\beta(n)-1}} \leq \\ &\binom{m_{\max}}{2} \frac{1}{n^{\beta(n)-1}} < \binom{m_{\max}}{2} \frac{1}{n^{\beta(n)-1}} \frac{1}{1 - \frac{\beta(n)(\beta(n)-1)}{2n}} \\ &< \frac{1}{3} n^{2c} \log(n) \left( \frac{1}{n} \right)^{\beta(n)-1} \end{aligned}$$

The third inequality follows from Observation 33 and  $n^{\beta(n)-1} - \frac{(\beta(n)-1)\beta(n)}{2} n^{\beta(n)-2} = n^{\beta(n)-1} \left( 1 - \frac{(\beta(n)-1)\beta(n)}{2n} \right)$ .

Since two terms in  $f \in \mathcal{R} \mathcal{F}^{n,k,m}$  share less than  $\beta(n)$  variables with high probability, a random  $f \in \mathcal{R} \mathcal{F}^{n,k,m}$  satisfies the large  $z$  property in being well behaved with high probability.  $\square$

Recalling Corollaries 42, 44, and 46 and the definition of “well behaved,” we note that  $f \in \mathcal{R} \mathcal{F}^{n,k,m}$  is well behaved with high probability.

## C Bounding $\Upsilon_s$

Next we present the proof of Lemma 28 that for  $f$  a well behaved monotone  $k$ -DNF function,  $m \leq 2^{k+1}c \log \log n$  and  $t \in f$  then  $|E_{1_t}^+ - E_{f \setminus \{t\}}^+| \geq 2^n \cdot \frac{1}{8 \log^{4c}(n)} \frac{1}{n^c}$ .

**Proof:** Divide  $f \setminus \{t\}$  into three disjoint sets,

- $T_{\text{disjoint}} = \{t' \in f \setminus \{t\} \mid t \cap t' = \emptyset\}$ ,
- $T_{\text{small}} = \{t' \in f \setminus \{t\} \mid 1 \leq |t' \cap t| \leq c\}$  and
- $T_{\text{not small}} = f \setminus (T_{\text{disjoint}} \cup T_{\text{small}})$ .

Looking only at examples in  $E_{1_t}^+$ , we now calculate the probability that each of these sets is not satisfied. Remembering that  $f$  is monotone, we note that if one set is not satisfied, it increases the chance another set is not satisfied. (i.e.  $\Pr_{e \in E} \{-t \mid \neg t'\} \geq \Pr_{e \in E} \{-t\}$  since if we know at least one variable is set to zero that increases the odds of another term to be set to zero if they share a variable.)

In the first case for  $T_{\text{disjoint}}$ ,

$$\begin{aligned} \Pr_{e \in E_{1_t}^+} \{\forall t' \in T_{\text{disjoint}}, \neg t'(e)\} &> (1 - \frac{1}{2^k})^m \\ &\geq (1 - \frac{1}{2^k})^{2^{k+1}c \log \log(n)} = \left(1 - \frac{1}{2^k}\right)^{2c \log \log(n)} \geq \\ &\frac{1}{4^{2c \log \log(n)}} = \frac{1}{\log^{4c}(n)}, \text{ by Observation 37.} \end{aligned}$$

In the second case, if  $t' \in T_{\text{small}}$  and  $r = t \cap t'$  with  $a_t$  such that  $X_{a_t} = r$  then by  $f$  being well behaved we know that  $|T_{a_t}| \leq 3m_{\max} \left(\frac{k|r|}{n|r|}\right)$ . Therefore

$$\begin{aligned} \Pr_{e \in E_{1_t}^+} \{\forall t' \in T_{\text{small}}, \neg t'(e)\} &> \prod_{r \subset t, 1 \leq |r| \leq c} \left(1 - \frac{2^{|r|}}{2^k}\right)^{3m_{\max} \left(\frac{k|r|}{n|r|}\right)} \\ &\geq \prod_{1 \leq |r| \leq c} \left(1 - \frac{2^{|r|}}{2^k}\right)^{2^{k+3}c \log \log(n) \left(\frac{k|r|}{n|r|}\right)^{\binom{k}{|r|}}} \\ &\geq \prod_{1 \leq |r| \leq c} \left(1 - \frac{2^{|r|}}{2^k}\right)^{2^{(k-|r|)} 2^{|r|+3}c \log \log(n) \left(\frac{k|r|}{n|r|}\right)^{\binom{k}{|r|}}} \\ &\geq \prod_{1 \leq |r| \leq c} \left(\frac{1}{4}\right)^{2^{|r|+3}c \log \log(n) \left(\frac{k|r|}{n|r|}\right)^{\binom{k}{|r|}}} \text{ By Obs. 37.} \\ &\geq \left(\frac{1}{4}\right)^{\frac{16c^2 \log \log(n) k^2}{n}}. \end{aligned}$$

The last inequality follows from noticing the product is maximized for  $|r| = 1$ , thus

$$\Pr_{e \in E_{1_t}^+} \{e \in \mathcal{R} E_{1_t}^+ \mid \forall t' \in T_{\text{small}}, \neg t'(e)\} > \frac{1}{4}.$$

We now bound the third case. Since  $f$  is well behaved, we know that a term in  $f$  overlaps another term by at most

$\beta(n)$  variables, and the number of terms overlapping by a set  $r \subset t$  in  $T_{\text{not small}}$  is at most  $\beta(n)$ . Therefore

$$\begin{aligned} \Pr_{e \in E_{1_t}^+} \{\forall t' \in T_{\text{not small}}, \neg t'(e)\} &> \prod_{r \subset t, c < |r| \leq \beta(n)} \left(1 - \frac{2^{|r|}}{2^k}\right)^{\beta(n)} \\ &> \left(1 - \frac{2^{\beta(n)}}{2^k}\right)^{\beta^2(n) \binom{k}{\beta(n)}} \geq \frac{1}{2}, \text{ (since } \binom{k}{|r|} \leq \binom{k}{\beta(n)} \text{).} \\ \text{Therefore (remembering } 2^k = n^c \text{)} &|\{e \in \mathcal{R} E_{1_t}^+ \mid \forall t' \in f \setminus \{t\}, \neg t'(e)\}| \\ &> 2^{n-k} \cdot \left(\frac{1}{4}\right) \left(\frac{1}{\log^{4c}(n)}\right) \left(\frac{1}{2}\right) = 2^n \cdot \frac{1}{8 \log^{4c}(n)} \frac{1}{n^c}. \end{aligned}$$

□

## APPENDIX 2

In the next two sections we present the standard arguments for the sake of completeness. In Section A we prove that we can sample to find a sufficient approximation to  $I_s(E^+)$ . In Section B we prove that our very straightforward algorithm runs in polynomial time and produces  $f$ .

### A Sampling and Approximating $I_s$

In Section 4, we proved we could determine if a set,  $s$ , is a subset of a term if  $c + 2 \leq |s| \leq \beta(n) + 1$  by computing  $I_s$ . Unfortunately, we cannot efficiently compute  $I_s$  since we cannot efficiently compute  $|E_{a_s}^+|$ . Instead, we show how to approximate  $I_s$ . We estimate this value by sampling  $g_s$  uniformly chosen labeled examples from  $E$ .

**Definition 47.** For  $s \subset X$ , let  $E_{\text{Sample}(g_s)} \subset E$  be a random sample of  $g_s$  labeled examples drawn uniformly from  $E$ .

**Definition 48.** Given  $E_{\text{Sample}(g_s)} \subset E$ , let  $E_{\text{Sample}(g_s)}^+ = E_{\text{Sample}(g_s)} \cap E^+$  be the set of positive examples in  $E_{\text{Sample}(g_s)}$ . Similarly, let  $\Upsilon_{\text{Sample}(g_s)} = E_{\text{Sample}(g_s)} \cap \Upsilon_s$  be the set of positive examples from  $E_{\text{Sample}(g_s)}$  which satisfy only terms in  $T_{1_s}$ .

**Observation 49.** Let  $s \subset X$ , we note that  $\mathbf{E}(|\Upsilon_{\text{Sample}(g_s)}|) = g_s \cdot \frac{|\Upsilon_s|}{|E|} = \frac{g_s}{2^n} |\Upsilon_s|$ .

Using sampled labeled examples, we compute the following function to approximate  $I_s$ .

**Definition 50.** Let  $s \subset X$ , we define  $I_{s, \text{Sample}(g_s)} = \sum_{e \in E_{\text{Sample}(g_s)}^+} (-1)^{\#_0(a_s(e))}$  to be our approximation of  $I_s$ , where  $a_s(e)$  is  $e|_s$ .

**Observation 51.** We note that the  $\mathbf{E}(I_{s, \text{Sample}(g_s)}) = \frac{g_s}{2^n} I_s$ .

This observation follows since the expected value of  $|E_{\text{Sample}(g_s)} \cap E_{a_s}^+|$  is  $g_s \frac{|E_{a_s}^+|}{|E|}$  and

$$\mathbf{E}(I_{s, \text{Sample}(g_s)}) = \sum_{a_s} (-1)^{\#_0(a_s)} g_s \frac{|E_{a_s}^+|}{|E|}.$$

Next, we bound how different our sampled  $I_{s, \text{Sample}(g_s)}$  is from the expected value. As we have not yet provided a lower tail bound, we state it next, as it is described in Canny [3].

**Lemma 52** (Chernoff). *Let  $\delta \in (0, 1]$ ,  $\mu$  be the expected value, and  $\chi$  be a series of independent Poisson trials then  $\Pr \{\chi < (1 - \delta)\mu\} < e^{-\mu\delta^2/2}$ .*

Applying the lower and upper Chernoff bounds from Lemmas 40 and 52, we prove that  $I_{s, \text{Sample}(g_s)}$  is within  $\frac{2g_s}{n^{c+1}}$  fraction of  $\mathbf{E}(I_{s, \text{Sample}(g_s)})$ .

**Lemma 53.** *For  $g_s = n^{2c+3}2^{k+|s|}$  and given access to examples drawn from a well behaved monotone  $k$ -DNF then  $|I_{s, \text{Sample}(g_s)} - \mathbf{E}(I_{s, \text{Sample}(g_s)})| < g_s \cdot \frac{2}{n^{c+1}}$  with probability  $1 - 4e^{-n/4}$ .*

**Proof:** To apply the Chernoff bounds, our main difficulty is our sum has both positive and negative values, we overcome this difficulty by bounding the positive and negative values separately. We define two indicator functions. Let  $r_{\text{even}}(e) = 1$  iff  $f(e) = 1$  and  $\#_0(e_{|s|})$  is even, and let  $r_{\text{odd}}(e) = 1$  iff  $f(e) = 1$  and  $\#_0(e_{|s|})$  is odd.

Let  $E_{\text{Sample}(g)}$  be a randomly generated set of  $g_s$  examples from  $E$ . Let  $X_{\text{even}} = \sum_{e \in E_{\text{Sample}(g_s)}} r_{\text{even}}(e)$ . (Similarly for  $X_{\text{odd}}$ .)

We observe that  $I_{s, \text{Sample}(g_s)} = X_{\text{even}} - X_{\text{odd}}$ .

If  $\exists a_s$  such that  $E_{a_s}^+ \neq \emptyset$ , then there is a term consistent with at least one  $a_s$ . This term satisfies the examples in  $E_{a_s}$  with probability at least  $\frac{1}{2^k}$ . There are  $2^{|s|}$  different  $a_s$ , thus if  $\#_0(a_s)$  is even, we expect at least  $\frac{1}{2^{|s|}2^k}$  fraction of total examples are set to one by  $r_{\text{even}}$ . Therefore in  $g_s = n^{2c+3}2^{k+|s|}$  examples, the expected value of the indicator function is either zero, or the expected value is at least  $n^{2c+3}$ . (Similarly for the case where  $\#_0(a_s)$  is odd.)

Using the Chernoff bounds with  $\delta = \frac{1}{n^{c+1}}$ , we bound  $\mathbf{E}(X_{\text{even}})$ , in the cases where the expected value is not zero.

$\Pr \{|X_{\text{even}} - (1 \pm \delta)\mathbf{E}(X_{\text{even}})\} \leq 2e^{-\frac{n^{2c+3}}{4n^{2c+2}}} = 2e^{-n/4}$ . (Similarly for  $\mathbf{E}(X_{\text{odd}})$ .) Consequently, the indicator functions will be  $\frac{1}{n^{c+1}}$  close to their respective expected value functions.

Therefore we know  $|(X_{\text{even}} - X_{\text{odd}}) - \mathbf{E}(I_{s, \text{Sample}(g_s)})| \leq \frac{1}{n^{c+1}}(\mathbf{E}(X_{\text{even}}) + \mathbf{E}(X_{\text{odd}})) \leq g_s \frac{2}{n^{c+1}}$ . Thus  $I_{s, \text{Sample}(g_s)}$  differs from  $\mathbf{E}(I_{s, \text{Sample}(g_s)})$  by at most  $g_s \cdot \frac{2}{n^{c+1}}$  with high probability.  $\square$

Using the previous Lemma 53, Theorem 26, and Observation 49, we note that we can determine if  $s \subset X$  is a subset of a term in a well behaved monotone  $k$ -DNF function by sampling labeled examples from the uniform distribution.

**Lemma 54.** *Let  $f$  be a well behaved monotone  $k$ -DNF formula,  $s \subset X$  where  $c + 2 \leq |s| \leq \beta(n) + 1$ , and  $g_s = n^{2c+3}2^{k+|s|}$ ,*

- if  $s \subset t \in f$  then  $I_{s, \text{Sample}(g_s)} > g_s \cdot \frac{1}{n^{c+\frac{1}{5}}}$  with probability  $1 - 4e^{-n/4}$ .
- If  $s \not\subset t \in f$  then  $I_{s, \text{Sample}(g_s)} < g_s \cdot \frac{1}{n^{c+\frac{1}{5}}}$  with probability  $1 - 4e^{-n/4}$ .

**Proof:** From Lemma 53 and Observation 51, we know

$$-\frac{2g_s}{n^{c+1}} + \frac{g_s}{2^n} I_s \leq I_{s, \text{Sample}(g_s)} \leq \frac{2g_s}{n^{c+1}} + \frac{g_s}{2^n} I_s$$

### Algorithm Learn Random Monotone DNF

1.  $S = \text{Distinguishing Subsets}$
2.  $f = \emptyset$
3. For  $s \in S$ 
  - (a)  $t = \emptyset$
  - (b) For  $x \in X$ 
    - If  $I_{s \cup \{x\}, \text{Sample}(g_s)} > g_s \cdot \frac{1}{n^{c+\frac{1}{5}}}$  then add  $x$  to  $t$
  - (c) add  $t$  to  $f$
4. Return  $f$

Figure 3:

### Function Distinguishing Subsets

1.  $S = \{s \subset X \mid |s| = c + 2, \text{ and } I_{s, \text{Sample}(g_s)} > g_s \cdot \frac{1}{n^{c+\frac{1}{5}}}\}$
2. For  $i = (c + 3)$  to  $\beta(n)$ 
  - (a)  $S' = \emptyset$
  - (b) For  $s \in S$  and  $x \in X$ 
    - If  $I_{s \cup \{x\}, \text{Sample}(g_s)} > g_s \cdot \frac{1}{n^{c+\frac{1}{5}}}$  then add  $(s \cup \{x\})$  to  $S'$
  - (c)  $S = S'$
3. Return  $S$

Figure 4:

with probability greater than  $1 - 4e^{-n/4}$ .

By Theorem 30, Observation 51, and Lemma 53 we know:

- if  $s \subset t \in f$  then  $I_s \geq 2^n \cdot \frac{1}{8 \log^{4c}(n)} \frac{1}{n^c} - 2^n \cdot \frac{4k \log^6(n)n^{2/3}}{n^{c+1}}$ . Thus,  $I_{s, \text{Sample}(g_s)} \geq -g_s \cdot \frac{2}{n^{c+1}} + g_s \cdot I_s \geq -g_s \cdot \frac{2}{n^{c+1}} + g_s \cdot \frac{1}{8 \log^{4c}(n)} \frac{1}{n^c} - g_s \cdot \frac{4k \log^6(n)n^{2/3}}{n^{c+1}} > g_s \cdot \frac{1}{n^{c+\frac{1}{5}}}$  with probability greater than  $1 - 4e^{-n/4}$ .
- If  $s \not\subset t \in f$  then  $I_s \leq 2^n \cdot \frac{4k \log^6(n)n^{2/3}}{n^{c+1}}$ . Thus,  $I_{s, \text{Sample}(g_s)} \leq g_s \cdot \frac{2}{n^{c+1}} + g_s \cdot I_s \leq g_s \cdot \frac{2}{n^{c+1}} + g_s \cdot \frac{4k \log^6(n)n^{2/3}}{n^{c+1}} < g_s \cdot \frac{1}{n^{c+\frac{1}{5}}}$  with probability greater than  $1 - 4e^{-n/4}$ .  $\square$

## B Learning Random Monotone DNF by Finding Terms in Polynomial Time

Next, we restate our algorithm to use  $I_{\text{Sample}(g_s)}$ .

Referring to our algorithm in Figures 3 and 4, the lemmas, and theorems in the previous sections, we prove our

algorithm discovers the unknown well behaved monotone  $k$ -DNF from random examples drawn from the uniform distribution with high probability in polynomial time. We show this by following the steps our algorithm takes; first our algorithm finds all  $(c+2)$ -sized subsets of  $s$  in time  $O(g_{c+2}n^{c+2})$  with probability greater than  $1 - 4n^{c+2}e^{-n/4}$ . Next, given all  $(c+2)$ -sized subsets of terms in  $f$ , our algorithm grows those subsets till they are of size  $\beta(n)$  with probability greater than  $1 - 4nmk^{\beta(n)}e^{-n/4}$  in time  $O(nmg_{\beta(n)}k^{\beta(n)})$ . Finally, given a subset of a term of size  $\beta(n)$ , our algorithm discovers all the variables in that term in time  $mng_{\beta(n)+1}k^{\beta(n)}$  with probability at least  $1 - mnk^{\beta(n)}(4e^{n/4})$ .

**Observation 55.** For  $s \subset X$ , computing  $I_{s, \text{Sample}(g_s)}$  takes time  $O(g_s)$ .

In step 1, our algorithm finds all the  $(c+2)$ -sized subsets of terms in  $f$ .

**Lemma 56.** Given a well behaved  $f \in \mathcal{F}^{n,k,m}$ , our function **Distinguishing Subsets** finds  $\{s \mid s \subset t \in f, |s| = c + 2\}$  in time  $O(g_{c+2}n^{c+2})$  with probability greater than  $1 - 4n^{c+2}e^{-n/4}$  in step 1.

**Proof:** Let  $s \subset X$  where  $|s| = c + 2$ . By Lemma 54, iff  $s \subset t \in f$  then  $I_{s, \text{Sample}(g_s)} \geq g_s \cdot \frac{1}{n^{c+\frac{1}{5}}}$  with probability greater than  $1 - 4e^{-n/4}$ . Function **Distinguishing Subsets** tests all subsets of size  $c + 2$ , thus our function has correctly selected the sets which are subset of terms in  $f$  with probability greater than  $1 - 4n^{c+2}e^{-n/4}$  in time  $O(g_{c+2}n^{c+2})$ .  $\square$

Having found all subsets of  $t \in f$  of size  $c + 2$  with high probability, our algorithm builds these sets till all the subsets of terms has size  $\beta(n)$ .

**Lemma 57.** Given a well behaved  $f \in \mathcal{F}^{n,k,m}$ , and  $T = \{s \mid s \subset t \in f, |s| = c + 2\}$ , function **Distinguishing Subsets** in step 2 returns  $\{s \mid s \subset t \in f, |s| = \beta(n)\}$  with probability greater than  $1 - 4mnk^{\beta(n)}e^{-n/4}$  in time bounded by  $O(nmg_{\beta(n)}k^{\beta(n)})$ .

**Proof:** Using the result of Lemma 54, each iteration of our loop is given a set  $S = \{s \mid s \subset t \in f, |s| = i\}$  and produces  $S' = \{s \mid s \subset t \in f, |s| = i + 1\}$  with probability more than  $1 - 4nm \binom{k}{i} e^{-n/4}$  for  $i = c+2 \dots \beta(n) - 1$  in time bounded by  $O(g_i n m k^i)$ . Thus in  $\beta(n) - 1 - (c+2)$  iterations our algorithm produces  $S = \{s \mid s \subset t \in f, |s| = \beta(n)\}$  with probability greater than  $1 - 4\beta(n)nmk^{\beta(n)-1}e^{-n/4}$  in time bounded by  $O(nmg_{\beta(n)}k^{\beta(n)-1})$ .  $\square$

Given all  $\beta(n)$ -sized subsets of  $t \in f$ , algorithm **Learn Random Monotone DNF** finds all the terms of  $f$ .

**Lemma 58.** Given a well behaved  $f \in \mathcal{F}^{n,k,m}$ , and  $S = \{s \mid s \subset t \in f, |s| = \beta(n)\}$  our algorithm, **Learn Random Monotone DNF**, finds  $f$  in time bounded by  $O(mng_{\beta(n)+1}k^{\beta(n)})$  with probability greater than  $1 - nmk^{\beta(n)}(4e^{-n/4})$  in step 3.

**Proof:** Algorithm **Learn Random Monotone DNF** uses Corollary 46 and Lemma 54.

Corollary 46 states that, for a well behaved monotone  $k$ -DNF,  $\forall s \in S$  where  $|s| \geq \beta(n)$  then  $|\{t \mid s \subset t \in f\}| \leq 1$ . Thus every  $s \in S$  is associated with at most one term  $t \in f$ .

Lemma 54 states that for a given  $s \subset X$  and  $x \in X$  where  $|s \cup \{x\}| = \beta(n) + 1$  iff  $I_{s \cup \{x\}, \text{Sample}(g_s)} \geq g_s \cdot \frac{1}{n^{c+\frac{1}{5}}}$  then  $s \cup \{x\} \subset t \in f$  with probability at least  $1 - 4e^{-n/4}$ . Thus for  $|s| = \beta(n)$ ,  $\exists! t$  such that  $s \subset t$ , we can determine if  $x \in t$  with high probability.

Combining these ideas, given  $s \in S$  we can find a term in the inside loop of step 3 by testing every  $x \in X$  to determine if  $\{x\} \cup s \subset t \in f$ , and thus find  $\{x \mid I_{s \cup \{x\}, \text{Sample}(g_s)} \geq g_s \cdot \frac{1}{n^{c+\frac{1}{5}}}\} = t \in f$  in time  $O(g_{\beta(n)+1}n)$  with probability greater than  $1 - 4ne^{-n/4}$ .

Together, the outside loop in step 3 selects every  $s \in S$  and the inside loop finds  $t$  where  $s \subset t$ . Since  $\forall t \in f$ , there exists  $s \in S$  such that  $s \subset t$ , Algorithm **Learn Random Monotone DNF** produces  $f$ .

The time it takes to do this is the time is bounded by  $O(g_{\beta(n)+1}nmk^{\beta(n)})$  with probability bounded by

$$1 - 4nmk^{\beta(n)}e^{-n/4}.$$

$\square$

**Theorem 59.** Given a well behaved  $f \in \mathcal{F}^{n,k,m}$ , Algorithm **Learn Random Monotone DNF** finds  $f$  in time bounded by  $O(mng_{\beta(n)+1}k^{\beta(n)})$  with probability greater than  $1 - 9mnk^{\beta(n)}e^{-n/4}$ .

**Proof:** Using Lemmas 56, 57, 58 we have proven that our algorithm finds all subsets of size  $c + 2$  of terms in  $f$  in Lemma 56, and having found these subsets it builds upon till our algorithm has found all subsets of terms of  $f$  of size  $\beta(n)$  in Lemma 57; it then uses the uniqueness of terms of size  $\beta(n)$  to find all the variables of a term in  $f$ ; thus finding the entire function.

The algorithm runs in time bounded by

$$O(mng_{\beta(n)+1}k^{\beta(n)})$$

with probability greater than  $1 - 9mnk^{\beta(n)}e^{-n/4}$ .  $\square$

---

# Polynomial regression under arbitrary product distributions

---

Eric Blais\* and Ryan O’Donnell and Karl Wimmer

Carnegie Mellon University

{eblais@cs, odonnell@cs, kwimmer@andrew}.cmu.edu

## Abstract

In recent work, Kalai, Klivans, Mansour, and Servedio [KKMS05] studied a variant of the “Low-Degree (Fourier) Algorithm” for learning under the uniform probability distribution on  $\{0, 1\}^n$ . They showed that the  $L_1$  polynomial regression algorithm yields *agnostic* (tolerant to arbitrary noise) learning algorithms with respect to the class of threshold functions — under certain restricted instance distributions, including uniform on  $\{0, 1\}^n$  and Gaussian on  $\mathbb{R}^n$ . In this work we show how *all* learning results based on the Low-Degree Algorithm can be generalized to give almost identical agnostic guarantees under *arbitrary* product distributions on instance spaces  $X_1 \times \dots \times X_n$ . We also extend these results to learning under *mixtures* of product distributions.

The main technical innovation is the use of (Hoeffding) orthogonal decomposition and the extension of the “noise sensitivity method” to arbitrary product spaces. In particular, we give a very simple proof that threshold functions over arbitrary product spaces have  $\delta$ -noise sensitivity  $O(\sqrt{\delta})$ , resolving an open problem suggested by Peres [Per04].

## 1 Introduction

In this paper we study binary classification learning problems over arbitrary instance spaces  $\mathcal{X} = X_1 \times \dots \times X_n$ . In other words, each instance has  $n$  “categorical attributes”, the  $i$ th attribute taking values in the set  $X_i$ . For now we assume that each  $X_i$  has cardinality at most  $\text{poly}(n)$ .<sup>1</sup>

It is convenient for learning algorithms to encode instances from  $\mathcal{X}$  as vectors in  $\{0, 1\}^{|X_1|+\dots+|X_n|}$  via the “one-out-of- $k$  encoding”; e.g., an attribute from  $X_1 = \{\text{red, green, blue}\}$  is replaced by one of  $(1, 0, 0)$ ,  $(0, 1, 0)$ , or  $(0, 0, 1)$ . Consider now the following familiar learning algorithm:

Given  $m$  examples of training data  $(\vec{x}_1, y_1), \dots, (\vec{x}_m, y_m) \in \mathcal{X} \times \{-1, 1\}$ ,

1. Expand each instance  $\vec{x}_i$  into a vector from  $\{0, 1\}^{|X_1|+\dots+|X_n|}$  via the “one-out-of- $k$ ” encoding.
2. Consider “features” which are products of up to  $d$  of the new 0-1 attributes.
3. Find the linear function  $W$  in the feature space that best fits the training labels under some loss measure  $\ell$ : e.g., squared loss, hinge loss, or  $L_1$  loss.
4. Output the hypothesis  $\text{sgn}(W - \theta)$ , where  $\theta \in [-1, 1]$  is chosen to minimize the hypothesis’ training error.

We will refer to this algorithm as “degree- $d$  polynomial regression (with loss  $\ell$ )”. When  $\ell$  is the hinge loss, this is equivalent to the soft margin SVM algorithm with the degree- $d$  polynomial kernel and no regularization [CV95].<sup>2</sup> When  $\ell$  is the squared loss and the data is drawn i.i.d. from the uniform distribution on  $\mathcal{X} = \{0, 1\}^n$ , the algorithm is effectively equivalent to the Low-Degree Algorithm of Linial, Mansour, and Nisan [LMN93] — see [KKMS05]. Using techniques from convex optimization (indeed, linear programming for  $L_1$  or hinge loss, and just basic linear algebra for squared loss), it is known that the algorithm can be performed in time  $\text{poly}(m, n^d)$ . For all known proofs of good generalization for the algorithm,  $m = n^{\Theta(d)}/\epsilon$  training examples are necessary (and sufficient). Hence we will view the degree- $d$  polynomial regression algorithm as requiring  $\text{poly}(n^d/\epsilon)$  time and examples. (Because of this, whether or not one uses the “kernel trick” is a moot point.)

Although SVM-based algorithms are very popular in practice, the scenarios in which they *provably* learn successfully are relatively few (see Section 1.2 below) — especially when there is error in the labels. Our goal in this paper is to broaden the class of scenarios in which learning with polynomial regression has provable, polynomial-time guarantees.

<sup>2</sup>Except for the minor difference of choosing an optimal  $\theta$  rather than fixing  $\theta = 0$ .

\*Supported in part by a scholarship from the Fonds québécois de recherche sur la nature et les technologies.

<sup>1</sup>Given real-valued attributes, the reader may think of bucketing them into  $\text{poly}(n)$  buckets.

## 1.1 The learning framework

We study binary classification learning in the natural “agnostic model” [KSS94] (sometimes described as the model with arbitrary classification noise). We assume access to training data drawn i.i.d. from some distribution  $\mathcal{D}$  on  $\mathcal{X}$ , where the labels are provided by an arbitrary unknown “target” function  $t : \mathcal{X} \rightarrow \{-1, 1\}$ . The task is to output a hypothesis  $h : \mathcal{X} \rightarrow \{-1, 1\}$  which is a good predictor on future examples from  $\mathcal{D}$ . We define the “error of  $h$ ” to be  $\text{err}(h) = \Pr_{\mathbf{x} \sim \mathcal{D}}[h(\mathbf{x}) \neq t(\mathbf{x})]$ .<sup>3</sup> We compare the error of an algorithm’s hypothesis with the best error achievable among functions in a fixed class  $\mathcal{C}$  of functions  $\mathcal{X} \rightarrow \{-1, 1\}$ . Define  $\text{Opt} = \inf_{f \in \mathcal{C}} \text{err}(f)$ . We say that an algorithm  $\mathcal{A}$  “agnostically learns with respect to  $\mathcal{C}$ ” if, given  $\epsilon > 0$  and access to training data, it outputs a hypothesis  $h$  which satisfies  $\mathbf{E}[\text{err}(h)] \leq \text{Opt} + \epsilon$ . Here the expectation is with respect to the training data drawn.<sup>4</sup> The running time (and number of training examples) used are measured as functions of  $n$  and  $\epsilon$ .

Instead of an instance distribution  $\mathcal{D}$  on  $\mathcal{X}$  and a target  $t : \mathcal{X} \rightarrow \{-1, 1\}$ , one can more generally allow a distribution  $\mathcal{D}'$  on  $\mathcal{X} \times \{-1, 1\}$ ; in this case,  $\text{err}(h) = \Pr_{(\mathbf{x}, y) \sim \mathcal{D}'}[h(\mathbf{x}) \neq y]$ . Our learning results also hold in this model just as in [KKMS05]; however we use the simpler definition for ease of presentation, except in Section 5.3.

In the special case when  $t$  is promised to be in  $\mathcal{C}$  we are in the scenario of PAC learning [Val84]. This corresponds to the case  $\text{Opt} = 0$ . Since  $\mathcal{C}$  is usually chosen (by necessity) to be a relatively simple class, the PAC model’s assumption that there is a perfect classifier in  $\mathcal{C}$  is generally considered somewhat unrealistic. This is why we work in the agnostic model.

Finally, since strong hardness results are known [KSS94, LBW95, KKMS05, GR06] for agnostic learning under general distributions  $\mathcal{D}$ , we are forced to make some distributional assumptions. The main assumption in this paper is that  $\mathcal{D}$  is a *product probability distribution* on  $\mathcal{X}$ ; i.e., the  $n$  attributes are independent. For a discussion of this assumption and extensions, see Section 1.3.

## 1.2 When polynomial regression works

Although the SVM algorithm is very popular in practice, the scenarios in which it provably learns successfully are relatively few. Let us consider the SVM algorithm with degree- $d$  polynomial kernel. The traditional SVM analysis is predicated on the assumption that the data is perfectly linearly separable in the polynomial feature space. Indeed, the heuristic arguments in support of good generalization are predicated on the data being separable *with large margin*. Even just the assumption of perfect separation may well be unreasonable. For example, suppose the target  $t$  is the very simple

function given by the intersection of two homogeneous linear threshold functions over  $\mathbb{R}^n$ ; i.e.,

$$t : \mathbb{R}^n \rightarrow \{-1, 1\}, \quad t(\mathbf{x}) = \text{sgn}(w_1 \cdot \mathbf{x}) \wedge \text{sgn}(w_2 \cdot \mathbf{x}).$$

It is known [MP69] that this target cannot be classified by the sign of a degree- $d$  polynomial in the attributes for *any* finite  $d$ ; this holds even when  $n = 2$ . Alternatively, when  $t$  is the intersection of two linear threshold functions over  $\{0, 1\}^n$ , it is not currently known if  $t$  can be classified by the sign of a degree- $d$  polynomial for any  $d < n - 1$ . [OS03]

Because of this problem, one usually considers the “soft margin SVM algorithm” [CV95]. As mentioned, when this is run with no “regularization”, the algorithm is essentially equivalent to degree- $d$  polynomial regression with hinge loss. To show that this algorithm even has a chance of learning efficiently, one must be able to show that simple target functions can at least be *approximately* classified by the sign of low-degree polynomials. Of course, even stating any such result requires distributional assumptions. Let us make the following definition:

**Definition 1.1** *Let  $\mathcal{D}$  be a probability distribution on  $\{0, 1\}^N$  and let  $t : \{0, 1\}^N \rightarrow \mathbb{R}$ . We say that  $t$  is  $\epsilon$ -concentrated up to degree  $d$  (under  $\mathcal{D}$ ) if there exists a polynomial  $p : \{0, 1\}^N \rightarrow \mathbb{R}$  of degree at most  $d$  which has squared loss at most  $\epsilon$  under  $\mathcal{D}$ ; i.e.,  $\mathbf{E}_{\mathbf{x} \sim \mathcal{D}}[(p(\mathbf{x}) - t(\mathbf{x}))^2] \leq \epsilon$ .*

It is well known that under the above conditions,  $h := \text{sgn}(p)$  has classification error at most  $\epsilon$  under  $\mathcal{D}$ . Further, it is relatively easy to show that if  $\mathcal{C}$  is a class of functions each of which is  $\epsilon$ -concentrated up to degree  $d$ , then the degree- $d$  polynomial regression algorithm with squared loss will PAC-learn  $\mathcal{C}$  to accuracy  $O(\epsilon)$  under  $\mathcal{D}$ .

The first result along these lines was due to Linial, Mansour, and Nisan [LMN93] who introduced the “Low-Degree Algorithm” for PAC-learning under the uniform distribution on  $\{0, 1\}^n$ . They showed that if  $f : \{0, 1\}^n \rightarrow \{-1, 1\}$  is computed by a circuit of size  $s$  and depth  $c$  then it is  $\epsilon$ -concentrated up to degree  $(O(\log(s/\epsilon)))^c$  under the uniform distribution. Some generalizations of this result [FJS91, Hås01] are discussed in Section 4.

Another result using this idea was due to Klivans, O’Donnell, and Servedio [KOS04]. They introduced the “noise sensitivity method” for showing concentration results under the uniform distribution on  $\{0, 1\}^n$ . In particular, they showed that any  $t : \{0, 1\}^n \rightarrow \{-1, 1\}$  expressible as a function of  $k$  linear threshold functions is  $\epsilon$ -concentrated up to degree  $O(k^2/\epsilon^2)$  under the uniform distribution.

These works obtained PAC learning guarantees for the polynomial regression algorithm — i.e., guarantees only holding under the somewhat unrealistic assumption that  $\text{Opt} = 0$ . A significant step towards handling noise was taken in [KKMS05]. Therein it was observed that low-degree  $L_2^2$ -approximability bounds imply  $L_1$ -approximability bounds (and hinge loss bounds), and further, such bounds imply that the polynomial regression algorithm works in the *agnostic* learning model. Specifically, their work contains the following theorem:

<sup>3</sup>In this paper, boldface denotes random variables.

<sup>4</sup>The definition of agnostic learning is sometimes taken to require error at most  $\text{Opt} + \epsilon$  with high probability, rather than in expectation. However this is known [KKMS05] to require almost negligible additional overhead.

**Theorem 1.2** ([KKMS05]) *Let  $\mathcal{D}$  be a distribution on  $\{0, 1\}^N$  and let  $\mathcal{C}$  be a class of functions  $\{0, 1\}^N \rightarrow \{-1, 1\}$  each of which is  $\epsilon^2$ -concentrated up to degree  $d$  under  $\mathcal{D}$ . Then the degree- $d$  polynomial regression algorithm with  $L_1$  loss (or hinge loss [Kal06]) uses  $\text{poly}(N^d/\epsilon)$  time and examples, and agnostically learns with respect to  $\mathcal{C}$  under  $\mathcal{D}$ .*

Thus one gets agnostic learning algorithms under the uniform distribution on  $\{0, 1\}^n$  with respect to the class of  $\text{AC}^0$  circuits (time  $n^{\text{poly}(\log(n/\epsilon))}$ ) and the class of functions of  $k$  thresholds (time  $n^{O(k^2/\epsilon^4)}$ ) — note that the latter is polynomial time assuming  $k$  and  $\epsilon$  are constants. Kalai et al. also obtained related results for agnostically learning with respect to single threshold functions under Gaussian and log-concave distributions on  $\mathbb{R}^n$ .

### 1.3 Overview of our learning results

We view the work of [KKMS05] as the first provable guarantee that one can learn interesting, broad classes of functions under the realistic noise model of agnostic learning (and in particular, that SVM-type methods can have this guarantee). One shortcoming of the present state of knowledge is that we have good concentration bounds for classes essentially only with respect to the uniform distribution on  $\{0, 1\}^n$  and the Gaussian distribution on  $\mathbb{R}^n$ .<sup>5</sup>

In this work we significantly broaden the class of distributions for which we can prove good concentration bounds, and hence for which we can prove the polynomial regression algorithm performs well. Roughly speaking, we show how to generalize any concentration result for the uniform distribution on  $\{0, 1\}^n$  into the same concentration result for *arbitrary product distributions*  $\mathcal{D}$  on instance spaces  $\mathcal{X} = X_1 \times \dots \times X_n$ .

We believe this is a significant generalization for several reasons. First, even just for the instance space  $\{0, 1\}^n$  the class of arbitrary product distributions is much more reasonable than the single distribution in which each attribute is 0 or 1 with probability exactly 1/2. Our results are even stronger than this, though: they give an algorithm that works simultaneously for any product distribution over *any* instance space  $\mathcal{X} = X_1 \times \dots \times X_n$  where each  $|X_i| \leq \text{poly}(n)$ .

Because we can handle non-binary attributes, the restriction to product spaces becomes much less severe. A common criticism of learning results under the uniform distribution or product distributions on  $\{0, 1\}^n$  is that they make the potentially unreasonable assumption that attributes are independent. However with our results, one can somewhat circumvent this. Suppose one believes that the attributes  $X_1, \dots, X_n$  are mostly independent, but some groups of them (e.g., height and weight) have mutual dependencies. One can then simply group together any dependent attribute sets  $X_{i_1}, \dots, X_{i_t}$  into a single “super-attribute” set  $(X_{i_1} \times \dots \times X_{i_t})$ . Assuming that this eliminates dependencies — i.e., the new (super-)attributes are all independent — and that each

<sup>5</sup>[FJS91] gives bounds for  $\text{AC}^0$  under constant-bounded product distributions on  $\{0, 1\}^n$ ; [KKMS05] gives inexplicit bounds for a single threshold function under log-concave distributions on  $\mathbb{R}^n$ .

$|X_{i_1} \times \dots \times X_{i_t}|$  is still at most  $\text{poly}(n)$ , one can proceed to use the polynomial regression algorithm. Here we see the usefulness of being able to handle arbitrary product distributions on arbitrary product sets.

In many reasonable cases our results can also tolerate the attribute sets  $X_i$  having superpolynomial size. What is really necessary is that the probability distribution on each  $X_i$  is mostly concentrated on polynomially many attributes. Indeed, we can further handle the common case when attributes are real-valued. As long as the probability distributions on real-valued attributes are not extremely skewed (e.g., Gaussian, exponential, Laplace, Pareto, chi-square, ...) our learning results go through after doing a naive “bucketing” scheme.

Finally, being able to learn under arbitrary product distributions opens the door to learning under *mixtures of product distributions*. Such mixtures — especially mixtures of Gaussians — are widely used as data distribution models in learning theory. We show that agnostic learning under mixtures can be reduced to agnostic learning under single product distributions. If the mixture distribution is precisely known to the algorithm, it can learn even under a mixture of polynomially many product distributions. Otherwise, when the mixture is unknown, we first need to use an algorithm for learning (or clustering) a mixture of product distributions from unlabeled examples. This is a difficult but well-studied problem. Using results of Feldman, O’Donnell, and Servedio [FOS05, FOS06] we can extend all of our agnostic learning results to learning under mixtures of constantly many product distributions with each  $|X_i| \leq O(1)$  and constantly many (axis-aligned) Gaussian distributions.

### 1.4 Outline of technical results

In Section 2 we recall the orthogonal decomposition of functions on product spaces, as well as the more recently-studied notions of concentration and noise sensitivity on such spaces. In particular, we observe that if one can prove a good noise sensitivity bound for a class  $\mathcal{C}$  under a product distribution  $\Pi$ , then [KKMS05] implies that the polynomial regression algorithm yields a good agnostic learner with respect to  $\mathcal{C}$  under  $\Pi$ .

Section 3 contains the key reduction from noise sensitivity in general product spaces to noise sensitivity under the uniform distribution on  $\{0, 1\}^n$ . It is carried out in the model case of linear threshold functions, which Peres [Per04] proved have  $\delta$ -noise sensitivity at most  $O(\sqrt{\delta})$ . We give a surprisingly simple proof of the following:

**Theorem 3.2** *Let  $f : \mathcal{X} \rightarrow \{-1, 1\}$  be a linear threshold function, where  $\mathcal{X} = X_1 \times \dots \times X_n$  has the product distribution  $\Pi = \pi_1 \times \dots \times \pi_n$ . Then  $\text{NS}_\delta(f) \leq O(\sqrt{\delta})$ .*

Proving this just in the case of a  $p$ -biased distribution on  $\{0, 1\}^n$  was an open problem suggested in [Per04]. This noise sensitivity bound thus gives us the following learning result:

**Theorem 3.4** Let  $\Pi = \pi_1 \times \cdots \times \pi_n$  be any product distribution over an instance space  $\mathcal{X} = X_1 \times \cdots \times X_n$ , where we assume  $|X_i| \leq \text{poly}(n)$  for each  $i$ . Let  $\mathcal{C}$  denote the class of functions of  $k$  linear threshold functions over  $\mathcal{X}$ . Taking  $d = O(k^2/\epsilon^4)$ , the degree- $d$  polynomial regression algorithm with  $L_1$  loss (or hinge loss) uses  $n^{O(k^2/\epsilon^4)}$  time and examples and agnostically learns with respect to  $\mathcal{C}$ .

In Section 4 we discuss how to extend concentration results for other concept classes from uniform on  $\{0, 1\}^n$  to arbitrary product distributions on product spaces  $\mathcal{X} = X_1 \times \cdots \times X_n$ . Of course, it's not immediately clear, given a concept class  $\mathcal{C}$  of functions on  $\{0, 1\}^n$ , what it even means for it to be generalized to functions on  $\mathcal{X}$ . We discuss a reasonable such notion based on one-out-of- $k$  encoding, and illustrate it in the case of  $AC^0$  functions. The idea in this section is simple: any concentration result under uniform on  $\{0, 1\}^n$  easily implies a (slightly weaker) noise sensitivity bound; this can be translated into the same noise sensitivity bound under any product distribution using the methods of Section 3. In turn, that implies a concentration bound in the general product space. As an example, we prove the following:

**Theorem 4.2** Let  $\mathcal{C}$  be the class of functions  $X_1 \times \cdots \times X_n \rightarrow \{-1, 1\}$  computed by unbounded fan-in circuit of size at most  $s$  and depth at most  $c$  (under the one-out-of- $k$  encoding). Assume  $|X_i| \leq \text{poly}(n)$  for each  $i$ . Let  $\Pi$  be any product distribution on  $X_1 \times \cdots \times X_n$ . Then polynomial regression agnostically learns with respect to  $\mathcal{C}$  under arbitrary product distributions in time  $n^{(O(\log(s/\epsilon)))^{c-1}/\epsilon^2}$ .

Section 5 describes extensions of our learning algorithm to cases beyond those in which one has exactly a product distribution on an instance space  $\mathcal{X} = X_1 \times \cdots \times X_n$  with each  $|X_i| \leq \text{poly}(n)$ : these extensions include distributions “bounded by” or “close to” product distributions, as well as certain cases when the  $X_i$ 's have superpolynomial cardinality or are  $\mathbb{R}$ . We end Section 5 with a discussion of learning under mixtures of product distributions. Here there is a distinction between learning when the mixture distribution is known to the algorithm and when it is unknown. In the former case we prove, e.g.:

**Theorem 5.16** Let  $\mathcal{D}$  be any known mixture of  $\text{poly}(n)$  product distributions over an instance space  $\mathcal{X} = X_1 \times \cdots \times X_n$ , where we assume  $|X_i| \leq \text{poly}(n)$  for each  $i$ . Then there is a  $n^{O(k^2/\epsilon^4)}$ -time algorithm for agnostically learning with respect to the class of functions of  $k$  linear threshold functions over  $\mathcal{X}$  under  $\mathcal{D}$ .

In the latter case, by relying on algorithms for learning mixture distributions from unlabeled data, we prove:

**Theorem 5.18** Let  $\mathcal{D}$  be any unknown mixture of  $O(1)$  product distributions over an instance space  $\mathcal{X} = X_1 \times \cdots \times X_n$ , where we assume either: a)  $|X_i| \leq O(1)$  for each  $i$ ; or b) each  $X_i = \mathbb{R}$  and each product distribution is a mixture of axis-aligned (poly( $n$ )-bounded) Gaussians. Then there is a  $n^{O(k^2/\epsilon^4)}$ -time algorithm for agnostically learning with respect to the class of functions of  $k$  linear threshold functions over  $\mathcal{X}$  under  $\mathcal{D}$ .

## 2 Product probability spaces

In this section we consider functions  $f : \mathcal{X} \rightarrow \mathbb{R}$ , where  $\mathcal{X} = X_1 \times \cdots \times X_n$  is a product set. We will also assume  $\mathcal{X}$  is endowed with some product probability distribution  $\Pi = \pi_1 \times \cdots \times \pi_n$ . All occurrences of  $\Pr[\cdot]$  and  $\mathbf{E}[\cdot]$  are with respect to this distribution unless otherwise noted, and we usually write  $\mathbf{x}$  for a random element of  $\mathcal{X}$  drawn from  $\Pi$ . For simplicity we assume that each set  $X_i$  is finite.<sup>6</sup> The vector space  $L^2(\mathcal{X}, \Pi)$  of all functions  $f : \mathcal{X} \rightarrow \mathbb{R}$  is viewed as an inner product space under the inner product  $\langle f, g \rangle = \mathbf{E}[f(\mathbf{x})g(\mathbf{x})]$ . We will also use the notation

$$\|f\|_2 = \sqrt{\langle f, f \rangle} = \sqrt{\mathbf{E}[f(\mathbf{x})^2]}.$$

### 2.1 Orthogonal decomposition

As each  $X_i$  is just an abstract set, there is not an inherent notion of a degree- $d$  polynomial on  $\mathcal{X}$ . Ultimately the polynomial regression algorithm identifies  $\mathcal{X}$  with a subset of  $\{0, 1\}^{|X_1| + \cdots + |X_n|}$  via the “one-out-of- $k$  encoding” and works with polynomials over this space. However to prove concentration results, we need to take a more abstract approach and consider the “(Hoeffding) orthogonal decomposition” of functions on product spaces; see [vM47, Hoe48, KR82, Ste86]. In this section we recall this notion with our own notation.

**Definition 2.1** We say a function  $f : X_1 \times \cdots \times X_n \rightarrow \mathbb{R}$  is a simple function of order  $d$  if it depends on at most  $d$  coordinates.

**Definition 2.2** We say a function  $f : X_1 \times \cdots \times X_n \rightarrow \mathbb{R}$  is a function of order  $d$  if it is a linear combination of simple functions of order  $d$ . The set of all such functions is a linear subspace of  $L^2(\mathcal{X}, \Pi)$  and we denote it by  $\mathcal{H}^{\leq d}(\mathcal{X}, \Pi)$ .

**Definition 2.3** We say a function  $f : X_1 \times \cdots \times X_n \rightarrow \mathbb{R}$  is a function of order exactly  $d$  if it is a function of order  $d$  and it is orthogonal to all functions of order  $d-1$ ; i.e.,  $\langle f, g \rangle = 0$  for all  $g \in \mathcal{H}^{\leq d-1}(\mathcal{X}, \Pi)$ . This is again a linear subspace of  $L^2(\mathcal{X}, \Pi)$  and we denote it by  $\mathcal{H}^{=d}(\mathcal{X}, \Pi)$ .

**Proposition 2.4** The space  $L^2(\mathcal{X}, \Pi)$  is the orthogonal direct sum of the  $\mathcal{H}^{=d}(\mathcal{X}, \Pi)$  spaces,

$$L^2(\mathcal{X}, \Pi) = \bigoplus_{d=0}^n \mathcal{H}^{=d}(\mathcal{X}, \Pi).$$

**Definition 2.5** By virtue of the previous proposition, every function  $f : X_1 \times \cdots \times X_n \rightarrow \mathbb{R}$  can be uniquely expressed as

$$f = f^{=0} + f^{=1} + f^{=2} + \cdots + f^{=n},$$

where  $f^{=d} : X_1 \times \cdots \times X_n \rightarrow \mathbb{R}$  denotes the projection of  $f$  into  $\mathcal{H}^{=d}(\mathcal{X}, \Pi)$ . We call  $f^{=d}$  the order  $d$  part of  $f$ . We will also write

$$f^{\leq d} = f^{=0} + f^{=1} + f^{=2} + \cdots + f^{=d}.$$

<sup>6</sup>In fact, we will only need that each  $L^2(X_i, \pi_i)$  has a countable basis.

In the sequel we will write simply  $\mathcal{H}^d$  in place of  $\mathcal{H}^d(\mathcal{X}, \Pi)$ , etc. Although we will not need it, we recall a further refinement of this decomposition:

**Definition 2.6** For each  $S \subseteq [n]$  we define  $\mathcal{H}^{\leq S}$  to be the subspace consisting of all functions depending only on the coordinates in  $S$ . We define  $\mathcal{H}^S$  to be the further subspace consisting of those functions in  $\mathcal{H}^{\leq S}$  that are orthogonal to all functions in  $\mathcal{H}^{\leq R}$  for each  $R \subsetneq S$ .

**Proposition 2.7** The space  $L^2(\mathcal{X}, \Pi)$  is the orthogonal direct sum of the  $\mathcal{H}^S$  spaces,  $L^2(\mathcal{X}, \Pi) = \bigoplus_{S \subseteq [n]} \mathcal{H}^S$ . Hence every function  $f : X_1 \times \dots \times X_n \rightarrow \mathbb{R}$  can be uniquely expressed as  $f = \sum_{S \subseteq [n]} f^S$ , where  $f^S : X_1 \times \dots \times X_n \rightarrow \mathbb{R}$  denotes the projection of  $f$  into  $\mathcal{H}^S$ . Denoting also  $f^{\leq S} = \sum_{R \subseteq S} f^R$  for the projection of  $f$  into  $\mathcal{H}^{\leq S}$ , we have the following interpretations:

$$f^{\leq S}(y_1, \dots, y_n) = \mathbf{E}[f(\mathbf{x}_1, \dots, \mathbf{x}_n) \mid \mathbf{x}_i = y_i \forall i \in S];$$

$$f^S(\mathbf{x}_1, \dots, \mathbf{x}_n) = \sum_{R \subseteq S} (-1)^{|S|-|R|} f^{\leq R}.$$

Finally, we connect the orthogonal decomposition of functions  $f : \mathcal{X} \rightarrow \mathbb{R}$  with their analogue under the one-out-of- $k$  encoding:

**Proposition 2.8** A function  $f : \mathcal{X} \rightarrow \mathbb{R}$  is of order  $d$  if and only if its analogue  $f : \{0, 1\}^{|X_1|+\dots+|X_n|} \rightarrow \mathbb{R}$  under the one-out-of- $k$  encoding is expressible as a polynomial of degree at most  $d$ .

## 2.2 Low-order concentration

As in the previous section we consider functions  $f : \mathcal{X} \rightarrow \mathbb{R}$  under a product distribution  $\Pi$ . We will be especially interested in classifiers, functions  $f : \mathcal{X} \rightarrow \{-1, 1\}$ . Our goal is to understand and develop conditions under which such  $f$  can be approximated in squared loss by low-degree polynomials.

By basic linear algebra, we have the following:

**Proposition 2.9** Given  $f : \mathcal{X} \rightarrow \mathbb{R}$ , the best order- $d$  approximator to  $f$  under squared loss is  $f^{\leq d}$ . I.e.,

$$\min_{g \text{ of order } d} \mathbf{E}[(f(\mathbf{x}) - g(\mathbf{x}))^2] = \|f - f^{\leq d}\|_2^2 = \sum_{i=d+1}^n \|f^{=i}\|_2^2.$$

**Definition 2.10** Given  $f : \mathcal{X} \rightarrow \mathbb{R}$  we say that  $f$  is  $\epsilon$ -concentrated up to order  $d$  if  $\sum_{i=d+1}^n \|f^{=i}\|_2^2 \leq \epsilon$ .

By Proposition 2.8 we conclude the following:

**Proposition 2.11** Let  $f : \mathcal{X} \rightarrow \mathbb{R}$  and identify  $f$  with a function  $\{0, 1\}^N \rightarrow \mathbb{R}$  under the one-out-of- $k$  encoding. Then there exists a polynomial  $p : \{0, 1\}^N \rightarrow \mathbb{R}$  of degree at most  $d$  which  $\epsilon$ -approximates  $f$  in squared loss under  $\Pi$  if and only if  $f$  is  $\epsilon$ -concentrated up to order  $d$ .

Combining this with the KKMS Theorem 1.2, we get the following learning result about polynomial regression:

**Theorem 2.12** Let  $\Pi = \pi_1 \times \dots \times \pi_n$  be a product distribution on  $\mathcal{X} = X_1 \times \dots \times X_n$ . Write  $N$  for the total number of possible attribute values,  $N = |X_1| + \dots + |X_n|$ . Let  $\mathcal{C}$  be a class of functions  $\mathcal{X} \rightarrow \{-1, 1\}$  each of which is  $\epsilon^2$ -concentrated up to order  $d$  under  $\Pi$ . Then the degree- $d$  polynomial regression algorithm with  $L_1$  loss (or hinge loss) uses  $\text{poly}(N^d/\epsilon)$  time and examples, and agnostically learns with respect to  $\mathcal{C}$  under  $\Pi$ .

We will now show how to prove low-order concentration results by extending the “noise sensitivity method” of [KOS04] to general product spaces.

## 2.3 Noise sensitivity

We recall the generalization of noise sensitivity [BKS99] to general product spaces, described in [MOO05].

**Definition 2.13** Given  $x \in X_1 \times \dots \times X_n$  and  $0 \leq \rho \leq 1$ , we define a  $\rho$ -noisy copy of  $x$  to be a random variable  $\mathbf{y}$  with distribution  $N_\rho(x)$ , where this denotes that each  $\mathbf{y}_i$  is chosen to equal  $x_i$  with probability  $\rho$  and to be randomly drawn from  $\pi_i$  with probability  $1 - \rho$ , independently across  $i$ .

**Definition 2.14** The noise operator  $T_\rho$  on functions  $f : \mathcal{X} \rightarrow \mathbb{R}$  is given by

$$(T_\rho f)(x) = \mathbf{E}_{\mathbf{y} \sim N_\rho(x)}[f(\mathbf{y})].$$

The noise stability of  $f$  at  $\rho$  is

$$\mathbb{S}_\rho(f) = \langle f, T_\rho f \rangle.$$

When  $f : \mathcal{X} \rightarrow \{-1, 1\}$  we also define the noise sensitivity of  $f$  at  $\delta \in [0, 1]$  to be

$$\mathbb{NS}_\delta(f) = \frac{1}{2} - \frac{1}{2} \mathbb{S}_{1-\delta}(f) = \Pr_{\substack{\mathbf{x} \sim \Pi \\ \mathbf{y} \sim N_{1-\delta}(\mathbf{x})}} [f(\mathbf{x}) \neq f(\mathbf{y})].$$

The connection between noise stability, sensitivity, and concentration comes from the following two facts:

**Proposition 2.15** ([MOO05]) For any  $f : \mathcal{X} \rightarrow \mathbb{R}$ ,

$$\mathbb{S}_\rho(f) = \sum_{i=0}^n \rho^i \|f^{=i}\|_2^2.$$

**Proposition 2.16** ([KOS04]) Suppose  $\mathbb{NS}_\delta(f) \leq \epsilon$ . Then  $f$  is  $\frac{2}{1-1/\epsilon}\epsilon$ -concentrated up to order  $1/\delta$ .

For example, Peres proved the following theorem:

**Theorem 2.17** ([Per04]) If  $f : \{0, 1\}^n \rightarrow \{-1, 1\}$  is a linear threshold function then

$$\mathbb{NS}_\delta(f) \leq O(1)\sqrt{\delta}$$

(under the uniform distribution on  $\{0, 1\}^n$ ). From [O’D03] we have that the  $O(1)$  can be taken to be  $\frac{5}{4}$  for every value of  $n$  and  $\delta$ .

It clearly follows that if  $f$  is any function of  $k$  linear threshold functions then  $\mathbb{NS}_\delta(f) \leq \frac{5}{4}k\sqrt{\delta}$ . Combining this with Proposition 2.16:

**Theorem 2.18** ([KOS04]) Let  $f : \{0, 1\}^n \rightarrow \{-1, 1\}$  be any function of  $k$  linear threshold functions. Then  $f$  is  $(4k/\sqrt{\delta})$ -concentrated up to order  $d$  under the uniform distribution, for any  $d \geq 1$ . In particular,  $f$  is  $\epsilon^2$ -concentrated up to order  $O(k^2/\epsilon^4)$ .

### 3 Noise sensitivity of threshold functions in product spaces

In this section we show that Peres's theorem can be extended to hold for linear threshold functions in all product spaces.

**Definition 3.1** We say a function  $f : X_1 \times \dots \times X_n \rightarrow \{-1, 1\}$  is a linear threshold function if its analogue  $f : \{0, 1\}^N \rightarrow \{-1, 1\}$  under one-out-of- $k$  encoding is expressible as a linear threshold function. Equivalently,  $f$  is a linear threshold function if there exist weight functions  $w_i : X_i \rightarrow \mathbb{R}$ ,  $i = 1 \dots n$ , and a number  $\theta \in \mathbb{R}$  such that

$$f(x_1, \dots, x_n) = \text{sgn} \left( \sum_{i=1}^n w_i(x_i) - \theta \right).$$

No version of Peres's Theorem 2.17 was previously known to hold even in the simple case of linear threshold functions on  $\{0, 1\}^n$  under a  $p$ -biased product distribution with  $p \neq 1/2$ . Understanding just this nonsymmetric case was left as an open question in [Per04]. We now show that threshold functions over general product spaces are no more noise sensitive than threshold functions over  $\{0, 1\}^n$  under the uniform distribution.

**Theorem 3.2** Let  $f : \mathcal{X} \rightarrow \{-1, 1\}$  be a linear threshold function, where  $\mathcal{X} = X_1 \times \dots \times X_n$  has the product distribution  $\Pi = \pi_1 \times \dots \times \pi_n$ . Then  $\text{NS}_\delta(f) \leq \frac{5}{4}\sqrt{\delta}$ .

**Proof:** For a pair of instances  $z_0, z_1 \in \mathcal{X}$  and a vector  $x \in \{0, 1\}^n$ , we introduce the notation  $z_x$  for the instance whose  $i$ th attribute  $(z_x)_i$  is the  $i$ th attribute of  $z_{x_i}$ . For any fixed  $z_0, z_1 \in \mathcal{X}$  we can define  $g_{z_0, z_1} : \{0, 1\}^n \rightarrow \{-1, 1\}$  such that  $g_{z_0, z_1}(x) = f(z_x)$ . Note that this function is a linear threshold function in the traditional binary sense.

Let  $z_0, z_1$  now denote independent random draws from  $\Pi$ , and let  $\mathbf{x}$  denote a uniformly random vector from  $\{0, 1\}^n$ . We have that  $z_{\mathbf{x}}$  is distributed as a random draw from  $\Pi$ . Further pick  $\mathbf{y} \in \{0, 1\}^n$  to be a  $\delta$ -noisy copy of  $\mathbf{x}$ , i.e.  $\mathbf{y} \sim N_\delta(\mathbf{x})$ . Then  $z_{\mathbf{y}}$  is distributed as  $N_\delta(z_{\mathbf{x}})$ . We now have

$$\begin{aligned} \text{NS}_\delta(f) &= \Pr_{z_0, z_1, \mathbf{x}, \mathbf{y}} [f(z_{\mathbf{x}}) \neq f(z_{\mathbf{y}})] \\ &= \mathbf{E}_{z_0, z_1} \left[ \Pr_{\mathbf{x}, \mathbf{y}} [f(z_{\mathbf{x}}) \neq f(z_{\mathbf{y}})] \right] \\ &= \mathbf{E}_{z_0, z_1} \left[ \Pr_{\mathbf{x}, \mathbf{y}} [g_{z_0, z_1}(\mathbf{x}) \neq g_{z_0, z_1}(\mathbf{y})] \right]. \end{aligned}$$

Once  $z_0$  and  $z_1$  are fixed, the quantity in the expectation is just the noise sensitivity at  $\delta$  of the binary linear threshold function  $g_{z_0, z_1}$ , which we can bound by  $\frac{5}{4}\sqrt{\delta}$  using Theorem 2.17. So

$$\begin{aligned} \text{NS}_\delta(f) &= \mathbf{E}_{z_0, z_1} \left[ \Pr_{\mathbf{x}, \mathbf{y}} [g_{z_0, z_1}(\mathbf{x}) \neq g_{z_0, z_1}(\mathbf{y})] \right] \\ &\leq \mathbf{E}_{z_0, z_1} \left[ \frac{5}{4}\sqrt{\delta} \right] = \frac{5}{4}\sqrt{\delta}, \end{aligned}$$

which is what we wanted to show.  $\square$

As with Theorem 2.18, we conclude:

**Theorem 3.3** Let  $f : \mathcal{X} \rightarrow \{-1, 1\}$  be any function of  $k$  linear threshold functions, where  $\mathcal{X} = X_1 \times \dots \times X_n$  has the product distribution  $\Pi = \pi_1 \times \dots \times \pi_n$ . Then  $f$  is  $(4k/\sqrt{d})$ -concentrated up to order  $d$ , for any  $d \geq 1$ . In particular,  $f$  is  $\epsilon^2$ -concentrated up to order  $O(k^2/\epsilon^4)$ .

By combining Theorem 3.3 with our main learning theorem, Theorem 2.12, we conclude:

**Theorem 3.4** Let  $\Pi = \pi_1 \times \dots \times \pi_n$  be any product distribution over an instance space  $\mathcal{X} = X_1 \times \dots \times X_n$ , where we assume  $|X_i| \leq \text{poly}(n)$  for each  $i$ . Let  $\mathcal{C}$  denote the class of functions of  $k$  linear threshold functions over  $\mathcal{X}$ . Taking  $d = O(k^2/\epsilon^4)$ , the degree- $d$  polynomial regression algorithm with  $L_1$  loss (or hinge loss) uses  $n^{O(k^2/\epsilon^4)}$  time and examples and agnostically learns with respect to  $\mathcal{C}$ .

### 4 Concentration for other classes under product distributions

In this section we illustrate how essentially any result about  $\epsilon$ -concentration of classes of functions under the uniform distribution on  $\{0, 1\}^n$  can be translated into a similar result for general product distributions. Besides linear threshold functions, the other main example of concentration comes from the original application of the Low Degree Algorithm [LMN93]: learning  $\text{AC}^0$  functions in quasipolynomial time. Recall that  $\text{AC}^0$  is the class of functions computed by unbounded fan-in circuits of constant depth and polynomial size. We will use this as a running example.

Suppose  $\mathcal{C}$  is a class of functions  $\mathcal{X} \rightarrow \{-1, 1\}$ , where  $\mathcal{X} = X_1 \times \dots \times X_n$ . As usual, under the one-out-of- $k$  encoding we can think of  $\mathcal{C}$  as a class of functions  $\{0, 1\}^N \rightarrow \{-1, 1\}$ . In our example, this gives a reasonable notion of "AC<sup>0</sup> circuits on general product sets  $\mathcal{X}$ ". Suppose further that  $\overline{\mathcal{C}} \supseteq \mathcal{C}$  is any class of functions  $\{0, 1\}^N \rightarrow \{-1, 1\}$  which is closed under negation of inputs and closed under fixing inputs to 0 or 1. In our example, the class of  $\text{AC}^0$  circuits indeed has this basic property (as does the more precisely specified class of all circuits with size at most  $s$  and depth at most  $c$ ).

Now by repeating the proof of Theorem 3.2, it is easy to see that any upper bound one can prove on the noise sensitivity of functions in  $\overline{\mathcal{C}}$  under the uniform distribution on  $\{0, 1\}^N$  immediately translates an identical bound on the noise sensitivity of functions in  $\mathcal{C}$  on  $\mathcal{X}$  under any product distribution. The only thing to notice is that the functions  $g_{z_0, z_1}$  arising in that proof will be in the class  $\overline{\mathcal{C}}$ . Thus we are reduced to proving noise sensitivity bounds for functions on  $\{0, 1\}^n$  under the uniform distribution.

Furthermore, any result on  $\epsilon$ -concentration of functions on  $\{0, 1\}^n$  under the uniform distribution can be easily translated into a noise sensitivity bound which is not much worse:

**Proposition 4.1** Suppose that  $f : \{0, 1\}^n \rightarrow \{-1, 1\}$  is  $\epsilon$ -concentrated up to degree  $d$  under the uniform distribution on  $\{0, 1\}^n$ . Then  $\text{NS}_{\epsilon/d}(f) \leq \epsilon$ .

**Proof:** Using traditional Fourier notation instead of orthogonal decomposition notation, we have

$$\begin{aligned} \mathbb{S}_{1-\epsilon/d}(f) &= \sum_{S \subseteq [n]} (1-\epsilon/d)^{|S|} \hat{f}(S)^2 \\ &\geq (1-\epsilon/d)^d (1-\epsilon) \geq (1-\epsilon)^2, \end{aligned}$$

where the first inequality used the fact that  $f$  is  $\epsilon$ -concentrated up to degree  $d$ . Thus

$$\mathbb{N}\mathbb{S}_{1-\epsilon/d}(f) = \frac{1}{2} - \frac{1}{2}\mathbb{S}_{1-\epsilon/d}(f) \leq \frac{1}{2} - \frac{1}{2}(1-\epsilon)^2 \leq \epsilon.$$

□

Finally, applying Proposition 2.16, we get  $O(\epsilon)$ -concentration up to order  $d/\epsilon$  for the original class  $\mathcal{C}$  of functions  $\mathcal{X} \rightarrow \{-1, 1\}$ , under any product distribution on  $\mathcal{X}$ . This leads to an agnostic learning result for  $\mathcal{C}$  under arbitrary product distributions which is the same as the one would get for  $\bar{\mathcal{C}}$  under the uniform distribution on  $\{0, 1\}^n$ , except for an extra factor of  $\epsilon$  in the running time's exponent.

For example, with regard to  $\text{AC}^0$  functions, [LMN93, Hås01] proved the following:

**Theorem 4.2** *Let  $f : \{0, 1\}^n \rightarrow \{-1, 1\}$  be computable by an unbounded fan-in circuit of size at most  $s$  and depth at most  $c$ . Then  $f$  is  $\epsilon$ -concentrated up to degree  $d = (O(\log(s/\epsilon)))^{c-1}$ .*

We therefore may conclude:

**Theorem 4.3** *Let  $\mathcal{C}$  be the class of functions  $X_1 \times \dots \times X_n \rightarrow \{-1, 1\}$  computed by unbounded fan-in circuit of size at most  $s$  and depth at most  $c$  (under the one-out-of- $k$  encoding). Assume  $|X_i| \leq \text{poly}(n)$  for each  $i$ . Let  $\Pi$  be any product distribution on  $X_1 \times \dots \times X_n$ . Then every  $f \in \mathcal{C}$  is  $\frac{2}{1-1/\epsilon}\epsilon$ -concentrated up to order  $d = (O(\log(s/\epsilon)))^{c-1}/\epsilon$ . As a consequence, polynomial regression agnostically learns with respect to  $\mathcal{C}$  under arbitrary product distributions in time  $n^{O(\log(s/\epsilon))^{c-1}/\epsilon^2}$ .*

This result should be compared to the following theorem from Furst, Jackson, and Smith [FJS91] for PAC-learning under bounded product distributions on  $\{0, 1\}^n$ :

**Theorem 4.4 ([FJS91])** *The class  $\mathcal{C}$  of functions  $\{0, 1\}^n \rightarrow \{-1, 1\}$  computed by unbounded fan-in circuit of size at most  $s$  and depth at most  $c$  can be PAC-learned under any product distribution in time  $n^{O((1/p)\log(s/\epsilon))^{c+O(1)}}$ , assuming the mean of each coordinate is in the range  $[p, 1-p]$ .*

The advantage of the result from [FJS91] is that it does not pay the extra  $1/\epsilon^2$  in the exponent. The advantages of our result is that it holds under arbitrary product distributions on product sets. (Our result is in the agnostic model, but the result of [FJS91] could also be by applying the results of [KKMS05].)

## 5 Extensions

### 5.1 Distributions close to or dominated by product distributions

We begin with some simple observations showing that the underlying distribution need not be *precisely* a product distribution. First, the following fact can be considered standard:

**Proposition 5.1** *Suppose that under distribution  $\mathcal{D}$ , algorithm  $\mathcal{A}$  agnostically learns with respect to class  $\mathcal{C}$ , using  $m$  examples to achieve error  $\epsilon$ . If  $\mathcal{D}'$  is any distribution satisfying  $\|\mathcal{D}' - \mathcal{D}\|_1 \leq \epsilon/m$ , then  $\mathcal{A}$  also agnostically learns under  $\mathcal{D}'$ , using  $m$  examples to achieve error  $2\epsilon + 2\epsilon/m \leq 4\epsilon$ .*

**Proof:** The key fact we use is that if  $\mathbf{X}$  is a random variable with  $|\mathbf{X}| \leq 1$  always, then  $|\mathbf{E}_{\mathcal{D}'}[\mathbf{X}] - \mathbf{E}_{\mathcal{D}}[\mathbf{X}]| \leq \|\mathcal{D}' - \mathcal{D}\|_1$ . This implies that for any hypothesis  $h$ ,  $|\text{err}_{\mathcal{D}'}(h) - \text{err}_{\mathcal{D}}(h)| \leq \epsilon/m$ . In particular, it follows that  $\text{Opt}_{\mathcal{D}'} \leq \text{Opt}_{\mathcal{D}} + \epsilon/m$ . Further, let  $\mathbf{h}$  be the random variable denoting the hypothesis  $\mathcal{A}$  produces when given examples from  $\mathcal{D}^{\otimes m}$ . By assumption, we have

$$\mathbf{E}_{\mathcal{D}^{\otimes m}}[\text{err}_{\mathcal{D}}(\mathbf{h})] \leq \text{Opt}_{\mathcal{D}} + \epsilon$$

which is at most  $\text{Opt}_{\mathcal{D}'} + \epsilon + \epsilon/m$ . Since  $\|\mathcal{D}'^{\otimes m} - \mathcal{D}^{\otimes m}\|_1 \leq m(\epsilon/m) = \epsilon$ , the key fact applied to  $\text{err}_{\mathcal{D}}(\mathbf{h})$  implies

$$\mathbf{E}_{\mathcal{D}'^{\otimes m}}[\text{err}_{\mathcal{D}}(\mathbf{h})] \leq \text{Opt}_{\mathcal{D}'} + \epsilon + \epsilon/m + \epsilon.$$

Finally, as we saw,  $\text{err}_{\mathcal{D}'}(\mathbf{h}) \leq \text{err}_{\mathcal{D}}(\mathbf{h}) + \epsilon/m$  always. Thus

$$\mathbf{E}_{\mathcal{D}'^{\otimes m}}[\text{err}_{\mathcal{D}'}(\mathbf{h})] \leq \text{Opt}_{\mathcal{D}'} + 2\epsilon + 2\epsilon/m,$$

completing the proof. □

We will use the above result later when learning under mixtures of product distributions.

A simple extension to the case when the distribution is “dominated” by a product distribution was already pointed out in [KKMS05]:

**Observation 5.2** *Let  $\mathcal{D}$  be a distribution on  $\mathcal{X}$  which is “ $C$ -dominated” by a product probability distribution  $\Pi = \pi_1 \times \dots \times \pi_n$ ; i.e., for all  $x \in \mathcal{X}$ ,  $\mathcal{D}(x) \leq C\Pi(x)$ . If  $f$  is  $\epsilon$ -concentrated up to degree  $d$  under  $\Pi$ , then  $f$  is  $C\epsilon$ -concentrated up to degree  $d$  under  $\mathcal{D}$ .*

Hence:

**Theorem 5.3** *Suppose we are in the setting of Theorem 3.4 except that  $\Pi$  is any distribution which is  $C$ -dominated by a product probability distribution. Then the degree- $d$  polynomial regression algorithm learns with respect to  $\mathcal{C}$  with  $d = O(C^2 k^2 / \epsilon^4)$  and hence  $n^{O(C^2 k^2 / \epsilon^4)}$  time and examples.*

### 5.2 Larger attribute domains

So far we have assumed that each attribute space  $X_i$  is only of polynomial cardinality. This can fairly easily be relaxed to the assumption that most of the probability mass in each  $(X_i, \pi_i)$  is concentrated on polynomially many atoms. Let us begin with some basic preliminaries:

**Notation 5.4** Given a distribution  $\pi$  on a set  $X$ , as well as a subset  $X' \subseteq X$ , we use the notation  $\pi'$  for the distribution on  $X'$  given by conditioning  $\pi$  on this set. (We always assume  $\pi(X') \neq 0$ .)

**Fact 5.5** Let  $\mathcal{X} = X_1 \times \dots \times X_n$  and let  $\Pi = \pi_1 \times \dots \times \pi_n$  be a product distribution on  $\mathcal{X}$ . Let  $X'_i \subseteq X_i$ ,  $i = 1 \dots n$ , and write  $\Pi'$  for the distribution  $\Pi$  conditioned on the set  $\mathcal{X}' = X'_1 \times \dots \times X'_n$ . Then  $\Pi'$  is the product distribution  $\pi'_1 \times \dots \times \pi'_n$ .

We now observe that if  $\mathcal{X}'$  is a “large” subset of  $\mathcal{X}$ , then any two functions which are close in  $L^2(\mathcal{X}, \Pi)$  are also close in  $L^2(\mathcal{X}', \Pi')$ :

**Proposition 5.6** In the setting of Fact 5.5, suppose that  $\Pr_{\mathbf{x}_i \sim \pi_i}[\mathbf{x}_i \notin X'_i] \leq 1/(2n)$  for all  $i$ . Then for any two functions  $f : \mathcal{X} \rightarrow \mathbb{R}$  and  $g : \mathcal{X} \rightarrow \mathbb{R}$ ,

$$\|f|_{\mathcal{X}'} - g|_{\mathcal{X}'}\|_{2, \Pi'}^2 \leq 2 \cdot \|f - g\|_{2, \Pi}^2$$

where  $f|_{\mathcal{X}'} : \mathcal{X}' \rightarrow \mathbb{R}$  denotes the restriction of  $f$  to  $\mathcal{X}'$ , and similarly for  $g|_{\mathcal{X}'}$ .

**Proof:** Writing  $h = f - g$ , we have

$$\begin{aligned} \|h\|_{2, \Pi}^2 &= \mathbf{E}_{\mathbf{x} \sim \Pi} [h(\mathbf{x})^2] \\ &= \Pr_{\mathbf{x} \sim \Pi} [\mathbf{x} \in \mathcal{X}'] \cdot \mathbf{E}_{\mathbf{x} \sim \Pi} [h(\mathbf{x})^2 \mid \mathbf{x} \in \mathcal{X}'] \\ &\quad + \Pr_{\mathbf{x} \sim \Pi} [\mathbf{x} \notin \mathcal{X}'] \cdot \mathbf{E}_{\mathbf{x} \sim \Pi} [h(\mathbf{x})^2 \mid \mathbf{x} \notin \mathcal{X}']. \end{aligned}$$

Using  $\mathbf{E}_{\mathbf{x} \sim \Pi} [h(\mathbf{x}) \mid \mathbf{x} \notin \mathcal{X}'] \geq 0$ , we have

$$\begin{aligned} \|h\|_{2, \Pi}^2 &\geq \Pr_{\mathbf{x} \sim \Pi} [\mathbf{x} \in \mathcal{X}'] \cdot \mathbf{E}_{\mathbf{x} \sim \Pi} [h(\mathbf{x})^2 \mid \mathbf{x} \in \mathcal{X}'] \\ &= \Pr_{\mathbf{x} \sim \Pi} [\mathbf{x} \in \mathcal{X}'] \cdot \mathbf{E}_{\mathbf{x} \sim \Pi'} [h(\mathbf{x})^2]. \end{aligned}$$

But by the union bound

$$\Pr_{\mathbf{x} \sim \Pi} [\mathbf{x} \notin \mathcal{X}'] \leq \sum_{i=1}^n \Pr_{\mathbf{x}_i \sim \Pi_i} [\mathbf{x}_i \notin X'_i] \leq n \cdot 1/(2n) = 1/2,$$

so  $\Pr_{\mathbf{x} \sim \Pi} [\mathbf{x} \in \mathcal{X}'] \geq 1/2$ . Thus

$$2 \cdot \|h\|_{2, \Pi}^2 \geq \mathbf{E}_{\mathbf{x} \sim \Pi'} [h(\mathbf{x})^2] = \|f|_{\mathcal{X}'} - g|_{\mathcal{X}'}\|_{2, \Pi'}^2,$$

completing the proof.  $\square$

**Corollary 5.7** In the setting of the previous proposition, if  $f$  is  $\epsilon$ -concentrated up to order  $d$  under  $\Pi$ , then  $f|_{\mathcal{X}'}$  is  $2\epsilon$ -concentrated up to order  $d$  under  $\Pi'$ .

**Proof:** It suffices to observe that if  $g : \mathcal{X} \rightarrow \mathbb{R}$  is a function of order  $d$ , then  $g|_{\mathcal{X}'}$  is also a function of order  $d$ .  $\square$

We can now describe an extended learning algorithm which works when the attribute spaces are mostly supported on sets of polynomial cardinality:

**Definition 5.8** We say that a finite probability space  $(X, \pi)$  is  $(\eta, r)$ -bounded if there exists a subset  $X' \subseteq X$  of cardinality at most  $|X'| \leq r$  such that  $\Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \notin X'] \leq \eta$ .

Our algorithm will learn whenever all  $n$  attribute sets are, say,  $(\epsilon/n, \text{poly}(n))$ -bounded. The first step of the algorithm will be to determine a set of attribute values which contain almost all of the probability mass.

**Lemma 5.9** Let  $(X, \pi)$  be an  $(\eta, r)$ -bounded probability space. Let  $\mathcal{Z}$  be a set of  $m = r \ln(r/\delta)/\eta$  samples drawn independently from  $\pi$ . Define  $Y$  to be the set  $\{x \in X : x \text{ was sampled in } \mathcal{Z}\}$ . Then with probability at least  $1 - \delta$ , the set  $Y$  satisfies  $\Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \notin Y] \leq 2\eta$ .

**Proof:** In fact, we will prove the slightly stronger statement that with probability at least  $1 - \delta$  the set  $Y$  satisfies  $\Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \notin Y \cap X'] \leq 2\eta$ , where  $X'$  is any set fulfilling the  $(\eta, r)$ -boundedness condition of  $(X, \pi)$ .

To prove the claim, we split the sampling procedure into  $r$  epochs, where we draw  $\ln(r/\delta)/\eta$  samples in each epoch. Let  $Y_i$  be the set of all atoms in  $X$  sampled among the first  $i$  epochs, with  $Y_0$  denoting the empty set. We will prove that with probability at least  $1 - \delta$ , the following holds for all epochs  $i \in [r]$ : either  $Y_{i-1}$  satisfies  $\Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \notin Y_{i-1} \cap X'] \leq 2\eta$ , or  $(Y_i \cap X') \setminus Y_{i-1} \neq \emptyset$  (i.e., we see a “new” atom from  $X'$  in the  $i$ th epoch).

Let’s first note that satisfying the above conditions implies that in the end  $\Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \notin Y \cap X'] \leq 2\eta$ . This is straightforward: if at any epoch  $Y_{i-1}$  satisfies  $\Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \notin Y_{i-1} \cap X'] \leq 2\eta$  then we’re done because  $Y \supseteq Y_{i-1}$ . Otherwise, in all  $r$  epochs we see a new atom from  $X'$ , and hence at the end of the sampling we will have seen  $r$  distinct atoms of  $X'$ ; then  $|X'| \leq r$  implies that our final  $Y \supseteq X'$ .

Now to complete the proof let us bound the probability that for a given  $i \in [r]$  the  $Y_{i-1}$  does not satisfy  $\Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \notin Y_{i-1} \cap X'] \leq 2\eta$  and we do not see a new element of  $X'$  in the  $i$ th epoch. Note that if  $\Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \notin Y_{i-1} \cap X'] > 2\eta$ , then the fact that  $\Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \notin X'] \leq \eta$  implies that  $\Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \in X' \setminus Y_{i-1}] > \eta$ . So the probability that we do not observe any element of  $X' \setminus Y_{i-1}$  in  $\ln(r/\delta)/\eta$  samples is

$$\begin{aligned} \Pr_{\mathbf{x} \sim \pi}[\mathbf{x} \notin X' \setminus Y_{i-1}]^{\ln(r/\delta)/\eta} &\leq (1 - \eta)^{\ln(r/\delta)/\eta} \\ &\leq e^{-\eta \ln(r/\delta)/\eta} = \delta/r. \end{aligned}$$

By applying the union bound, we see that there is probability at most  $\delta$  that any of the  $r$  epochs fails, so we’re done.  $\square$

We now give our extended learning algorithm:

1. Draw a set  $\mathcal{Z}_1$  of  $m_1$  unlabeled examples.
2. Draw a set  $\mathcal{Z}_2$  of  $m_2$  labeled examples.
3. Delete from  $\mathcal{Z}_2$  any instance/label pair where the instance contains an attribute value not appearing in  $\mathcal{Z}_1$ .
4. Run the degree- $d$  polynomial regression algorithm on  $\mathcal{Z}_2$ .

**Theorem 5.10** Let  $\Pi = \pi_1 \times \dots \times \pi_n$  be a product distribution on the set  $\mathcal{X} = X_1 \times \dots \times X_n$  and assume each probability space  $(X_i, \pi_i)$  is  $(\epsilon/n, r)$ -bounded. Write  $N = nr$ . Let  $\mathcal{C}$  be a class of functions  $\mathcal{X} \rightarrow \{-1, 1\}$  each of which is  $\epsilon^2$ -concentrated up to order  $d$ . Set  $m_1 = \text{poly}(N/\epsilon)$  and  $m_2 = \text{poly}(N^d/\epsilon)$ . The above algorithm uses  $\text{poly}(N^d/\epsilon)$  time and examples and agnostically learns with respect to  $\mathcal{C}$  under  $\Pi$ .

**Proof:** For simplicity we will equivalently prove that the algorithm outputs a hypothesis with error at most  $\text{Opt} + O(\epsilon)$ , rather than  $\text{Opt} + \epsilon$ .

We first want to establish that with probability at least  $1 - \epsilon$ , the set of attributes observed in the sample  $\mathcal{Z}_1$  covers almost all of the probability mass of  $\Pi$ . For each  $i \in [n]$ , let  $X'_i$  be the set of attribute values from  $X_i$  observed in at least one of the samples in  $\mathcal{Z}_1$ . Using the fact that each  $(X_i, \pi_i)$  is  $(\epsilon/n, r)$ -bounded, Lemma 5.9 implies that for sufficiently large  $m_1 = \text{poly}(N/\epsilon) \log(1/\epsilon)$ , each  $X'_i$  will satisfy  $\Pr_{\mathbf{x}_i \sim \pi_i}[\mathbf{x}_i \notin X'_i] \leq 2\epsilon/n$  except with probability at most  $\epsilon/n$ . Applying the union bound, all  $X'_i$  simultaneously satisfy the condition with probability at least  $1 - \epsilon$ . We henceforth assume this happens. Writing  $\mathcal{X}' = X'_1 \times \dots \times X'_n$ , we note that, by the union bound,  $\Pr_{\mathbf{x} \sim \Pi}[\mathbf{x} \notin \mathcal{X}'] \leq 2\epsilon$ .

The second thing we establish is that we do not throw away too many examples in Step 3 of the algorithm. We have just observed that the probability a given example in  $\mathcal{Z}_2$  is deleted is at most  $2\epsilon$ . We may assume  $2\epsilon \leq 1/2$ , and then a Chernoff bound (and  $m_2 \gg \log(1/\epsilon)$ ) easily implies that with probability at least  $1 - \epsilon$ , at most, say, two-thirds of all examples are deleted. Assuming this happens, we have that even after deletion,  $\mathcal{Z}_2$  still contains at least  $\text{poly}(N^d/\epsilon)$  many examples.

We now come to the main part of the proof, which is based on the observation that the undeleted examples in  $\mathcal{Z}_2$  are distributed as i.i.d. draws from the restricted product distribution  $\Pi'$  gotten by conditioning  $\Pi$  on  $\mathcal{X}'$ . Thus we are in a position to apply our main learning result, Theorem 2.12. The polynomial regression part of the above algorithm indeed uses  $\text{poly}(N^d/\epsilon)$  time and examples, and it remains to analyze the error of the hypothesis it outputs.

First, we use the fact that each function  $f$  in  $\mathcal{C}$  is  $\epsilon^2$ -concentrated up to order  $d$  to conclude that each function  $f|_{\mathcal{X}'}$  in “ $\mathcal{C}|_{\mathcal{X}'}$ ” is  $2\epsilon^2$ -concentrated up to order  $d$ . This uses Proposition 5.6 and the fact that we may assume  $2\epsilon \leq 1/2$ . Next, the guarantee of Theorem 2.12 is that when learning the target classifier  $t$  (viewed as a function  $\mathcal{X} \rightarrow \{-1, 1\}$  or  $\mathcal{X}' \rightarrow \{-1, 1\}$ ), the expected error under  $\Pi'$  of the hypothesis  $h$  produced is at most  $\text{Opt}' + O(\epsilon)$ , where

$$\text{Opt}' = \min_{f \in \mathcal{C}|_{\mathcal{X}'}} \Pr_{\mathbf{x} \sim \Pi'} [f(\mathbf{x}) \neq t(\mathbf{x})].$$

By definition, there is a function  $f \in \mathcal{C}$  satisfying

$$\Pr_{\mathbf{x} \sim \Pi} [f(\mathbf{x}) \neq t(\mathbf{x})] = \text{Opt}.$$

Since  $\Pr_{\mathbf{x} \sim \Pi}[\mathbf{x} \notin \mathcal{X}'] \leq 2\epsilon$ , it is easy to see that  $f|_{\mathcal{X}'}$  has error at most  $\text{Opt} + 2\epsilon$  on  $t$  under  $\Pi'$ . Thus  $\text{Opt}' \leq \text{Opt} + 2\epsilon$ ,

and we conclude that the expected error under  $\Pi'$  of  $h$  on  $t$  is at most  $\text{Opt} + 2\epsilon + O(\epsilon) = \text{Opt} + O(\epsilon)$ . Finally, the same observation implies that the expected error under  $\Pi$  of  $h$  on  $t$  is at most  $\text{Opt} + 2\epsilon + O(\epsilon) = \text{Opt} + O(\epsilon)$ .

We have thus established that with probability at least  $1 - 2\epsilon$ , the polynomial regression part of the above algorithm outputs a hypothesis with expected error at most  $\text{Opt} + O(\epsilon)$ . It follows that the overall expected error is at most  $\text{Opt} + O(\epsilon)$ , as desired.  $\square$

### 5.3 Real-valued attributes

We next consider the particular case of learning with respect to linear threshold functions, but when some of the attributes are *real-valued*. This case is relatively easily handled by discretizing the ranges of the distributions and using the previously discussed techniques. Our approach works for a very wide variety of distributions on  $\mathbb{R}$ ; these distributions need not even be continuous. We only need the distributions to satisfy “polynomial boundedness and anti-concentration” bounds.

**Definition 5.11** We say that a distribution  $\mathcal{D}$  over  $\mathbb{R}$  is  $B$ -polynomially bounded if for all  $\eta > 0$ , there is an interval  $I$  of length at most  $\text{poly}(B/\eta)$  such that  $\Pr_{\mathbf{x} \sim \mathcal{D}}[\mathbf{x} \notin I] \leq \eta$ .

**Definition 5.12** Given a real-valued random variable  $x$  with distribution  $\mathcal{D}$ , recall that the Lévy (anti-)concentration function  $Q(x; \lambda)$  is defined by

$$Q(x; \lambda) = \sup_{t \in \mathbb{R}} \Pr_{\mathbf{x} \sim \mathcal{D}} [\mathbf{x} \in [t - \lambda/2, t + \lambda/2]].$$

We say that  $\mathcal{D}$  has  $B$ -polynomial anti-concentration if  $Q(\mathcal{D}; \lambda) \leq \text{poly}(B) \cdot \lambda^c$  for some positive  $c > 0$ . Note that if  $\mathcal{D}$  is a continuous distribution with pdf everywhere at most  $B$  then it has  $B$ -polynomial anti-concentration (with  $c = 1$  in fact).

Having polynomial boundedness and concentration is an extremely mild condition; for example, the following familiar continuous distributions are all  $B$ -polynomial bounded and have  $B$ -polynomial anti-concentration: *Gaussians* with  $1/B \leq \sigma^2 \leq B$ ; *exponential* distributions with  $1/B \leq \lambda \leq B$ ; *Laplace* distributions with scale parameter with  $1/B \leq b \leq B$ ; *Pareto* distributions with shape parameter  $1/B \leq k \leq B$ ; *chi-square* distributions with variance  $1/B \leq \sigma^2 \leq B$  (for 1 degree of freedom, the anti-concentration “ $c$ ” needs to be  $1/2$ ); etc.

(Furthermore, in most cases even the condition on the parameter being in  $[1/B, B]$  can be eliminated. For example, suppose the first coordinate has a Gaussian distribution with standard deviation  $\sigma$ . With  $O(\log(1/\delta))$  examples, one can with probability at least  $1 - \delta$  estimate  $\sigma$  to within a multiplicative factor of 2. Having done so, one can multiply all examples’ first coordinate by an appropriate constant so as to get a Gaussian distribution with standard deviation in  $[1/2, 2]$ . Further, this does not change the underlying agnostic learning problem, since the class of linear threshold functions is closed under scaling a coordinate. For clarity of exposition, we leave further considerations of this sort to the

reader.)

We now describe the effect that discretizing a real-valued distribution can have with respect to functions of linear threshold functions. It is convenient to switch from working with a distribution on  $\mathcal{X}$  and target function  $\mathcal{X} \rightarrow \{-1, 1\}$  to just having a distribution  $\mathcal{D}$  on  $\mathcal{X} \times \{-1, 1\}$  — see the discussion after definition of agnostic learning in Section 1.1. As usual, assume that  $\mathcal{X} = X_1 \times \dots \times X_n$  is a product set and that the marginal distribution of  $\mathcal{D}$  on  $\mathcal{X}$  is a product distribution.

Suppose we have one coordinate with a real-valued distribution; without loss of generality, say  $X_1 = \mathbb{R}$ , and write  $\mathcal{D}_1$  for the marginal distribution of  $\mathcal{D}$  on  $X_1$ . When we refer to a “linear threshold function” on  $\mathcal{X}$ , we assume that the “weight function”  $w_1 : X_1 \rightarrow \mathbb{R}$  for coordinate 1 is just  $w_1(x_1) = c_1 x_1$  for some nonzero constant  $c_1$ .

**Lemma 5.13** *Let  $\mathcal{C}$  denote the class of functions of  $k$  linear threshold functions over  $\mathcal{X}$ . As usual, write*

$$\text{Opt} = \inf_{f \in \mathcal{C}} \text{err}_{\mathcal{D}}(f), \quad \text{where } \text{err}_{\mathcal{D}}(f) = \Pr_{(x,y) \sim \mathcal{D}} [f(x) \neq y].$$

Consider discretizing  $X_1 = \mathbb{R}$  by mapping each  $x_1 \in \mathbb{R}$  to  $\text{rd}_{\tau}(x_1)$ , the nearest integer multiple of  $\tau$  to  $x_1$ . Write  $X'_1 = \tau\mathbb{Z}$  and let  $\mathcal{D}'$  denote the distribution on  $X'_1 \times X_2 \times \dots \times X_n \times \{-1, 1\}$  induced from  $\mathcal{D}$  by the discretization.<sup>7</sup> Write  $\text{Opt}'$  for the quantity analogous to  $\text{Opt}$  for  $\mathcal{D}'$ . Then if  $\mathcal{D}_1$  has  $B$ -polynomial anti-concentration, it holds that  $\text{Opt}' \leq \text{Opt} + k \cdot \text{poly}(B) \cdot \tau^{\Omega(1)}$ .

**Proof:** It suffices to show that for any  $f \in \mathcal{C}$ ,

$$\begin{aligned} k \cdot \text{poly}(B) \cdot \tau^{\Omega(1)} &\geq |\text{err}_{\mathcal{D}}(f) - \text{err}_{\mathcal{D}'}(f)| \\ &= \left| \Pr_{(x,y) \sim \mathcal{D}} [f(x) \neq y] - \Pr_{(x,y) \sim \mathcal{D}'} [f(x) \neq y] \right|. \end{aligned}$$

Writing  $\Pi$  for the marginal of  $\mathcal{D}$  on  $\mathcal{X}$ , we can prove the above by proving

$$\Pr_{\mathbf{x} \sim \Pi} [f(\mathbf{x}) \neq f(\text{rd}_{\tau}(\mathbf{x}_1), \mathbf{x}_2, \dots, \mathbf{x}_n)] \leq k \cdot \text{poly}(B) \cdot \tau^{\Omega(1)}.$$

Since  $f$  is a function of some  $k$  linear threshold functions, by the union bound it suffices to show

$$\Pr_{\mathbf{x} \sim \Pi} [h(\mathbf{x}) \neq h(\text{rd}_{\tau}(\mathbf{x}_1), \mathbf{x}_2, \dots, \mathbf{x}_n)] \leq \text{poly}(B) \cdot \tau^{\Omega(1)}$$

for any linear threshold function  $h$ . We can do this by showing

$$\Pr_{\substack{\mathbf{x}_1 \sim \mathcal{D}_1 \\ \mathbf{Y}}} [\text{sgn}(c_1 \mathbf{x}_1 + \mathbf{Y}) \neq \text{sgn}(c_1 \text{rd}_{\tau}(\mathbf{x}_1) + \mathbf{Y})] \leq \text{poly}(B) \cdot \tau^{\Omega(1)},$$

where  $\mathbf{Y}$  is the random variable distributed according to the other part of the linear threshold function  $h$ . Note that  $\mathbf{Y}$  and  $\mathbf{x}_1$  are independent because  $\Pi$  is a product distribution. Now since  $|\mathbf{x}_1 - \text{rd}_{\tau}(\mathbf{x}_1)|$  is always at most  $\tau/2$ , we can only have  $\text{sgn}(c_1 \mathbf{x}_1 + \mathbf{Y}) \neq \text{sgn}(c_1 \text{rd}_{\tau}(\mathbf{x}_1) + \mathbf{Y})$  if

$$|c_1 \mathbf{x}_1 + \mathbf{Y}| \leq |c_1| \tau/2 \iff |\mathbf{x}_1 + \mathbf{Y}/c_1| \leq \tau/2.$$

<sup>7</sup>This can lead to inconsistent labels, which is why we switched to  $\mathcal{D}$  rather than have a target function.

It is an easy and well-known fact that if  $\mathbf{x}$  and  $\mathbf{y}$  are independent random variables then  $Q(\mathbf{x} + \mathbf{y}; \lambda) \leq Q(\mathbf{x}; \lambda)$ ; hence

$$\Pr_{\substack{\mathbf{x}_1 \sim \mathcal{D}_1 \\ \mathbf{Y}}} [|\mathbf{x}_1 + \mathbf{Y}/c_1| \leq \tau/2] \leq Q(\mathbf{x}_1; \tau/2).$$

But  $\mathcal{D}_1$  has  $B$ -polynomial anti-concentration, so  $Q(\mathbf{x}_1; \tau/t) \leq \text{poly}(B) \cdot \tau^{\Omega(1)}$ , as needed.  $\square$

By repeating this lemma up to  $n$  times, it follows that even if all  $n$  coordinate distributions are real-valued, so long as they have  $\text{poly}(n)$ -polynomial anti-concentration we will suffer little error. Specifically (assuming  $k \leq \text{poly}(n)$  as well), by taking  $\tau = \text{poly}(\epsilon/n)$  we get that discretization only leads to an additional error of  $\epsilon$ .

Finally, note that if a distribution  $\mathcal{D}_i$  is  $\text{poly}(n)$ -polynomially bounded then its discretized version is  $(\epsilon/n, \text{poly}(n/\epsilon))$ -bounded in the sense of Section 5.2; this lets us apply Theorem 5.10. Summarizing:

**Theorem 5.14** *Let  $\Pi = \pi_1 \times \dots \times \pi_n$  be a product distribution on the set  $\mathcal{X} = X_1 \times \dots \times X_n$ . For the finite  $X_i$ 's, assume each is  $(\epsilon/n, \text{poly}(n/\epsilon))$ -bounded. For the real  $X_i$ 's, assume the associated  $\pi_i$  is  $\text{poly}(n)$ -polynomially bounded and has  $\text{poly}(n)$ -polynomial anti-concentration. Let  $\mathcal{C}$  denote the class of functions of at most  $k \leq \text{poly}(n)$  linear threshold functions over  $\mathcal{X}$ . Then there is a  $\text{poly}(n/\epsilon)^{k^2/\epsilon^4}$  time algorithm which agnostically learns with respect to  $\mathcal{C}$  under  $\Pi$ .*

## 5.4 Mixtures of product distributions

So far we have only considered learning under distributions  $\mathcal{D}$  that are product distributions. In this section we show how to handle the commonly-studied case of mixtures of product distributions.

The first step is to show a generic learning-theoretic reduction: Roughly speaking, if we can agnostically learn with respect to any one of a family of distributions, then we can agnostically learn with respect to a *known* mixture of distributions from this family — even a mixture of polynomially many such distributions. (In our application the family of distributions will be the product distributions, but our reduction does not rely on this.) Although the following theorem uses relatively standard ideas, we do not know if it has appeared previously in the literature:

**Theorem 5.15** *Let  $\mathfrak{D}$  be a family of distributions over an instance space  $\mathcal{X}$ . There is a generic reduction from the problem of agnostically learning under a known mixture of  $c$  distributions from  $\mathfrak{D}$  to the problem of agnostically learning under a single known distribution from  $\mathfrak{D}$ . The reduction incurs a running time slowdown of  $\text{poly}(cT)/\gamma$  for an additional error of  $\gamma$ , where  $T$  denotes the maximum time needed to compute  $\mathcal{D}(x)$  for a mixture component  $\mathcal{D}$ .*

**Proof:** Suppose we are agnostically learning (with respect to some class  $\mathcal{C}$ ) under the distribution  $\mathcal{D}$  which is a mixture of  $c$  distributions  $\mathcal{D}_1, \dots, \mathcal{D}_c$  with mixing weights  $p_1, \dots, p_c$ . We make the assumption that the learning algorithm knows each of the mixing weights  $p_i$ , each of the distributions  $\mathcal{D}_i$ ,

and can compute any of the probabilities  $\mathcal{D}_i(x)$  in time  $T$ . We assume in the following that the  $\mathcal{D}_i$ 's are discrete distributions, but the case of absolutely continuous distributions could be treated in essentially the same way.

First, we claim that the algorithm can simulate learning under any of the single distributions  $\mathcal{D}_i$ , with slowdown  $\text{poly}(cT)/p_i$ . This is a standard proof based on rejection sampling: given an example  $x$ , the algorithm retains it with probability

$$r_i(x) := p_i \frac{\mathcal{D}_i(x)}{\mathcal{D}(x)}, \quad (1)$$

a quantity the algorithm can compute in time  $\text{poly}(cT)$ . One can check that this leads to the correct distribution  $\mathcal{D}_i$  on instances. The probability of retaining an example is easy seen to be precisely  $1/p_i$ , leading to the stated slowdown.

The main part of the proof now involves showing that if the algorithm agnostically learns under each  $\mathcal{D}_i$ , it can combine the hypotheses produced into an overall hypothesis which is good under  $\mathcal{D}$ . We will deal with the issue of running time (in particular, very small  $p_i$ 's) at the end of the proof. Let  $\text{Opt}$  denote the minimal error achievable among functions in  $\mathcal{C}$  under  $\mathcal{D}$ , and write  $\text{Opt}_i$  for the analogous quantity under  $\mathcal{D}_i$ ,  $i = 1 \dots c$ . Since one could use the same  $f \in \mathcal{C}$  for each  $\mathcal{D}_i$ , clearly  $\text{Opt} \geq \sum_{i=1}^c p_i \text{Opt}_i$ . By reduction, the algorithm produces hypotheses  $\mathbf{h}_1, \dots, \mathbf{h}_c$  satisfying  $\mathbf{E}[\text{err}_{\mathcal{D}_i}(\mathbf{h}_i)] \leq \text{Opt}_i + \epsilon$ .

We allow our overall algorithm to output a *randomized* hypothesis  $\mathbf{h}$ . We will then show that  $\mathbf{E}[\text{err}_{\mathcal{D}}(\mathbf{h})] \leq \text{Opt} + \epsilon$ , where the expectation is over the subalgorithms' production of the  $\mathbf{h}_i$ 's plus the "internal coins" of  $\mathbf{h}$ . Having shown this, it follows that our algorithm could equally well produce a deterministic hypothesis, just by (randomly) fixing a setting of  $\mathbf{h}$ 's internal coins as its last step.

Assume for a moment that the subalgorithms' hypotheses are fixed,  $h_1, \dots, h_c$ . The randomized overall hypothesis  $\mathbf{h} : \mathcal{X} \rightarrow \{-1, 1\}$  is defined by taking  $\mathbf{h}(x) = h_i(x)$  with probability exactly  $r_i(x)$ , where the probabilities  $r_i(x)$  are as defined in (1). (Note that they indeed sum to 1 and are computable in time  $\text{poly}(cT)$ .) Writing  $t$  for the target function, we compute:

$$\begin{aligned} & \mathbf{E}_{\mathbf{h}'\text{'s coins}} [\text{err}_{\mathcal{D}}(\mathbf{h})] \\ = & \mathbf{E}_{\mathbf{x} \sim \mathcal{D}} \left[ \mathbf{Pr}_{\mathbf{h}'\text{'s coins}} [\mathbf{h}(\mathbf{x}) \neq t(\mathbf{x})] \right] \\ = & \mathbf{E}_{\mathbf{x} \sim \mathcal{D}} \left[ \sum_{i: h_i(\mathbf{x}) \neq t(\mathbf{x})} r_i(\mathbf{x}) \right] \\ = & \mathbf{E}_{\mathbf{x} \sim \mathcal{D}} \left[ \sum_{i: h_i(\mathbf{x}) \neq t(\mathbf{x})} p_i(\mathbf{x}) \frac{\mathcal{D}_i(\mathbf{x})}{\mathcal{D}(\mathbf{x})} \right] \\ = & \sum_{x \in \mathcal{X}} \sum_{i: h_i(x) \neq t(x)} p_i(x) \mathcal{D}_i(x) \\ = & \sum_{i=1}^c p_i \sum_{x: h_i(x) \neq t(x)} \mathcal{D}_i(x) = \sum_{i=1}^c p_i \text{err}_{\mathcal{D}_i}(h_i). \end{aligned}$$

We now take the expectation over the production of the sub-hypotheses and conclude

$$\begin{aligned} \mathbf{E}_{\mathbf{h}} [\text{err}_{\mathcal{D}}(\mathbf{h})] &= \sum_{i=1}^c p_i \mathbf{E}[\text{err}_{\mathcal{D}_i}(\mathbf{h}_i)] \leq \sum_{i=1}^c p_i (\text{Opt}_i + \epsilon) \\ &= \sum_{i=1}^c p_i \text{Opt}_i + \epsilon \leq \text{Opt} + \epsilon, \quad (2) \end{aligned}$$

as claimed.

It remains to deal with small  $p_i$ 's and analyze the running time slowdown. We modify the overall algorithm so that it only simulates and learns under  $\mathcal{D}_i$  if  $p_i \geq \gamma/c$ . Thus the simulation slowdown we incur is only  $\text{poly}(cT)/\gamma$ , as desired. For any  $i$  with  $p_i < \gamma/c$  we use an arbitrary hypothesis  $h_i$  in the above analysis and assume only  $\text{err}_{\mathcal{D}_i}(h_i) \leq 1$ . It is easy to see that this incurs an additional error in (2) of at most  $\sum_{i: p_i < \gamma/c} p_i \leq \gamma$ , as necessary.  $\square$

Combining Theorem 5.15 with, say, Theorem 3.4 (for simplicity), we may conclude:

**Theorem 5.16** *Let  $\mathcal{D}$  be any known mixture of  $\text{poly}(n)$  product distributions over an instance space  $\mathcal{X} = X_1 \times \dots \times X_n$ , where we assume  $|X_i| \leq \text{poly}(n)$  for each  $i$ . Then there is a  $n^{O(k^2/\epsilon^4)}$ -time algorithm for agnostically learning with respect to the class of functions of  $k$  linear threshold functions over  $\mathcal{X}$  under  $\mathcal{D}$ .*

When the mixture of product distributions is not known a priori, we can first run an algorithm for learning mixtures of product distributions from unlabeled examples. For example, Feldman, O'Donnell, and Servedio [FOS05] proved the following:

**Theorem 5.17 ([FOS05])** *Let  $\mathcal{D}$  be an unknown mixture of  $c = O(1)$  many product distributions over an instance space  $\mathcal{X} = X_1 \times \dots \times X_n$ , where we assume  $|X_i| \leq O(1)$  for each  $i$ . There is an algorithm which, given i.i.d. examples from  $\mathcal{D}$  and  $\eta > 0$ , runs in time  $\text{poly}(n/\eta) \log(1/\delta)$  and with probability at least  $1 - \delta$  outputs the parameters of a mixture of  $c$  product distributions  $\mathcal{D}'$  satisfying  $\|\mathcal{D}' - \mathcal{D}\|_1 \leq \eta$ .*

(The theorem was originally stated in terms of KL-divergence but also holds with  $L_1$ -distance [FOS05].) In [FOS06] the same authors gave an analogous result for the case when each  $X_i = \mathbb{R}$  and each product distribution is a product of Gaussians with means and variances in  $[1/\text{poly}(n), \text{poly}(n)]$ .

We conclude:

**Theorem 5.18** *Let  $\mathcal{D}$  be any unknown mixture of  $O(1)$  product distributions over an instance space  $\mathcal{X} = X_1 \times \dots \times X_n$ , where we assume either: a)  $|X_i| \leq O(1)$  for each  $i$ ; or b) each  $X_i = \mathbb{R}$  and each product distribution is a mixture of axis-aligned  $\text{poly}(n)$ -bounded Gaussians. Then there is a  $n^{O(k^2/\epsilon^4)}$ -time algorithm for agnostically learning with respect to the class of functions of  $k$  linear threshold functions over  $\mathcal{X}$  under  $\mathcal{D}$ .*

**Proof:** First use the results of [FOS05, FOS06] with  $\eta = \epsilon/n^{O(k^2/\epsilon^4)}$ , producing a known mixture distribution  $\mathcal{D}'$  with  $\|\mathcal{D}' - \mathcal{D}\|_1 \leq \epsilon/n^{O(k^2/\epsilon^4)}$ . Then run the algorithm from Theorem 5.18. The conclusion now follows from Proposition 5.1.  $\square$

## 6 Conclusions

In this work, we have shown how to perform agnostic learning under arbitrary product distributions and even under limited mixtures of product distributions. The main technique was showing that noise sensitivity bounds under the uniform distribution on  $\{0, 1\}^n$  yield the same noise sensitivity bounds under arbitrary product distributions. The running time and examples required by our algorithm are virtually the same as those required for learning under the uniform distribution on  $\{0, 1\}^n$ .

While we have established many interesting scenarios for which polynomial regression works, there is still significant room for extension. One direction is to seek out new concept classes and/or distributions for which polynomial regression achieves polynomial-time agnostic learning. Our work has dealt mostly in the case where all the attributes are mutually independent; it would be especially interesting to get learning under discrete distributions that are far removed from this assumption.

## References

- [BKS99] Itai Benjamini, Gil Kalai, and Oded Schramm. Noise sensitivity of Boolean functions and applications to percolation. *Publ. Math. de l'IHÉS*, 90(1):5–43, 1999.
- [CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [FJS91] Merrick Furst, Jeffrey Jackson, and Sean Smith. Improved learning of  $AC^0$  functions. In *Proc. 4th Workshop on Comp. Learning Theory*, pages 317–325, 1991.
- [FOS05] Jonathan Feldman, Ryan O'Donnell, and Rocco Servedio. Learning mixtures of product distributions over discrete domains. In *Proc. 46th IEEE Symp. on Foundations of Comp. Sci.*, pages 501–510, 2005.
- [FOS06] Jonathan Feldman, Ryan O'Donnell, and Rocco Servedio. Pac learning mixtures of gaussians with no separation assumption. In *Proc. 19th Workshop on Comp. Learning Theory*, pages 20–34, 2006.
- [GR06] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. In *Proc. 47th IEEE Symp. on Foundations of Comp. Sci.*, pages 543–552, 2006.
- [Hås01] J. Håstad. A slight sharpening of LMN. *J. of Computing and Sys. Sci.*, 63(3):498–508, 2001.
- [Hoe48] Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *Ann. Math. Stat.*, 19(3):293–325, 1948.
- [Kal06] Adam Kalai. Machine learning theory course notes. <http://www.cc.gatech.edu/~atk/teaching/mlt06/lectures/mlt-06-10.pdf>, 2006.
- [KKMS05] Adam Kalai, Adam Klivans, Yishay Mansour, and Rocco Servedio. Agnostically learning halfspaces. In *Proc. 46th IEEE Symp. on Foundations of Comp. Sci.*, pages 11–20, 2005.
- [KOS04] Adam Klivans, Ryan O'Donnell, and Rocco Servedio. Learning intersections and thresholds of halfspaces. *J. of Computing and Sys. Sci.*, 68(4):808–840, 2004.
- [KR82] Samuel Karlin and Yosef Rinott. Applications of Anova type decompositions for comparisons of conditional variance statistics including jack-knife estimates. *Ann. Stat.*, 10(2):485–501, 1982.
- [KSS94] Michael Kearns, Robert Schapire, and Linda Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2):115–141, 1994.
- [LBW95] Wee Sun Lee, Peter Bartlett, and Robert Williamson. On efficient agnostic learning of linear combinations of basis functions. In *Proc. 8th Workshop on Comp. Learning Theory*, pages 369–376, 1995.
- [LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40(3):607–620, 1993.
- [MOO05] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. In *Proc. 46th IEEE Symp. on Foundations of Comp. Sci.*, pages 21–30, 2005.
- [MP69] Marvin Minsky and Seymour Papert. *Perceptrons*. MIT Press, 1969.
- [O'D03] Ryan O'Donnell. *Computational aspects of noise sensitivity*. PhD thesis, MIT, 2003.
- [OS03] Ryan O'Donnell and Rocco Servedio. New degree bounds for polynomial threshold functions. In *Proc. 35th ACM Symp. on the Theory of Computing*, pages 325–334, 2003.
- [Per04] Y. Peres. Noise stability of weighted majority. [arXiv:math/0412377v1](https://arxiv.org/abs/math/0412377v1), 2004.
- [Ste86] J. Michael Steele. An Efron-Stein inequality for nonsymmetric statistics. *Ann. Stat.*, 14(2):753–758, 1986.
- [Val84] Leslie Valiant. A theory of the learnable. *Comm. of the ACM*, 27(11):1134–1142, 1984.
- [vM47] Richard von Mises. On the asymptotic distribution of differentiable statistical functions. *Ann. Math. Stat.*, 18(3):309–348, 1947.

---

# How Local Should a Learning Method Be?

---

Alon Zakai \*

Interdisciplinary Center for Neural Computation  
Hebrew University of Jerusalem  
Jerusalem, Israel 91904  
alon.zakai@mail.huji.ac.il

Ya'acov Ritov \*

Department of Statistics  
Hebrew University of Jerusalem  
Jerusalem, Israel 91905  
yaacov.ritov@huji.ac.il

## Abstract

We consider the question of why modern machine learning methods like support vector machines outperform earlier nonparametric techniques like k-NN. Our approach investigates the *locality* of learning methods, i.e., the tendency to focus mainly on the close-by part of the training set when constructing a new guess at a particular location. We show that, on the one hand, we can expect *all* consistent learning methods to be local in some sense; hence if we consider consistency a desirable property then a degree of locality is unavoidable. On the other hand, we also claim that earlier methods like k-NN are local in a more strict manner which implies performance limitations. Thus, we argue that a degree of locality is necessary but that this should not be overdone. Support vector machines and related techniques strike a good balance in this matter, which we suggest may partially explain their good performance in practice.

## 1 Introduction

It is commonly seen in practice that modern methods in machine learning – such as kernel machines and more specifically support vector machines – outperform older techniques in nonparametric statistics such as k-NN [for a concrete example, see, e.g., Joa98]. The main approaches to explaining this phenomenon are margin-based bounds on the generalization error and that margin maximization in effect minimizes the VC dimension, again, arriving at a favorable bound on the generalization error [Vap98, STC00]. In this work we consider an alternative approach to investigating this matter, in hopes of showing the underlying issues in a different light.

We will focus on **local learning**, i.e., the property of a learning method that it uses mainly the close-by part of the training set to construct new guesses. That is, when an estimate is generated at a point  $x$  using a training set  $S_n = \{(x_i, y_i)\}_{i=1..n}$  (i.e., we are trying to guess a corresponding value of  $y$  for  $x$ , using  $x$  and the training set), then a local method is one that is influenced mostly by the points

$(x_i, y_i)$  for which  $x_i$  is close to  $x$ . Many classical methods in nonparametric statistics are clearly of this sort, e.g., k-NN. This is often stated as a detriment of such methods, in particular since local learning is susceptible to the curse of dimensionality – in high-dimensional spaces, one needs a great many points in the training set in order for a sufficient amount to end up close-by to the point  $x$  currently being estimated. On the other hand, methods like support vector machines appear non-local in their definition – the separating hyperplane is determined by the entire training set, and furthermore does not depend on the particular point we intend to estimate at – and thus one might suspect that the superior performance of support vector machines and related techniques is connected to this matter.

However, whether this is the case is not immediately obvious. In fact, we might suspect that many kernel machines behave locally: consider that a typical kernel machine can be written as  $\sum_i \alpha_i y_i k(x_i, x)$ , where  $k$  is a kernel function, e.g., the RBF kernel  $k_{\text{RBF}}(x, x') = \exp(-\gamma \|x - x'\|^2)$  (we do not write the sign operation, which would appear here if our goal is classification and not regression, but the issue is the same in that case). This appears outwardly similar to weighted k-NN, whose general form is  $\frac{\sum_i y_i s(x_i, x)}{\sum_i s(x_i, x)}$ , where the sum is taken over the  $K$  nearest neighbors of  $x$  and  $s$  is a similarity measure; in fact, we can take  $s = k_{\text{RBF}}$ . Also similar in form is another classical statistical technique, kernel estimators, which can be written as  $\frac{\sum_i y_i k(\|x_i - x\|)}{\sum_i k(\|x_i - x\|)}$  where  $k: \mathbb{R} \rightarrow \mathbb{R}$  has compact support. It appears that the main difference between kernel machines and the earlier techniques lies in the coefficients; for kernel machines,  $\alpha_i$  is determined in a manner based on the entire training set, and not just the local subset of it. Perhaps, then, this might lead to non-local behavior of some sort, and in conclusion it is not immediately obvious whether kernel machines behave locally or not. We are therefore in need of an analysis to give us an answer.

For convenience, we will from now on refer to kernel machines as **'modern methods'**; we mean mainly support vector machines and related techniques, specifically, ones that use both maximal-margin separation and the 'kernel trick' [Vap98], but to a lesser degree also boosting [FS99], which similarly appears to have good performance due to margin maximization [SFBL97]. By **'classical methods'** we will refer to older techniques studied in the statistical literature, the prime examples of which are, as mentioned in the previous paragraph, k-NN and kernel estimators; another example

---

\*We would like to acknowledge support for this project from the Israel Science Foundation as well as partial support from NSF grant DMS-0605236.

is local regression [CL95]. Using this terminology, our goal is to explain, at least in part, the performance advantage of modern methods over classical ones.

As we have seen, we are in need of an analysis to tell us whether modern methods behave locally or not. One such analysis was carried out in [BDR06], with the conclusion that kernel machines do in fact behave locally in some sense. If so, then it might appear that being local cannot explain the performance advantage of modern methods over classical ones, since apparently both approaches have that property. We will argue against this notion, while at the same time agreeing with the results in [BDR06]. Specifically, we will first show that indeed both modern and classical methods behave locally in some sense, but that the underlying cause is the property of consistency, i.e., that the method is able to successfully learn given any distribution (we leave a formal definition to the next section). Importantly, however, classical methods are local in a far stricter manner, and we will show that such strict locality implies performance limitations. Thus, a degree of locality is necessary for consistency, but is detrimental if taken to excess. We hypothesize that modern methods in fact strike a good balance in this matter, which may help to explain their superior performance.

In more general terms, based on our results we will argue in the discussion (Section 7) that the challenge in devising useful learning methods is to combine a local aspect, which is necessary for consistency, with a global aspect, which is useful for improving performance; the prime example of such a performance-improving global aspect is of course maximal-margin separation. To see this point, consider first that k-NN is defined in a simple manner that immediately ensures it is local (from its very definition), which allows it to be consistent, as we will see, with little additional work. However, it is hard to incorporate into such a local technique a non-local regularization method like maximal-margin separation. On the other hand, if we start with a method using maximal-margin separation then it is not trivial to ensure that it behaves locally, which we will see is a precondition for consistency. In other words, we want our learning methods to (1) be local, so that they may be consistent, and (2) apply a global regularization method, since this improves performance in practice. Devising a method having both of these properties at the same time is not trivial to achieve, but support vector machines and related methods do manage to do so: on the one hand they utilize maximal-margin separation, while on the other via the ‘kernel trick’ they end up having sufficiently local behavior in order to be consistent (assuming we choose an appropriate kernel and so forth). We will discuss this argument more at length in the discussion at the end of this work.

Technically speaking, the analysis that we conduct in order to arrive at the conclusions just mentioned is based on definitions inspired by those in [ZR07]. The main difference from that work is that local behavior was defined there by comparing a method’s response on the entire training set to the ‘local training set’, which contains only the close-by points. This approach has the advantage of having practical applications in that it can answer what might occur if we ‘localize’, say, a support vector machine (i.e., show it only close-by points, as done in k-NN). However, the comparison

of a method’s response on two training sets of different size (the local one is in all reasonable cases smaller) has the disadvantage that it is hard to talk about subtle degrees of locality, since the change in the size of the training set introduces a source of variability. Our goal in the present work is in fact to speak about such differences of degree. We therefore define locality differently, by considering changes to far-off points instead of removing them from the training set, which keeps the size of the training set fixed. Why this is helpful will become clear later on.

The structure of the rest of this work is as follows. In Section 2 we describe the formal setting of the problem and lay out notation. In Section 3 we define locality and other concepts and give an overview of our results. In Section 4 we present our results for consistency and its connection to weak locality. In Section 5 we turn to strict locality and its drawbacks. In Section 6 we deal with the application of our results to classification. Finally, in Section 7 we summarize and discuss our results.

## 2 Formal Setting

We now complete the formal description of the setting. We are given an i.i.d sample  $S_n = \{(x_i, y_i)\}_{i=1..n}$  from some distribution  $P$ ; then a new (independent) pair  $(x, y)$  is drawn from the same  $P$  and our goal is to predict  $y$  when shown only  $x$ . Our prediction (also called estimate, or guess) of  $y$  is written  $f(S_n, x)$ , some measurable function that depends both on the training set and the point to be estimated (note that this notation, where the training set and new observation are of equal standing as inputs to  $f$ , is slightly atypical, but is very convenient in our setting, as will become clear later). We call  $f$  a *learning method* (sometimes *method* or *estimator*); note that it can produce guesses for any size training set and any  $x$ . One specific context is that of classification (also called pattern recognition) where  $y \in \{-1, +1\}$ ; we call learning methods in this context classifiers and call  $y$  the class. While our results apply to classification, we will not focus on it in most of this work, since a regression-type setting is simpler to deal with. Later on, in Section 6, we will show how to apply our results to classification.

Our goal is to estimate  $f^*(x) = E(y|x)$ , that is, the expected value of  $y$  conditioned on  $x$ , or the regression of  $y$  on  $x$ . Our hope is that  $f(S_n, x)$  is close to  $f^*(x)$ . We say that  $f$  is **consistent** on a distribution  $P$  iff

$$L_{n,P}(f) \equiv E_{S_n, x \sim P} |f(S_n, x) - f^*(x)| \xrightarrow{n \rightarrow \infty} 0$$

where the expected value is taken over all training sets  $S_n$  and observations  $x$  both distributed according to  $P$ . We will omit  $P$  from  $L_{n,P}$  when the distribution is clear from the context, and we will generally further shorten our notation to write expressions of the form

$$L_n(f) \equiv E_{S_n, x} |f(S_n, x) - f^*(x)|$$

when, again, the distribution is clear from the context. (Note that the choice of the absolute value in the  $L_n$  loss – i.e., the  $\mathcal{L}_1$  norm – is only for convenience; our results hold in the more typical  $\mathcal{L}_2$  norm as well.)

If a method is consistent on all  $P$  then we call it consistent (this is sometimes called *universal consistency*). Importantly for us, methods like support vector machines and

boosting are consistent [Ste02, Zha04], or at least can be if the parameters are chosen accordingly. In fact such choices often turn out to lead to good performance in practice, and therefore we are interested in consistent versions of modern methods. We will return to this matter in the discussion.

The following notation will be used. Denote by  $\mu_P$  (or just  $\mu$ , if  $P$  is clear from the context) the marginal measure on  $x$  of a distribution  $P$ . We denote random variables by, for example,  $(x, y) \sim P$  and  $S_n \sim P$  where the latter indicates a random i.i.d sample of  $n$  elements from the distribution  $P$ . We will often abbreviate and write  $x \sim P$  where we mean  $x \sim \mu_P$ . To prevent confusion we always use  $x$  and  $y$  to indicate a pair  $(x, y)$  sampled from  $P$ .

As already implied, we write expected values in the form  $E_{v \sim V} H(v)$  where  $v$  is a random variable distributed according to  $V$ . We denote probabilities by, e.g.,  $P_{v \sim V}(U(v))$ , which is the probability of an event  $U(v)$  taken over a random variable  $v$ . In both cases we will omit  $V$  when it is clear from the context.

For any set  $B \subseteq X$ , denote by  $P_B$  the conditioning of  $P$  on  $B$ , that is, the conditioning of  $\mu_P$  on  $B$  (and the limiting of  $f^*$ 's domain to  $B$ ). We denote  $B_{x,r} = \{x' \in \mathbb{R}^d : \|x - x'\| \leq r\}$ , the ball of radius  $r$  around  $x$  (using the Euclidean norm). Let  $P_{x,r} = P_{B_{x,r}}$ .

Finally, we make the following assumptions which are mainly for convenience. Our distributions  $P$  are on  $(X, Y)$  where  $X \subset \mathbb{R}^d, Y \subset \mathbb{R}$  (i.e., we work in Euclidean spaces). We assume that  $X, Y$  are bounded

$$\sup_{x \in X} \|x\|, \sup_{y \in Y} |y| \leq M_1$$

for some  $M_1 > 0$  which is the same for all distributions. Thus, when we say ‘all distributions’ we mean all distributions bounded by the same value of  $M_1$ . We also assume that our learning methods return bounded responses,  $|f(S_n, x)| \leq M_2$  for some  $M_2 > 0$ .<sup>1</sup> Let  $M$  be a constant fulfilling  $M \geq M_1, M_2$ .

### 3 Definitions and Overview

We will now define the main concepts that concern us, starting first with some convenient notation. For any training set  $S_n = \{(x_i, y_i)\}$  and values  $x \in X, r \geq 0, \{\tilde{y}_i\} \in Y^n$ , let  $S_n(x, r, \{\tilde{y}_i\}) =$

$$\left\{ \left( x_i, 1\{\|x - x_i\| \leq r\}y_i + 1\{\|x - x_i\| > r\}\tilde{y}_i \right) \right\}$$

That is,  $S_n(x, r, \{\tilde{y}_i\})$  does not change the locations  $x_i$ , and has the original  $y$  values  $y_i$  close-by to  $x$  (up to distance  $r$ ), while replacing far-off labels with  $\tilde{y}_i$ . We can now define one sense of locality: we call a method  $f$  **local** on a distribution  $P$  iff there exists a sequence  $\{R_n\}, R_n \searrow 0$ , for which

$$E_{\{x_i\}, x} \sup_{\{y_i\}, \{\tilde{y}_i\}} |f(S_n, x) - f(S_n(x, R_n, \{\tilde{y}_i\}), x)| \xrightarrow{n \rightarrow \infty} 0$$

<sup>1</sup>Note that this is a minor assumption since for essentially all modern and classical methods we have  $\sup_x |f(S_n, x)| \leq C \cdot \max_i |y_i|$  for some  $C > 0$ , and the  $y_i$  values are already assumed to be bounded. Furthermore, we are concerned with consistent methods, i.e., that behave similarly to  $f^*$  in the limit, and  $f^*$  is bounded.

(Here  $S_n = \{(x_i, y_i)\}$ , following our usual notation, i.e.,  $S_n$  is constructed from  $\{x_i\}$  in the expectation and  $\{y_i\}$  in the sup.)

This definition is fairly straightforward: a method is local if, asymptotically speaking, it returns very similar results when we change far-off labels. Thus, the method is influenced mainly by the close-by part of the training set, which is the intuition behind a local method. Note that, since  $R_n \rightarrow 0$ , in effect the method is influenced only by the local part of the training set in a strong sense. Note also that the definition speaks of locality on a single distribution  $P$ ; as with consistency, if a method  $f$  is local on all  $P$  then we say that  $f$  is local (i.e., if  $P$  is not specified, we mean all  $P$ ). We will use this convention with other definitions as well.

It turns out that there are more useful ways to define locality, for reasons which we will see later. One such definition of locality is weaker than that given before, and one is stronger. We start with the weaker:

**Definition 1** Call a method  $f$  **weakly local** on a distribution  $P$  iff, for every distribution  $\tilde{P}$  for which  $\mu_P = \mu_{\tilde{P}}$ , there exists a sequence  $\{R_n\}, R_n \searrow 0$  for which

$$E_{\{x_i\}, x} E_{\{y_i\} \sim P | \{x_i\}, \{\tilde{y}_i\} \sim \tilde{P} | \{x_i\}} |f(S_n, x) - f(S_n(x, R_n, \{\tilde{y}_i\}), x)| \xrightarrow{n \rightarrow \infty} 0$$

(Here  $\{y_i\} \sim P | \{x_i\}$  means that each  $y_i$  has distribution  $P$  conditioned on  $x_i$ .)

Thus, a weakly local method is one which, if we replace far-off labels with labels from another distribution, is asymptotically not influenced by that change (note that we keep  $\mu$ , the measure on  $x$ , fixed; we care only about changes to  $y$  values). This definition is weaker than the one given before in that instead of the supremum over all  $y$ , we sample alternate  $y$  values from a fixed distribution. However, since we require that this property occur for *all* distributions  $\tilde{P}$ , we still have the essential behavior of being most influenced by the close-by part of the training set.

In one of our main results we will see that all consistent learnings methods are in fact very close to being weakly local (we will require a minor technical relaxation of the definition given above). Hence this is true for, e.g., support vector machines, assuming the kernel and parameters ensure consistency.

It is obvious that classical methods like k-NN and kernel estimators are weakly local, both because they are consistent [see DGKL94, GKP84, respectively], and by direct inspection, see Section 5. However, they seem to be local in a stronger sense than that appearing in weak locality. In fact they have the following stronger property:

**Definition 2** Call a method  $f$  **strictly local** on a distribution  $P$  iff there exists a sequence  $\{R_n\}, R_n \searrow 0$ , for which

$$P_{\{x_i\}, x} \left( \forall \{y_i\}, \{\tilde{y}_i\} \quad f(S_n, x) = f(S_n(x, R_n, \{\tilde{y}_i\}), x) \right) \xrightarrow{n \rightarrow \infty} 1$$

Thus, a strictly local estimator is one for which we can replace far-off labels and this, with probability going to 1, will not affect our estimates at all (this is easily seen to be

stronger than the original definition of locality due to the boundedness assumption on  $f$ ). Note that we can consider stricter notions of locality, however, this definition is strict enough, since classical methods fulfill it.

We will see later in Section 5 that, unlike classical methods, many (if not all) modern methods are **not** strictly local, and that this has potentially important consequences, since strictly local methods have performance limitations.

In [ZR07] similar definitions appeared. In that work, local behavior was defined by comparing  $f$ 's response to the response it would have given had far-off points been removed from the training set, whereas in the definitions given above we consider changes to their  $y$  values instead. As mentioned in the introduction, the reason for this is the need to consider varying degrees of locality. In our definitions, we can either change the  $y$  values to values sampled from a fixed distribution (weak locality) or consider all possible changes (locality, and, in a stronger sense, strict locality). We will see that these differences can in fact be of importance. A further reason for preferring our definitions over ones in which far-off examples are removed is that the latter approach changes the size of the training set, and in a data-dependent manner. This introduces a source of variability which then makes it hard to talk about concepts like strict locality, where we require that with high probability there be no change in the response; if  $n$  changes, this itself may cause an alteration (e.g., this occurs in the common case where a regularization constant is used whose value depends on  $n$ ). Alternatively, we might have removed a fixed number of far-off observations depending on  $n$  (as in k-NN, in fact), but this causes other inconveniences in that the radius in which the remaining observations lie is now a random variable (which is, as before, a source of variability). Replacing far-off  $y$  values, as we have chosen to do, therefore seems the most productive choice.

We now survey other related work. Research regarding locality was done in the context of learning methods that work by minimizing a loss function. Such loss functions can be 'localized' by re-weighting them so that close-by points are more influential; see [BV92], [VB93] for such an approach in the setting of Empirical Risk Minimization [ERM; Vap98] and [CL95] and references therein for the specific case of linear regression; see [AMS97] for a survey of applications in this area. The approach we follow differs from this one in that we focus on consistency in the sense of asymptotically arriving at the lowest possible loss achievable by any measurable function – i.e., in the nonparametric sense – and not in the sense of minimizing the loss within a set of finite VC dimension. The nonparametric sense is, we believe, the one most relevant to locality, and the best context in which to compare modern and classical methods.

We now briefly summarize our two main results. First, regarding the connection between consistency and weak locality, let us consider now a property weaker than consistency. Define the means of  $f$  and  $f^*$  by

$$\begin{aligned} E_n(f) &\equiv E_{n,P}(f) \equiv E_{S_n,x} f(S_n, x) \\ E(f^*) &\equiv E_P(f^*) \equiv E_x f^*(x) = E_x E(y|x) = Ey \end{aligned}$$

the latter expression which is just the global mean of  $y$ , and define  $f, f^*$ 's Mean Absolute Deviations (MADs) by

$$\begin{aligned} \text{MAD}_n(f) &\equiv \text{MAD}_{n,P}(f) \equiv E_{S_n,x} |f(S_n, x) - E_n(f)| \\ \text{MAD}(f^*) &\equiv \text{MAD}_P(f^*) \equiv E_x |f^*(x) - E(f^*)| \end{aligned}$$

(we prefer the MAD over the variance due to the choice of the  $\mathcal{L}_1$  norm). We define

**Definition 3** Call a method  $f$  **Weakly Consistent in Mean (WCM)** iff there exists a function  $H : \mathbb{R} \rightarrow \mathbb{R}$ ,  $H(0) = 0$ ,  $\lim_{t \rightarrow 0} H(t) = 0$ , for which,  $\forall P$ ,

$$\left\{ \begin{array}{l} \limsup_{n \rightarrow \infty} |E_n(f) - E(f^*)| \\ \limsup_{n \rightarrow \infty} \text{MAD}_n(f) \end{array} \right\} \leq H(\text{MAD}(f^*))$$

(Note that the same  $H$  is used for all  $P$ .)

A WCM learning method is required only to do 'reasonably' well in estimating the global properties of the distribution – the mean and MAD, which are two scalar values – in a way that depends on the MAD, i.e., on the difficulty; we only require that performance be good when the learning task is overall quite easy, in the sense of  $f^*(x)$  being almost constant. Note that when  $H(\text{MAD}(f^*)) \geq 2M$  we require nothing of  $f$  for such  $f^*$  (since  $|f|, |f^*| \leq M$ ), and that also for small  $\text{MAD}(f^*)$  we may allow the MAD of  $f$  to be significantly larger than that of  $f^*$  (consider, for example,  $H(t) = c \cdot (\sqrt{t} + t)$  for large  $c > 0$ ).

It is easy to see that WCM is weaker than consistency and implied by it. Assuming consistency,

$$\begin{aligned} |E_n(f) - E(f^*)| &= |E_{S_n,x}(f(S_n, x) - f^*(x))| \\ &\leq |E_{S_n,x} |f(S_n, x) - f^*(x)| \quad (1) \\ &= |L_n(f)| \rightarrow 0 \end{aligned}$$

(note that here even  $H(t) \equiv 0$  would have worked), and

$$\begin{aligned} \limsup_n \text{MAD}_n(f) &= \limsup_n E_{S_n,x} |f(S_n, x) - E_n(f)| \\ &\leq \limsup_n \left\{ E_{S_n,x} |f(S_n, x) - f^*(x)| \right. \\ &\quad \left. + E_x |f^*(x) - E(f^*)| \right. \\ &\quad \left. + |E(f^*) - E_n(f)| \right\} \\ &= \text{MAD}(f^*) \end{aligned}$$

using the consistency of  $f$  and (1); thus,  $H(t) = t$  shows that the WCM property holds for all consistent methods.

We can now ask, what is missing in WCM that is present in consistency? Since WCM is a 'global' property (concerned only with two scalar values that are functions of the entire space), it seems apparent that what is missing in WCM is some 'local' aspect, i.e., of correctly learning in each small area separately. We will see that in fact a property very similar to weak locality can fill that role; we will call that definition Uniform Approximate Weak Locality (UAWL). We will then prove that consistency is logically equivalent to the combination of UAWL and WCM. From our definitions it will be easy to see that the UAWL and WCM properties are 'independent' in the sense that neither implies the other. Thus, we can see consistency as comprised of two independent properties, which might be presented as

$$\text{Consistency} \iff \text{UAWL} \oplus \text{WCM}$$

Thus, our first conclusion is that a form of local behavior is fundamental to consistency; any consistent method must be in a sense local, no matter how it is defined. In fact, the difference between consistency and locality comes down to the additional requirement in consistency that we also are not far from estimating global properties of the distribution, as formalized by the WCM property.<sup>2</sup> This means that if we start with a method defined in an explicitly local manner, like k-NN, then we get ‘for free’ the property of UAWL. Then all we need to do to get consistency is to ensure the WCM property, which is relatively simple (we just need the scalar value representing our global mean to converge to the accurate one, and our MAD to not be too large). Since consistency is a desirable property, this explains some of the attractiveness of classical methods: achieving consistency with them is relatively simple.

Our second main result will show the drawbacks of this simplicity of classical methods, and will concern strict locality. To show the limitations of strict locality, we define the following property: call a method  $g$  **preferable** to another method  $f$ , over a set of distributions  $\mathcal{P}$ , iff, for every  $P \in \mathcal{P}$ ,

$$L_n(g) < L_n(f)$$

for large enough  $n$  (possibly depending on  $P$ ). That is, no matter what the true distribution is out of those in  $\mathcal{P}$ ,  $g$  is eventually better than  $f$ . Our claim is then that, for every strictly local method  $f$ , we can always construct a non-strictly local  $g$  which is preferable to  $f$ . For convenience we will show this on a specific example, but argue that the result is a quite general one.

#### 4 Weak Locality and Consistency

As hinted at before, it turns out that a slight complication of our definition of weak locality is necessary. To present the improved definition, we start with some preparatory notation. For any  $q \geq 0$  and distribution  $P$ , let

$$\bar{f}^q(S_n, x) = E_{x' \sim P_{x,q}} f(S_n, x')$$

That is,  $\bar{f}^q$  applies a ‘smoothing’ operation performed around the  $x$  being estimated (recall that  $P_{x,q}$  is  $P$  conditioned on the ball of radius  $q$  around  $x$ ). Note that if  $q = 0$  then we interpret the expected value as a delta function and we get  $\bar{f}^0 = f$ . Note also that we require the actual unknown distribution  $P$  in the definition of  $\bar{f}^q$ , i.e.,  $\bar{f}^q$  cannot be directly implemented in practice –  $\bar{f}^q$  is a construction for theoretical purposes.

We define the following set of sequences:

$$\mathcal{T} = \{\{T_n\} : T_n \searrow 0\}$$

and, for any sequence  $T = \{T_n\} \in \mathcal{T}$ , we define the set of its infinite subsequences and selection functions on them by

$$\mathcal{R}(T) = \{\{R_n\} : \{R_n\} \subseteq T, R_n \searrow 0\}$$

$$\mathcal{Q}(T) = \{Q : T \rightarrow T : Q(T_n) = o(T_n)\}$$

<sup>2</sup>Note that we need both the mean and the MAD to behave in an appropriate way, as appearing in the definition of the WCM property, because if only the mean is accurate then due to the variance we may estimate the global properties very poorly.

We now motivate these definitions. First, regarding  $\mathcal{T}$ : instead of allowing any possible value in  $[0, \infty)$  for  $R_n$  and  $Q$ , we limit them to a countable set  $\mathcal{T}$ . The reason for this is that due to  $[0, \infty)$  being an uncountable set it is not clear to the authors if additional conditions are not required to prove our results in that case. In any event, a countable set of possible values is of sufficient interest for any practical learning-theoretical purpose, since we end up using only a countable number of  $R_n, Q$  values (since  $n \in \mathbb{N}$ ). Note that the set of possible values  $\mathcal{T}$  can be chosen in whatever manner is desired, so long as this is done in advance.

$\mathcal{R}(T)$  contains **localizing sequences**, sequences of radii that determine how far off we alter the data shown when we perform  $S_n(x, R_n, \{\tilde{y}_i\})$ . We require that  $R_n \searrow 0$ , as we are interested in learning methods that focus on the truly local part of the training set, i.e., having radius 0 asymptotically.

$\mathcal{Q}(T)$  contains functions of the possible values  $T$  that become negligibly small when  $T_n$  is small. We will use the values  $Q(R_n)$  to determine radii on which to smooth, via  $Q(R_n)$ , which we might call the **smoothing radius**; note that since  $Q(R_n) = o(R_n)$ , we smooth on a radius much smaller than  $R_n$ , hence this is a fairly minor operation.

Finally, we define

$$\mathcal{R}^+(T) = \{\{R_n\} : \{R_n\} \subseteq T\}$$

$$\mathcal{Q}^+(T) = \{Q : T \rightarrow T\}$$

which are the same as before, but without the requirement of converging to 0. We now arrive at our main definition for this section, whose description is unavoidably technical:

**Definition 4** Call a learning method  $f$  **Uniformly Approximately Weakly Local (UAWL)** iff

$$\forall P, \tilde{P}, \mu_P = \mu_{\tilde{P}}$$

$$\forall T \in \mathcal{T}$$

$$\exists Q \in \mathcal{Q}(T)$$

$$\forall Q' \in \mathcal{Q}^+(T), Q' \geq Q$$

$$\exists \{R_n\} \in \mathcal{R}(T)$$

$$\forall \{R'_n\} \in \mathcal{R}^+(T), R'_n \geq R_n$$

$$E_{\{x_i\}, x} E_{\{y_i\} \sim P | \{x_i\}, \{\tilde{y}_i\} \sim \tilde{P} | \{x_i\}}$$

$$|f(S_n, x) - \bar{f}^{Q'(R'_n)}(S_n(x, R'_n, \{\tilde{y}_i\}), x)| \xrightarrow{n \rightarrow \infty} 0$$

(Here the expression  $R'_n \geq R_n$  simply implies an inequality for the entire series, i.e., for all  $n$ .  $Q' \geq Q$  implies  $Q'(T_k) \geq Q(T_k)$  for all  $k$ .)

Thus, a UAWL method returns similar values even when we replace far-off data with different values of  $y$ ; essentially the same idea as with weak locality, but allowing for minor smoothing, and requiring uniformity in  $Q, R_n$ . With a UAWL method, loosely speaking, for any large enough  $Q, R_n$  we get local behavior. Note that the notion of  $R_n$  being large enough is a natural one since taking  $R_n$  to 0 very quickly is problematic (doing so may lead to us getting few or no points in radius  $R_n$ , i.e., few or no points from the important distribution).

The reason for including smoothing in this definition is that, if all we assume is that learning methods are measurable

(and not smooth in some strong sense), then odd counterexamples exist to the connection between locality and consistency; see [ZR07] for details. By incorporating smoothing in our definition we remove the need to require it of the learning methods we consider, which lets us apply our results to any method known to be consistent. The reason for the second new aspect in this definition, that of allowing all large-enough  $Q', R'_n$ , is that this leads to an exact equivalence with consistency, as we will see in Theorem 5; furthermore, it would be odd for the locality of a method to depend much on the specific  $Q, R_n$  used for it. To make the matter concrete, note that the proof of Theorem 5 requires using the same  $Q, R_n$  over multiple distributions; without allowing all large-enough  $Q', R'_n$  there exist odd counterexamples in which each distribution has some appropriate  $Q, R_n$  but none exist that are appropriate for all of them simultaneously.

Our result for consistency is the following:

**Theorem 5** *A learning method  $f$  is consistent iff  $f$  is both UAWL and WCM.*

We prove the  $\Leftarrow$  direction, that UAWL and WCM imply consistency, in Appendix A. Note that it is clear from the proof that we can replace  $\tilde{P}$  in the definition of UAWL with all distributions having  $y$  constant, but we believe the definition given before is clearer.

For the  $\Rightarrow$  direction, that consistency implies UAWL and WCM, it is immediately obvious that consistency implies WCM. Regarding UAWL, a proof of a slightly simpler claim (without uniformity in  $R_n, Q$ ) appears in [ZR07]; using methods from other proofs in [ZR07], it is trivial to extend the proof to showing uniformity as well. For completeness we give a brief sketch of the proof appearing there: for fixed  $r, q$  instead of  $R_n, Q$ , we can use the consistency of  $f$  on the effective distributions seen (i.e., distributions that are altered to  $\tilde{P}$  far away from  $x$ ) to see that the appropriate loss converges to 0, for every  $x$  separately. Since, again for every  $x$ , the overall loss converges to 0, this also occurs in the area with radius  $q$ , which is the one relevant to us. We then take  $R_n, Q$  to 0 slowly enough to complete the proof.

Theorem 5 can be summarized as follows:

$$\text{Consistency} \iff \text{UAWL} \oplus \text{WCM}$$

Here we use the symbol  $\oplus$  because each of the two properties UAWL and WCM can exist without the other: consider the following two methods,

$$f_y(S_n, x) = \frac{1}{n} \sum_{i=1}^n y_i \quad f_0(S_n, x) = 0$$

$f_y$  (called thus because it considers only the  $y$  values) is WCM, since  $E_n(f_y) \rightarrow E(f^*)$  and clearly  $f_y$ 's MAD converges to 0. (In fact,  $f_y$  is WCM with  $H \equiv 0$ , i.e., in the strongest sense. That is, there are even 'weaker' methods that are WCM.) On the other hand,  $f_y$  is clearly not UAWL (consider, e.g., two distributions having  $f^*(x) \equiv -1, f^*(x) \equiv +1$ ). On the other hand,  $f_0$  is trivially UAWL, but not WCM.

## 5 Strict Locality

In this section we will deal with strict locality and its consequences.

It is immediately clear that kernel estimators are strictly local (use  $R_n$  equal to the bandwidth, and recall that  $k$  has compact support). For k-NN things are less obvious, but still fairly simple: k-NN is consistent if the number of neighbors  $k_n$  fulfills  $k_n \rightarrow \infty, \frac{k_n}{n} \rightarrow 0$  [DGKL94]. From inspecting the proof of consistency it is clear that these conditions ensure that the  $k_n$  neighbors will fall in an area of radius going to zero, with probability going to 1. Thus (unsurprisingly) k-NN is strictly local: just like kernel estimators, it completely ignores far-off points, but it does so with very high probability instead of certainty (since there is always a chance, even though it becomes negligibly small, that we will need to look far for the  $k_n$  nearest neighbors).

We have seen that any consistent method must be in some sense local, specifically, UAWL. We can now ask, must a consistent method also be strictly local? It turns out that the answer is no. Consider, for example, kernel ridge regression [SGV98], which can be written in the kernel-induced space (via a transformation  $\phi$ ) as

$$L(w) = \frac{1}{n} \sum_i (w' \phi(x_i) - y_i)^2 + \lambda \|w\|^2 \quad (2)$$

It is clear that under mild regularity conditions we will not get strict locality, since any change to the  $y_i$  values can cause a change to the resulting  $w$ , as is obvious from looking at the solution to (2); thus, kernel ridge regression is not strictly local. It appears clear that a similar phenomenon occurs for other types of kernel machines, as well as methods such as boosting (but we do not supply a formal proof), simply because there is always the possibility of influence by far-off points (as is also clear from these methods minimizing a global loss function which is an average of losses at individual points; any change to a point influences the overall loss, with potential consequences on the entire space). While the influence of far-off points wanes as  $n$  converges to infinity – which is necessary, as we have seen, in order for the method to be consistent – the far-off points are not simply ignored as with classical methods like k-NN. There is always the possibility of being influenced by the farther points, even if this is a rare occurrence.

We will now see that the property of potentially being influenced by far-off points can, in fact, be important. The reason is that strictly local methods have performance limitations. As is well known, to talk in a meaningful way about performance, we cannot make comparisons on the set of all distributions [see, e.g., DGL96]. We therefore consider limited sets of distributions, as is done in the minimax setting in statistics. We first begin with a brief reminder of the setting and how minimax losses can be achieved.

Assume for simplicity a Lipschitz set of functions  $f^* \in \mathcal{L}(L)$  on  $[0, 1]^d$ ,

$$|f^*(x_1) - f^*(x_2)| \leq L \cdot \|x_1 - x_2\|$$

and take  $x$  uniform on  $X = [0, 1]^d$ ; let  $y = f^*(x) + \epsilon, \epsilon \sim N(0, \sigma^2)$ . Consider a simple kernel estimator with radius  $r$ ,

$$f(S_n, x) = \frac{\sum_i 1\{|x_i - x| \leq r\} y_i}{\sum_i 1\{|x_i - x| \leq r\}}$$

For every  $x_0$ , we receive on average on the order of  $nr^d$  points in radius  $r$  to estimate  $f^*(x_0)$ , so we can estimate

$Ef^*(x)$  in that area up to precision  $\frac{\sigma}{\sqrt{nr^d}}$ . It is also clear that  $Ef^*(x)$  differs from  $f^*(x_0)$  by up to  $Lr$ , giving us roughly  $L_n(f) \leq \frac{\sigma}{\sqrt{nr^d}} + Lr$ , an example of a bias-variance trade-off (the bias is due to estimating  $Ef^*(x)$  on  $B_{x_0,r}$  and not  $f^*(x_0)$  directly, and the variance is due to having only the order of  $nr^d$  points). From this simple analysis it can be concluded that a choice of  $r = r_n = O(n^{-1/(d+2)})$  is appropriate, and that this will give us a loss of  $O(n^{-1/(d+2)})$ . This is in fact the minimax rate, i.e., the best-possible achievable rate, as shown in [Sto80, Sto82].

Importantly, notice how we must consider close-by points in order to arrive at the rate: if we look only at points at distance  $r$  or more, then  $f^*(x_0)$  may differ by up to  $Lr$  and we would not be able to overcome this issue in a minimax sense. Furthermore, it is also obvious from the analysis that the close-by points are enough in order to achieve the rate, i.e., to be up to a constant factor of the actual minimax loss. This can be directly seen by the equality of the bias and variance factors when we minimize their sum.

Thus, even a strictly local method like kernel estimators can achieve the minimax rate; in that sense, there is nothing to improve upon. In the example above the rate is  $n^{-1/(d+2)}$ , and kernel estimators can achieve it, but we have no assurance that they do so with a low constant factor; since such constant factors are hard to analyze, they are for the most part ignored in statistics. While this is reasonable in the sense that the rate is arguably the most important aspect in an asymptotic analysis, in actual practice – i.e., when working with some fixed finite  $n$  – the constant factor can be critical, since for fixed finite  $n$  we do not care about the asymptotic rate but only about the actual value of  $L_n$ . We will now make such a comparison of the actual values of  $L_n$  and claim that strictly local methods are limited in their ability to minimize it.

As defined previously, call a method  $g$  **preferable** to another method  $f$ , over a set of distributions  $\mathcal{P}$ , iff, for every  $P \in \mathcal{P}$ ,

$$L_n(g) < L_n(f)$$

for large enough  $n$  (possibly depending on  $P$ ). We will now see that in fact it is simple to construct a method preferable to any strictly local method, thus showing that strict locality brings with it performance limitations. The reason for the limitation is easy to see: by completely ignoring far-off points, there is no ability to adapt to rare occurrences in which those far-off points are in fact necessary for good performance. In statistical terms, while we have lower bias with the close-by points, we have lower variance with the farther-off ones due to their greater number. On average we prefer to balance these two out, as shown above, but in specific cases we can do better than such an average; consider, for example, the unlikely but possible case where the close-by points have bizarre values (e.g., their empirical variance is much larger than  $\sigma^2$  in the example above); in such a case, based on the empirical sample we can tell that it would probably be better to focus on slightly farther off points. That is, while on average the close-by points are most relevant, there is a minority of cases in which they are in fact misleading, and in at least some of those cases we can tell when they occur, at least with high probability. We will now formalize this notion in a concrete result in a specific setting. While only one

example, the underlying issue just mentioned should hold in a wide range of cases.

The following definition will make our result easier to state: call a method  $f$  **reasonable** iff, when all  $y_i$  in  $S_n$  have the same value,  $f$  returns that value. Note that practically every existing learning method has this property, including those of interest to us, and that in fact all consistent methods must have this property in an asymptotic sense in order to be consistent on distributions having a constant value of  $y$ . Then we claim the following:

**Proposition 6** *Let  $\mathcal{L}$  be the following set of distributions. Assume  $X = [0, 1]$  and that  $\mu$  is uniform on  $X$ . Let  $Y = \{-1, +1\}$ , assume that all  $f^*(x)$  are Lipschitz with constant  $\leq L$ , and that*

$$\mu\left(f^*(x) \in \{-1, 0, +1\}\right) = 0 \quad (3)$$

*Assume that  $f$  is a strictly local method and that  $f$  is reasonable. Then there exists a reasonable method  $g$  for which, for every  $P \in \mathcal{L}$ , for large enough  $n$  we have*

$$L_n(g) < L_n(f)$$

*That is,  $g$  is preferable to  $f$ .*

(Note that the assumption (3) is for convenience, and leaves us to deal with the most interesting cases.) The proof of the proposition appears in Appendix B.

Thus, any strictly local method can be improved upon due to its ignorance of far-off points. Given that support vector machines and other techniques used in machine learning are in fact local but not strictly local, there is the possibility (which we concede that we only argue towards, but do not prove) that this helps to explain their performance advantage over classical methods which are strictly local.

## 6 Classification

We will now show how our results apply to classification. First, we note that many theoretical analyses of classification methods such as support vector machines and boosting in fact work on the real-valued response of such methods, i.e., before the sign operation; see, e.g., [Zha04, BJM06]. In that sense these classification methods are treated similarly to regression estimators, and our results are of relevance to them. However, this connection is only an informal one, and therefore in this section we will show how it can be formalized.

In classification [see, e.g., DGL96] we deal with learning methods  $c(S_n, x)$  which return values in  $\{-1, +1\}$ . The loss of interest is the 0-1 loss,

$$R_{0-1}(c) = P(c(S_n, x) \neq y) = E_{S_n, (x, y)} 1\{c(S_n, x) \neq y\}$$

which is usually compared to the lowest possible loss (also known as the Bayesian loss), giving the excess loss, which is well-known to be equivalent to

$$\tilde{L}_n(c) \equiv E_{S_n, x} |c(S_n, x) - c^*(x)| \cdot |2\eta(x) - 1|$$

where  $\eta(x) = P(y = 1|x)$  and  $c^*(x) = \text{sign}(f^*(x))$ . This differs from the loss  $L_n$  studied in the main part of this work,

but as shown in [ZR07], consistency-related results such as Theorem 5 can be adapted to classification, using a method that we now briefly summarize. The idea is to note that

$$\begin{aligned}
\tilde{L}_n(c) &\equiv E_{S_n, x} |c(S_n, x) - c^*(x)| \cdot |2\eta(x) - 1| \\
&= E_{S_n, x} |c(S_n, x) - c^*(x)| \cdot |f^*(x)| \\
&= E_{S_n, x} |c(S_n, x) \cdot |f^*(x)| - f^*(x)| \\
&\equiv E_{S_n, x} |f_c^*(S_n, x) - f^*(x)| \\
&= L_n(f_c^*)
\end{aligned} \tag{4}$$

where we define  $f_c^*(S_n, x) \equiv c(S_n, x) \cdot |f^*(x)|$ . Now, a classifier  $c$  can be seen as estimating  $\text{sign}(f^*)$ . For every such  $c$  we define a learning method  $f_c$  that estimates  $f^*$ , by

$$f_c(S_n, x) = c(S_n, x) f_{|\cdot|}(S_n, x)$$

where  $f_{|\cdot|}$  is the absolute value of some pre-determined consistent method, i.e., a consistent estimator of  $|f^*|$  (that is,  $c$  estimates the sign of  $f^*$  and  $f_{|\cdot|}$  estimates the absolute value; together they estimate  $f^*$ ). It is then straightforward to show that  $c$  is consistent (as a classifier) on a set of distributions precisely when  $f_c$  is consistent (as a regression-type estimator) on that same set, since  $f_c$  is asymptotically equivalent to  $f_c^*$ , and using  $\tilde{L}_n(c) = L_n(f_c^*)$  from (4).

Regarding our result for strict locality, Proposition 6, the proof can be modified to apply to classification as follows. First, note that already  $Y = \{-1, +1\}$ , and that if we replace  $f$  with a classifier  $c$  (i.e., a function into  $\{-1, +1\}$ ) then  $g$  defined in the proof is also a classifier (in fact, the setting was chosen for its relevance to classification). Denote  $d = g$  to avoid confusion; thus, our goal is to show that  $\tilde{L}_n(d) - \tilde{L}_n(c) < 0$ . Now, as shown in (4) we have  $\tilde{L}_n(c) = L_n(f_c^*)$ , so our goal is to evaluate  $L_n(f_d^*) - L_n(f_c^*)$ . Note that, when event  $A$  occurs as defined in the proof, then instead of a response of 1 for  $f^*$  we now have a response of 1 for  $c$ , giving an overall response of  $f_c^*(S_n, x) = |f^*(x)|$  (and vice versa for a response of  $-1$ ), which leads to replacing  $|1 - f^*(x)|$  with  $|f^*(x)| - f^*(x)$  and of  $|-1 - f^*(x)|$  with  $|-f^*(x) - f^*(x)|$ . In (5) we then get

$$\left| |f^*(x) - f^*(x)| - \left| -|f^*(x)| - f^*(x) \right| \right| = -2f^*(x)$$

and  $-2f^*(x)$  happens to be the exact same result as in the original proof. All the rest of the proof can remain as before, thus proving the claim in the context of classification.

## 7 Discussion

We have argued that (1) some degree of locality is unavoidable in learning, but that (2) if this is taken to an extreme then it brings with it performance limitations. We speculate that the superior performance of modern methods over classical ones may, in part, be due to the former striking a proper balance in this matter.

Regarding the unavoidability of local learning, this is a direct result of locality being implied by consistency. In fact, in consistency we require the ability to do well on *all* distributions, which includes distributions that only differ in very

small localized ways. Thus, a consistent method must end up trusting only close-by points. The only way to avoid this issue is to dismiss consistency as a useful property. While in theory such an approach might make sense – say, if we know in advance that the true distribution belongs to some limited set – in practice many effective methods in machine learning are useful precisely because they make as few as possible assumptions on the distribution. In fact, this is the reason non-parametric methods are often more effective on real-world problems than parametric ones. Thus, generally speaking, consistency appears to be a property that we cannot easily discard. Since consistency implies a form of locality, locality is unavoidable as well.

As we have seen, the difference between consistency and the relevant form of locality, UAWL, turns out to be a fairly minor property, WCM. This means that if one of our goals is consistency then it makes sense to focus on achieving the UAWL property, since it is generally more difficult to ensure than WCM (ensuring WCM amounts to checking that two scalar values are within some reasonable bound). This may explain the historical appearance of and focus on classical methods like k-NN and kernel estimators: by defining them in an explicitly local manner, which is simple to do, the UAWL property is easily taken care of. Consequently, defining such local methods is convenient and proving their consistency relatively easy as well.

Such definitions, however, make the resulting methods not only local in the necessary sense, but also *strictly* local. As we have seen, strict locality is not necessary for consistency and in fact implies some limitations on performance. Thus, being motivated by convenient definitions and proofs may lead to deficits in practice.

On the other hand, we can start with improving real-world performance. The primary method of doing so which we intend here is maximal-margin separation, which turns out to be very effective in practice, and has an appealing geometric intuition (keeping the classes as far apart as possible). This approach is clearly not a local one, since the maximal-margin hyperplane depends on the entire training set. Furthermore, in some sense it is reasonable to expect an effective regularization technique to in fact be non-local: if, as in soft-margin support vector machines, we consider the sum of deviations across the margin (i.e., of observations on the wrong side of it), then it would be hard to do so in a local manner. That is, if we expect to allow some total amount of deviations based on some rationale, it is hard to enforce this locally; if we do work locally, then we need to apply the same approach in every area, instead of being able to accept more deviations in some areas in return for smaller deviations elsewhere as well as a larger overall margin.

Thus, techniques like maximal-margin separation are effective and desirable, but non-local in their definition. This appears problematic if we also want the property of consistency, which as we have seen requires a degree of locality. Hence, in devising learning methods we come up against a difficulty: we want our learning methods to (1) be local, so that they may be consistent, but we also want to (2) apply some performance-improving technique like maximal-margin separation, which is non-local.

We can now try to explain the success of modern ma-

chine learning methods by their combining these two properties in an effective manner: by using the ‘kernel trick’ and choosing a universal kernel [Ste02] we can get sufficiently local behavior for consistency, while at the same time we are still applying the maximal-margin principle in a global manner, thus improving performance. It is this combined approach which may be missing from classical methods.<sup>3</sup>

## A Proof of $\Leftarrow$ in Theorem 5

Denote  $S_n(x, r, a) = S_n(x, r, \{a_i\})$  where  $a_i = a$ , i.e.,  $S_n(x, r, a)$  replaces the  $y$  values of all far-off points with  $a$ .

Fix some  $T \in \mathcal{T}$  and some  $r, q \in T$ . For any  $\alpha \in \mathbb{R}$ , we have the trivial fact that

$$\begin{aligned} |f(S_n, x) - f^*(x)| &\leq \\ &|f(S_n, x) - \bar{f}^q(S_n(x, r, \alpha), x)| \\ &+ |\bar{f}^q(S_n(x, r, \alpha), x) - f^*(x)| \end{aligned}$$

Let  $A = \{\alpha_m\}$  be a countable set and let  $m_n$  be a sequence. Write

$$\begin{aligned} &|f(S_n, x) - f^*(x)| \\ &\leq \inf_{m \leq m_n} \left( |f(S_n, x) - \bar{f}^q(S_n(x, r, \alpha_m), x)| \right. \\ &\quad \left. + |\bar{f}^q(S_n(x, r, \alpha_m), x) - f^*(x)| \right) \\ &\leq \sup_{m \leq m_n} |f(S_n, x) - \bar{f}^q(S_n(x, r, \alpha_m), x)| \\ &\quad + \inf_{m \leq m_n} |\bar{f}^q(S_n(x, r, \alpha_m), x) - f^*(x)| \end{aligned}$$

and thus

$$\begin{aligned} &E_{S_n, x} |f(S_n, x) - f^*(x)| \\ &\leq E_{S_n, x} \sup_{m \leq m_n} |f(S_n, x) - \bar{f}^q(S_n(x, r, \alpha_m), x)| \\ &\quad + E_{S_n, x} \inf_{m \leq m_n} |\bar{f}^q(S_n(x, r, \alpha_m), x) - f^*(x)| \\ &\leq \sum_{m \leq m_n} E_{S_n, x} |f(S_n, x) - \bar{f}^q(S_n(x, r, \alpha_m), x)| \\ &\quad + E_{S_n, x} \inf_{m \leq m_n} |\bar{f}^q(S_n(x, r, \alpha_m), x) - f^*(x)| \end{aligned}$$

By the UAWL property, for any  $\alpha_m \in A$  we have

$$E_{S_n, x} |f(S_n, x) - \bar{f}^{Q(R_n)}(S_n(x, R_n, \alpha_m), x)| \rightarrow 0$$

for appropriate  $Q, R_n$ , since  $S_n(x, R_n, \alpha)$  can be seen as sampled from a situation where  $\bar{P}$  in the definition of UAWL has  $y$  constant and equal to  $\alpha$ . This is then true in particular for  $Q \equiv q, R_n \equiv r$ , since by keeping these values fixed they necessarily eventually become appropriate in the sense of the definition of UAWL (i.e., as constants, they eventually become larger than the sequences from the definition of UAWL – both of which tend to 0 – that we compare them with in order to check if they are appropriate). It is therefore also clear that there exists a sequence  $m_n \rightarrow \infty$  for which

$$\sum_{m \leq m_n} E_{S_n, x} |f(S_n, x) - \bar{f}^q(S_n(x, r, \alpha_m), x)| \rightarrow 0$$

<sup>3</sup>Note that an additional advantage of kernel machines is that we can easily make them non-consistent, by choosing an appropriate kernel, i.e., a non-universal one.

(by taking  $m_n \rightarrow \infty$  slowly enough, e.g., by keeping  $m_n = k$  fixed and raising it to  $k + 1$  only when the sum of the first  $k + 1$  elements will, for all  $n' \geq n$ , be smaller than  $k^{-1}$ , which must eventually occur since the sum is of elements converging to 0). For this  $m_n$  we therefore have

$$\begin{aligned} &\limsup_{n \rightarrow \infty} E_{S_n, x} |f(S_n, x) - f^*(x)| \\ &\leq \limsup_{n \rightarrow \infty} E_{S_n, x} \inf_{m \leq m_n} |\bar{f}^q(S_n(x, r, \alpha_m), x) - f^*(x)| \end{aligned}$$

We now pick  $A = \{\alpha_m\}$  to be dense in  $[-M, M]$  (recall that  $M$  is a bound on  $f^*$  and  $f$ ), and turn to analyzing the expression on the last line. Fix some  $x \in \text{supp}(P)$ , and consider the expression corresponding to  $x$  in the expected value. Then for large enough  $n$  we can find some  $m(x) \in \{1, \dots, m_n\}$  for which  $|\alpha_{m(x)} - E_{P_{x, r}}(f^*)| < \epsilon$ , for any  $\epsilon > 0$  (due to  $A$  being dense). Then

$$\begin{aligned} &E_{S_n} \inf_{m \leq m_n} |\bar{f}^q(S_n(x, r, \alpha_m), x) - f^*(x)| \\ &\leq E_{S_n} |\bar{f}^q(S_n(x, r, \alpha_{m(x)}), x) - f^*(x)| \\ &\leq E_{S_n} |E_{x' \sim P_{x, q}} [f(S_n(x, r, \alpha_{m(x)}), x') - f^*(x')]| \\ &\quad + |E_{x' \sim P_{x, q}} f^*(x') - f^*(x)| \\ &\leq E_{S_n} E_{x' \sim P_{x, q}} |f(S_n(x, r, \alpha_{m(x)}), x') - f^*(x')| \\ &\quad + E_{x' \sim P_{x, q}} |f^*(x') - f^*(x)| \end{aligned}$$

The expression on the last line converges to 0 (for almost all  $x$ ) when  $q \rightarrow 0$ , by the corollary to the following lemma:

**Lemma 7** [ [Dev81]; Lemma 1.1] *For any distribution  $P$  and measurable  $g$ , if  $E_{x \sim P} |g(x)| < \infty$  then*

$$\lim_{q \rightarrow 0} E_{x' \sim P_{x, q}} g(x') = g(x)$$

for almost all  $x$ .

**Corollary 8** *For any distribution  $P$  and measurable  $g$ , if  $E_{x \sim P} |g(x)| < \infty$  then*

$$\lim_{q \rightarrow 0} E_{x' \sim P_{x, q}} |g(x') - g(x)| = 0$$

for almost all  $x$ .

Thus, we arrive at

$$\begin{aligned} &\limsup_n E_{S_n} \inf_{m \leq m_n} |\bar{f}^q(S_n(x, r, \alpha_m), x) - f^*(x)| \\ &\leq \limsup_n E_{S_n} E_{x' \sim P_{x, q}} |f(S_n(x, r, \alpha_{m(x)}), x') - f^*(x')| \\ &\quad + \epsilon_1 \end{aligned}$$

where  $\epsilon_1 > 0$  can be made arbitrarily small by picking  $q$  small enough.

Note that we can see  $S_n(x, r, \alpha)$  as sampled from the distribution  $P_{x, r, \alpha}$ , by which we mean a distribution having the same  $\mu$  as  $P$ , equal to  $P$  on  $B_{x, r}$ , and having constant  $y$

equal to  $\alpha$  elsewhere. Then

$$\begin{aligned}
& E_{S_n} E_{x' \sim P_{x,q}} |f(S_n(x, r, \alpha_{m(x)}), x') - f^*(x')| \\
&= \frac{1}{\mu(B_{x,q})} E_{S_n, x' \sim P} \\
&\quad |f(S_n(x, r, \alpha_{m(x)}), x') - f^*(x')| 1_{\{x' \in B_{x,q}\}} \\
&\leq \frac{1}{\mu(B_{x,q})} E_{S_n, x' \sim P} |f(S_n(x, r, \alpha_{m(x)}), x') - f^*(x')| \\
&= \frac{1}{\mu(B_{x,q})} E_{S_n, x' \sim P_{x,r,\alpha_{m(x)}}} |f(S_n, x') - f^*(x')| \\
&= \frac{1}{\mu(B_{x,q})} E_{S_n, x' \sim P_{x,r,\alpha_{m(x)}}} \\
&\quad |f(S_n, x') - E_n(f) + E_n(f) - E(f^*) + \\
&\quad\quad E(f^*) - f^*(x')| \\
&\leq \frac{1}{\mu(B_{x,q})} \left[ \text{MAD}_{n, P_{x,r,\alpha_{m(x)}}}(f) + \right. \\
&\quad \left. |E_{n, P_{x,r,\alpha_{m(x)}}}(f) - E_{P_{x,r,\alpha_{m(x)}}}(f^*)| + \right. \\
&\quad \left. \text{MAD}_{P_{x,r,\alpha_{m(x)}}}(f^*) \right]
\end{aligned}$$

where the expected values  $E_n(f)$ ,  $E(f^*)$  on the equation before last are w.r.t  $P_{x,r,\alpha_{m(x)}}$  (the omission is for clarity).

Using the WCM property, we can therefore bound

$$\begin{aligned}
& \limsup_n E_{S_n} \inf_{m \leq m_n} |\bar{f}^q(S_n(x, r, \alpha_m), x) - f^*(x)| \\
&\leq \frac{1}{\mu(B_{x,q})} \left[ \text{MAD}_{P_{x,r,\alpha_{m(x)}}}(f^*) + \right. \\
&\quad \left. 2H \left( \text{MAD}_{P_{x,r,\alpha_{m(x)}}}(f^*) \right) \right] + \epsilon_1
\end{aligned}$$

We now turn to consider the MAD of  $P_{x,r,\alpha_{m(x)}}$ . Notice first that

$$\text{MAD}(P_{x,r,\alpha_{m(x)}}) \leq \text{MAD}(P_{x,r}) + \epsilon$$

because  $|\alpha_{m(x)} - E_{P_{x,r}}(f^*)| < \epsilon$ . Consider now the effect of changing  $r$ . First, by Lemma 7 we have, for almost every  $x$ ,

$$\lim_{r \rightarrow 0} E_{x' \sim P_{x,r}} f^*(x') = f^*(x)$$

so, for almost every  $x$ ,

$$\begin{aligned}
& \lim_{r \rightarrow 0} \text{MAD}_{P_{x,r}}(f^*) \\
&= \lim_{r \rightarrow 0} E_{x' \sim P_{x,r}} |f^*(x') - E_{x'' \sim P_{x,r}} f^*(x'')| \\
&\leq \lim_{r \rightarrow 0} E_{x' \sim P_{x,r}} |f^*(x') - f^*(x)| + \\
&\quad |f^*(x) - E_{x'' \sim P_{x,r}} f^*(x'')| \\
&= \lim_{r \rightarrow 0} E_{x' \sim P_{x,r}} |f^*(x') - f^*(x)| \\
&= 0
\end{aligned}$$

using Corollary 7 for the last equality, and thus

$$\limsup_{r \rightarrow 0} \text{MAD}(P_{x,r,\alpha_{m(x)}}) \leq \epsilon$$

We can pick  $q$  to make  $\epsilon_1$  arbitrarily small, and then  $r$  to make  $\text{MAD}(P_{x,r,\alpha_{m(x)}})$  arbitrarily small as well (note that

we thus counter the  $\frac{1}{\mu(B_{x,q})}$  factor), and therefore, for almost every  $x$ ,

$$\lim_{n \rightarrow \infty} E_{S_n} |f(S_n, x) - f^*(x)| = 0$$

where we also use the continuity of  $H$  at 0. This in turn implies, along with the dominated convergence theorem, that

$$\lim_{n \rightarrow \infty} L_n(f) = 0$$

thus proving that  $f$  is consistent.

## B Proof of Proposition 6

Denote  $S_n \cap B = \{(x_i, y_i) \in S_n : x_i \in B\}$ . Fix some  $P \in \mathcal{L}$  as in the statement of the proposition. Let  $R_n$  be the radii from the definition of strict locality for  $f$ . Note that, by the definition of strict locality, we can replace  $R_n$  with any  $R'_n \geq R_n$ ,  $R'_n \searrow 0$  and strict locality will still hold. WLOG we can therefore assume that  $nR_n \rightarrow \infty$ .

Define  $R(S_n, x, r)$  as the property that

$$\forall \{y_i\}, \{\tilde{y}_i\} \quad f(S_n, x) = f(S_n(x, r, \{\tilde{y}_i\}), x)$$

That is,  $R$  is the property that strict locality in fact occurs; by the definition of strict locality we know that the probability of  $R(S_n, x, R_n)$  rises to 1.

We define the following additional properties. Denote by  $A_X = A_X(x, r)$  the property that  $x \in (\sqrt{r}, 1 - \sqrt{r})$  (which makes sense for  $r \leq 1/4$ , and is indeed the case concerning us as the values replacing  $r$  will tend to 0). Denote by  $A_0 = A_0(S_n, x, r)$  the property that

$$|S_n \cap B_{x,r}| \in [nr, 3nr]$$

$$|S_n \cap (B_{x,\sqrt{r}} \setminus B_{x,r})| \in [n\sqrt{r}, 3n\sqrt{r}]$$

and that  $R(S_n, x, r)$  holds. Note that  $A_0(S_n, x, R_n)$  occurs with probability going to 1, due to the marginal distribution  $\mu$  being uniform on  $[0, 1]$  (and using Bernstein's Inequality), i.e.,  $A_0$  implies that the number of observations in the regions  $B_{x,R_n}, B_{x,\sqrt{R_n}}$  are in the ranges of values we would expect them to be, up to a constant. The additional requirement that  $R(S_n, x, R_n)$  holds does not change the probability of  $A_0(S_n, x, R_n)$  going to 1, since the probability of  $R(S_n, x, R_n)$  goes to 1.

Define also  $A_+(S_n, x, r)$  as the property where  $A_X(x, r)$ ,  $A_0(S_n, x, r)$  hold, and in addition we have

$$(x_i, y_i) \in S_n \cap B_{x,r} \quad \longrightarrow \quad y_i = -1$$

$$(x_i, y_i) \in S_n \cap (B_{x,\sqrt{r}} \setminus B_{x,r}) \quad \longrightarrow \quad y_i = +1$$

i.e., the majority of points in  $B_{x,\sqrt{r}}$  have label +1, while the minority in the smaller enclosed region  $B_{x,r}$  have label -1, and strict locality occurs. Hence if  $f$  were applied to  $S_n, x$ , its response would be -1 (due to  $f$  being reasonable), despite the numerous slightly farther-off points with label +1. Likewise define  $A_-(S_n, x, r)$  as the same property with reversed signs. Finally, let  $A(S_n, x, r)$  be the property that either  $A_+(S_n, x, r)$  or  $A_-(S_n, x, r)$  holds. Note that for small  $R_n$  we expect that the probability of  $A(S_n, x, R_n)$  be very small, i.e., it is an odd occurrence.

We define a new method  $g$  as follows:

$$g(S_n, x) = \begin{cases} f(S_n, x) & \neg A(S_n, x, R_n) \\ -f(S_n, x) & A(S_n, x, R_n) \end{cases}$$

(i.e., we return one value if the property  $A(S_n, x, R_n)$  holds, and another otherwise). That is, on ‘normal’ training sets  $g$  is the same as  $f$ ; however, on odd training sets with property  $A$ ,  $g$  guesses the opposite of  $f$ : it trusts the large number of points within radius  $(R_n, \sqrt{R_n})$  over the smaller number in radius  $(0, R_n)$ ;  $g$  also behaves the same as  $f$  for  $x$  close to the boundaries 0, 1 and only changes  $f$ 's behavior when  $g$  takes into account the points in radius  $R_n$  and ignores the rest. Note that  $g$  is strictly local, like  $f$ , albeit with larger radius. This suffices to prove the proposition and thus make the claim that strict locality has performance limitations, since it shows that we would always want to raise  $R_n$  to improve performance. In fact we can continue to raise  $R_n$  while the close-by points comprise an ‘odd’ training set in a sense similar to that mentioned above, which will lead to a non-strictly local method (since we may end up with large  $R_n$ , even  $O(1)$ , albeit with small probability).

We will now prove that  $g$  has the property described in the proposition, i.e., that it is preferable to  $f$ . Consider some fixed  $x \in (0, 1)$ , then the corresponding element for  $x$  from the loss  $L_n(g) = E_{S_n, x} |g(S_n, x) - f^*(x)|$  obeys

$$\begin{aligned} E_{S_n} |g(S_n, x) - f^*(x)| &= \\ E_{S_n} 1\{A(S_n, x, R_n)\} |g(S_n, x) - f^*(x)| &+ \\ + E_{S_n} 1\{\neg A(S_n, x, R_n)\} |g(S_n, x) - f^*(x)| \end{aligned}$$

The last expression is equal to

$$E_{S_n} 1\{\neg A(S_n, x, R_n)\} |f(S_n, x) - f^*(x)|$$

so when comparing  $L_n(g)$  to  $L_n(f)$  it cancels out. We are left with evaluating

$$l_x(g) \equiv E_{S_n} 1\{A(S_n, x, R_n)\} |g(S_n, x) - f^*(x)|$$

which we compare to

$$l_x(f) \equiv E_{S_n} 1\{A(S_n, x, R_n)\} |f(S_n, x) - f^*(x)|$$

As mentioned before, when  $A_+(S_n, x, r)$  holds then  $f$  returns  $-1$ , because  $f$  considers only the points in radius  $r$ , all of whom have label  $-1$ , and because  $f$  is reasonable. Consequently in this case  $g$  returns  $+1$ , and vice versa for  $A_-$ . To consider the difference  $L_n(g) - L_n(f)$ , which we want to prove is negative, we can then write

$$\begin{aligned} l_x(g) - l_x(f) &= E_{S_n} 1\{A(S_n, x, R_n)\} \\ &\quad (|g(S_n, x) - f^*(x)| - |f(S_n, x) - f^*(x)|) \\ &= E_{S_n} 1\{A_+(S_n, x, R_n)\} \\ &\quad (|1 - f^*(x)| - |1 - f^*(x)|) \\ &\quad + E_{S_n} 1\{A_-(S_n, x, R_n)\} \\ &\quad (|1 - f^*(x)| - |1 - f^*(x)|) \\ &= E_{S_n} |1 - f^*(x)| \\ &\quad (1\{A_+(S_n, x, R_n)\} - 1\{A_-(S_n, x, R_n)\}) \\ &\quad - E_{S_n} |1 - f^*(x)| \\ &\quad (1\{A_+(S_n, x, R_n)\} - 1\{A_-(S_n, x, R_n)\}) \\ &= E_{S_n} [1\{A_+(S_n, x, R_n)\} - 1\{A_-(S_n, x, R_n)\}] \\ &\quad (|1 - f^*(x)| - |1 - f^*(x)|) \\ &= -2f^*(x) \left[ P_{S_n}(A_+(S_n, x, R_n)) \right. \\ &\quad \left. - P_{S_n}(A_-(S_n, x, R_n)) \right] \end{aligned} \quad (5)$$

Our goal is to show that the expected value over  $x$  of this last expression is negative. For convenience we will write  $A(S_n, x, R_n) \equiv A, A_+(S_n, x, R_n) \equiv A_+$  and likewise for  $A_-$ . Note that, for any  $x$  fulfilling  $1\{A_X\}$ , we have that the probability of  $A_0$  converges to 1 as mentioned before. Thus, we are left to consider the sign of

$$-E_x 1\{A_X\} f^*(x) \left[ P_{S_n}(A_+|A_0) - P_{S_n}(A_-|A_0) \right]$$

Denote

$$F_n(K, k) = P_{S_n}(A_+|A_0, K, k) - P_{S_n}(A_-|A_0, K, k)$$

where  $K$  is the number of observations in radius  $(R_n, \sqrt{R_n})$  and  $k$  is the number in  $(0, R_n)$ , both around  $x$ ; hence the relevant set of values for  $K$  is  $[n\sqrt{R_n}, 3n\sqrt{R_n}]$ , and for  $k$  is  $[nR_n, 3nR_n]$ . Note that  $F_n$  depends on  $x$ , but we omit it for clarity for reasons which will soon be obvious.

Let  $p_n(K, k)$  be the probability of the values  $K, k$  for any  $x$  fulfilling  $1\{A_X\}$ . Then

$$\begin{aligned} -E_x 1\{A_X\} f^*(x) \left[ P_{S_n}(A_+|A_0) - P_{S_n}(A_-|A_0) \right] \\ = - \sum_{K, k} p_n(K, k) E_x 1\{A_X\} f^*(x) F_n(K, k) \end{aligned}$$

where the sum is over the set of relevant values for  $K, k$  as mentioned before.

We will now show that large enough  $n$  we have, for all relevant  $K, k$ , that  $E_x 1\{A_X\} f^*(x) F_n(K, k) > 0$ ; note that this is enough to finish the proof.

Consider some fixed  $K, k$  and some fixed  $x$  fulfilling  $1\{A_X\}$ . Assume WLOG that  $0 < f^*(x) < 1$  (due to the symmetry in the problem, the other case arrives at the same result). Using the Lipschitz property of  $f^*$ , and since  $P(y = 1|x) = \frac{1}{2}(1 + f^*(x))$ , we can bound the conditional probabilities on  $A_0, K, k$  (and assuming  $x$  fulfills  $1\{A_X\}$ ) in the following manner (note that the conditional probabilities only depend on the behavior of  $y_i$  values):

$$\begin{aligned} P_{S_n}(A_+|A_0, K, k) &\geq \\ &\frac{(1 + f^*(x) - \sqrt{R_n}L)^K (1 - f^*(x) - R_nL)^k}{2^{K+k}} \\ P_{S_n}(A_-|A_0, K, k) &\leq \\ &\frac{(1 + f^*(x) + R_nL)^k (1 - f^*(x) + \sqrt{R_n}L)^K}{2^{K+k}} \end{aligned}$$

Note that these bounds depend only on  $f^*(x)$  and not  $x$  itself. Note also that in particular

$$F_n(K, k) \geq - \frac{(2 + R_nL)^k (1 + \sqrt{R_n}L)^K}{2^{K+k}} \quad (6)$$

Now, consider  $2^{K+k} E_x 1\{A_X\} f^*(x) F_n(K, k)$ . We claim that according to the bounds above, for every  $x$  fulfilling  $1\{A_X\}$  we have

$$\inf_{K, k} 2^{K+k} F_n(K, k) \rightarrow \infty \quad (7)$$

where the infimum is taken over all relevant  $K, k$ . To see this, recall the assumption that  $0 < f^*(x) < 1$ , and consider

the behavior of the bound for  $2^{K+k}P_{S_n}(A_+|A_0, K, k)$ : by taking the logarithm we get

$$K \log(1 + f^*(x) - \sqrt{R_n L}) + k \log(1 - f^*(x) - R_n L)$$

which clearly converges to infinity, even when taking the infimum over  $K, k$ , since  $R_n \rightarrow 0$  and all relevant  $K$  converge to infinity faster than all  $k$  (recall the ranges of values of  $K, k$ , and that  $nR_n \rightarrow \infty$ , so they all converge to infinity). Similarly we can see that  $2^{K+k}P_{S_n}(A_-|A_0, K, k)$  converges to 0, thus showing (7).

In a similar manner we can see that, for every  $x$  fulfilling  $1\{A_X\}$ , for large enough  $n$  we have

$$\inf_{K,k} 2^{K+k} F_n(K, k) > \sup_{K,k} (2 + R_n L)^k \left(1 + \sqrt{R_n L}\right)^K \quad (8)$$

Note that the RHS is related to the lower bound of  $F_n(K, k)$  as shown in (6).

Taken together, the facts just stated imply that the measure of points  $x$  fulfilling both (7) and (8) converges to 1 (formally, using the dominated convergence theorem on the identifier function on that set). Due to (6), it is clear that the values of the other points cannot overcome them from causing the overall integral to be positive, and we conclude that  $E_x 1\{A_X\} f^*(x) F_n(K, k) > 0$  for large enough  $n$  in a manner that does not depend upon  $K, k$  (since we have used the sup, inf over relevant  $K, k$  values), proving the result.

## References

- [AMS97] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.
- [BDR06] Y. Bengio, O. Delalleau, and N. Le Roux. The curse of highly variable functions for local kernel machines. In *Advances in Neural Information Processing Systems 18*, pages 107–114. MIT Press, Cambridge, MA, 2006.
- [BJM06] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.
- [BV92] L. Bottou and V. N. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.
- [CL95] W. Cleveland and C. Loader. Smoothing by local regression: Principles and methods. Technical report, AT&T Bell Laboratories, Murray Hill, NY., 1995. Available at <http://citeseer.ist.psu.edu/194800.html>.
- [Dev81] L. Devroye. On the almost everywhere convergence of nonparametric regression function estimates. *Annals of Statistics*, 9:1310–1319, 1981.
- [DGKL94] L. Devroye, L. Györfi, A. Krzyżak, and G. Lugosi. On the strong universal consistency of nearest neighbor regression function estimates. *Annals of Statistics*, 22:1371–1385, 1994.
- [DGL96] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, NY, 1996.
- [FS99] Y. Freund and R. Schapire. A short introduction to boosting. *J. Japan. Soc. for Artif. Intel.*, 14(5):771–780, 1999.
- [GKP84] W. Greblicki, A. Krzyżak, and M. Pawlak. Distribution-free pointwise consistency of kernel regression estimate. *Annals of Statistics*, 12(4):1570–1575, 1984.
- [Joa98] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, volume 1398, pages 137–142. Springer Verlag, Heidelberg, DE, 1998.
- [SFBL97] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. In *Proc. 14th International Conference on Machine Learning*, pages 322–330. Morgan Kaufmann, 1997.
- [SGV98] G. Saunders, A. Gammernan, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proc. 15th International Conf. on Machine Learning*, pages 515–521. Morgan Kaufmann, San Francisco, CA, 1998.
- [STC00] J. Shawe-Taylor and N. Cristianini. *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, 2000.
- [Ste02] I. Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, 18:768–791, 2002.
- [Sto80] C. J. Stone. Optimal rates of convergence for nonparametric estimators. *Annals of Statistics*, 8:1348–1360, 1980.
- [Sto82] C. J. Stone. Optimal global rates of convergence for nonparametric regression. *Annals of Statistics*, 10:1040–1053, 1982.
- [Vap98] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, NY, 1998.
- [VB93] V. N. Vapnik and L. Bottou. Local algorithms for pattern recognition and dependencies estimation. *Neural Computation*, 5(6):893–909, 1993.
- [Zha04] T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–134, 2004.
- [ZR07] A. Zakai and Y. Ritov. Local behavior of consistent learning methods, 2007. Submitted for Publication.

---

# Learning coordinate gradients with multi-task kernels

---

Yiming Ying and Colin Campbell\*

Department of Engineering Mathematics, University of Bristol  
Queens Building, University Walk, Bristol, BS8 1TR, UK.  
{enxyy, C.Campbell}@bristol.ac.uk

## Abstract

Coordinate gradient learning is motivated by the problem of variable selection and determining variable covariation. In this paper we propose a novel unifying framework for coordinate gradient learning (MGL) from the perspective of multi-task learning. Our approach relies on multi-task kernels to simulate the structure of gradient learning. This has several appealing properties. Firstly, it allows us to introduce a novel algorithm which appropriately captures the inherent structure of coordinate gradient learning. Secondly, this approach gives rise to a clear algorithmic process: a computational optimization algorithm which is memory and time efficient. Finally, a statistical error analysis ensures convergence of the estimated function and its gradient to the true function and true gradient. We report some preliminary experiments to validate MGL for variable selection as well as determining variable covariation.

## 1 Introduction

Let  $X \subseteq \mathbb{R}^d$  be compact,  $Y \subseteq \mathbb{R}$ ,  $Z = X \times Y$ , and  $\mathbb{N}_n = \{1, 2, \dots, n\}$  for any  $n \in \mathbb{N}$ . A common theme in machine learning is to learn a target function  $f_* : X \rightarrow Y$  from a finite set of input/output samples  $\mathbf{z} = \{(x_i, y_i) : i \in \mathbb{N}_m\} \subseteq Z$ . However, in many applications, we not only wish to learn the target function, but also want to find which variables are salient and how these variables interact with each other. This problem has practical motivations: to facilitate data visualization and dimensionality reduction, for example. Such a motivation is important when there are many redundant variables and we wish to find the salient features among these. These problems can occur in many contexts. For example, with gene expression array datasets, the vast majority of features may be redundant to a classification task and we need to find a small set of genuinely distinguishing features. These motivations have driven the design of various statistical and machine learning models [8, 11, 21, 22] for variable (feature) selection.

Here, we build on previous contributions [15, 16, 17] by addressing coordinate gradient learning and its use for variable

selection and covariation learning, the interaction between variables. Specifically, for any  $x \in X$ , we denote  $x$  by  $(x^1, x^2, \dots, x^d)$ . The target is to learn the gradient of  $f_*$  (if it exists) denoted by a vector-valued function  $\nabla f_*(x) = (\frac{\partial f_*}{\partial x^1}, \dots, \frac{\partial f_*}{\partial x^d})$ . The intuition behind gradient learning for variable selection and coordinate covariation is the following. The inner product between components of  $\nabla f_*$  indicates the interaction between coordinate variables. Specific norms of  $\frac{\partial f_*}{\partial x^p}$  can indicate the salience of the  $p$ -th variable: the smaller the norm is, the less important this variable will be.

In this paper we propose a novel unifying formulation of coordinate gradient learning from the perspective of multi-task learning. Learning multiple tasks together has been extensively studied both theoretically and practically in several papers [1, 2, 5, 7, 13, 14]. One way to frame this problem is to learn a vector-valued function where each of its components is a real-valued function and corresponds to a particular task. A key objective in this formulation is to capture an appropriate structure among tasks so that common information is shared across tasks. Here we follow this methodology and employ a vector-valued function  $\vec{f} = (f_1, \vec{f}_2) = (f_1, f_2, \dots, f_{d+1})$ , where  $f_1$  is used to simulate  $f_*$ , and  $\vec{f}_2$  is used to simulate its gradient  $\nabla f_*$ . We assume that  $\vec{f}$  comes from a vector-valued reproducing kernel Hilbert space (RKHS) associated with multi-task (matrix-valued) kernels, see [14]. The rich structure of RKHS space reflects the latent structure of multi-task gradient learning, i.e. the pooling of information across components (tasks) of  $\nabla f_*$  using multi-task kernels.

The paper is organized as follows. In Section 2, we first review the definition of multi-task kernels and vector-valued RKHS. Then, we propose a unifying formulation of coordinate gradient learning from the perspective of multi-task learning which is referred to as multi-task gradient learning (MGL). The choices of multi-task kernels motivate different learning models [11, 15, 16, 17]. This allows us to introduce a novel choice of multi-task kernel which reveals the inherent structure of gradient learning. Kernel methods [19, 20] usually enjoy the representer theorem which paves the way for designing efficient optimization algorithms. In Section 3 we explore a representer theorem for MGL algorithms. Subsequently, in Section 4 we discuss computational optimization approaches for MGL algorithms, mainly focusing on least square loss and the SVM counterpart for gradient learning.

\*We acknowledge support from EPSRC grant EP/E027296/1.

A statistical error analysis in Section 5 ensures the convergence of the estimated function and its gradient to the true function and true gradient. Finally, in Section 6 preliminary numerical experiments are reported to validate our proposed approach.

### 1.1 Related work

A number of machine learning and statistical models have been proposed for variable (feature) selection. Least absolute shrinkage and selection operator (LASSO) [21] and basis pursuit denoising [8] suggest use of  $\ell^1$  regularization to remove redundant features. Weston *et al* [22] introduced a method for selecting features by minimizing bounds on the leave-one-out error.

Guyon *et al* [11] proposed recursive feature elimination (RFE) which used a linear kernel SVM: variables with least influence on the weights  $\frac{1}{2}\|w\|^2$  are considered least important. Although these algorithms are promising, there remain unresolved issues. For example, they do not indicate variable covariation and the extension of these algorithms to the non-linear case was marginally discussed. Our method outlined here covers variable covariation and nonlinear feature selection. As such, in Section 2, we show that RFE-SVM is a special case of our multi-task formulation.

Motivated by the Taylor expansion of a function at samples  $\{x_i : i \in \mathbb{N}_m\}$ , Mukherjee *et al* [15, 16, 17] proposed an algorithm for learning the gradient function. They used the norm of its components for variable (feature) selection and spectral decomposition of the covariance of the learned gradient function for dimension reduction [16]. Specifically, let  $\mathcal{H}_G$  be a scalar RKHS (see e.g. [3]) and use  $f_1 \in \mathcal{H}_G$  to simulate  $f_*$ . For any  $p \in \mathbb{N}_d$ , a function  $f_{p+1} \in \mathcal{H}_G$  is used to learn  $\partial f_*/\partial x^p$ . The results presented by Mukherjee *et al* are quite promising both theoretically and practically, but there is no pooling information shared across the components of the gradient. This may lead to less accurate approximation to the true gradient. We will address all these issues in our unifying framework.

## 2 Multi-task kernels and learning gradients

In this section we formulate the gradient learning problem from the perspective of multi-task learning. Specifically, we employ a vector-valued RKHS to simulate the target function and its gradient. The abundant structure of vector-valued RKHS enables us to couple information across components of the gradient in terms of multi-task kernels.

### 2.1 Multi-task model for gradient learning

We begin with a review of the definition of multi-task kernels and introduce vector-valued RKHS (see [14] and the reference therein). Throughout this paper, we use the notation  $\langle \cdot, \cdot \rangle$  and  $\|\cdot\|$  to denote the standard Euclidean inner product and norm respectively.

**Definition 1** *We say that a function  $\mathcal{K} : X \times X \rightarrow \mathbb{R}^{d+1} \times \mathbb{R}^{d+1}$  is a multi-task (matrix-valued) kernel on  $X$  if, for any  $x, t \in X$ ,  $\mathcal{K}(x, t)^T = \mathcal{K}(t, x)$ , and it is positive semi-definite, i.e., for any  $m \in \mathbb{N}$ ,  $\{x_j \in X : j \in \mathbb{N}_m\}$  and  $\{y_j \in \mathbb{R}^{d+1} :$*

$j \in \mathbb{N}_m\}$  there holds

$$\sum_{i,j \in \mathbb{N}_m} \langle y_i, \mathcal{K}(x_i, x_j) y_j \rangle \geq 0. \quad (1)$$

In the spirit of Moore-Aronszajn's theorem, there exists a one-to-one correspondence between the multi-task kernel  $\mathcal{K}$  with property (1) and a vector-valued RKHS of functions  $\vec{f} : X \rightarrow \mathbb{R}^{d+1}$  with norm  $\langle \cdot, \cdot \rangle_{\mathcal{K}}$  denoted by  $\mathcal{H}_{\mathcal{K}}$ , see e.g. [14]. Moreover, for any  $x \in X$ ,  $y \in \mathbb{R}^{d+1}$  and  $\vec{f} \in \mathcal{H}_{\mathcal{K}}$ , we have the reproducing property

$$\langle \vec{f}(x), y \rangle = \langle \vec{f}, \mathcal{K}_x y \rangle_{\mathcal{K}} \quad (2)$$

where  $\mathcal{K}_x y : X \rightarrow \mathbb{R}^{d+1}$  is defined, for any  $t \in X$ , by  $\mathcal{K}_x y(t) := \mathcal{K}(t, x) y$ .

In the following we describe our multi-task kernel-based framework for gradient learning. Following Mukherjee *et al* [15, 17], the derivation of gradient learning can be motivated by the Taylor expansion of  $f_*$ :  $f_*(x_i) \approx f_*(x_j) + \nabla f_*(x_j)(x_i - x_j)^T$ . Since we wish to learn  $f_*$  with  $f_1$  and  $\nabla f_*$  with  $\vec{f}_2$ , replacing  $f_*(x_i)$  by  $y_i$ , the error<sup>1</sup>

$$y_i \approx f_1(x_j) + \vec{f}_2(x_j)(x_i - x_j)^T$$

is expected to be small whenever  $x_i$  is close to  $x_j$ . To enforce the constraint that  $x_i$  is close to  $x_j$ , we introduce a weight function produced by a Gaussian with deviation  $s$  defined by

$w_{ij} = \frac{1}{s^{d+2}} e^{-\frac{\|x_i - x_j\|^2}{2s^2}}$ . This implies that  $w_{ij} \approx 0$  if  $x_i$  is far away from  $x_j$ .

We now propose the following multi-task formulation for gradient learning (MGL):

$$\begin{aligned} \vec{f}_z &= \arg \min_{\vec{f} \in \mathcal{H}_{\mathcal{K}}} \left\{ \frac{1}{m^2} \sum_{i,j} w_{ij} L(y_i, \right. \\ &\left. f_1(x_j) + \vec{f}_2(x_j)(x_i - x_j)^T) + \lambda \|\vec{f}\|_{\mathcal{K}}^2 \right\}. \end{aligned} \quad (3)$$

where  $L : \mathbb{R} \times \mathbb{R} \rightarrow [0, \infty)$  is a prescribed loss function and  $\lambda$  is usually called the *regularization parameter*. The minimum is taken over a vector-valued RKHS with multi-task kernel  $\mathcal{K}$ . The first component  $f_{1,z}$  of the minimizer  $\vec{f}_z$  of the above algorithm is used to simulate the target function and the other components  $\vec{f}_{2z} := (f_{2,z}, \dots, f_{d+1,z})$  to learn its gradient function. In Section 6, we will discuss how to use the solution  $\vec{f}_z$  for variable selection as well as covariation measurement.

Different choice of loss functions yield different gradient learning algorithms. For instance, if the loss function  $L(y, t) = (y - t)^2$  then algorithm (3) leads to the least-square multi-task gradient learning (LSMGL):

$$\begin{aligned} \arg \min_{\vec{f} \in \mathcal{H}_{\mathcal{K}}} &\left\{ \frac{1}{m^2} \sum_{i,j \in \mathbb{N}_m} w_{ij} [y_i - f_1(x_j) \right. \\ &\left. - \vec{f}_2(x_j)(x_i - x_j)^T]^2 + \lambda \|\vec{f}\|_{\mathcal{K}}^2 \right\}. \end{aligned} \quad (4)$$

In classification, the choice of loss function  $L(y, t) = (1 - yt)_+$  in algorithm (3) yields the support vector machine for multi-task gradient learning (SVMML):

<sup>1</sup>Our form of Taylor expansion is slightly different from that used in [15, 17]. However, the essential idea is the same.

$$\arg \min_{\substack{\vec{f} \in \mathcal{H}_{\mathcal{K}} \\ b \in \mathbb{R}}} \left\{ \frac{1}{m^2} \sum_{i,j \in \mathbb{N}_m} w_{ij} [1 - y_i (f_1(x_j) + b + \vec{f}_2(x_j)(x_i - x_j)^T)]_+ + \lambda \|\vec{f}\|_{\mathcal{K}}^2 \right\}. \quad (5)$$

Here,  $f_1(x) + b$  is used to learn the target function and  $\vec{f}_2$ , simulating the gradient of the target function. Hence  $b$  plays the same role of offset as in the standard SVM formulation. In this case, at each point  $x_i$  the error between the output  $y_i$  and  $f(x_i)$  is now replaced by the error between  $y_i$  and the first order Taylor expansion of  $f(x_i)$  at  $x_j$ , i.e.,  $f_1(x_j) + \vec{f}_2(x_j)(x_i - x_j)^T$ .

## 2.2 Choice of multi-task kernels

We note that if  $\mathcal{K}$  is a diagonal matrix-valued kernel, then each component of a vector-valued function in the associated RKHS of  $\mathcal{K}$  can be represented, independently of the other components, as a function in the RKHS of a scalar kernel. Consequently, for a scalar kernel  $G$  if we choose the multi-task kernel  $\mathcal{K}$  given, for any  $x, t \in X$ , by  $\mathcal{K}(x, t) = G(x, t)I_{d+1}$  then the MGL algorithm (3) is reduced to the gradient learning algorithm proposed in [15, 16, 17] using  $(d+1)$ -folds of scalar RKHS. There, under some conditions on the underlying distribution  $\rho$ , it has been proven that  $f_{1,z} \rightarrow f_*$  and  $\vec{f}_{2z} \rightarrow \nabla f_*$  when the number of samples tends to infinity. Although their results are promising both theoretically and practically, a more inherent structure would be  $\vec{f}_{2z} = \nabla f_{1,z}$ . In our MGL framework (3), we can recover this structure by choosing the multi-task kernel appropriately.

Our alternative choice of multi-task kernel is stimulated by the Hessian of Gaussian kernel proposed in [7]. For any scalar kernel  $G$  and any  $x, t \in X$ , we introduce the function

$$\mathcal{K}(x, t) = \begin{pmatrix} G(x, t), & (\nabla_t G(x, t))^T \\ \nabla_x G(x, t), & \nabla_x^2 G(x, t) \end{pmatrix} \quad (6)$$

which we will show to be a multi-task kernel. To see this, let  $\ell^2$  be the Hilbert space with norm  $\|w\|_{\ell^2}^2 = \sum_{j=1}^{\infty} w_j^2$ . Suppose that  $G$  has a feature representation, i.e.,  $G(x, t) = \langle \phi(x), \phi(t) \rangle_{\ell^2}$  and, for any  $f \in \mathcal{H}_G$ , there exists a vector  $w \in \ell^2$  such that  $f(x) = \langle w, \phi(x) \rangle_{\ell^2}$  and

$$\|f\|_G = \|w\|_{\ell^2}.$$

Indeed, if the input space  $X$  is compact and  $G : X \times X \rightarrow \mathbb{R}$  is a Mercer kernel, i.e., it is continuous, symmetric and positive semi-definite, then, according to Mercer theorem,  $G$  always has the above feature representation (see e.g. [9]). Now we have the following proposition about  $\mathcal{K}$  defined by equation (6). Let  $\tilde{e}_p$  be the  $p$ -th coordinate basis in  $\mathbb{R}^{d+1}$ .

**Theorem 2** *For any smooth scalar Mercer kernel  $G$ , define function  $\mathcal{K}$  by equation (6). Then,  $\mathcal{K}$  is a multi-task kernel and, for any  $\vec{f} = (f_1, \vec{f}_2) \in \mathcal{H}_{\mathcal{K}}$  there holds*

$$\vec{f}_2 = \nabla f_1. \quad (7)$$

**Proof:** Since  $G$  is a scalar kernel, for any  $x, t \in X$  we have that  $G(x, t) = G(t, x)$ . Therefore,  $\mathcal{K}(x, t)^T = \mathcal{K}(t, x)$ . Moreover,  $G$  is assumed to be a Mercer kernel which implies that it has a feature representation  $G(x, t) = \langle \phi(x), \phi(t) \rangle_{\ell^2}$ . Consequently,  $\nabla_t G(x, t) = \langle \phi(x), \nabla \phi(t) \rangle_{\ell^2}$  and  $\nabla_x G(x, t) = \langle \nabla \phi(x), \phi(t) \rangle_{\ell^2}$ , and  $\nabla_x^2 G(x, t) = \langle \nabla \phi(x), \nabla \phi(t) \rangle_{\ell^2}$ . Then, we introduce, for any  $w \in \ell^2, x \in X, \mathbf{y} \in \mathbb{R}^{d+1}$ , the feature map  $\Phi(x) : \ell^2 \rightarrow \mathbb{R}^{d+1}$  defined by

$$\Phi(x)w := (\langle \phi(x), w \rangle_{\ell^2}, \langle \partial_1 \phi(x), w \rangle_{\ell^2}, \dots, \langle \partial_d \phi(x), w \rangle_{\ell^2})^T.$$

Its adjoint map  $\Phi^*$  is given, for any  $t \in X$  and  $\mathbf{y} \in \mathbb{R}^{d+1}$ , by  $\Phi^*(t)\mathbf{y} := \phi(x)\mathbf{y}^1 + \sum_{p \in \mathbb{N}_d} \partial_p \phi(x)\mathbf{y}^{p+1}$ . Hence,  $\mathcal{K}(x, t)\mathbf{y} = \Phi(x)\Phi^*(t)\mathbf{y}$ . Consequently, for any  $m \in \mathbb{N}$ , any  $i, j \in \mathbb{N}_m$  and  $\mathbf{y}_i, \mathbf{y}_j \in \mathbb{R}^{d+1}$ , it follows  $\sum_{i,j \in \mathbb{N}_m} \langle \mathbf{y}_i, \mathcal{K}(x_i, x_j)\mathbf{y}_j \rangle = \|\sum_{i \in \mathbb{N}_m} \Phi^*(x_i)\mathbf{y}_i\|_{\ell^2}^2$  is nonnegative which tells us that  $\mathcal{K}$  is a multi-task kernel.

We turn to the second assertion. When  $\vec{f}$  is in the form of a finite combination of kernel section  $\{\mathcal{K}_{x,\mathbf{y}} : \mathbf{y} \in \mathbb{R}^{d+1}, x \in X\}$ , the second assertion follows directly from the definition of  $\mathcal{K}$ . For the general case, we use the fact that the vector-valued RKHS is the closure of the span of kernel sections, see [14]. To this end, assume that there exists a sequence  $\{\vec{f}_j = (f_1^j, f_2^j, \dots, f_{d+1}^j)\}$  of finite combination of kernel sections such that  $\vec{f}_j \rightarrow \vec{f} \in \mathcal{H}_{\mathcal{K}}$  w.r.t. the RKHS norm. Hence, by the reproducing property (2), for any  $x \in X$  and  $p \in \mathbb{N}_d$ ,  $|f_{p+1}^j(x) - f_{p+1}(x)| = |\langle \tilde{e}_{p+1}, \vec{f}_j(x) - \vec{f}(x) \rangle| = |\langle \vec{f}_j - \vec{f}, \mathcal{K}_x \tilde{e}_{p+1} \rangle_{\mathcal{K}}| \leq \|\vec{f}_j - \vec{f}\|_{\mathcal{K}} \sqrt{\tilde{e}_{p+1}^T \mathcal{K}(x, x) \tilde{e}_{p+1}}$  which tends to zeros as  $j$  tends to infinity. Consequently, it follows, for any  $x \in X$ ,

$$f_{p+1}^j(x) \rightarrow f_{p+1}(x), \quad \text{as } j \rightarrow \infty. \quad (8)$$

Let  $\delta_p \in \mathbb{R}^d$  be a vector with its  $p$ -th component  $\delta > 0$  and others equal zero. Applying the reproducing property (2) yields that

$$\begin{aligned} & \left| \frac{[f_1^j(x+\delta_p) - f_1^j(x)] - [f_1(x+\delta_p) - f_1(x)]}{\delta} \right| \\ &= \left| \frac{1}{\delta} \langle \vec{f}_j - \vec{f}, \mathcal{K}_{x+\delta_p} \tilde{e}_1 - \mathcal{K}_x \tilde{e}_1 \rangle_{\mathcal{K}} \right| \\ &\leq \|\vec{f}_j - \vec{f}\|_{\mathcal{K}} \langle \tilde{e}_1, \frac{1}{\delta^2} [\mathcal{K}(x+\delta_p, x+\delta_p) + \mathcal{K}(x, x) - \mathcal{K}(x, x+\delta_p) - \mathcal{K}(x+\delta_p, x)] \tilde{e}_1 \rangle^{\frac{1}{2}} \\ &= \|\vec{f}_j - \vec{f}\|_{\mathcal{K}} \left( \frac{1}{\delta^2} [G(x+\delta_p, x+\delta_p) + G(x, x) - G(x, x+\delta_p) - G(x+\delta_p, x)] \right)^{\frac{1}{2}}. \end{aligned}$$

Since  $G$  is smooth and  $X$  is compact there exists an absolute constant  $\tilde{c} > 0$  such that, for any  $\delta > 0$ , the above equation is furthermore bounded by

$$\left| \frac{[f_1^j(x+\delta_p) - f_1^j(x)] - [f_1(x+\delta_p) - f_1(x)]}{\delta} \right| \leq \tilde{c} \|\vec{f}_j - \vec{f}\|_{\mathcal{K}}.$$

Consequently, letting  $\delta \rightarrow 0$  in the above equation it follows  $|\partial_p f_1^j(x) - \partial_p f_1(x)| \rightarrow 0$  as  $j$  tends to infinity. Combining this with equation (8) and the fact that  $f_{p+1}^j(x) = \partial_p f_1^j(x)$  implies that  $\partial_p f_1(x) = f_{p+1}(x)$  which completes the theorem.  $\blacksquare$

The scalar kernel  $G$  plays the role of a *hyper-parameter* to produce the multi-task kernel  $\mathcal{K}$  given by equation (6). By the above theorem, if we choose  $\mathcal{K}$  to be defined by equation (6) then any solution  $f_{\mathbf{z}} = (f_{1,\mathbf{z}}, \vec{f}_{2\mathbf{z}})$  of algorithm (3) enjoys the structure  $\vec{f}_{2\mathbf{z}} = \nabla f_{1,\mathbf{z}}$ .

Further specifying the kernel  $G$  in the definition (6) of multi-task kernel  $\mathcal{K}$ , we can recover the RFE feature ranking algorithm for a linear SVM [11]. To see this, let  $G$  be a linear kernel. In the next section, we will see that, for any solution  $\vec{f}_{\mathbf{z}}$  of MGL algorithm (3), there exists  $\{c_{j,\mathbf{z}} \in \mathbb{R}^{d+1} : j \in \mathbb{N}_m\}$  such that  $\vec{f}_{\mathbf{z}} = \sum_{j \in \mathbb{N}_m} \mathcal{K}_{x_j} c_{j,\mathbf{z}}$ . Since  $G$  is linear, combining this with Theorem 2 we know that  $f_{1,\mathbf{z}}(x) = W_{\mathbf{z}}^T x$  with  $W_{\mathbf{z}} = \sum_j (x_j^T, 1) c_{j,\mathbf{z}} \in \mathbb{R}$  and  $\vec{f}_{2\mathbf{z}} = \nabla f_{1,\mathbf{z}} = W_{\mathbf{z}}^T$ . Consequently, in the case we have that

$$f_{1,\mathbf{z}}(x_j) + \vec{f}_{2\mathbf{z}}(x_j)(x_i - x_j) = W_{\mathbf{z}}^T x_i = f_{1,\mathbf{z}}(x_i).$$

Moreover, by the reproducing property (2) we can check that

$$\|\vec{f}_{\mathbf{z}}\|_{\mathcal{K}}^2 = \|f_{1,\mathbf{z}}\|_G^2 = \|W_{\mathbf{z}}\|^2.$$

Putting the above equations together, in this special case we know that the SVMMLG algorithm (5) is reduced, with the choice of  $w_{ij} = 1$ , to the classical learning algorithm:

$$\min_{W \in \mathbb{R}^d} \left\{ \frac{1}{m} \sum_{i \in \mathbb{N}_m} (1 - y_i(W^T x_i + b))_+ + \lambda \|W\|^2 \right\}.$$

Hence, our formulation of gradient learning (3) can be regarded as a generalization of RFE-SVM [11] to the nonlinear case.

In the subsequent sections we discuss a general representation theorem and computational optimization problems motivated by MGL algorithms.

### 3 Representer theorem

In this section we investigate the representer theorem for the MGL algorithm (3). This forms a foundation for the derivation of a computationally efficient algorithm for MGL in Section 4.

Recall that  $\tilde{e}_p$  is the  $p$ -th coordinate basis in  $\mathbb{R}^{d+1}$  and, for any  $x \in \mathbb{R}^d$ , denote the vector  $\tilde{x}^T$  by  $(0, x^T)$ . By the reproducing property (2), we have that  $f_1(x_j) = \langle \vec{f}(x_j), \tilde{e}_1 \rangle = \langle \vec{f}, \mathcal{K}_{x_j} \tilde{e}_1 \rangle_{\mathcal{K}}$  and likewise,  $\vec{f}_2(x_j)(x_i - x_j)^T = \langle \vec{f}(x_j), \tilde{x}_i - \tilde{x}_j \rangle = \langle \vec{f}, \mathcal{K}_{x_j}(\tilde{x}_i - \tilde{x}_j) \rangle_{\mathcal{K}}$ . Then, the algorithm (3) can be rewritten by

$$\arg \min_{\vec{f} \in \mathcal{H}_{\mathcal{K}}} \left\{ \frac{1}{m^2} \sum_{i,j \in \mathbb{N}_m} w_{ij} L(y_i, \langle \vec{f}, \mathcal{K}_{x_j}(\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j) \rangle_{\mathcal{K}}) + \lambda \|\vec{f}\|_{\mathcal{K}}^2 \right\}. \quad (9)$$

In analogy with standard kernel methods [19, 20], we have the following representer theorem for MGL by using the properties of multi-task kernels.

**Theorem 3** *For any multi-task kernel  $\mathcal{K}$ , consider the gradient learning algorithm (3). Then, there exists representer coefficients  $\{c_{j,\mathbf{z}} \in \mathbb{R}^{d+1} : j \in \mathbb{N}_m\}$  such that*

$$\vec{f}_{\mathbf{z}} = \sum_{j \in \mathbb{N}_m} \mathcal{K}_{x_j} c_{j,\mathbf{z}}$$

and, for every  $j \in \mathbb{N}_m$ , the representer coefficient  $c_{j,\mathbf{z}} \in \text{span}\{\tilde{e}_1, \tilde{x}_i : i \in \mathbb{N}_m\}$ .

**Proof:** We can write any minimizer  $\vec{f}_{\mathbf{z}} \in \mathcal{H}_{\mathcal{K}}$  as  $\vec{f}_{\mathbf{z}} = \vec{f}_{\parallel} + \vec{f}_{\perp}$  where  $\vec{f}_{\parallel}$  is in the span  $\{\mathcal{K}_{x_j} \tilde{e}_1, \mathcal{K}_{x_j} \tilde{x}_i, i, j \in \mathbb{N}_m\}$  and  $\vec{f}_{\perp}$  is perpendicular to this span space. By the reproducing property (2), we have that  $\langle \vec{f}(x_j), \tilde{e}_1 + \tilde{x}_i - \tilde{x}_j \rangle = \langle \vec{f}, \mathcal{K}_{x_j}(\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j) \rangle_{\mathcal{K}} = \langle \vec{f}_{\parallel}, \mathcal{K}_{x_j}(\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j) \rangle_{\mathcal{K}}$ . Hence,  $\vec{f}_{\perp}$  makes no contribution to the loss function in the MGL algorithm (9) (i.e. algorithm (3)). However, the norm  $\|\vec{f}\|_{\mathcal{K}}^2 = \|\vec{f}_{\parallel}\|_{\mathcal{K}}^2 + \|\vec{f}_{\perp}\|_{\mathcal{K}}^2 > \|f_{\parallel}\|_{\mathcal{K}}^2$  unless  $f_{\perp} = 0$ . This implies, any solution  $\vec{f}_{\mathbf{z}}$  belongs to the span space  $\{\mathcal{K}_{x_j} \tilde{e}_1, \mathcal{K}_{x_j} \tilde{x}_i, i, j \in \mathbb{N}_m\}$  and the corresponding representer coefficients belong to the span of  $\{\tilde{e}_1, \tilde{x}_i : i \in \mathbb{N}_m\}$ . ■

The representer theorem above tells us that the optimal solution  $\vec{f}_{\mathbf{z}}$  of algorithm (3) lives in the finite span of training samples which paves the way for designing efficient optimization algorithms for multi-task gradient learning.

## 4 Optimization and solution

In this section, by the above representer theorem, we explore efficient algorithms for computing the representer coefficients. For clarity, we mainly focus on least-square multi-task gradient learning algorithms (LSMGL). At the end of this section, the support vector machine for gradient learning (SVMMLG) in classification will be briefly discussed. One can apply the subsequent procedures to other loss functions.

### 4.1 Computation of representer coefficients

To specify the solution of LSMGL, we denote the column vector  $C_{\mathbf{z}} \in \mathbb{R}^{m(d+1)}$  by consecutively concatenating all column vectors  $\{c_{j,\mathbf{z}} \in \mathbb{R}^{d+1} : j \in \mathbb{N}_m\}$  and, likewise we define a column vector  $\mathbb{Y} \in \mathbb{R}^{m(d+1)}$  by concatenating column vectors  $\{y_i \in \mathbb{R}^{d+1} : i \in \mathbb{N}_m\}$ . Moreover, we introduce an  $m(d+1) \times m(d+1)$  matrix by concatenating all  $(d+1) \times (d+1)$  matrix  $\mathcal{K}(x_i, x_j)$  denoted by

$$\mathcal{K}_{\mathbf{x}} = (\mathcal{K}(x_i, x_j))_{i,j \in \mathbb{N}_m}.$$

Finally, we introduce a system of equations

$$m^2 \lambda c_j + B_j \sum_{l \in \mathbb{N}_m} \mathcal{K}(x_j, x_l) c_l = y_j, \quad \forall j \in \mathbb{N}_m \quad (10)$$

where  $B_j = \sum_{i \in \mathbb{N}_m} w_{ij} (\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j)(\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j)^T$ ,  $y_j = \sum_{i \in \mathbb{N}_m} w_{ij} y_i (\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j)$ .

We now can solve the LSMGL algorithm by the following theorem.

**Theorem 4** *For any  $j \in \mathbb{N}_m$ , the vectors  $B_j, y_j$  be defined by equation (10). Then, the representer coefficients  $C_{\mathbf{z}}$  for the solution of the LSMGL algorithm are given by the following equation*

$$\mathbb{Y} = \left( m^2 \lambda I_{m(d+1)} + \text{diag}(B_1, \dots, B_m) \mathcal{K}_{\mathbf{x}} \right)_{i,j=1}^m C_{\mathbf{z}}. \quad (11)$$

**Proof:** By Theorem 3, there exists  $\{c_{j,\mathbf{z}} \in \mathbb{R}^{d+1} : j \in \mathbb{N}_m\}$  such that  $\vec{f}_{\mathbf{z}} = \sum_{j \in \mathbb{N}_m} \mathcal{K}_{x_j} c_{j,\mathbf{z}}$ . However, taking the

functional derivative of algorithm (3) with respect to  $f$  yields that  $\frac{1}{m^2} \sum_{i,j \in \mathbb{N}_m} w_{ij} (\langle \vec{f}_z, \mathcal{K}_{x_j}(\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j) \rangle_{\mathcal{K}} - y_i) \mathcal{K}_{x_j}(\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j) + \lambda \vec{f}_z = 0$  which means that  $c_{j,z} = \frac{1}{m^2 \lambda} \sum_{i \in \mathbb{N}_m} w_{ij} (y_i - \langle \vec{f}_z, \mathcal{K}_{x_j}(\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j) \rangle_{\mathcal{K}}) (\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j)$ . Equivalently, equation (10) holds true, and hence completes the assertion. ■

Solving equation (11) involves the inversion of an  $m(d+1) \times m(d+1)$  matrix whose time complexity is usually  $O((md)^3)$ . However, it is computationally prohibitive since the coordinate (feature) dimension  $d$  is very large in many applications. Fortunately, as suggested in Theorem 3, the representer coefficients  $\{c_{j,z} : j \in \mathbb{N}_m\}$  can be represented by the span of column vectors of matrix

$$\tilde{M}_x = \{\tilde{e}_1, \tilde{x}_1, \dots, \tilde{x}_{m-1}, \tilde{x}_m\}.$$

This observation suggests the possibility of reduction of the original high dimensional problem in  $\mathbb{R}^{d+1}$  to the low dimensional space spanned by  $\tilde{M}_x$ . This low dimensional space can naturally be introduced by singular vectors of  $\tilde{M}_x$ .

To this end, we consider the representation of the matrix  $\tilde{M}_x$  by its singular vectors. It will be proven to be useful to represent matrix  $\tilde{M}_x$  from the singular value decomposition (SVD) of the data matrix defined by

$$M_x = [x_1, x_2, \dots, x_{m-1}, x_m].$$

Apparently, the rank  $s$  of  $M_x$  is at most  $\min(m, d)$ . The SVD of  $M_x$  tells us that there exists orthogonal matrices  $V_{d \times d}$  and  $U_{m \times m}$  such that

$$\begin{aligned} M_x &= [V_1, \dots, V_d] \Sigma \begin{bmatrix} U_1 \\ \vdots \\ U_m \end{bmatrix}^T \\ &= [V_1, \dots, V_s] (\beta_1, \dots, \beta_m) \end{aligned} \quad (12)$$

Here, the  $d \times m$  matrix  $\Sigma = \begin{bmatrix} \text{diag}\{\sigma_1, \dots, \sigma_s\} & 0 \\ 0 & 0 \end{bmatrix}$ . For any  $j \in \mathbb{N}_m$ , we use the notation  $U_j = (U_{1j}, \dots, U_{mj})$  and  $\beta_j^T = (\sigma_1 U_{1j}, \sigma_2 U_{2j}, \dots, \sigma_s U_{sj}) \in \mathbb{R}^s$ . From now on we also denote

$$\mathcal{V} = [V_1, \dots, V_s] \quad (13)$$

Hence, we have, for any  $j \in \mathbb{N}_m$ , that  $x_j = \mathcal{V} \beta_j$ .

We are now ready to specify the representation of  $\tilde{M}_x$  from the above SVD of  $M_x$ . To see this, for any  $l \in \mathbb{N}_s$  and  $j \in \mathbb{N}_m$ , let  $\tilde{V}_l^T = (0, V_l^T)$ ,  $\tilde{\beta}_j^T = (0, \beta_j^T)$ . In addition, we introduce the  $(d+1) \times (s+1)$  matrix

$$\tilde{\mathcal{V}} = \begin{pmatrix} 1 & 0 \\ 0 & \mathcal{V} \end{pmatrix} = [\tilde{e}_1, \tilde{V}_1, \dots, \tilde{V}_s] \quad (14)$$

which induces a one-to-one mapping  $\tilde{\mathcal{V}} : \mathbb{R}^{s+1} \rightarrow \mathbb{R}^{d+1}$  defined, for any  $\beta \in \mathbb{R}^{s+1}$ , by  $x = \tilde{\mathcal{V}} \beta \in \mathbb{R}^{d+1}$  since column vectors in  $\tilde{\mathcal{V}}$  are orthogonal to each other. Consequently, it follows that

$$\tilde{M}_x = \tilde{\mathcal{V}} [e_1, \tilde{\beta}_1, \dots, \tilde{\beta}_m],$$

where  $e_1$  is the standard first coordinate basis in  $\mathbb{R}^{s+1}$ . Equivalently, for any  $i, j \in \mathbb{N}_m$ ,

$$\tilde{e}_1 = \tilde{\mathcal{V}} e_1, \quad \tilde{x}_j = \tilde{\mathcal{V}} \tilde{\beta}_j. \quad (15)$$

We now assemble all material to state the reduced system associated with equation (11). For this purpose, firstly we define the kernel  $\tilde{\mathcal{K}}$ , for any  $x, t \in X$ , by

$$\tilde{\mathcal{K}}(x, t) := \tilde{\mathcal{V}}^T \mathcal{K}(x, t) \tilde{\mathcal{V}},$$

and introduce the  $m(s+1) \times m(s+1)$  matrix by catenating all  $(s+1) \times (s+1)$  matrices  $\tilde{\mathcal{K}}(x_i, x_j)$ :

$$\tilde{\mathcal{K}}_x = (\tilde{\mathcal{K}}(x_i, x_j))_{i,j \in \mathbb{N}_m}. \quad (16)$$

Secondly, for any  $j \in \mathbb{N}_m$ , set  $\mathcal{B}_j = \sum_{i \in \mathbb{N}_m} w_{ij} (e_1 + \tilde{\beta}_i - \tilde{\beta}_j)(e_1 + \tilde{\beta}_i - \tilde{\beta}_j)^T$  and  $\mathcal{Y}_j = \sum_{i \in \mathbb{N}_m} w_{ij} y_i (e_1 + \tilde{\beta}_i - \tilde{\beta}_j)$ . Thirdly, associated with the system (10), for any  $j \in \mathbb{N}_m$  and  $\gamma_j \in \mathbb{R}^{s+1}$ , we define the system in reduced low dimensional space  $\mathbb{R}^{s+1}$

$$m^2 \lambda \gamma_j + \mathcal{B}_j \sum_{l \in \mathbb{N}_m} \tilde{\mathcal{K}}(x_j, x_l) \gamma_l = \mathcal{Y}_j. \quad (17)$$

Finally, in analogy with the notation  $\mathbb{Y}$ , the column vector  $\mathcal{Y} \in \mathbb{R}^{m(s+1)}$  is defined by successively catenating column vectors  $\{\mathcal{Y}_i \in \mathbb{R}^{s+1} : i \in \mathbb{N}_m\}$ . Likewise we can define  $\gamma_z$  by catenating column vectors  $\{\gamma_{j,z} \in \mathbb{R}^{s+1} : j \in \mathbb{N}_m\}$ . With the above preparation we have the following result.

**Theorem 5** *If the  $\{\gamma_{j,z} \in \mathbb{R}^{s+1} : j \in \mathbb{N}_m\}$  is the solution of system (17), i.e.,*

$$\mathcal{Y} = (m^2 \lambda I_{m(s+1)} + \text{diag}(\mathcal{B}_1, \dots, \mathcal{B}_m) \tilde{\mathcal{K}}_x) \gamma, \quad (18)$$

*then the coefficient  $C_z$  defined, for any  $j \in \mathbb{N}_m$ , by  $c_{j,z} = \tilde{\mathcal{V}} \gamma_{j,z}$  is one of the solution of system (11), and thus yields representation coefficients of the solution  $\vec{f}_z$  for the LSMGL algorithm (3).*

**Proof:** Let  $\gamma_z$  is the solution of system (17). Since  $\tilde{\mathcal{V}}$  is orthogonal, the system (17) is equivalent to the following equation

$$m^2 \lambda \tilde{\mathcal{V}} \gamma_{j,z} + \tilde{\mathcal{V}} \mathcal{B}_j \sum_{l \in \mathbb{N}_m} \tilde{\mathcal{K}}(x_j, x_l) \gamma_{l,z} = \tilde{\mathcal{V}} \mathcal{Y}_j.$$

Recall, for any  $j \in \mathbb{N}_m$ , that  $B_j = \tilde{\mathcal{V}} \mathcal{B}_j \tilde{\mathcal{V}}^T$ ,  $\tilde{\mathcal{V}} e_1 e_1^T \tilde{\mathcal{V}}^T = \tilde{e}_1 \tilde{e}_1^T$ ,  $\tilde{\mathcal{V}}^T \tilde{\mathcal{V}} = I_{s+1}$ , and  $Y_j = \tilde{\mathcal{V}} \mathcal{Y}_j$ . Hence, the above system is identical to system (11) (i.e. system (10)) with  $C_j$  replaced by  $\tilde{\mathcal{V}} \gamma_{j,z}$  which completes the assertion. ■

We end this subsection with a brief discussion of the solution of the SVMGL algorithm. Since the hinge loss is not differentiable, we cannot use the above techniques to derive an optimization algorithm for SVMGL. Instead, we can consider its dual problem. To this end, we introduce slack variables  $\{\xi_{ij} : i, j \in \mathbb{N}_m\}$  and rewrite SVMGL as follows:

$$\begin{cases} \arg \min_{\vec{f}, \xi, b} \left\{ \frac{1}{m^2} \sum_{i,j \in \mathbb{N}_m} w_{ij} \xi_{ij} + \lambda \|\vec{f}\|_{\mathcal{K}}^2 \right\} \\ \text{s.t.} \quad y_i (\langle \vec{f}, \mathcal{K}_{x_j}(\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j) \rangle_{\mathcal{K}} + b) \geq 1 - \xi_{ij}, \\ \quad \xi_{ij} \geq 0, \quad \forall i, j \in \mathbb{N}_m. \end{cases} \quad (19)$$

- Given a multi-task kernel  $\mathcal{K}$  produced by scalar kernel  $G$ ,  $\lambda > 0$  and inputs  $\{(x_i, y_i) : i \in \mathbb{N}_m\}$
1. Compute projection mapping  $\tilde{\mathcal{V}}$  and reduced vector  $\tilde{\beta}_j$  from equations (12), (13), and (14).
  2. Compute  $\tilde{\mathcal{K}}(x_j, x_i) = \tilde{\mathcal{V}}^T \mathcal{K}(x_j, x_i) \tilde{\mathcal{V}}$  (i.e. equation (16))
  3. Solving equation (18) to get coefficient  $\gamma_{\mathbf{z}}$ , see Theorem 5 (equivalently, equation (25) when  $G$  is linear or RBF kernel, see Theorem 6).
  4. Output vector-valued function:  $\vec{f}_{\mathbf{z}}(\cdot) = \sum_{j \in \mathbb{N}_m} \mathcal{K}(\cdot, x_j) (\tilde{\mathcal{V}} \gamma_{j, \mathbf{z}})$ .
  5. Compute variable covariance and ranking variables using Proposition 1 in Section 6.

Table 1: Pseudo-code for least square multi-task gradient learning

Parallel to the derivation of the dual problem of standard SVM (e.g. [20, 23]), using Lagrangian theory we can obtain the following dual problem of SVMMLG:

$$\left\{ \begin{array}{l} \arg \max_{\alpha} \sum_{i,j \in \mathbb{N}_m} \alpha_{ij} - \frac{1}{4m^2 \lambda} \sum_{i,j,i',j' \in \mathbb{N}_m} \alpha_{ij} y_i \alpha_{i'j'} y_{i'} \\ \times [(\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j)^T \mathcal{K}(x_j, x_{j'}) (\tilde{e}_1 + \tilde{x}_{i'} - \tilde{x}_{j'})] \\ \text{s.t. } \sum_{i,j \in \mathbb{N}_m} y_i \alpha_{ij} = 0, 0 \leq \alpha_{ij} \leq w_{ij}, \forall i, j \in \mathbb{N}_m. \end{array} \right. \quad (20)$$

Moreover, if the solution of dual problem is  $\alpha_{\mathbf{z}} = \{\alpha_{ij, \mathbf{z}} : i, j \in \mathbb{N}_m\}$  then the solution of SVMMLG can be represented by

$$\vec{f}_{\mathbf{z}} = \frac{1}{2m^2 \lambda} \sum_{i,j \in \mathbb{N}_m} y_i \alpha_{ij, \mathbf{z}} \mathcal{K}_{x_j} (\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j).$$

Note that

$$((\tilde{e}_1 + \tilde{x}_i - \tilde{x}_j)^T \mathcal{K}(x_j, x_{j'}) (\tilde{e}_1 + \tilde{x}_{i'} - \tilde{x}_{j'}))_{(i,j),(i',j')}$$

is a scalar kernel matrix with double indices  $(i, j)$  and  $(i', j')$ . Then, when the number of samples is small the dual problem of SVMMLG can be efficiently solved by quadratic programming with  $\alpha \in \mathbb{R}^{m^2}$ .

## 4.2 Further low dimensional formulation

Consider the multi-task kernel  $\mathcal{K}$  defined by equation (6) with scalar kernel  $G$ . In this section, by further specifying  $G$  we show that LSMGL algorithm (4) with input/output  $\{(x_i, y_i) : i \in \mathbb{N}_m\}$  can be reduced to its low dimensional formulation with input/output  $\{(\beta_i, y_i) : i \in \mathbb{N}_m\}$ , where  $\beta_j$  is defined by equation (12). This clarification will provide a more computationally efficient algorithm.

To this end, consider the scalar kernel  $G$  defined on  $\mathbb{R}^d \times \mathbb{R}^d$ . By definition of kernels (called restriction theorem in [3]), we can see that  $G$  is also a reproducing kernel on  $\mathbb{R}^s \times \mathbb{R}^s$ . Hence,  $\mathcal{K}$  defined by equation (6) on  $\mathbb{R}^d$  is also a multi-task kernel on the underlying space  $\mathbb{R}^s$ ; we use the same notation  $\mathcal{K}$  when no confusion arises. Therefore, associated with the LSMGL algorithm (4) in  $\mathbb{R}^d$  we have an LSMGL in low dimensional input space  $\mathbb{R}^s$ :

$$\vec{g}_{\mathbf{z}} = \arg \min_{\vec{g} \in \mathcal{H}_{\mathcal{K}}} \left\{ \frac{1}{m^2} \sum_{i,j} w_{ij} [y_i - g_1(\beta_j) - \vec{g}_2(\beta_j) (\beta_i - \beta_j)^T]^2 + \lambda \|\vec{g}\|_{\mathcal{K}}^2 \right\}. \quad (21)$$

In analogy with the derivation of the system (10), for any  $j \in \mathbb{N}_m$  and  $\gamma_j \in \mathbb{R}^{s+1}$ , we know that the representer coefficients of the LSMGL algorithm (21) in reduced low dimensional space  $\mathbb{R}^{s+1}$  satisfy that

$$m^2 \lambda \gamma_j + \mathcal{B}_j \sum_{l \in \mathbb{N}_m} \mathcal{K}(\beta_j, \beta_l) \gamma_l = \mathcal{Y}_j. \quad (22)$$

We are now ready to discuss the relation between representer coefficients of the LSMGL algorithm (4) and those of reduced LSMGL algorithm (21). For this purpose, let  $G$  to satisfy, for any  $d \times s$  matrix  $V$ , that  $V^T V = I_s$  and  $\beta, \beta' \in \mathbb{R}^s$ , that

$$G(V\beta, V\beta') = G(\beta, \beta'). \quad (23)$$

There exists abundant functions  $G$  satisfying the above property. For instance, linear product kernel  $G(x, t) = x^T t$ , Gaussian kernel  $G(x, t) = e^{-\|x-t\|^2/2\sigma}$ , and sigmoid kernel  $G(x, t) = \tanh(ax^T t + r)$  with parameters  $a, r \in \mathbb{R}$ . More generally, kernel  $G$  satisfies property (23) if it is produced by a radial basis function (RBF)  $h : (0, \infty) \rightarrow \mathbb{R}$  defined, for any  $x, t \in X$ , by

$$G(x, t) = h(\|x - t\|^2). \quad (24)$$

We say a function  $h : (0, \infty) \rightarrow \mathbb{R}$  is *complete monotone* if it is smooth and, for any  $r > 0$  and  $k \in \mathbb{N}$ ,  $(-1)^k f^{(k)}(r) \geq 0$ . Here  $h^{(k)}$  denotes the  $k$ -th derivative of  $h$ . According to the well-known Schoenberg's theorem [18], if  $h$  is complete monotone then the function  $G$  defined by equation (24) is positive semi-definite, and hence becomes a scalar kernel. For instance, the choice of  $h(t) = e^{-\frac{t}{2\sigma}}$  with standard deviation  $\sigma > 0$  and  $h(t) = (\sigma^2 + \|x - t\|^2)^{-\alpha}$  with parameter  $\alpha > 0$  yield Laplacian kernel and inverse polynomial kernel respectively.

Now we are in a position to summarize the reduction theorem for multi-task kernels (6) produced by scalar kernels  $G$  satisfying (23). Here we also use the convention that  $\mathcal{K}_{\beta} = (\mathcal{K}(\beta_i, \beta_j))_{i,j \in \mathbb{N}_m}$ .

**Theorem 6** *Let  $G$  have the property (23) and  $\mathcal{K}$  be defined by equation (6). Suppose  $\{\gamma_{j, \mathbf{z}} : j \in \mathbb{N}_m\}$  are the representer coefficients of algorithm (21), i.e.,  $\gamma_{\mathbf{z}}$  solves the equation*

$$\mathcal{Y} = \left( m^2 \lambda I_{m(s+1)} + \text{diag}(\mathcal{B}_1, \dots, \mathcal{B}_m) \mathcal{K}_{\beta} \right) \gamma_{\mathbf{z}}, \quad (25)$$

*Then, the representer coefficients  $\{c_{j, \mathbf{z}} : j \in \mathbb{N}_m\}$  of algorithm (4) are given by*

$$c_{j, \mathbf{z}} = \tilde{\mathcal{V}} \gamma_{j, \mathbf{z}}.$$

**Proof:** Suppose the multi-task kernel  $\mathcal{K}$  is produced by  $G$  with property (23). Recall that  $\mathcal{V}^T \mathcal{V} = I_s$  with  $\mathcal{V}$  given by (14). Then, kernel  $\mathcal{K}$  satisfies, for any  $x = \mathcal{V}\beta$  and  $t = \mathcal{V}\beta'$  with  $\mathcal{V}$ , that

$$\begin{aligned} \mathcal{K}(x, t) &= \mathcal{K}(\mathcal{V}\beta_i, \mathcal{V}\beta_j) \\ &= \begin{pmatrix} G(\beta, \beta'), & (\mathcal{V}\nabla_{\beta_i} G(\beta_i, \beta_j))^T \\ \mathcal{V}\nabla_{\beta_j} G(\beta_i, \beta_j), & \mathcal{V}(\nabla_{\beta_i} \nabla_{\beta_j} G(\beta_i, \beta_j))\mathcal{V}^T \end{pmatrix}. \end{aligned}$$

Hence, it follows, for any  $x_i, x_j \in X$  and  $i, j \in \mathbb{N}_m$ , that

$$\tilde{\mathcal{K}}(x_i, x_j) = \begin{pmatrix} 1 & 0 \\ 0 & \mathcal{V} \end{pmatrix}^T \mathcal{K}(x_i, x_j) \begin{pmatrix} 1 & 0 \\ 0 & \mathcal{V} \end{pmatrix} = \mathcal{K}(\beta_i, \beta_j).$$

Therefore, the system (17) is identical to the system (22). Consequently, the desired assertion follows directly from Theorem 5.  $\blacksquare$

Equipped with Theorem 6, the time complexity and computer memory can be further reduced by directly computing  $m(s+1) \times m(s+1)$  matrix  $\mathcal{K}_\beta$  instead of first computing  $m(d+1) \times m(d+1)$  matrix  $\mathcal{K}_x$  and then  $\tilde{\mathcal{K}}_x$  in Theorem 5. Theorem 6 also gives an appealing insight into multi-task gradient learning framework. Roughly speaking, learning gradient in the high dimensional space is equivalent to learning them in the low dimensional projection space spanned by the input data.

## 5 Statistical error analysis

In this section we give an error analysis for least square MGL algorithms. Our target is to show that the learned vector-valued function from our algorithm statistically converges to the true function and true gradient.

For the least square loss, denote by  $\rho_X(\cdot)$  the marginal distribution on  $X$  and, for any  $x \in X$ , let  $\rho(\cdot|x)$  to be the conditional distribution on  $Y$ . Then, the target function is the regression function  $f_\rho$  minimizing the generalization error

$$\mathcal{E}(f) = \int_{\mathcal{X}} (y - f(x))^2 d\rho(x, y).$$

Specifically, the regression function is defined, for any  $x \in X$ , by

$$f_\rho(x) = \arg \min_{t \in \mathbb{R}} \int_Y (y - t)^2 \rho(y|x) = \int_Y y d\rho(y|x).$$

Hence, in this case the purpose of error analysis is to show that solution  $\vec{f}_z$  of LSMGL algorithm (4) statistically converges to  $\vec{f}_\rho = (f_\rho, \nabla f_\rho)$  as  $m \rightarrow \infty$ ,  $s = s(m) \rightarrow 0$  and  $\lambda = \lambda(m) \rightarrow 0$ .

To this end, we introduce some notations and the following hypotheses are assumed to be true throughout this section. Firstly, we assume that  $Y \subseteq [-M, M]$  with  $M > 0$ . Since  $X$  is compact the diameter of  $X$  denoted by  $D = \sup_{x, u \in X} \|x - u\|$  is finite. Secondly, denote by  $L_{\rho_X}^2$  the space of square integral functions  $\vec{f} : X \rightarrow \mathbb{R}^{d+1}$  with norm  $\|\vec{f}\|_{\rho_X}^2 = \int_X \|\vec{f}(x)\|^2 d\rho_X(x)$ . Finally, denote the boundary of  $X$  by  $\partial X$ . We assume, for some constant  $c_\rho > 0$ , that the marginal distribution satisfies that

$$\rho_X(x \in X : \text{dist}(x, \partial X) < s) \leq c_\rho s, \quad \forall 0 < s < D. \quad (26)$$

and, for some parameter  $0 < \theta \leq 1$ , the density function  $p(x)$  of  $\rho_X$  satisfies  $\theta$ -Hölder continuous condition, i.e., for any  $x, u \in X$  there holds

$$|p(x) - p(u)| \leq c_\rho \|x - u\|^\theta, \quad \forall x, u \in X. \quad (27)$$

Of course,  $p$  is a bounded function on  $X$  since it is continuous and  $X$  is compact. For instance, if the boundary of  $X$  is piecewise smooth and  $\rho_X$  is the uniform distribution over  $X$  then the marginal distribution  $\rho_X$  satisfies conditions (26) and (27) with parameter  $\theta = 1$ .

We are ready to present our statistical error analysis of LSMGL algorithms. Recall here we used the notation  $\vec{f}_\rho = (f_\rho, \nabla f_\rho)$ .

**Theorem 7** *Suppose that the marginal distribution  $\rho_X$  satisfies (26) and (27). For any multi-task kernel  $\mathcal{K}$ , let  $\vec{f}_z$  be the solution of LSMGL algorithm. If  $\vec{f}_\rho \in \mathcal{H}_\mathcal{K}$  then there exists a constant  $c$  such that, for any  $m \in \mathbb{N}$ , with the choice of  $\lambda = s^{2\theta}$  and  $s = m^{-\frac{1}{3(d+2)+4\theta}}$ , there holds*

$$\mathbb{E}[\|\vec{f}_z - \vec{f}_\rho\|_{\rho_X}^2] \leq cm^{-\frac{\theta}{3(d+2)+4\theta}}.$$

*If moreover  $\rho_X$  is a uniform distribution then, choosing  $\lambda = s^\theta$  and  $s = m^{-\frac{1}{3(d+2)+5\theta}}$ , there holds*

$$\mathbb{E}[\|\vec{f}_z - \vec{f}_\rho\|_{\rho_X}^2] \leq cm^{-\frac{\theta}{3(d+2)+\theta}}.$$

The proof of this theorem needs several steps which are postponed to the appendix. More accurate error rates in terms of probability inequality are possible using techniques in [17, 15]. It would also be interesting to extend this theorem to other loss functions such as the SVMML algorithm.

## 6 Experimental validation

In this section we will only preliminarily validate the MGL algorithm (3) on the problem of variable selection and covariance measurement.

By the representer Theorem 3 in Section 3, the solution of MGL denoted by  $\vec{f}_z = (f_{1,z}, f_{2z}) = (f_{1,z}, f_{2,z}, \dots, f_{d+1,z})$  can be rewritten as  $\vec{f}_z = \sum_{j \in \mathbb{N}_m} \mathcal{K}_{x_j} c_{j,z}$ . Since it only belongs to a vector-valued RKHS  $\mathcal{H}_\mathcal{K}$ , we need to find a common criterion inner product (norm)  $\langle \cdot, \cdot \rangle_r$  to measure each component of the learned gradient  $\vec{f}_{2z} = (f_{2,z}, \dots, f_{d+1,z})$ . Once we find the criterion inner product  $\langle \cdot, \cdot \rangle_r$ , we can use the coordinate covariance

$$\text{Cov}(\vec{f}_{2z}) = \left( \langle f_{p+1,z}, f_{q+1,z} \rangle_r \right)_{p,q \in \mathbb{N}_d} \quad (28)$$

to measure how the variables covary. Also, the variable (feature) ranking can be done according to the following relative magnitude of norm of each component of  $\vec{f}_{2z}$ :

$$s_p = \frac{\|f_{p+1,z}\|_r}{(\sum_{q \in \mathbb{N}_d} \|f_{q+1,z}\|_r^2)^{1/2}}. \quad (29)$$

If the scalar kernel  $G$  is a linear kernel then every component of  $\vec{f}_{2z}$  is a constant. In this case, we can choose the standard Euclidean inner product to be the criterion inner product (norm). When the kernel  $G$  is an RBF kernel, we show in the following proposition that we can select the criterion inner product  $\langle \cdot, \cdot \rangle_r$  to be the RKHS inner product  $\langle \cdot, \cdot \rangle_G$  in  $\mathcal{H}_G$ . The computation is summarized in the following proposition.

**Proposition 1** Suppose the scalar kernel  $G$  has a feature representation and the multi-task kernel  $\mathcal{K}$  is defined by equation (6). Then, for any solution  $\vec{f}_{\mathbf{z}} = \sum_{j \in \mathbb{N}_m} \mathcal{K}_{x_j} c_{j,\mathbf{z}} \in \mathcal{H}_{\mathcal{K}}$  of MGL algorithm (3), the following hold true.

1. If  $G$  is a linear kernel then the coordinate covariance is defined by

$$\text{Cov}(\vec{f}_{2\mathbf{z}}) = \vec{f}_{2\mathbf{z}}^T \vec{f}_{2\mathbf{z}} = \sum_{i,j \in \mathbb{N}_m} (x_i, I_d) c_{i,\mathbf{z}} c_{j,\mathbf{z}}^T (x_j, I_d)^T.$$

Moreover, for LSMGL algorithm the above equation can be more efficiently computed by

$$\text{Cov}(\vec{f}_{2\mathbf{z}}) = \mathcal{V} \left[ \sum_{i,j \in \mathbb{N}_m} (\beta_i, I_s) \gamma_i \gamma_j^T (\beta_j, I_s)^T \right] \mathcal{V}^T.$$

2. If  $G$  is a smooth RBF kernel then  $f_{p+1,\mathbf{z}} \in \mathcal{H}_G$  and the coordinate covariance  $\text{Cov}(\vec{f}_{2\mathbf{z}}) = ((f_{p+1,\mathbf{z}}, f_{q+1,\mathbf{z}})_G)_{p,q \in \mathbb{N}_d}$  can be computed by

$$(f_{p+1,\mathbf{z}}, f_{q+1,\mathbf{z}})_G = C_{\mathbf{z}}^T (\mathbb{K}_{pq}(x_i - x_j))_{i,j=1}^m C_{\mathbf{z}}, \quad (30)$$

where the kernel matrix  $\mathbb{K}_{pq}(x_i - x_j)$  defined, for any  $i, j \in \mathbb{N}_m$ , by

$$\begin{pmatrix} -(\partial_{p_q}^2 G)(x_i - x_j), & ((\nabla \partial_{p_q}^2 G)(x_i - x_j))^T \\ -(\nabla \partial_{p_q}^2 G)(x_i - x_j), & (\nabla^2 \partial_{p_q}^2 G)(x_i - x_j) \end{pmatrix}.$$

The proof is postponed to the appendix where the computation of  $\mathbb{K}_{pq}$  is also given if  $G$  is a Gaussian.

We run our experiment on two artificial datasets and one gene expression dataset following [17]. In the first experiment, the target function  $f_\rho : \mathbb{R}^d \rightarrow \mathbb{R}$  with notation  $x = (x^1, \dots, x^d) \in \mathbb{R}^d$  and  $d = 80$ . The output  $y$  is contaminated by a Gaussian noise

$$y = f_\rho(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_y).$$

As depicted in Figure 1 (leftmost), the thirty inputs whose relevant features are  $[1, 10] \cup [11, 20] \cup [41, 50]$  are generated as follows:

1. For samples from 1 to 10,  $x^p \sim \mathcal{N}(1, 0.05)$ , for  $p \in [1, 10]$  and  $x^p \sim \mathcal{N}(0, 0.1)$ , for  $p \in [11, 80]$ .
2. For samples from 11 to 30,  $x^p \sim \mathcal{N}(1, 0.05)$ , for  $p \in [11, 20]$  and  $x^p \sim \mathcal{N}(0, 0.1)$ , for  $p \in [1, 10] \cup [31, 80]$ .
3. For samples from 11 to 30, features are in the form of  $x^p \sim \mathcal{N}(1, 0.05)$ , for  $p \in [41, 50]$ , and  $x^p \sim \mathcal{N}(0, 0.1)$ , for  $p \in [1, 40] \cup [51, 80]$ .

We let the regression function  $f_\rho$  to be a linear function. Specifically, we choose a noise parameter  $\sigma_y = 3$  and the regression function is defined by  $f_\rho(x_i) = w_1^T x_i$  for  $i \in [1, 10]$ ,  $f_\rho(x_i) = w_2^T x_i$  for  $i \in [11, 20]$ , and  $f_\rho(x_i) = w_3^T x_i$  for  $i \in [21, 30]$ , where,  $w_1^k = 2 + 0.5 \sin(2\pi k/10)$  for  $k \in [1, 10]$  and otherwise zero,  $w_2^k = -2 - 0.5 \sin(2\pi k/10)$  for  $k \in [11, 20]$  and zero otherwise. The vector  $w_3$  is defined by  $w_3^k = -2 - 0.5 \sin(2\pi k/10)$  for  $k \in [41, 50]$  and zero otherwise.

In this linear case, we use the kernel  $G(x, t) = x^T t$  as a basic scalar kernel and the multi-task kernel  $\mathcal{K}$  defined by (6) in LSMGL algorithm (4). As in [11, 15], the regularization parameter  $\lambda$  is set to be a fixed number such as 0.1 (variation

in this parameter made little difference to feature ranking). The parameter  $s$  in the weight coefficients  $w_{ij}$  is set to be the median pairwise distance between inputs. In Figure 1, the result of LSMGL is shown in (b) for variable covariation and in (c) for feature selection respectively. We also ran the algorithm (3) with the choice of kernel  $\mathcal{K}(x, t) = G(x, t) I_{d+1}$  ([15, 16, 17]). The results are shown in (d) and (e) of Figure 1. We see that both algorithms worked well. The LSMGL algorithm works slightly better: the reason maybe be that it captures the inherent structure of gradient learning as mentioned before. We also ran LSMGL algorithm on this dataset; the result is no essentially different from SVMMLG.

In the second experiment, we use the SVMMLG algorithm for classification. For this dataset, only the first two features are relevant to the classification task. The remaining 78 redundant features are distributed according to a small Gaussian random deviate. The distribution for the first two features is shown in (f). In SVMMLG, the parameter  $s$  and  $\lambda$  are the same as those in the first example. The scalar kernel is set to be a Gaussian  $G(x, t) = e^{-\|x-t\|^2/2\sigma^2}$  where  $\sigma$  is also the median pairwise distance between inputs. The feature selection results for the SVMMLG algorithm are illustrated respectively in (g) and (h) with different choices of multi-task kernels  $\mathcal{K}$  given by equation (6) and  $\mathcal{K}(x, t) = G(x, t) I_{d+1}$ . Both algorithms picked up the two important features.

Finally, we apply our LSMGL algorithm to a well-studied expression dataset. This dataset has two classes: acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL), see e.g. [10]. There are a total of 7129 genes (variables) and 72 patients, split into a training set of 38 examples and a test set of 34 examples. In the training set, 27 examples belong to ALL and 11 belong to AML, and the test set is composed of 20 ALL and 14 AML. Various variable selection algorithms have been applied to this dataset by choosing features based on training set, and then performing classification on the test set with the selected features. We ran LSMGL with the choice of multi-task  $\mathcal{K}$  given by equation (6) where  $G$  is a linear kernel. The solution  $\vec{f}_{\mathbf{z}}$  is learned from the training set for ranking the genes according to the values of  $s_p$  defined by equation (29). Then, ridge regression is run on the training set with truncated features to build a classifier to predict the labels on the test set. The regularization parameter of LSMGL is fixed to be 0.1 while the regularization parameter in ridge regression is tuned using leave-one-out cross-validation in the training set. The test error with selected top ranked genes is reported in Table 2. The classification accuracy is quite comparable to the gradient learning algorithm using individual RKHSs [15, 17]. However, [11, 15, 17] did the recursive techniques to rank features and employed SVM for classification while our method showed that ridge regression for classification and non-recursive technique for feature ranking also worked well in this data set. It would be interesting to further explore this issue.

The preliminary experiments above validated our proposed MGL algorithms. However further experiments need to be performed to evaluate our multi-task framework for gradient learning.

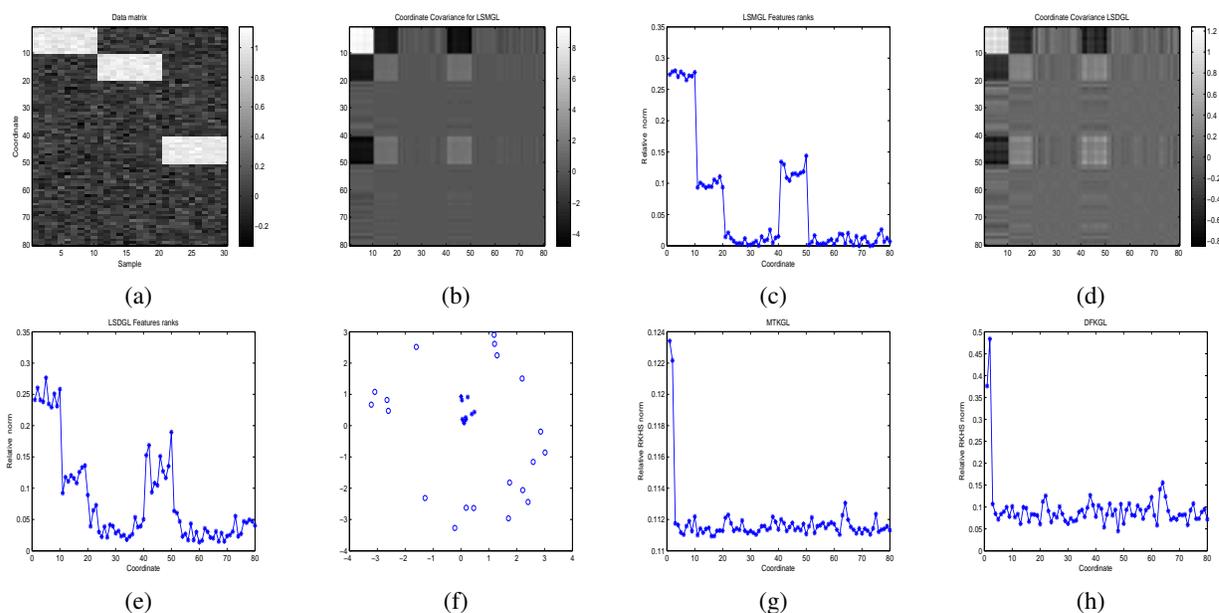


Figure 1: LSMGL and SVMML feature ranking

genes	10	40	80	100	200	500
test error	2	1	0	0	1	1
genes	1000	2000	3000	4000	6000	7129
test error	2	1	1	1	1	1

Table 2: Number of test error using ridge regression algorithm versus the number of top ranked genes selected by LSMGL algorithm.

## 7 Conclusions

In this paper, our main contribution was to provide a novel unifying framework for gradient learning from the perspective of multi-task learning. Various variable selection methods in the literature can be recovered by the choice of multi-task kernels. More importantly, this framework allows us to introduce a novel choice of multi-task kernel to capture the inherent structure of gradient learning. An appealing representer theorem was presented which facilitates the design of efficient optimization algorithms, especially for datasets with high dimension and few training examples. Finally, a statistical error analysis was provided to ensure the convergence of the learned function to true function and true gradient.

Here we only preliminarily validated the method. A more extensive benchmark study remains to be pursued. In future we will explore more experiments on biomedical datasets and compare our MGL algorithms with previous related methods for feature selection, such as those in [21, 22] etc. It will be interesting to implement different loss functions in the MGL algorithms for regression and classification, apply the spectral decomposition of the gradient outer products to dimension reduction (see e.g. [16]), and possible use for network inference from the covariance of the learned gradient

function.

## References

- [1] R. K. Ando & T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Machine Learning Research*, 1817–1853, 2005
- [2] A. Argyriou, T. Evgeniou, & M. Pontil. Multi-task feature learning. *NIPS*, 2006.
- [3] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.* 68: 337–404, 1950.
- [4] P. L. Bartlett & S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *J. of Machine Learning Research*, 3:463–482, 2002.
- [5] S. Ben-David & R. Schuller. Exploiting task relatedness for multiple task learning. *COLT*, 2003.
- [6] D. R. Chen, Q. Wu, Y. Ying, & D. X. Zhou. Support vector machine soft margin classifiers: Error analysis. *J. of Machine Learning Research*, 5:1143–1175, 2004.
- [7] A. Caponnetto, C.A. Micchelli, M. Pontil, & Y. Ying. Universal multi-task kernels, Preprint, 2007.
- [8] S.S. Chen, D.L. Donoho & M.A. Saunders. Atomic decomposition pursuits. *SIAM J. of Scientific Computing*, 20: 33-61,1999.
- [9] F. Cucker & S. Smale. On the mathematical foundations of learning. *Bull. Amer. Math. Soc.* 39: 149, 2001.
- [10] T. R. Golub et. al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286: 531-537, 1999.
- [11] I. Guyon, J. Weston, S. Barnhill, & V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning* 46: 389–422, 2002.
- [12] V.I. Koltchinskii & D. Panchenko. Rademacher processes and bounding the risk of function learning. In *J.*

- Wellner E. Gine, D. Mason, editor, High Dimensional Probability II, pages 443-459, 2000.
- [13] T. Evgeniou, C. A. Micchelli & M. Pontil. Learning multiple tasks with kernel methods. *J. Machine Learning Research*, 6: 615–637, 2005.
- [14] C. A. Micchelli & M. Pontil. On learning vector-valued functions. *Neural Computation*, 17: 177-204, 2005.
- [15] S. Mukherjee & Q. Wu. Estimation of gradients and coordinate covariation in classification. *J. of Machine Learning Research* 7: 2481-2514, 2006.
- [16] S. Mukherjee, Q. Wu, & D. X. Zhou. Learning gradients and feature selection on manifolds. Preprint, 2007.
- [17] S. Mukherjee & D. X. Zhou. Learning coordinate covariances via gradients. *J. of Machine Learning Research* 7: 519-549, 2006.
- [18] I. J. Schoenberg. Metric spaces and completely monotone functions, *Ann. of Math.* 39: 811-841, 1938.
- [19] B. Schölkopf & A. J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, MA, USA, 2002.
- [20] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, 2004.
- [21] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.* 58: 267-288, 1996.
- [22] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, & V. Vapnik. Feature selection for SVMs, *NIPS*, 2001.
- [23] V. N. Vapnik. *Statistical learning theory*. Wiley, New York, 1998.

## Appendix

Let  $G$  be a scalar kernel, we use the convention  $\partial_p^{(2)}G$  to denote the  $p$ -th partial derivative of  $G$  with respect to the second argument, and so is the gradient  $\nabla^{(2)}G$ .

### Proof of Proposition 1.

When  $G$  is a linear kernel, by the definition (6) of multi-task kernel  $\mathcal{K}$ , we have that  $\vec{f}_{2\mathbf{z}} = \sum_{j \in \mathbb{N}_m} (x_j, I_d) c_{j,\mathbf{z}}$  which implies that

$$\text{Cov}(\vec{f}_{2\mathbf{z}}) = \vec{f}_{2\mathbf{z}} \vec{f}_{2\mathbf{z}}^T = \sum_{i,j \in \mathbb{N}_m} (x_i, I_d) c_{i,\mathbf{z}} c_{j,\mathbf{z}}^T (x_j, I_d)^T.$$

For the LSMGL algorithm, in Section 4 we showed that  $c_{j,\mathbf{z}} = \tilde{\mathcal{V}} \gamma_{j,\mathbf{z}}$  and, for any  $j \in \mathbb{N}_m$ ,  $x_j = \mathcal{V} \beta_j$ , the above equation can be further simplified to the following:

$$\text{Cov}(\vec{f}_{2\mathbf{z}}) = \mathcal{V} \left[ \sum_{i,j \in \mathbb{N}_m} (\beta_i, I_s) \gamma_{i,\mathbf{z}} \gamma_{j,\mathbf{z}}^T (\beta_j, I_s)^T \right] \mathcal{V}^T.$$

When  $G$  is an RBF kernel, for any  $x, t \in \mathbb{R}^d$  and  $p, q \in \mathbb{N}_d$ ,  $G(x, t) = G(x - t)$  and  $\partial_{t_q} \partial_{x_p} G(x, t) = -(\partial_p \partial_q G)(x, t)$ . Hence, for any  $p \in \mathbb{N}_d$  and  $x \in \mathbb{R}^d$ , we have that

$$f_{p+1,\mathbf{z}}(x) = \sum_{j \in \mathbb{N}_m} (-\partial_p^{(2)} G(x, x_j), -\nabla^{(2)} \partial_p^{(2)} G(x, x_j)) c_j$$

Since  $G$  has a feature representation, i.e., for any  $x, t \in X$ , there holds that  $G(x, t) = \langle \phi(x), \phi(t) \rangle_{\ell^2}$ . Also, observe that  $\partial_p^{(2)} G(x, x_j) = \langle \phi(x), (\partial_p \phi)(x_j) \rangle_{\ell^2}$  and  $\partial_q^{(2)} \partial_p^{(2)} G(x, x_j) =$

$\langle \phi(x), (\partial_p \partial_q \phi)(x_j) \rangle_{\ell^2}$ . Denote  $c_j$  by  $(c_j^1, \dots, c_j^{d+1})^T$ . Consequently, for any  $p \in \mathbb{N}_d$ ,  $f_{p+1,\mathbf{z}}(\cdot) = \langle w_p, \phi(\cdot) \rangle_{\ell^2}$  with  $w_p = -\sum_{j \in \mathbb{N}_m} (\partial_p \phi(x_j) c_j^1 + \sum_{q \in \mathbb{N}_d} \partial_q \partial_p \phi(x_j) c_j^{q+1})$ . Therefore, for any  $p, q \in \mathbb{N}_d$  we have that  $f_{p+1,\mathbf{z}} \in \mathcal{H}_G$  and

$$\langle f_{p+1,\mathbf{z}}, f_{q+1,\mathbf{z}} \rangle_G = \langle w_p, w_q \rangle_{\ell^2} = \sum_{i,j \in \mathbb{N}_m} c_i^T \mathbb{K}_{pq}(x_i - x_j) c_j$$

which completes the assertion.  $\square$

### Computation of kernel $\mathbb{K}$ .

If  $G$  is a Gaussian kernel with standard variation  $\sigma$ , that is, for any  $x, t \in \mathbb{R}^d$ ,  $G(x, t) = G(x - t) = e^{-\frac{\|x-t\|^2}{2\sigma^2}}$ , the computation of  $\mathbb{K}$  is listed as follows.

1.  $(\partial_{pq}^2 G)(x) = \left[ \frac{x_p x_q}{\sigma^2} - \frac{\delta_{pq}}{\sigma} \right] G(x)$
2. For any  $q' \in \mathbb{N}_d$ ,  $\partial_{q'} \partial_p \partial_q G(x) = \left[ \frac{x_q \sigma_{pq'} + x_p \delta_{qq'} + x_{q'} \delta_{pq} - \frac{x_p x_q x_{q'}}{\sigma^3} \right] G(x)$ . Hence,

$$\nabla \partial_{pq}^2 G(x) = \left[ \frac{e_p x_q + e_q x_p}{\sigma^2} + \frac{x \delta_{pq}}{\sigma^2} - \frac{x_p x_q x}{\sigma^3} \right] G(x)$$

3. For any  $p', q' \in \mathbb{N}_d$ ,  $\partial_{p'} \partial_{q'} \partial_{pq}^2 G(x) = G(x) \left[ \frac{\delta_{p'q'} \delta_{qp'}}{\sigma^2} + \frac{\delta_{p'p'} \delta_{q'q'} + \delta_{p'q'} \delta_{pq}}{\sigma^2} - \frac{1}{\sigma^3} (x_q x_{q'} \sigma_{pp'} + x_p x_q \delta_{p'q'} + x_p x_{q'} \delta_{p'q}) + \frac{x_{p'}}{\sigma} \left( \frac{x_p x_q x_{q'}}{\sigma^3} - \frac{x_q \delta_{pq'} + x_{q'} \delta_{pq} + x_p \delta_{qq'}}{\sigma^2} \right) \right]$ . Hence,

$$\begin{aligned} \nabla^2 \partial_{pq}^2 G(x) &= G(x) \left[ \frac{e_q^T e_p + e_p^T e_q + \delta_{pq} I_d}{\sigma^2} \right. \\ &\quad - \frac{1}{\sigma^3} (x_q (e_p x^T + x e_p^T) + x_p (x e_q^T + e_q x^T)) \\ &\quad \left. - \frac{x_p x_q I_d}{\sigma^3} + \frac{x x^T}{\sigma^3} \left( \frac{x_p x_q}{\sigma} - \sigma_{pq} \right) \right]. \end{aligned}$$

### Proof of Theorem 7

We turn our attention to the proof of Theorem 7. We begin with some notations and background materials. First denote by  $\text{Er}_{\mathbf{z}}$  the empirical loss in LSMGL algorithm, i.e.,

$$\text{Er}_{\mathbf{z}}(\vec{f}) = \frac{1}{m^2} \sum_{i,j} w(x_i - x_j) \times (y_i - f_1(x_j) - \vec{f}_2(x_j)(x_i - x_j)^T)^2,$$

and the modified form of its expectation

$$\text{Er}(\vec{f}) = \int_Z \int_X w(x - u) \left[ y - f_1(u) - \vec{f}_2(u)(x - u) \right]^2 d\rho(x, y) d\rho_X(u).$$

Since the Gaussian weight  $w(x - u) = w_s(x - u)$  is dependent on  $s$ , the above definition of  $\text{Er}(\vec{f})$  is depending on the parameter  $s$ . In addition, define the Lipschitz constant  $|\nabla f_\rho|_{\text{Lip}}$  to be the minimum constant  $c$  such that  $\|\nabla f_\rho(x + u) - \nabla f_\rho(x)\| \leq c\|u\|$ ,  $\forall x, u \in X$ . We say that  $\nabla f_\rho$  is Lipschitz continuous if  $|\nabla f_\rho|_{\text{Lip}}$  is finite.

The error analysis here is divided into two main steps motivated by the techniques in [15]. The first step is to bound the square error  $\|\vec{f}_{\mathbf{z}} - \vec{f}_\rho\|_{\rho_X}^2$  by the excess error  $\text{Er}(\vec{f}_{\mathbf{z}}) - \text{Er}(\vec{f}_\rho)$ . In the second step, we employ standard error decomposition [6] and Rademacher complexities [4, 12] to estimate the excess error. These two steps will be respectively stated in the

following two propositions. Before we do that, we introduce an auxiliary functional  $\mathcal{Q}_s$  defined by

$$\begin{aligned} \mathcal{Q}_s(\vec{f}, \vec{f}_\rho) &= \int \int w(x-u) [f_\rho(u) - f_1(u) \\ &+ (\vec{f}_2(u) - \nabla f_\rho(u))(u-x)^T]^2 d\rho_X(x) d\rho_X(u). \end{aligned}$$

We are ready to present the first step of the error analysis: bounding the square error  $\|\vec{f}_z - \vec{f}_\rho\|_{\rho_X}^2$  by the excess error  $\text{Er}(\vec{f}_z) - \text{Er}(\vec{f}_\rho)$  which is stated as the following proposition.

**Proposition 2** *If  $0 < s, \lambda < 1$  then there exists a constant  $c'_\rho$  such that*

$$\begin{aligned} \mathbb{E} \left[ \|\vec{f}_z - \vec{f}_\rho\|_{\rho_X}^2 \right] &\leq c'_\rho \left( \min[s^{-\theta}, \max_{x \in X} p^{-1}(x)] \right. \\ &\quad \times \mathbb{E}[\text{Er}(\vec{f}_z) - \text{Er}(\vec{f}_\rho)] \\ &\quad \left. + s^\theta (\mathbb{E}[\|\vec{f}_z\|_{\mathcal{K}}^2] + \|\vec{f}_\rho\|_\infty + |\nabla f_\rho|_{\text{Lip}}) \right). \end{aligned}$$

The proof of this proposition follows directly from the following Lemmas 8 and 9. For this purpose, let the subset  $X_s$  of  $X$  be

$$X_s = \{u \in X : \text{dist}(u, \partial X) > s, |p(u)| \geq (1 + c_\rho)s^\theta\} \quad (31)$$

and

$$c_\rho(s) := \min\{p(x) : \|u-x\| \leq s, u \in X_s\}.$$

Recall that  $\tilde{e}_1$  is the first coordinate basis in  $\mathbb{R}^{d+1}$  and, for any  $x \in \mathbb{R}^d$ ,  $\tilde{x}^T = (0, x)^T \in \mathbb{R}^{d+1}$ .

**Lemma 8** *If  $0 < s < 1$  then there exists a constant  $c'_\rho$  such that*

$$\begin{aligned} \mathbb{E} \left[ \|\vec{f}_z - \vec{f}_\rho\|_{\rho_X}^2 \right] &\leq c'_\rho \left( s^\theta [\mathbb{E}[\|\vec{f}_z\|_{\mathcal{K}}^2] + \|\vec{f}_\rho\|_\infty] \right. \\ &\quad \left. + \min[s^{-\theta}, \max_{x \in X} p^{-1}(x)] \mathbb{E} \left[ \mathcal{Q}_s(\vec{f}_z, \vec{f}_\rho) \right] \right). \end{aligned}$$

**Proof:** Write  $\|\vec{f}_z - \vec{f}_\rho\|_{\rho_X}^2$  by

$$\begin{aligned} \|\vec{f}_z - \vec{f}_\rho\|_{\rho_X}^2 &= \int_{X \setminus X_s} \|\vec{f}_z(u) - \vec{f}_\rho(u)\|^2 d\rho_X(u) \\ &\quad + \int_{X_s} \|\vec{f}_z(u) - \vec{f}_\rho(u)\|^2 d\rho_X(u) \end{aligned} \quad (32)$$

By the definition of  $X_s$ , we have that  $\rho_X(X \setminus X_s) \leq c_\rho s + c_\rho(1 + c_\rho)|X|s^\theta \leq c'_\rho s^\theta$  where  $|X|$  is the Lebesgue measure of  $X$ . Hence, the first term of the above equation is bounded by

$$2c'_\rho (\|\vec{f}_z\|_\infty^2 + \|\vec{f}_\rho\|_\infty^2) s^\theta \leq 2c'_\rho (\|\vec{f}_z\|_{\mathcal{K}}^2 + \|\vec{f}_\rho\|_\infty^2) s^\theta.$$

For the second term on the right-hand side of equation (32), observe that, for any  $u \in X_s$ ,  $\text{dist}(u, \partial X) > s$  and  $\{u : \|u-x\| \leq s, x \in X_s\} \subseteq X$ . Moreover, for any  $x \in X$  such that  $\|u-x\| \leq s$ , by the definition of  $X_s$  there holds

$$p(x) = p(u) - (p(u) - p(x)) \geq (1 + c_\rho)s^\theta - c_\rho \|u-x\|^\theta \geq s^\theta.$$

Consequently, it follows that

$$c_\rho(s) \geq \max(s^\theta, \min_{x \in X} p(x)), \quad (33)$$

and

$$\begin{aligned} \mathcal{Q}_s(\vec{f}_z, \vec{f}_\rho) &= \int_X \int_X w(x-u) \\ &\quad \times [(\vec{f}_\rho(u) - \vec{f}_z(u))(\tilde{e}_1 + \tilde{u} - \tilde{x})]^2 d\rho_X(x) d\rho_X(u) \\ &\geq \int_{X_s} \left[ \int_{\|u-x\| \leq s} ((\vec{f}_\rho(u) - \vec{f}_z(u))(\tilde{e}_1 + \tilde{u} - \tilde{x}))^2 \right. \\ &\quad \left. \times d\rho_X(x) \right] d\rho_X(u) \\ &\geq c_\rho(s) \int_{X_s} \left[ \int_{\|u-x\| \leq s} ((\vec{f}_\rho(u) - \vec{f}_z(u))(\tilde{e}_1 + \tilde{u} - \tilde{x}))^2 \right. \\ &\quad \left. \times dx \right] d\rho_X(u). \end{aligned} \quad (34)$$

The integral w.r.t.  $x$  on the right-hand side of the above inequality can be written as  $(\vec{f}_\rho(u) - \vec{f}_z(u))W(s)(\vec{f}_\rho(u) - \vec{f}_z(u))^T$  with  $(d+1) \times (d+1)$  matrix  $W(s)$  defined by

$$W(s) = \int_{\|u-x\| \leq s} [(\tilde{e}_1 + \tilde{u} - \tilde{x})(\tilde{e}_1 + \tilde{u} - \tilde{x})^T] dx.$$

Here,  $(\tilde{e}_1 + \tilde{u} - \tilde{x})(\tilde{e}_1 + \tilde{u} - \tilde{x})^T$  equals that

$$\begin{pmatrix} 1 & (u-x)^T \\ u-x & (u-x)(u-x)^T \end{pmatrix}$$

Observe that  $\int_{\|u-x\| \leq s} w(x-u) dx = s^{-2} \int_{\|t\| \leq 1} e^{-\frac{\|t\|^2}{2}} dt$  and  $\int_{\|u-x\| \leq s} w(x-u)(u-x) dx = 0$ . In addition, for any  $p \neq q \in \mathbb{N}_d$ ,  $\int_{\|u-x\| \leq s} w(x-u)(u^p - x^p)(u^q - x^q) dx = 0$  and  $\int_{\|u-x\| \leq s} w(x-u)(x^p - u^p)^2 dx = \int_{\|t\| \leq 1} e^{-\frac{\|t\|^2}{2}} (t^p)^2 dt$ . From the above observations, there exists a constant  $c$  such that

$$(\vec{f}_\rho(u) - \vec{f}_z(u))W(s)(\vec{f}_\rho(u) - \vec{f}_z(u))^T \geq c \|\vec{f}_\rho(u) - \vec{f}_z(u)\|^2.$$

Recalling the definition of  $W(s)$  and substituting this back into equation (34) implies, for any  $0 < s < 1$ , that

$$c c_\rho(s) \int_{X_s} \|\vec{f}_z(u) - \vec{f}_\rho(u)\|^2 d\rho_X(u) \leq \mathcal{Q}_s(\vec{f}_z, \vec{f}_\rho).$$

Plugging this into equation (34), the desired estimate follows from the estimation of  $c_\rho(s)$ , i.e., equation (33).  $\blacksquare$

Now we can bound  $\mathcal{Q}_s$  by the following lemma.

**Lemma 9** *If  $0 < s < 1$  then there exists a constant  $c$  such that, for any  $\vec{f} \in \mathcal{H}_{\mathcal{K}}$ , the following equations hold true.*

1.  $\mathcal{Q}_s(\vec{f}, \vec{f}_\rho) \leq c \left( s^2 |\nabla f_\rho|_{\text{Lip}}^2 + [\text{Er}(\vec{f}) - \text{Er}(\vec{f}_\rho)] \right)$ .
2.  $\text{Er}(\vec{f}) - \text{Er}(\vec{f}_\rho) \leq c \left( s^2 |\nabla f_\rho|_{\text{Lip}}^2 + \mathcal{Q}_s(\vec{f}, \vec{f}_\rho) \right)$ .

**Proof:** Observe that  $[y - f_1(u) - \vec{f}_2(u)(x-u)^T]^2 = [y - f_\rho(u) - \nabla f_\rho(u)(x-u)^T]^2 + 2[y - f_\rho(u) - \nabla f_\rho(u)(x-u)^T][f_\rho(u) - f_1(u) + (\vec{f}_2(u) - \nabla f_\rho(u))(u-x)^T] + [f_\rho(u) - f_1(u) + (\vec{f}_2(u) - \nabla f_\rho(u))(u-x)^T]^2$ . Then, taking the integral of both sides of the above equality and using the fact that  $f_\rho(x) = \int_Y y d\rho_X(x)$  we have that

$$\begin{aligned} \text{Er}(\vec{f}) - \text{Er}(\vec{f}_\rho) &= \mathcal{Q}_s(\vec{f}, \vec{f}_\rho) + 2 \int_X \int_X w(x-u) [f_\rho(x) \\ &\quad - f_\rho(u) - \nabla f_\rho(u)(x-u)^T] [f_\rho(u) - f_1(u) + \\ &\quad (\vec{f}_2(u) - \nabla f_\rho(u))(u-x)^T] d\rho_X(x) d\rho_X(u) \\ &\geq \mathcal{Q}_s(\vec{f}, \vec{f}_\rho) - 2 \left( \int_X \int_X w(x-u) [f_\rho(x) - f_\rho(u) \right. \\ &\quad \left. - \nabla f_\rho(u)(x-u)^T]^2 d\rho_X(x) d\rho_X(u) \right)^{\frac{1}{2}} \left( \mathcal{Q}_s(\vec{f}, \vec{f}_\rho) \right)^{\frac{1}{2}}. \end{aligned}$$

Applying the inequality, for any  $a, b > 0$ , that  $-2a^2 - \frac{1}{2}b^2 \leq -2ab$ , from the above equality we further have that

$$\begin{aligned} \text{Er}(\vec{f}) - \text{Er}(\vec{f}_\rho) &\geq \frac{1}{2} \mathcal{Q}_s(\vec{f}, \vec{f}_\rho) - 2 \int_X \int_X w(x-u) \\ &\quad [f_\rho(x) - f_\rho(u) - \nabla f_\rho(u)(x-u)^T]^2 d\rho_X(x) d\rho_X(u). \end{aligned} \quad (35)$$

Likewise,

$$\begin{aligned} \text{Er}(\vec{f}) - \text{Er}(\vec{f}_\rho) &= \mathcal{Q}_s(\vec{f}, \vec{f}_\rho) + 2 \int_X \int_X w(x-u) [f_\rho(x) \\ &\quad - f_\rho(u) - \nabla f_\rho(u)(x-u)^T] [f_\rho(u) - f_1(u) + \\ &\quad (\vec{f}_2(u) - \nabla f_\rho(u))(u-x)^T] d\rho_X(x) d\rho_X(u) \\ &\leq \mathcal{Q}_s(\vec{f}, \vec{f}_\rho) + 2 \left( \int_X \int_X w(x-u) [f_\rho(x) - f_\rho(u) \right. \\ &\quad \left. - \nabla f_\rho(u)(x-u)^T]^2 d\rho_X(x) d\rho_X(u) \right)^{\frac{1}{2}} \left( \mathcal{Q}_s(\vec{f}, \vec{f}_\rho) \right)^{\frac{1}{2}}. \end{aligned}$$

Applying the inequality  $2ab \leq a^2 + b^2$  to the above inequality yields that

$$\begin{aligned} \text{Er}(\vec{f}) - \text{Er}(\vec{f}_\rho) &\leq 2\mathcal{Q}_s(\vec{f}, \vec{f}_\rho) + \int_X \int_X w(x-u) \\ &\quad [f_\rho(x) - f_\rho(u) - \nabla f_\rho(u)(x-u)^T]^2 d\rho_X(x) d\rho_X(u) \end{aligned} \quad (36)$$

However,  $|f_\rho(x) - f_\rho(u) - \nabla f_\rho(u)(x-u)^T| = \left| \int_0^1 (\nabla f_\rho(tx + (1-t)u) - \nabla f_\rho(u))(x-u)^T dt \right| \leq |\nabla f_\rho|_{\text{Lip}} \|x-u\|^2$  and the density  $p(x)$  of  $\rho_X$  is a bounded function since we assume it is  $\theta$ -Hölder continuous and  $X$  is compact. Therefore,

$$\begin{aligned} &\int_X \int_X w(x-u) [f_\rho(x) - f_\rho(u) - \nabla f_\rho(u)(x-u)^T]^2 \\ &\quad \times d\rho_X(x) d\rho_X(u) \\ &\leq \|p\|_\infty |\nabla f_\rho|_{\text{Lip}}^2 \left[ \int_{\mathbb{R}^d} \frac{1}{s^{d+2}} e^{-\frac{\|x\|^2}{2s^2}} \|x\|^4 dx \right] \\ &\leq c \|p\|_\infty |\nabla f_\rho|_{\text{Lip}}^2 s^{2\theta}. \end{aligned}$$

Putting this into Equations (35) and (36) and arranging the terms involved yields the desired result.  $\blacksquare$

From Property (1) of Lemma 9, for any  $\vec{f} \in \mathcal{H}_\mathcal{K}$  we have that

$$\text{Er}(\vec{f}) - \text{Er}(\vec{f}_\rho) \geq -cs^2 |\nabla f_\rho|_{\text{Lip}}^2. \quad (37)$$

We now turn our attention to the second step of the error analysis: the estimation of the excess error  $\text{Er}(\vec{f}_\mathbf{z}) - \text{Er}(\vec{f}_\rho) + \lambda \|\vec{f}_\mathbf{z}\|_{\mathcal{K}}^2$ . To do this, let

$$\vec{f}_\lambda = \arg \inf_{\vec{f} \in \mathcal{H}_\mathcal{K}} \left\{ \text{Er}(\vec{f}) + \lambda \|\vec{f}\|_{\mathcal{K}}^2 \right\}$$

By the *error decomposition* technique in [6], we get the following estimation.

**Proposition 3** *If  $\vec{f}_\lambda$  is defined above then there exists a constant  $c$  such that*

$$\begin{aligned} \text{Er}(\vec{f}_\mathbf{z}) - \text{Er}(\vec{f}_\rho) + \lambda \|\vec{f}_\mathbf{z}\|_{\mathcal{K}}^2 &\leq \mathcal{S}(\mathbf{z}) \\ &\quad + c(s^2 |\nabla f_\rho|_{\text{Lip}}^2 + \mathcal{A}(\lambda, s)), \end{aligned}$$

where

$$\mathcal{S}(\mathbf{z}) = \text{Er}(\vec{f}_\mathbf{z}) - \text{Er}_\mathbf{z}(\vec{f}_\mathbf{z}) + \text{Er}_\mathbf{z}(\vec{f}_\lambda) - \text{Er}(\vec{f}_\lambda)$$

is referred to the *sample error* and

$$\mathcal{A}(\lambda, s) = \inf_{\vec{f} \in \mathcal{H}_\mathcal{K}} \left\{ \mathcal{Q}_s(\vec{f}, \vec{f}_\rho) + \lambda \|\vec{f}\|_{\mathcal{K}}^2 \right\}$$

is called the *approximation error*.

**Proof:** Note that  $\text{Er}(\vec{f}_\mathbf{z}) - \text{Er}(\vec{f}_\rho) + \lambda \|\vec{f}_\mathbf{z}\|_{\mathcal{K}}^2 = [\text{Er}(\vec{f}_\mathbf{z}) - \text{Er}_\mathbf{z}(\vec{f}_\mathbf{z}) + \text{Er}_\mathbf{z}(\vec{f}_\lambda) - \text{Er}(\vec{f}_\lambda)] + [(\text{Er}_\mathbf{z}(\vec{f}_\mathbf{z}) + \lambda \|\vec{f}_\mathbf{z}\|_{\mathcal{K}}^2) - (\text{Er}_\mathbf{z}(\vec{f}_\lambda) + \lambda \|\vec{f}_\lambda\|_{\mathcal{K}}^2)] + [\text{Er}(\vec{f}_\lambda) - \text{Er}(\vec{f}_\rho) + \lambda \|\vec{f}_\lambda\|_{\mathcal{K}}^2]$ . By the definition of  $\vec{f}_\mathbf{z}$ , we know that the second term in parenthesis on the right-hand side of the above equation is negative. Hence, by the definition of  $f_\lambda$ , we get that  $\text{Er}(\vec{f}_\mathbf{z}) - \text{Er}(\vec{f}_\rho) + \lambda \|\vec{f}_\mathbf{z}\|_{\mathcal{K}}^2 \leq \mathcal{S}(\mathbf{z}) + \inf_{f \in \mathcal{H}_\mathcal{K}} \{\text{Er}(\vec{f}) - \text{Er}(\vec{f}_\rho) + \lambda \|\vec{f}\|_{\mathcal{K}}^2\}$ . By the property (2) in Lemma 9, we also have, for any  $\vec{f} \in \mathcal{H}_\mathcal{K}$ , that  $\text{Er}(\vec{f}) - \text{Er}(\vec{f}_\rho) \leq c \left( (s^2 |\nabla f_\rho|_{\text{Lip}}^2 + \mathcal{Q}_s(\vec{f}, \vec{f}_\rho)) \right)$  which implies that

$$\inf \{ \text{Er}(\vec{f}) - \text{Er}(\vec{f}_\rho) + \lambda \|\vec{f}\|_{\mathcal{K}}^2 \} \leq c \left( s^2 |\nabla f_\rho|_{\text{Lip}}^2 + \mathcal{A}(\lambda, s) \right).$$

This completes the proposition.  $\blacksquare$

Now it suffice to estimate the sample error  $\mathcal{S}(\mathbf{z})$ . To this end, observe that  $\text{Er}_\mathbf{z}(\vec{f}_\mathbf{z}) + \lambda \|\vec{f}_\mathbf{z}\|_{\mathcal{K}}^2 \leq \text{Er}_\mathbf{z}(0) + \lambda \|0\|_{\mathcal{K}}^2 \leq \frac{M^2}{s^{d+2}}$  which implies that  $\|\vec{f}_\mathbf{z}\|_{\mathcal{K}} \leq Ms^{-(d+2)/2}$ . Likewise,  $\text{Er}(\vec{f}_\lambda) + \lambda \|\vec{f}_\lambda\|_{\mathcal{K}}^2 \leq \text{Er}(0) + \lambda \|0\|_{\mathcal{K}}^2 \|\vec{f}_\lambda\|_{\mathcal{K}} \leq \frac{M^2}{s^{d+2}}$  which tells us that  $\|\vec{f}_\lambda\|_{\mathcal{K}} \leq Ms^{-(d+2)/2}$ . Using these bounds on  $\|\vec{f}_\mathbf{z}\|_{\mathcal{K}}$  and  $\|\vec{f}_\lambda\|_{\mathcal{K}}$ , we can use the Rademacher averages (see e.g. [4, 12]) for its definition and properties) to get the following estimation for the sample error.

**Lemma 10** *For any  $0 < \lambda < 1$ , there exists a constant  $c$  such that, for any  $m \in \mathbb{N}$ , there holds*

$$\mathbb{E}[\mathcal{S}(\mathbf{z})] \leq c \left( \frac{1}{s^{2(d+2)} \lambda m} + \frac{1}{s^{3(d+2)/2} \sqrt{\lambda m}} \right).$$

Since the proof of the above lemma is rather a standard approach and indeed parallel to the proof of Lemma 26 (replacing  $r = Ms^{-(d+2)/2}$  there) in the appendix of [15], for the simplicity we omit the details here.

We have assembled all the materials to prove Theorem 7.

**Proof of Theorem 7**

Since we assume that  $\vec{f}_\rho \in \mathcal{H}_\mathcal{K}$ , for any  $|\partial_\rho f_\rho(x+u) - \partial_\rho \vec{f}_\rho(u)| = |\langle \tilde{e}_{p+1}, \vec{f}_\rho(x+u) - \vec{f}_\rho(u) \rangle| = |\langle \vec{f}_\rho, \mathcal{K}_{x+u} \tilde{e}_{p+1} - \mathcal{K}_u \tilde{e}_{p+1} \rangle_{\mathcal{K}}| \leq \|\vec{f}_\rho\|_{\mathcal{K}} \left[ \tilde{e}_{p+1}^T (\mathcal{K}(x+u, x+u) + \mathcal{K}(u, u) - \mathcal{K}(x+u, u) - \mathcal{K}(u, x+u)) \tilde{e}_{p+1} \right]^{\frac{1}{2}} \leq c \|\vec{f}_\rho\|_{\mathcal{K}} \|u\|$ , and hence  $|\nabla f_\rho|_{\text{Lip}} \leq c \|\vec{f}_\rho\|_{\mathcal{K}}$ . Moreover,

$$\mathcal{A}(\lambda, s) \leq \mathcal{Q}(\vec{f}_\rho, \vec{f}_\rho) + \lambda \|\vec{f}_\rho\|_{\mathcal{K}}^2 = \lambda \|\vec{f}_\rho\|_{\mathcal{K}}^2.$$

Hence, we know from Proposition 3 and equation (37) that  $\lambda \|\vec{f}_\mathbf{z}\|_{\mathcal{K}}^2 \leq \mathcal{S}(\mathbf{z}) + c'(s^2 + \lambda)$ .

Combining the above equations with Propositions 2 and 3, there exists a constant  $c$  such that

$$\begin{aligned} \mathbb{E}[\|\vec{f}_\mathbf{z} - \vec{f}_\rho\|_{\rho_X}^2] &\leq c \left( [\min(s^{-\theta}, \max_{x \in X} p^{-1}(x)) + \frac{s^\theta}{\lambda}] \right. \\ &\quad \left. \times [\mathbb{E}[\mathcal{S}(\mathbf{z})] + s^2 + \lambda] + s^\theta \right). \end{aligned}$$

If we choose  $\lambda = s^{2\theta}$  and  $s = m^{-\frac{1}{3(d+2)+4\theta}}$  yields the first assertion.

If  $\rho_X$  is the uniform distribution over  $X$ , then we have that  $\min(s^{-\theta}, \min_{x \in X} p(x)) = 1$ . Hence, choosing  $\lambda = s^\theta$  and  $s = m^{-\frac{1}{3(d+2)+5\theta}}$  we have the desired second assertion. This completes the theorem.

---

# Sparse Recovery in Large Ensembles of Kernel Machines

---

Vladimir Koltchinskii\*

School of Mathematics, Georgia Institute of Technology  
vlad@math.gatech.edu

Ming Yuan†

School of Industrial and Systems Engineering, Georgia Institute of Technology  
myuan@iyse.gatech.edu

## Abstract

A problem of learning a prediction rule that is approximated in a linear span of a large number of reproducing kernel Hilbert spaces is considered. The method is based on penalized empirical risk minimization with  $\ell_1$ -type complexity penalty. Oracle inequalities on excess risk of such estimators are proved showing that the method is adaptive to unknown degree of “sparsity” of the target function.

## 1 Introduction

Let  $(X, Y)$  be a random couple in  $S \times T$ , where  $(S, \mathcal{S}), (T, \mathcal{T})$  are measurable spaces. Usually,  $T$  is either a finite set, or a subset of  $\mathbb{R}$  (in the first case,  $T$  can be also identified with a finite subset of  $\mathbb{R}$ ). Most often,  $S$  is a compact domain in a finite dimensional Euclidean space, or a compact manifold. Let  $P$  denote the distribution of  $(X, Y)$  and  $\Pi$  denote the distribution of  $X$ . In a general framework of prediction,  $X$  is an observable instance and  $Y$  is an unobservable label which is to be predicted based on an observation of  $X$ . Let  $\ell : T \times \mathbb{R} \mapsto \mathbb{R}_+$  be a loss function. It will be assumed in what follows that, for all  $y \in T$ , the function  $\ell(y; \cdot)$  is convex. Given  $f : S \mapsto \mathbb{R}$ , denote

$$(\ell \bullet f)(x, y) := \ell(y, f(x))$$

and define the (true) risk of  $f$  as

$$\mathbb{E}\ell(Y; f(X)) = P(\ell \bullet f).$$

The prediction problem then can be formulated as convex risk minimization problem with the optimal prediction rule  $f_*$  defined as

$$f_* := \operatorname{argmin}_{f: S \mapsto \mathbb{R}} P(\ell \bullet f)$$

where the minimum is taken over all measurable functions  $f : S \mapsto \mathbb{R}$ . It will be assumed in what follows that

$f_*$  exists and it is uniformly bounded. We shall also assume the uniqueness of  $f_*$  in the following discussion.

In the case when the distribution  $P$  of  $(X, Y)$  is unknown, it has to be estimated based on the training data which (in the simplest case) consists of  $n$  independent copies  $(X_1, Y_1), \dots, (X_n, Y_n)$  of  $(X, Y)$ . Let  $P_n$  denote the empirical distribution based on the training data. Then the risk  $P(\ell \bullet f)$  can be estimated by the empirical risk

$$n^{-1} \sum_{j=1}^n \ell(Y_j, f(X_j)) = P_n(\ell \bullet f).$$

The direct minimization of the empirical risk over a large enough family of function  $f : S \mapsto \mathbb{R}$  almost inevitably leads to overfitting. To avoid it, a proper complexity regularization is needed. In this paper, we will study a problem in which the unknown target function  $f_*$  is being approximated in a linear span  $\mathcal{H}$  of a large dictionary consisting of  $N$  reproducing kernel Hilbert spaces (RKHS)  $\mathcal{H}_1, \dots, \mathcal{H}_N$ . It will be assumed that we are given  $N$  symmetric nonnegatively definite kernels  $K_j : S \times S \mapsto \mathbb{R}$ ,  $j = 1, \dots, N$  and that  $\mathcal{H}_j$  is the RKHS generated by  $K_j : \mathcal{H}_j = \mathcal{H}_{K_j}$ . Suppose, for simplicity, that

$$K_j(x, x) \leq 1, \quad x \in S, \quad j = 1, \dots, N.$$

The space

$$\mathcal{H} := \text{l.s.} \left( \bigcup_{j=1}^N \mathcal{H}_j \right)$$

consists of all functions  $f : S \mapsto \mathbb{R}$  that have the following (possibly, non-unique) additive representation

$$f = f_1 + \dots + f_N, \quad f_j \in \mathcal{H}_j, \quad f_j \in \mathcal{H}_j, \quad j = 1, \dots, N$$

and it can be naturally equipped with the  $\ell_1$ -norm:

$$\begin{aligned} \|f\|_{\ell_1} &:= \|f\|_{\ell_1(\mathcal{H})} := \inf \left\{ \sum_{j=1}^N \|f_j\|_{\mathcal{H}_j} : f \right. \\ &= \left. \sum_{j=1}^N f_j, f_j \in \mathcal{H}_j, j = 1, \dots, N \right\}. \end{aligned}$$

Additive models are a well-known special case of this formulation. In additive models,  $S$  is a subset of

---

\*Partially supported by NSF grant DMS-0624841.

†Partially supported by NSF grant DMS-0624841.

$\mathbb{R}^N$ , i.e.,  $X = (x_1, \dots, x_N)'$ , and  $\mathcal{H}_j$  represents a functional space defined over  $x_j$ . Several approaches have been proposed recently to exploit the sparsity in additive models (Lin and Zhang, 2006; Ravikumar et al., 2007; Yuan, 2007). In this paper, we consider an extension of  $\ell_1$  penalization technique to a more general class of problem.

In particular, we study the following penalized empirical risk minimization problem:

$$\hat{f}^\varepsilon := \operatorname{argmin}_{f \in \mathcal{H}} \left[ P_n(\ell \bullet f) + \varepsilon \|f\|_{\ell_1} \right], \quad (1.1)$$

where  $\varepsilon > 0$  is a small regularization parameter. Equivalently, this problem can be written as

$$(\hat{f}_1^\varepsilon, \dots, \hat{f}_N^\varepsilon) := \operatorname{argmin}_{f_j \in \mathcal{H}_j, j=1, \dots, N} \left[ P_n(\ell \bullet (f_1 + \dots + f_N)) + \varepsilon \sum_{j=1}^N \|f_j\|_{\mathcal{H}_j} \right]. \quad (1.2)$$

According to the representer theorem (Wahba, 1990), the components of the minimizer  $\hat{f}_j^\varepsilon$  have the following representation:

$$\hat{f}_j^\varepsilon(x) = \sum_{i=1}^n \hat{c}_{ij} K_j(X_i, x)$$

for some real vector  $\hat{c}_j = (\hat{c}_{ij} : i = 1, \dots, n)$ . In other words, (1.2) can be rewritten as a finite dimensional convex minimization problem over  $(c_{ij} : i = 1, \dots, n; j = 1, \dots, N)$ .

It is known (see, e.g., Micchelli and Pontil, 2005) that

$$\|f\|_{\ell_1(\mathcal{H})} = \inf \left\{ \|f\|_K : K \in \operatorname{conv}\{K_j : j = 1, \dots, N\} \right\},$$

where  $\|\cdot\|_K$  denote the RKHS-norm generated by symmetric nonnegatively definite kernel  $K$  and

$$\operatorname{conv}\{K_j : j = 1, \dots, N\} := \left\{ \sum_{j=1}^N c_j K_j : c_j \geq 0, \sum_{j=1}^N c_j = 1 \right\}.$$

Therefore (1.2) can be also written as

$$(\hat{f}^\varepsilon, \hat{K}^\varepsilon) := \operatorname{argmin}_{K \in \operatorname{conv}(K_j, j=1, \dots, N)} \operatorname{argmin}_{f \in \mathcal{H}_K} \left[ P_n(\ell \bullet f) + \varepsilon \|f\|_K \right], \quad (1.3)$$

leading to an interpretation of the problem as the one of learning not only the target function  $f_*$ , but also the kernel  $K$  in the convex hull of a given dictionary of kernels (which can be viewed as ‘‘aggregation’’ of kernel machines). Similar problems have been studied recently by Bousquet et al. (2003), Cramer et al. (2003), Lanckriet et al. (2004), Micchelli and Pontil (2005) and Srebro and Ben-David (2006) among others.

The choice of  $\ell_1$ -norm for complexity penalization is related to our interest in the case when the total number  $N$  of spaces  $\mathcal{H}_j$  in the dictionary is very large, but the target function  $f_*$  can be approximated reasonably well by functions from relatively small number  $d$  of such spaces. The  $\ell_1$ -penalization technique has been commonly used to recover sparse solutions in the case of simple dictionaries that consist of one-dimensional spaces  $\mathcal{H}_j$  (see, e.g. Koltchinskii (2007) and references therein). The goal is to extend this methodology to more general class of problems that include aggregation of large ensembles of kernel machines and sparse additive models. In the case of additive models with the quadratic loss, (1.1) becomes the so-called COSSO estimate recently introduced by Lin and Zhang (2006).

For  $f \in \mathcal{H}$ , define the excess risk of  $f$  as

$$\mathcal{E}(f) = P(\ell \bullet f) - P(\ell \bullet f_*) = P(\ell \bullet f) - \inf_{g: S \rightarrow \mathbb{R}} P(\ell \bullet g).$$

Our main goal is to control the excess risk of  $\hat{f}^\varepsilon$ ,  $\mathcal{E}(\hat{f}^\varepsilon)$ .

Throughout the paper, we shall also make the following assumption

$$n^\gamma \leq N \leq e^n$$

for some  $\gamma > 0$ .

It will also be assumed that the loss function  $\ell$  satisfies the following properties: for all  $y \in T$ ,  $\ell(y, \cdot)$  is twice differentiable,  $\ell''_u$  is a uniformly bounded function in  $T \times \mathbb{R}$ ,

$$\sup_{y \in T} \ell(y; 0) < +\infty, \quad \sup_{y \in T} \ell''_u(y; 0) < +\infty$$

and

$$\tau(R) := \frac{1}{2} \inf_{y \in T} \inf_{|u| \leq R} \ell''_u(y, u) > 0, \quad R > 0. \quad (1.4)$$

We also assume without loss of generality that, for all  $R$ ,  $\tau(R) \leq 1$ . These assumptions imply that

$$|\ell'_u(y, u)| \leq L_1 + L|u|, \quad y \in T, u \in \mathbb{R}$$

with some constants  $L_1, L \geq 0$  (if  $\ell'_u$  is uniformly bounded, one can take  $L = 0$ ).

The following bound on the excess risk holds under the assumptions on the loss:

$$\begin{aligned} & \tau(\|f\|_\infty \vee \|f_*\|_\infty) \|f - f_*\|_{L_2(\Pi)}^2 \\ & \leq \mathcal{E}(f) \leq C \|f - f_*\|_{L_2(\Pi)}^2 \end{aligned} \quad (1.5)$$

with a constant  $C > 0$  depending only on  $\ell$ . This bound easily follows from a simple argument based on Taylor expansion and it will be used later in the paper.

The quadratic loss  $\ell(y, u) := (y - u)^2$  in the case when  $T \subset \mathbb{R}$  is a bounded set is one of the main examples of such loss functions. In this case,  $\tau(R) = 1$  for all  $R$ . In regression problems with a bounded response variable, more general loss functions of the form  $\ell(y, u) := \phi(y - u)$  can be also used, where  $\phi$  is an even non-negative convex twice continuously differentiable function with  $\phi''$  uniformly bounded in  $\mathbb{R}$ ,  $\phi(0) = 0$  and

$\phi''(u) > 0$ ,  $u \in \mathbb{R}$ . In classification problems, the loss function of the form  $\ell(y, u) = \phi(yu)$  is commonly used, with  $\phi$  being a nonnegative decreasing convex twice continuously differentiable function such that, again,  $\phi''$  is uniformly bounded in  $\mathbb{R}$  and  $\phi''(u) > 0$ ,  $u \in \mathbb{R}$ . The loss function  $\phi(u) = \log_2(1 + e^{-u})$  (often referred to as the logit loss) is a specific example.

We will assume in what follows that  $\mathcal{H}$  is dense in  $L_2(\Pi)$ , which, together with (1.5), implies that

$$\inf_{f \in \mathcal{H}} P(\ell \bullet f) = \inf_{f \in L_2(\Pi)} P(\ell \bullet f) = P(\ell \bullet f_*).$$

We also need several basic facts about RKHS which can be found in, for example, Wahba (1990). Let  $K$  be a symmetric nonnegatively definite kernel on  $S \times S$  with

$$\sup_{x \in S} K(x, x) \leq 1$$

and  $\mathcal{H}_K$  be the corresponding RKHS. Given a probability measure  $\Pi$  on  $S$ , let  $\phi_k, k \geq 1$  be the orthonormal system of functions in  $L_2(\Pi)$  such that the following spectral representation (as in Mercer's theorem) holds:

$$K(x, y) = \sum_{k=1}^{\infty} \lambda_k \phi_k(x) \phi_k(y), \quad x, y \in S,$$

which is true under mild regularity conditions. Without loss of generality we can and do assume that  $\{\lambda_k\}$  is a decreasing sequence,  $\lambda_k \rightarrow 0$ . It is well known that for  $f, g \in \mathcal{H}_K$ ,

$$\langle f, g \rangle_{\mathcal{H}_K} = \sum_{k=1}^{\infty} \frac{\langle f, \phi_k \rangle_{L_2(\Pi)} \langle g, \phi_k \rangle_{L_2(\Pi)}}{\lambda_k}.$$

Denote  $H_D \subset \mathcal{H}_K$  the linear span of functions  $f \in \mathcal{H}_K$  such that

$$\sum_{k=1}^{\infty} \frac{\langle f, \phi_k \rangle_{L_2(\Pi)}^2}{\lambda_k^2} < \infty$$

and let  $D : H_D \mapsto L_2(\Pi)$  be a linear operator defined as follows:

$$Df := \sum_{k=1}^{\infty} \frac{\langle f, \phi_k \rangle_{L_2(\Pi)}}{\lambda_k} \phi_k, \quad f \in H_D.$$

Then we obviously have

$$\langle f, g \rangle_{\mathcal{H}_K} = \langle Df, g \rangle_{L_2(\Pi)}, \quad f \in H_D, g \in \mathcal{H}_K.$$

Given a dictionary  $\{\mathcal{H}_1, \dots, \mathcal{H}_N\}$  of RKHS, one can quite similarly define spectral representations of kernels  $K_j$  with nonincreasing sequences of eigenvalues  $\{\lambda_k^{(j)} : k \geq 1\}$  and orthonormal in  $L_2(\Pi)$  eigenfunctions  $\{\phi_k^{(j)} : k \geq 1\}$ . This also defines spaces  $H_{D_j}$  and linear operators  $D_j : H_{D_j} \mapsto L_2(\Pi)$  such that

$$\langle f, g \rangle_{\mathcal{H}_j} = \langle D_j f, g \rangle_{L_2(\Pi)}, \quad f \in H_{D_j}, g \in \mathcal{H}_{K_j}.$$

## 2 Bounding the $\ell_1$ -norm

Our first goal is to derive upper bounds on  $\|\hat{f}^\varepsilon\|_{\ell_1}$  that hold with a high probability. In what follows we use the notation

$$(\ell' \bullet f)(x, y) := \ell'_u(y, f(x)),$$

where  $\ell'_u(y, u)$  is the derivative of  $\ell$  with respect to the second variable.

**Theorem 1** *There exists a constant  $D > 0$  depending only on  $\ell$  such that for all  $A \geq 1$  and for all  $\varepsilon > 0$  and  $f \in \mathcal{H}$  satisfying the condition*

$$\varepsilon \geq D \|\ell' \bullet f\|_\infty \sqrt{\frac{A \log N}{n}} \bigvee 4 \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P((\ell' \bullet f)h_k)|,$$

the following bound holds

$$\mathbb{P} \left\{ \|\hat{f}^\varepsilon\|_{\ell_1} \geq 3\|f\|_{\ell_1} \right\} \leq N^{-A}. \quad (2.1)$$

In particular, if  $\varepsilon \geq D \|\ell' \bullet f^{\varepsilon/4}\|_\infty \sqrt{\frac{A \log N}{n}}$ , then

$$\mathbb{P} \left\{ \|\hat{f}^\varepsilon\|_{\ell_1} \geq 3\|f^{\varepsilon/4}\|_{\ell_1} \right\} \leq N^{-A}. \quad (2.2)$$

**Proof.** By the definition of  $\hat{f}^\varepsilon$ , for all  $f \in \mathcal{H}$ ,

$$P_n(\ell \bullet \hat{f}^\varepsilon) + \varepsilon \|\hat{f}^\varepsilon\|_{\ell_1} \leq P_n(\ell \bullet f) + \varepsilon \|f\|_{\ell_1}.$$

The convexity of the functional  $f \mapsto P_n(\ell \bullet f)$  implies that

$$P_n(\ell \bullet \hat{f}^\varepsilon) - P_n(\ell \bullet f) \geq P_n((\ell' \bullet f)(\hat{f}^\varepsilon - f)).$$

As a result,

$$\begin{aligned} \varepsilon \|\hat{f}^\varepsilon\|_{\ell_1} &\leq \varepsilon \|f\|_{\ell_1} + P_n((\ell' \bullet f)(f - \hat{f}^\varepsilon)) \\ &\leq \varepsilon \|f\|_{\ell_1} + \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P_n((\ell' \bullet f)h_k)| \times \\ &\quad \times \|\hat{f}^\varepsilon - f\|_{\ell_1}. \end{aligned}$$

It follows that

$$\begin{aligned} &\left( \varepsilon - \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P_n((\ell' \bullet f)h_k)| \right) \|\hat{f}^\varepsilon\|_{\ell_1} \\ &\leq \left( \varepsilon + \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P_n((\ell' \bullet f)h_k)| \right) \|f\|_{\ell_1}. \end{aligned}$$

Under the assumption

$$\varepsilon > \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P_n((\ell' \bullet f)h_k)|,$$

this yields

$$\|\hat{f}^\varepsilon\|_{\ell_1} \leq \frac{\varepsilon + \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P_n((\ell' \bullet f)h_k)|}{\varepsilon - \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P_n((\ell' \bullet f)h_k)|} \|f\|_{\ell_1}. \quad (2.3)$$

Note that

$$\begin{aligned} & \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P_n((\ell' \bullet f)h_k)| \\ \leq & \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |(P_n - P)(\ell' \bullet f)h_k| + \\ & + \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P((\ell' \bullet f)h_k)|. \end{aligned}$$

Also, for any  $i = 1, \dots, N$

$$\begin{aligned} & \sup_{\|h_i\|_{\mathcal{H}_i} \leq 1} |(P_n - P)(\ell' \bullet f)h_i| \\ = & \sup_{\|h_i\|_{\mathcal{H}_i} \leq 1} \left| n^{-1} \sum_{j=1}^n \left( (\ell' \bullet f)(X_j, Y_j) \langle h_i, K_i(X_j, \cdot) \rangle_{\mathcal{H}_i} \right. \right. \\ & \left. \left. - \mathbb{E}(\ell' \bullet f)(X_j, Y_j) \langle h_i, K_i(X_j, \cdot) \rangle_{\mathcal{H}_i} \right) \right| \\ = & \left\| n^{-1} \sum_{j=1}^n \left( (\ell' \bullet f)(X_j, Y_j) K_i(X_j, \cdot) \right. \right. \\ & \left. \left. - \mathbb{E}(\ell' \bullet f)(X_j, Y_j) K_i(X_j, \cdot) \right) \right\|_{\mathcal{H}_i}. \end{aligned}$$

Using Bernstein's type inequality in Hilbert spaces, we are easily getting the bound

$$\begin{aligned} & \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |(P_n - P)(\ell' \bullet f)h_k| \leq \\ & C \|\ell' \bullet f\|_{\infty} \left( \sqrt{\frac{A \log N}{n}} \vee \frac{A \log N}{n} \right) \end{aligned}$$

with probability at least  $1 - N^{-A}$ . As soon as

$$\varepsilon \geq 4C \|\ell' \bullet f\|_{\infty} \left( \sqrt{\frac{A \log N}{n}} \vee \frac{A \log N}{n} \right)$$

and

$$\varepsilon \geq 4 \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P((\ell' \bullet f)h_k)|,$$

we get

$$\max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P_n((\ell' \bullet f)h_k)| \leq \varepsilon/2,$$

and it follows from (2.3) that with probability at least  $1 - N^{-A}$

$$\|\hat{f}^{\varepsilon}\|_{\ell_1} \leq \frac{\varepsilon + \varepsilon/2}{\varepsilon - \varepsilon/2} \|f\|_{\ell_1} = 3\|f\|_{\ell_1},$$

implying (2.1).

In particular, we can use in (2.1)  $f := f^{\varepsilon/4}$ . Then, by the necessary conditions of extremum in the definition of  $f^{\varepsilon/4}$ ,

$$\max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P((\ell' \bullet f^{\varepsilon/4})h_k)| \leq \frac{\varepsilon}{4},$$

and the second bound follows. ■

We now provide an alternative set of conditions on  $\varepsilon$  so that (2.1) holds. By the conditions on the loss,

$$\|\ell' \bullet f\|_{\infty} \leq C(1 + L\|f\|_{\infty}) \leq C(1 + L\|f\|_{\ell_1})$$

with constants  $C, L$  depending only on  $\ell$  (if  $\ell'$  is uniformly bounded,  $L = 0$ ).

Since, by the necessary conditions of minimum at  $f_*$ ,

$$P((\ell' \bullet f_*)h_k) = 0, \quad h_k \in \mathcal{H}_k, k = 1, \dots, N,$$

we also have

$$\begin{aligned} & \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P((\ell' \bullet f)h_k)| \\ = & \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P((\ell' \bullet f) - (\ell' \bullet f_*)h_k)| \\ \leq & C\|f - f_*\|_{L_2(\Pi)} \end{aligned}$$

where we used the fact that  $\ell'_u(y, u)$  is Lipschitz with respect to  $u$ . Therefore, the condition on  $\varepsilon$  in (2.1) is satisfied if

$$\varepsilon \geq D(1 + \|f\|_{\ell_1}) \sqrt{\frac{A \log N}{n}}$$

and

$$\|f - f_*\|_{L_2(\Pi)} \leq \varepsilon/D$$

with a properly chosen  $D$  (depending only on  $\ell$ ).

### 3 Oracle inequalities

In what follows we will assume that  $R > 0$  is such that

$$\|\hat{f}^{\varepsilon}\|_{\ell_1} \leq R$$

with probability at least  $1 - N^{-A}$ . In particular, if  $\bar{f} \in \mathcal{H}$  satisfies the assumption of Theorem 1, i.e.,

$$\varepsilon \geq D\|\ell' \bullet \bar{f}\|_{\infty} \sqrt{\frac{A \log N}{n}} \vee 4 \max_{1 \leq k \leq N} \sup_{\|h_k\|_{\mathcal{H}_k} \leq 1} |P((\ell' \bullet \bar{f})h_k)|,$$

then one can take  $R = 3\|\bar{f}\|_{\ell_1}$ .

We need some measures of dependence (in a probabilistic sense) between the spaces  $\mathcal{H}_j, j = 1, \dots, N$ . In the case of a simple dictionary  $\{h_1, \dots, h_N\}$  consisting of  $N$  functions (equivalently,  $N$  one-dimensional spaces) the error of sparse recovery depends on the Gram matrix of the dictionary in the space  $L_2(\Pi)$  (see, e.g., Koltchinskii (2007)). A similar approach is taken here. Given  $h_j \in \mathcal{H}_j, j = 1, \dots, N$  and  $J \subset \{1, \dots, N\}$ , denote by  $\kappa(\{h_j : j \in J\})$  the minimal eigenvalue of the Gram matrix  $(\langle h_i, h_j \rangle_{L_2(\Pi)})_{i,j \in J}$  and  $\bar{\kappa}(\{h_j : j \in J\})$  its maximal eigenvalue. Let

$$\kappa(J) := \inf \left\{ \kappa(\{h_j : j \in J\}) : h_j \in \mathcal{H}_j, \|h_j\|_{L_2(\Pi)} = 1 \right\}$$

and

$$\bar{\kappa}(J) := \sup \left\{ \kappa(\{h_j : j \in J\}) : h_j \in \mathcal{H}_j, \|h_j\|_{L_2(\Pi)} = 1 \right\}$$

Also, denote  $L_J$  the linear span of subspaces  $\mathcal{H}_j, j \in J$ . Let

$$\begin{aligned} \rho(J) := \sup \left\{ \frac{\langle f, g \rangle_{L_2(\Pi)}}{\|f\|_{L_2(\Pi)} \|g\|_{L_2(\Pi)}} : f \in L_J, g \in L_{J^c}, \right. \\ \left. f \neq 0, g \neq 0 \right\}. \end{aligned}$$

In what follows, we will consider a set  $\mathcal{O} = \mathcal{O}(M_1, M_2)$  of functions (more precisely, their additive representations)  $f = f_1 + \dots + f_N \in \mathcal{H}$ ,  $f_j \in \mathcal{H}_j$ ,  $j = 1, \dots, N$  that will be called “admissible oracles”. Let  $J_f := \{j : f_j \neq 0\}$  and suppose the following assumptions hold:

**O1.** The “relevant” part  $J_f$  of the dictionary satisfies the condition

$$\frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1 - \rho^2(J_f))} \leq M_1.$$

**O2.** For some  $\beta > 1/2$  and for all  $j \in J_f$

$$\lambda_k^{(j)} \leq M_2 k^{-2\beta}, \quad k = 1, 2, \dots$$

Recall that  $D_j$  is the linear operator defined in the first section. Denote

$$\zeta(f) := \frac{1}{\text{card}(J_f)} \sum_{j \in J_f} \frac{\|D_j f_j\|_{L_2(\Pi)}^2}{\|f_j\|_{\mathcal{H}_j}^2}.$$

We are now in the position to state the main result of this paper.

**Theorem 2** *There exist constants  $D, L$  depending only on  $\ell$  ( $L = 0$  if  $\ell'_u$  is uniformly bounded) such that for all  $A \geq 1$ , for all  $f \in \mathcal{O}$  with  $\text{card}(J_f) = d$  and for all*

$$\varepsilon \geq D(1 + LR) \sqrt{\frac{\log N}{n}}$$

with probability at least  $1 - N^{-A}$

$$\begin{aligned} & \mathcal{E}(\hat{f}^\varepsilon) + 2\varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\ & \leq 7\mathcal{E}(f) + K \left[ \frac{d^{(2\beta-1)/(2\beta+1)}}{n^{2\beta/(2\beta+1)}} + \zeta(f)d\varepsilon^2 + \frac{A \log N}{n} \right], \end{aligned}$$

where  $K$  is a constant depending on  $\ell, R, M_1, M_2, \|f\|_\infty$  and  $\|f_*\|_\infty$ .

The meaning of this result can be described as follows. Suppose there exists an oracle  $f$  such that the excess risk of  $f$  is small (i.e.,  $f$  provides a good approximation of  $f_*$ ); the set  $J_f$  is small (i.e.,  $f$  has a sparse representation in the dictionary); the condition (O1) is satisfied, i.e. the relevant part of the dictionary is “well posed” in the sense that the spaces  $\mathcal{H}_j, j \in J_f$  are not “too dependent” among themselves and with the rest of the spaces in the dictionary; the condition (O2) is satisfied, which means “sufficient smoothness” of functions in the spaces  $\mathcal{H}_j, j \in J_f$ ; finally, the components  $f_j, j \in J_f$  of the oracle  $f$  are even smoother in the sense that the quantities  $\frac{\|D_j f_j\|_{L_2(\Pi)}}{\|f_j\|_{\mathcal{H}_j}}, j \in J_f$  are properly bounded. Then the excess risk of the empirical solution  $\hat{f}^\varepsilon$  is controlled by the excess risk of the oracle as well as by its degree of sparsity  $d$  and, at the same time,  $\hat{f}^\varepsilon$  is approximately sparse in the sense that

$\sum_{j \notin J_f} \|f_j^\varepsilon\|_{\mathcal{H}_j}$  is small. In other words, the solution obtained via  $\ell_1$ -penalized empirical risk minimization is adaptive to sparsity (at least, subject to constraints described above).

**Proof.** Throughout the proof we fix representations  $f = f_1 + \dots + f_N$  and  $\hat{f}^\varepsilon = \hat{f}_1^\varepsilon + \dots + \hat{f}_N^\varepsilon$  (and we use (1.2) to define  $\hat{f}_j^\varepsilon$ ). The definition of  $\hat{f}_j^\varepsilon$  implies that for all  $f \in \mathcal{H}$ ,

$$P_n(\ell \bullet \hat{f}^\varepsilon) + \varepsilon \|\hat{f}^\varepsilon\|_{\ell_1} \leq P_n(\ell \bullet f) + \varepsilon \|f\|_{\ell_1}.$$

Therefore,

$$\begin{aligned} & \mathcal{E}(\hat{f}^\varepsilon) + \varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\ & \leq \mathcal{E}(f) + \varepsilon \sum_{j \in J_f} (\|f_j\|_{\mathcal{H}_j} - \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j}) \\ & \quad + (P - P_n)(\ell \bullet f - \ell \bullet \hat{f}^\varepsilon). \end{aligned}$$

We first show that

$$\begin{aligned} & \mathcal{E}(\hat{f}^\varepsilon) + 2\varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\ & \leq 3\mathcal{E}(f) + \frac{2\zeta(f)d}{\tau \kappa(J_f)(1 - \rho^2(J_f))} \varepsilon^2 \\ & \quad + 2(P - P_n)(\ell \bullet f - \ell \bullet \hat{f}^\varepsilon), \end{aligned}$$

where

$$\tau = \tau(\|f\|_\infty \vee \|\hat{f}^\varepsilon\|_\infty \vee \|f_*\|_\infty).$$

Let  $s_j(f_j)$  be a subgradient of  $f_j \mapsto \|f_j\|_{\mathcal{H}_j}$  at  $f_j \in \mathcal{H}_j$ , i.e.  $s_j(f_j) = \frac{f_j}{\|f_j\|_{\mathcal{H}_j}}$  if  $f_j \neq 0$  and  $s_j(f_j)$  is an arbitrary vector with  $\|s_j(f_j)\|_{\mathcal{H}_j} \leq 1$  otherwise. Then we have

$$\begin{aligned} \|f_j\|_{\mathcal{H}_j} - \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} & \leq \langle s_j(f_j), f_j - \hat{f}_j^\varepsilon \rangle_{\mathcal{H}_j} \\ & = \langle D_j s_j(f_j), f_j - \hat{f}_j^\varepsilon \rangle_{L_2(\Pi)} \\ & \leq \|D_j s_j(f_j)\|_{L_2(\Pi)} \|f_j - \hat{f}_j^\varepsilon\|_{L_2(\Pi)}. \end{aligned}$$

It follows that

$$\begin{aligned} & \mathcal{E}(\hat{f}^\varepsilon) + \varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\ & \leq \mathcal{E}(f) + \varepsilon \left( \sum_{j \in J_f} \|D_j s_j(f_j)\|_{L_2(\Pi)}^2 \right)^{1/2} \times \\ & \quad \times \left( \sum_{j \in J_f} \|f_j - \hat{f}_j^\varepsilon\|_{L_2(\Pi)}^2 \right)^{1/2} \\ & \quad + (P - P_n)(\ell \bullet f - \ell \bullet \hat{f}^\varepsilon). \end{aligned}$$

It can also be shown that (see Koltchinskii, 2007, Proposition 1)

$$\begin{aligned} & \left( \sum_{j \in J_f} \|f_j - \hat{f}_j^\varepsilon\|_{L_2(\Pi)}^2 \right)^{1/2} \\ & \leq \sqrt{\frac{1}{\kappa(J_f)(1 - \rho^2(J_f))}} \|f - \hat{f}^\varepsilon\|_{L_2(\Pi)}. \end{aligned}$$

This allows us to write

$$\begin{aligned} & \mathcal{E}(\hat{f}^\varepsilon) + \varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\ & \leq \mathcal{E}(f) + \varepsilon \sqrt{\frac{\zeta(f)d}{\kappa(J_f)(1-\rho^2(J_f))}} \|f - \hat{f}^\varepsilon\|_{L_2(\Pi)} \\ & \quad + (P - P_n)(\ell \bullet f - \ell \bullet \hat{f}^\varepsilon). \end{aligned}$$

Then, using the bounds

$$\|f - \hat{f}^\varepsilon\|_{L_2(\Pi)} \leq \|f - f_*\|_{L_2(\Pi)} + \|\hat{f}^\varepsilon - f_*\|_{L_2(\Pi)}$$

and

$$\mathcal{E}(f) \geq \tau \|f - f_*\|_{L_2(\Pi)}^2, \quad \mathcal{E}(\hat{f}^\varepsilon) \geq \tau \|\hat{f}^\varepsilon - f_*\|_{L_2(\Pi)}^2,$$

we get

$$\begin{aligned} & \mathcal{E}(\hat{f}^\varepsilon) + \varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\ & \leq \mathcal{E}(f) + \varepsilon \sqrt{\frac{\zeta(f)d}{\kappa(J_f)(1-\rho^2(J_f))}} \times \\ & \quad \times \left( \sqrt{\frac{\mathcal{E}(f)}{\tau}} + \sqrt{\frac{\mathcal{E}(\hat{f}^\varepsilon)}{\tau}} \right) \\ & \quad + (P - P_n)(\ell \bullet f - \ell \bullet \hat{f}^\varepsilon). \end{aligned}$$

Applying the inequality  $ab \leq a^2/2 + b^2/2$ , we show that

$$\begin{aligned} & \varepsilon \sqrt{\frac{\zeta(f)d}{\kappa(J_f)(1-\rho^2(J_f))}} \sqrt{\frac{\mathcal{E}(f)}{\tau}} \\ & \leq \frac{\mathcal{E}(f)}{2} + \frac{\zeta(f)d}{2\tau\kappa(J_f)(1-\rho^2(J_f))} \varepsilon^2. \end{aligned}$$

Similarly,

$$\begin{aligned} & \varepsilon \sqrt{\frac{\zeta(f)d}{\kappa(J_f)(1-\rho^2(J_f))}} \sqrt{\frac{\mathcal{E}(\hat{f}^\varepsilon)}{\tau}} \\ & \leq \frac{\mathcal{E}(\hat{f}^\varepsilon)}{2} + \frac{\zeta(f)d}{2\tau\kappa(J_f)(1-\rho^2(J_f))} \varepsilon^2. \end{aligned}$$

This leads to the following bound

$$\begin{aligned} & \mathcal{E}(\hat{f}^\varepsilon) + \varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\ & \leq \mathcal{E}(f) + \frac{\mathcal{E}(\hat{f}^\varepsilon)}{2} + \frac{\zeta(f)d}{2\tau\kappa(J_f)(1-\rho^2(J_f))} \varepsilon^2 \\ & \quad + \frac{\mathcal{E}(f)}{2} + \frac{\zeta(f)d}{2\tau\kappa(J_f)(1-\rho^2(J_f))} \varepsilon^2 \\ & \quad + (P - P_n)(\ell \bullet f - \ell \bullet \hat{f}^\varepsilon). \end{aligned}$$

It easily follows that

$$\begin{aligned} & \mathcal{E}(\hat{f}^\varepsilon) + 2\varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\ & \leq 3\mathcal{E}(f) + \frac{2\zeta(f)d}{\tau\kappa(J_f)(1-\rho^2(J_f))} \varepsilon^2 \\ & \quad + 2(P - P_n)(\ell \bullet f - \ell \bullet \hat{f}^\varepsilon). \end{aligned}$$

Denote

$$\alpha_n(\delta, \Delta, R) := \sup \left\{ |(P_n - P)(\ell \bullet g - \ell \bullet f)| : \|g - f\|_{L_2(\Pi)} \leq \delta, \sum_{j \notin J_f} \|g_j\|_{\mathcal{H}_j} \leq \Delta, \|g\|_{\ell_1} \leq R \right\}.$$

If  $\|\hat{f}^\varepsilon\|_{\ell_1} \leq R$  (which holds with probability at least  $1 - N^{-A}$ ), then we have

$$\begin{aligned} & \mathcal{E}(\hat{f}^\varepsilon) + 2\varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\ & \leq 3\mathcal{E}(f) + \frac{2\zeta(f)d}{\tau\kappa(J_f)(1-\rho^2(J_f))} \varepsilon^2 \\ & \quad + 2\alpha_n \left( \|\hat{f}^\varepsilon - f\|_{L_2(\Pi)}, \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j}, R \right) \end{aligned}$$

with  $\tau = \tau(R \vee \|f\|_\infty \vee \|f_*\|_\infty)$ . We use Lemma 8 to get

$$\begin{aligned} & \mathcal{E}(\hat{f}^\varepsilon) + 2\varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\ & \leq 3\mathcal{E}(f) + \frac{2\zeta(f)d}{\tau\kappa(J_f)(1-\rho^2(J_f))} \varepsilon^2 \\ & \quad + C(1 + LR) \left[ \sqrt{\frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))}} \times \right. \\ & \quad \times \|\hat{f}^\varepsilon - f\|_{L_2(\Pi)} \sqrt{\frac{dm}{n}} + R \sqrt{\frac{\max_{j \in J_f} \sum_{k > m} \lambda_k^{(j)}}{n}} + \\ & \quad + R \sqrt{\frac{\max_{j \in J_f} \lambda_m^{(j)}}{n}} \sqrt{\frac{\log d}{n}} + \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \times \\ & \quad \times \sqrt{\frac{\log(N-d)+1}{n}} \left. \right] + C(1 + LR) \times \\ & \quad \times \|\hat{f}^\varepsilon - f\|_{L_2(\Pi)} \sqrt{\frac{A \log N}{n}} \\ & \quad + CR(1 + LR) \frac{A \log N}{n} \end{aligned} \tag{3.1}$$

(Lemma 8 can be used only under the assumption  $R \leq e^N$ ; however, for very large  $R > e^N$ , the proof of the inequality of the theorem is very simple). Recall that

$$\|\hat{f}^\varepsilon - f\|_{L_2(\Pi)} \leq \sqrt{\frac{\mathcal{E}(\hat{f}^\varepsilon)}{\tau}} + \sqrt{\frac{\mathcal{E}(f)}{\tau}}.$$

Under the assumption

$$\varepsilon \geq C(1 + LR) \sqrt{\frac{\log N}{n}},$$

we get

$$\begin{aligned}
& \mathcal{E}(\hat{f}^\varepsilon) + \varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\
\leq & 3\mathcal{E}(f) + 2 \frac{2\zeta(f)d}{\tau\kappa(J_f)(1-\rho^2(J_f))} \varepsilon^2 + \\
& C(1+LR) \left[ \sqrt{\frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))}} \times \right. \\
& \times \left( \sqrt{\frac{\mathcal{E}(\hat{f}^\varepsilon)}{\tau}} + \sqrt{\frac{\mathcal{E}(f)}{\tau}} \right) \sqrt{\frac{dm}{n}} + R \times \\
& \times \left. \sqrt{\frac{\max_{j \in J_f} \sum_{k>m} \lambda_k^{(j)}}{n}} + R \sqrt{\frac{\max_{j \in J_f} \lambda_m^{(j)}}{n}} \sqrt{\frac{\log d}{n}} \right] \\
& + C(1+LR) \left( \sqrt{\frac{\mathcal{E}(\hat{f}^\varepsilon)}{\tau}} + \sqrt{\frac{\mathcal{E}(f)}{\tau}} \right) \sqrt{\frac{A \log N}{n}} \\
& + CR(1+LR) \frac{A \log N}{n}. \tag{3.2}
\end{aligned}$$

Then we have

$$\begin{aligned}
& C(1+LR) \sqrt{\frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))}} \sqrt{\frac{\mathcal{E}(\hat{f}^\varepsilon)}{\tau}} \sqrt{\frac{dm}{n}} \\
\leq & \frac{1}{4} \mathcal{E}(\hat{f}^\varepsilon) + 2C^2(1+LR)^2 \frac{\bar{\kappa}(J_f)}{\tau\kappa(J_f)(1-\rho^2(J_f))} \frac{dm}{n}
\end{aligned}$$

and

$$\begin{aligned}
& C(1+LR) \sqrt{\frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))}} \sqrt{\frac{\mathcal{E}(f)}{\tau}} \sqrt{\frac{dm}{n}} \\
\leq & \frac{1}{4} \mathcal{E}(f) + 2C^2(1+LR)^2 \frac{\bar{\kappa}(J_f)}{\tau\kappa(J_f)(1-\rho^2(J_f))} \frac{dm}{n}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
& C(1+LR) \sqrt{\frac{\mathcal{E}(\hat{f}^\varepsilon)}{\tau}} \sqrt{\frac{A \log N}{n}} \\
\leq & \frac{1}{4} \mathcal{E}(\hat{f}^\varepsilon) + 2C^2(1+LR)^2 \frac{A \log N}{n}
\end{aligned}$$

and

$$\begin{aligned}
& C(1+LR) \sqrt{\frac{\mathcal{E}(f)}{\tau}} \sqrt{\frac{A \log N}{n}} \\
\leq & \frac{1}{4} \mathcal{E}(f) + 2C^2(1+LR)^2 \frac{A \log N}{n}.
\end{aligned}$$

This yields the following bound

$$\begin{aligned}
& \frac{1}{2} \mathcal{E}(\hat{f}^\varepsilon) + \varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\
\leq & \frac{7}{2} \mathcal{E}(f) + 2 \frac{2\zeta(f)d}{\tau\kappa(J_f)(1-\rho^2(J_f))} \varepsilon^2 + \\
& 4C^2(1+LR)^2 \left[ \frac{\bar{\kappa}(J_f)}{\tau\kappa(J_f)(1-\rho^2(J_f))} \frac{dm}{n} \right. \\
& + R \sqrt{\frac{\max_{j \in J_f} \sum_{k>m} \lambda_k^{(j)}}{n}} + \\
& R \sqrt{\frac{\max_{j \in J_f} \lambda_m^{(j)}}{n}} \sqrt{\frac{\log d}{n}} \left. \right] + 4C^2(1+LR)^2 \times \\
& \times \frac{A \log N}{n} + CR(1+LR) \frac{A \log N}{n}. \tag{3.3}
\end{aligned}$$

It remains to take

$$\begin{aligned}
m := & \frac{n^{1/(2\beta+1)}}{d^{2/(2\beta+1)}} \left( \frac{(1+LR)^2}{R\tau} \right)^{-2/(2\beta+1)} \times \\
& \times \left( \frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))} \right)^{-2/(2\beta+1)}
\end{aligned}$$

to get the following bound (with some constant  $C > 0$ )

$$\begin{aligned}
& \mathcal{E}(\hat{f}^\varepsilon) + 2\varepsilon \sum_{j \notin J_f} \|\hat{f}_j^\varepsilon\|_{\mathcal{H}_j} \\
\leq & 7\mathcal{E}(f) + 8 \frac{\zeta(f)d}{\tau\kappa(J_f)(1-\rho^2(J_f))} \varepsilon^2 + \\
& + C \left( \frac{(1+LR)^2}{\tau} \right)^{(2\beta-1)/(2\beta+1)} \times \\
& \times \left( \frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))} \right)^{(2\beta-1)/(2\beta+1)} \times \\
& \times \frac{d^{(2\beta-1)/(2\beta+1)}}{n^{2\beta/2\beta+1}} + \left( 4C^2(1+LR)^2 + \right. \\
& \left. + CR(1+LR) \right) \frac{A \log N}{n}, \tag{3.4}
\end{aligned}$$

which implies the result. ■

## 4 Appendix

The Rademacher process is defined as

$$R_n(g) := n^{-1} \sum_{j=1}^n \varepsilon_j g(X_j)$$

where  $\{\varepsilon_j\}$  are i.i.d. Rademacher random variables independent of  $\{X_j\}$ .

We will need several bounds for Rademacher processes indexed by functions from RKHS (some of them are well known; see, e.g., Mendelson (2002) and Blanchard, Bousquet and Massart (2007)). We state them without proofs for brevity.

First we consider a single RKHS  $\mathcal{H}_K$  where  $K$  is a kernel with eigenvalues  $\lambda_k$  and eigenfunctions  $\phi_k$  (in  $L_2(\Pi)$ ).

**Lemma 3** *The following bound holds:*

$$\mathbb{E} \sup_{\|h\|_{\mathcal{H}_K} \leq 1} |R_n(h)| \leq \sqrt{\frac{\sum_{k=1}^{\infty} \lambda_k}{n}}.$$

Let  $m \geq 1$ . Denote by  $L$  the linear span of the functions  $\{\phi_k : k = 1, \dots, m\}$  and by  $L^\perp$  the closed linear span (in  $L_2(\Pi)$ ) of the functions  $\{\phi_k : k \geq m+1\}$ .  $P_L, P_{L^\perp}$  will denote orthogonal projectors in  $L_2(\Pi)$  on the corresponding subspaces.

**Lemma 4** *For all  $m \geq 1$ ,*

$$\mathbb{E} \sup_{\|h\|_{\mathcal{H}_K} \leq 1} |R_n(P_{L^\perp} h)| \leq \sqrt{\frac{\sum_{k=m+1}^{\infty} \lambda_k}{n}}.$$

We now turn to the case of a dictionary  $\{\mathcal{H}_j : j = 1, \dots, N\}$  of RKHS with kernels  $\{K_j : j = 1, \dots, N\}$ . As before, denote  $\{\lambda_k^{(j)} : k \geq 1\}$  the eigenvalues (arranged in decreasing order) and  $\{\phi_k^{(j)} : k \geq 1\}$  the  $L_2(\Pi)$ -orthonormal eigenfunctions of  $K_j$ . The following bounds will be needed in this case.

**Lemma 5** *With some numerical constant  $C$ ,*

$$\mathbb{E} \max_{1 \leq j \leq N} \sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(h_j)| \leq \sqrt{\frac{\max_{1 \leq j \leq N} \sum_{k=1}^{\infty} \lambda_k^{(j)}}{n}} + C \sqrt{\frac{\log N}{n}}.$$

**Proof.** We use bounded difference inequality to get for each  $j = 1, \dots, N$  with probability at least  $1 - e^{-t - \log N}$

$$\sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(h_j)| \leq \mathbb{E} \sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(h_j)| + \frac{C\sqrt{t + \log N}}{\sqrt{n}}.$$

By the union bound, this yields with probability at least  $1 - Ne^{-t - \log N} = 1 - e^{-t}$

$$\max_{1 \leq j \leq N} \sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(h_j)| \leq \max_{1 \leq j \leq N} \mathbb{E} \sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(h_j)| + \frac{C\sqrt{t}}{\sqrt{n}} + \frac{C\sqrt{\log N}}{\sqrt{n}},$$

which holds for all  $t > 0$  and implies that

$$\mathbb{E} \max_{1 \leq j \leq N} \sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(h_j)| \leq \max_{1 \leq j \leq N} \mathbb{E} \sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(h_j)| + \frac{C\sqrt{\log N}}{\sqrt{n}}$$

with a properly chosen constant  $C > 0$ . Note that, by Lemma 3,

$$\mathbb{E} \sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(h_j)| \leq \sqrt{\frac{\sum_{k=1}^{\infty} \lambda_k^{(j)}}{n}},$$

which implies the result.  $\blacksquare$

As before, denote  $L_j, L_j^\perp$  the subspaces of  $L_2(\Pi)$  spanned on  $\{\phi_k^{(j)} : k \leq m\}$  and  $\{\phi_k^{(j)} : k > m\}$ , respectively,  $P_{L_j}, P_{L_j^\perp}$  being the corresponding orthogonal projections. Recall that sequence  $\{\lambda_k^{(j)}\}$  is nonincreasing. The following statement is a uniform version of Lemma 4.

**Lemma 6** *With some numerical constant  $C$ ,*

$$\begin{aligned} & \mathbb{E} \max_{1 \leq j \leq N} \sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(P_{L_j^\perp} h_j)| \\ & \leq 2 \sqrt{\frac{\max_{1 \leq j \leq N} \sum_{k=m+1}^{\infty} \lambda_k^{(j)}}{n}} \\ & \quad + 2 \sqrt{\frac{\max_{1 \leq j \leq N} \lambda_m^{(j)}}{n}} \sqrt{\frac{\log N + C}{n}} + 2 \frac{\log N + C}{n}. \end{aligned}$$

**Lemma 7** *The following bound holds:*

$$\begin{aligned} & \mathbb{E} \sup \left\{ |R_n(g - f)| : \|g - f\|_{L_2(\Pi)} \leq \delta, \|g\|_{\ell_1} \leq R, \right. \\ & \left. \sum_{j \notin J_f} \|g_j\|_{\mathcal{H}_j} \leq \Delta \right\} \leq C \sqrt{\frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1 - \rho^2(J_f))}} \delta \sqrt{\frac{dm}{n}} \\ & + 2R \sqrt{\frac{\max_{j \in J_f} \sum_{k>m} \lambda_k^{(j)}}{n}} + CR \sqrt{\frac{\max_{j \in J_f} \lambda_m^{(j)}}{n}} \sqrt{\frac{\log d}{n}} \\ & \quad + C\Delta \sqrt{\frac{\log(N - d) + 1}{n}}. \end{aligned}$$

**Proof.** First note that

$$\begin{aligned} & \mathbb{E} \sup \left\{ \left| R_n \left( \sum_{j=1}^N (g_j - f_j) \right) \right| : \|g - f\|_{L_2(\Pi)} \leq \delta, \|g\|_{\ell_1} \leq R, \right. \\ & \left. \sum_{j \notin J_f} \|g_j\|_{\mathcal{H}_j} \leq \Delta \right\} \leq \mathbb{E} \sup \left\{ \left| R_n \left( \left| \sum_{j \in J_f} (g_j - f_j) \right| \right) \right| : \right. \\ & \left. \|g - f\|_{L_2(\Pi)} \leq \delta, \|g\|_{\ell_1} \leq R, \sum_{j \in J_f} \|g_j\|_{\mathcal{H}_j} \leq \Delta \right\} + \\ & \mathbb{E} \sup \left\{ \left| R_n \left( \sum_{j \notin J_f} (g_j - f_j) \right) \right| : \|g - f\|_{L_2(\Pi)} \leq \delta, \|g\|_{\ell_1} \leq R, \right. \\ & \left. \sum_{j \notin J_f} \|g_j\|_{\mathcal{H}_j} \leq \Delta \right\}. \end{aligned}$$

The second term can be bounded as follows:

$$\begin{aligned} & \mathbb{E} \sup \left\{ \left| R_n \left( \sum_{j \notin J_f} g_j \right) \right| : \|g - f\|_{L_2(\Pi)} \leq \delta, \|g\|_{\ell_1} \leq R, \right. \\ & \left. \sum_{j \notin J_f} \|g_j\|_{\mathcal{H}_j} \leq \Delta \right\} \leq \mathbb{E} \sup \left\{ \left| R_n \left( \sum_{j \notin J_f} \|g_j\|_{\mathcal{H}_j} h_j \right) \right| : \right. \\ & \left. \sum_{j \notin J_f} \|g_j\|_{\mathcal{H}_j} \leq \Delta, \|h_j\|_{\mathcal{H}_j} \leq 1 \right\} \leq \\ & \Delta \mathbb{E} \max_{j \notin J_f} \sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(h_j)| \leq C \Delta \sqrt{\frac{\log(N-d)+1}{n}}, \end{aligned}$$

where we used Lemma 5. As to the first term, we use the bound

$$\begin{aligned} & \mathbb{E} \sup \left\{ \left| R_n \left( \sum_{j \in J_f} (g_j - f_j) \right) \right| : \|g - f\|_{L_2(\Pi)} \leq \delta, \|g\|_{\ell_1} \leq R, \right. \\ & \left. \sum_{j \in J_f} \|g_j\|_{\mathcal{H}_j} \leq \Delta \right\} \leq \mathbb{E} \sup \left\{ \left| R_n \left( \sum_{j \in J_f} P_{L_j} (g_j - f_j) \right) \right| : \right. \\ & \left. \|g - f\|_{L_2(\Pi)} \leq \delta \right\} \\ & + \mathbb{E} \sup \left\{ \left| R_n \left( \sum_{j \in J_f} P_{L_j^\perp} (g_j - f_j) \right) \right| : \|g\|_{\ell_1} \leq R \right\}. \end{aligned}$$

Note that

$$\begin{aligned} & \left\| \sum_{j \in J_f} P_{L_j} (g_j - f_j) \right\|_{L_2(\Pi)}^2 \leq \bar{\kappa}(J_f) \sum_{j \in J_f} \left\| P_{L_j} (g_j - f_j) \right\|_{L_2(\Pi)}^2 \\ & \leq \bar{\kappa}(J_f) \sum_{j \in J_f} \|g_j - f_j\|_{L_2(\Pi)}^2 \leq \frac{\bar{\kappa}(J_f)}{\kappa(J_f)} \left\| \sum_{j \in J_f} (g_j - f_j) \right\|_{L_2(\Pi)}^2 \\ & \leq \frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))} \left\| \sum_{j=1}^n (g_j - f_j) \right\|_{L_2(\Pi)}^2 \leq \\ & \frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))} \delta^2. \end{aligned}$$

Also,  $\sum_{j \in J_f} P_{L_j} (g_j - f_j)$  takes values in the linear span of  $\bigcup_{j \in J_f} L_j$  whose dimension  $\leq dm$ . This yields the following bound

$$\begin{aligned} & \mathbb{E} \sup \left\{ \left| R_n \left( \sum_{j \in J_f} P_{L_j} (g_j - f_j) \right) \right| : \|g - f\|_{L_2(\Pi)} \leq \delta \right\} \\ & \leq C \sqrt{\frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))}} \delta \sqrt{\frac{dm}{n}}. \end{aligned}$$

Finally, we use Lemma 6 to get

$$\begin{aligned} & \mathbb{E} \sup \left\{ \left| R_n \left( \sum_{j \in J_f} P_{L_j^\perp} (g_j - f_j) \right) \right| : \|g\|_{\ell_1} \leq R \right\} \\ & \leq \mathbb{E} \sup \left\{ \left| R_n \left( \sum_{j \in J_f} \|g_j - f_j\|_{\mathcal{H}_j} P_{L_j^\perp} h_j \right) \right| : \|g\|_{\ell_1} \leq R, \right. \\ & \left. \|h_j\|_{\mathcal{H}_j} \leq 1, j = 1, \dots, N \right\} \\ & \leq 2R \mathbb{E} \max_{j \in J_f} \sup_{\|h_j\|_{\mathcal{H}_j} \leq 1} |R_n(P_{L_j^\perp} h_j)| \\ & \leq 2R \sqrt{\frac{\max_{j \in J_f} \sum_{k>m} \lambda_k^{(j)}}{n}} + C \sqrt{\frac{\max_{j \in J_f} \lambda_m^{(j)}}{n}} \sqrt{\frac{\log d}{n}}. \end{aligned}$$

Combining the above bounds we get

$$\begin{aligned} & \mathbb{E} \sup \left\{ |R_n(g - f)| : \|g - f\|_{L_2(\Pi)} \leq \delta, \|g\|_{\ell_1} \leq R, \right. \\ & \left. \sum_{j \notin J_f} \|g_j\|_{\mathcal{H}_j} \leq \Delta \right\} \leq C \sqrt{\frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))}} \delta \sqrt{\frac{dm}{n}} \\ & + 2R \sqrt{\frac{\max_{j \in J_f} \sum_{k>m} \lambda_k^{(j)}}{n}} + CR \sqrt{\frac{\max_{j \in J_f} \lambda_m^{(j)}}{n}} \sqrt{\frac{\log d}{n}} \\ & + C \Delta \sqrt{\frac{\log(N-d)+1}{n}}. \blacksquare \end{aligned}$$

Recall that

$$\begin{aligned} & \alpha_n(\delta, \Delta, R) := \\ & \sup \left\{ |(P_n - P)(\ell \bullet g - \ell \bullet f)| : g \in \mathcal{G}(\delta, \Delta, R) \right\}, \end{aligned}$$

where

$$\begin{aligned} & \mathcal{G}(\delta, \Delta, R) := \\ & \left\{ g : \|g - f\|_{L_2(\Pi)} \leq \delta, \sum_{j \notin J_f} \|g_j\|_{\mathcal{H}_j} \leq \Delta, \|g\|_{\ell_1} \leq R \right\}. \end{aligned}$$

We will assume that  $R \leq e^N$  (recall also the assumption  $N \geq n^\gamma$ ).

**Lemma 8** *There exist constants  $C, L$  depending only on the loss  $\ell$  ( $L = 0$  if  $\ell'$  is bounded) such that for all*

$$n^{-1/2} \leq \delta \leq 2R, \quad n^{-1/2} \leq \Delta \leq R \quad (4.1)$$

*and for all  $A \geq 1$  the following bound holds with probability at least  $1 - N^{-A}$ :*

$$\begin{aligned} & \alpha_n(\delta, \Delta, R) \leq C(1 + LR) \left[ \sqrt{\frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1-\rho^2(J_f))}} \times \right. \\ & \left. \delta \sqrt{\frac{dm}{n}} + R \sqrt{\frac{\max_{j \in J_f} \sum_{k>m} \lambda_k^{(j)}}{n}} + \right. \\ & \left. R \sqrt{\frac{\max_{j \in J_f} \lambda_m^{(j)}}{n}} \sqrt{\frac{\log d}{n}} + \Delta \sqrt{\frac{\log(N-d)+1}{n}} \right] \\ & + C(1 + LR) \delta \sqrt{\frac{A \log N}{n}} + CR(1 + LR) \frac{A \log N}{n}. \quad (4.2) \end{aligned}$$

**Proof.** First note that, by Talagrand's concentration inequality, with probability at least  $1 - e^{-t}$

$$\alpha_n(\delta; \Delta; R) \leq 2 \left[ \mathbb{E} \alpha_n(\delta; \Delta, R) + C(1 + LR) \delta \sqrt{\frac{t}{n}} + \frac{CR(1 + LR)t}{n} \right].$$

To apply Talagrand's inequality we used the assumptions on the loss function. It follows from these assumptions that for all  $g \in \mathcal{G}(\delta, \Delta, R)$

$$\|\ell \bullet g - \ell \bullet f\|_{L_2(\Pi)} \leq C(1 + LR) \|g - f\|_{L_2(\Pi)} \leq C(1 + LR) \delta$$

and also

$$\|\ell \bullet g - \ell \bullet f\|_\infty \leq CR(1 + LR).$$

Next, by symmetrization inequality,

$$\mathbb{E} \alpha_n(\delta; \Delta, R) \leq 2 \mathbb{E} \sup \left\{ |R_n((\ell \bullet g - \ell \bullet f) : g \in \mathcal{G}(\delta, \Delta, R))|. \right\}.$$

We write  $u = g - f$  and

$$\ell \bullet g - \ell \bullet f = \ell \bullet (f + u) - \ell \bullet f$$

and observe that the function

$$[-R, R] \ni u \mapsto \ell \bullet (f + u) - \ell \bullet f$$

is Lipschitz with constant  $C(1 + LR)$ . This allows us to use Rademacher contraction inequality (Ledoux and Talagrand, 1991) to get

$$\mathbb{E} \alpha_n(\delta; \Delta, R) \leq C(1 + LR) \times \mathbb{E} \sup \left\{ \left| R_n(g - f) \right| : g \in \mathcal{G}(\delta, \Delta, R) \right\}.$$

The last expectation can be further bounded by Lemma 7. As a result, we get the following bound that holds with probability at least  $1 - e^{-t}$ :

$$\begin{aligned} \alpha_n(\delta; \Delta, R) &\leq C(1 + LR) \left[ \sqrt{\frac{\bar{\kappa}(J_f)}{\kappa(J_f)(1 - \rho^2(J_f))}} \delta \sqrt{\frac{dm}{n}} \right. \\ &+ R \sqrt{\frac{\max_{j \in J_f} \sum_{k > m} \lambda_k^{(j)}}{n}} + R \sqrt{\frac{\max_{j \in J_f} \lambda_m^{(j)}}{n}} \sqrt{\frac{\log d}{n}} + \\ &\quad \Delta \sqrt{\frac{\log(N - d) + 1}{n}} \left. \right] + C(1 + LR) \delta \sqrt{\frac{t}{n}} \\ &\quad + \frac{CR(1 + LR)t}{n} =: \tilde{\beta}_n(\delta, \Delta, R; t). \end{aligned} \quad (4.3)$$

The next goal is to make the bound uniform in

$$n^{-1/2} \leq \delta \leq 2R \quad \text{and} \quad n^{-1/2} \leq \Delta \leq R. \quad (4.4)$$

To this end, consider

$$\delta_j := 2R2^{-j}, \quad \Delta_j := R2^{-j}.$$

We will replace  $t$  by  $t + 2 \log \log(2R\sqrt{n})$  and use bound (4.3) for all  $\delta = \delta_j$  and  $\Delta = \Delta_k$  satisfying the conditions (4.4). By the union bound, with probability at least

$$\begin{aligned} 1 - \log(R\sqrt{n}) \log(2R\sqrt{n}) \exp \left\{ -t - 2 \log \log(2R\sqrt{n}) \right\} \\ \geq 1 - e^{-t}, \end{aligned}$$

the following bound holds for all  $\delta_j, \Delta_k$  satisfying (4.4):

$$\alpha_n(\delta_j, \Delta_k, R) \leq \tilde{\beta}_n \left( \delta_j, \Delta_k, R; t + 2 \log \log \left( \frac{2R}{\sqrt{n}} \right) \right).$$

It is enough now to substitute in the above bound  $t := A \log N$  and to use the fact that the functions  $\alpha_n(\delta, \Delta, R)$  and  $\tilde{\beta}_n(\delta, \Delta, R; t)$  are nondecreasing with respect to  $\delta$  and  $\Delta$ . Together with the conditions  $R \leq e^N$  and  $N \geq n^\gamma$ , this implies the claim. ■

## References

- [1] Bousquet, O. and Herrmann, D. (2003), On the complexity of learning the kernel matrix, In: *Advances in Neural Information Processing Systems 15*.
- [2] Blanchard, G., Bousquet, O. and Massart, P. (2008), Statistical performance of support vector machines, *Annals of Statistics*, **36**, 489-531.
- [3] Crammer, K., Keshet, J. and Singer, Y. (2003), Kernel design using boosting, In: *Advances in Neural Information Processing Systems 15*.
- [4] Koltchinskii, V. (2008), Sparsity in penalized empirical risk minimization, *Ann. Inst. H. Poincaré*, to appear.
- [5] Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. and Jordan, M. (2004), Learning the kernel matrix with semidefinite programming, *Journal of Machine Learning Research*, **5**, 27-72.
- [6] Ledoux, M. and Talagrand, M. (1991), *Probability in Banach Spaces*, Springer, New York.
- [7] Lin, Y. and Zhang, H. (2006), Component selection and smoothing in multivariate nonparametric regression, *Annals of Statistics*, **34**, 2272-2297.
- [8] Micchelli, C. and Pontil, M. (2005), Learning the kernel function via regularization, *Journal of Machine Learning Research*, **6**, 1099-1125.
- [9] Mendelson, S. (2002) Geometric parameters of kernel machines, In: *COLT 2002*, Lecture Notes in Artificial Intelligence, 2375, Springer, 29-43.
- [10] Ravikumar, P., Liu, H., Lafferty, J. and Wasserman, L. (2007), SpAM: sparse additive models, to appear in: *Advances in Neural Information Processing Systems (NIPS 07)*.
- [11] Srebro, N. and Ben-David, S. (2006), Learning bounds for support vector machines with learned kernels, In: *Proceedings of 19th Annual Conference on Learning Theory (COLT 2006)*, 169-183.
- [12] Wahba, G (1990), *Spline Models for Observational Data*, Philadelphia: SIAM.
- [13] Yuan, M. (2007), Nonnegative garrote component selection in functional ANOVA models, in *Proceedings of AI and Statistics (AISTAT 07)*, 656-662.

---

# More Efficient Internal-Regret-Minimizing Algorithms

---

Amy Greenwald, Zheng Li, and Warren Schudy  
Brown University, Providence, RI 02912  
{amy,ws}@cs.brown.edu and zheng@dam.brown.edu

## Abstract

Standard no-internal-regret (NIR) algorithms compute a fixed point of a matrix, and hence typically require  $O(n^3)$  run time per round of learning, where  $n$  is the dimensionality of the matrix. The main contribution of this paper is a novel NIR algorithm, which is a simple and straightforward variant of a standard NIR algorithm. However, rather than compute a fixed point every round, our algorithm relies on power iteration to estimate a fixed point, and hence runs in  $O(n^2)$  time per round.

Nonetheless, it is not enough to look only at the per-round run time of an online learning algorithm. One must also consider the algorithm's convergence rate. It turns out that the convergence rate of the aforementioned algorithm is slower than desired. This observation motivates our second contribution, which is an analysis of a multithreaded NIR algorithm that trades-off between its run time per round of learning and its convergence rate.

## 1 Introduction

An *online decision problem* (ODP) consists of a series of rounds, during each of which an agent chooses one of  $n$  pure actions and receives a reward corresponding to its choice. The agent's objective is to maximize its cumulative rewards. It can work towards this goal by abiding by an *online learning algorithm*, which prescribes a possibly mixed action (i.e., a probability distribution over the set of pure actions) to play each round, based on past actions and their corresponding rewards. The success of such an algorithm is typically measured in a worst-case fashion: specifically, an adversary chooses the sequence of rewards that the agent faces. Hence, the agent—the protagonist—*must* randomize its play; otherwise, it can easily be exploited by the adversary.

The observation that an ODP of this nature can be used to model a single player's perspective in a repeated game has spawned a growing literature connecting computational learning theory—specifically, the subarea of regret minimization—and game theory—specifically, the

subarea of learning in repeated games. Both groups of researchers are interested in designing algorithms by which an agent can learn from its past actions, and the rewards associated with those actions, to play actions now and in the future that yield high rewards. More specifically, the entire sequence of actions should yield low regret for not having played otherwise, or equivalently, near equilibrium behavior.

In a seminal paper by Foster and Vohra [FV97], it was established that the empirical distribution of the joint play of a particular class of online learning algorithms, called no-internal-regret (NIR) learners, converges to the set of correlated equilibria in repeated matrix games. However, standard NIR learning algorithms (see Cesa-Bianchi and Lugosi [CBL06] and Blum and Mansour [BM05])<sup>1</sup>—including the algorithm proposed by Foster and Vohra (hereafter, FV)—involve a fixed point calculation during each round of learning, an operation that is cubic<sup>2</sup> in the number of pure actions available to the player. Knowing that fixed point calculations are expensive, Hart and Mas-Colell [HMC00] describe “a simple adaptive procedure” (hereafter, HM) that also achieves the aforementioned convergence result. HM's per-round run time is linear in the number of pure actions.

It is well-known [HMC00] that HM does not exhibit no internal regret in the usual sense, meaning against an *adaptive* adversary—one that can adapt in response to the protagonist's “realized” pure actions (i.e., those that result from sampling his mixed actions). Still, in a recent paper, Cahn [Cah04] has shown that HM's algorithm does exhibit no internal regret against an adversary that is “not too sophisticated.” In this paper, we use the terminology *nearly oblivious* to refer to this

---

<sup>1</sup>The former reference is to a book that surveys the field; the latter reference is to a paper that includes a black-box method for constructing NIR learners from another class of learners called no-external-regret learners.

<sup>2</sup>Strassen [Str69] devised an  $O(n^{2.81})$ -time algorithm for matrix-matrix multiplication, based on which a fixed point can be computed with the same run time [CLRS01]. Copper-Smith and Winograd [CW87] devised an  $O(n^{2.36})$ -time algorithm for matrix-matrix multiplication, but unlike Strassen's result their result is impractical. For better pedagogy, we quote the “natural”  $O(n^3)$  runtime in most of our discussions rather than these better bounds.

type of adversary, because the “not-too-sophisticated” condition is a weakening of the usual notion of an *oblivious* adversary—one who chooses the sequence of rewards after the protagonist chooses its online learning algorithm, but before the protagonist realizes any of its pure actions. Since an oblivious adversary is also nearly oblivious, Cahn’s result implies that HM exhibits no internal regret against an oblivious adversary.

As alluded to above, both FV and HM (and all the algorithms studied in this paper) learn a mixed action each round, and then play a pure action: i.e., a sample from that mixed action. One important difference between them, however, which can be viewed at least as a partial explanation of their varying strengths, is that FV maintains as its state the mixed action it learns, whereas HM maintains as its state the pure action it plays. Intuitively, the latter cannot exhibit no internal regret against an adaptive adversary because an adaptive adversary can exploit any dependencies between the consecutively sampled pure actions.

Young [You04] proposes, but does not analyze rigorously, a variant of HM he calls Incremental Conditional Regret Matching (ICRM), which keeps track of a mixed action instead of a pure action, and hence exhibits no internal regret against an adaptive adversary.<sup>3</sup> ICRM has quadratic run time each round. To motivate ICRM, recall that standard NIR algorithms involve a fixed-point calculation. Specifically, they rely on solutions to equations of the form  $q = qP_t$ , where  $P_t$  is a stochastic matrix that encodes the learner’s regrets for its actions through time  $t$ . Rather than solve this equation exactly, ICRM takes  $q_{t+1} \leftarrow q_t P_t$  as an iterative approximation of the desired fixed point.

The regret matrix  $P_t$  used in ICRM (and HM) depends on a parameter  $\mu$  that is strictly larger than the maximum regret per round. This makes ICRM less intuitive than it could be. We show that the same idea also works when the normalizing factor  $\mu t$  is replaced by the actual total regret experienced by the learner. This simplifies the algorithm and eliminates the need for the learner to know or estimate a bound on the rewards. We call our algorithm Power Iteration (PI),<sup>4</sup> because another more intuitive way to view it is as a modification of a standard NIR algorithm (e.g., Greenwald, *et al.* [GJMar]) with its fixed-point calculation replaced by power iteration. Once again, the first (and primary) contribution of this paper is a proof that *using power iteration to estimate a fixed point, which costs only  $O(n^2)$  per round, suffices to achieve no-internal-regret against an adaptive adversary.*

Although our PI algorithm is intuitive, the proof that the idea pans out—that PI exhibits NIR against an adaptive adversary—is non-trivial (which may be why

<sup>3</sup>Our analytical tools can be used to establish Young’s claim rigorously.

<sup>4</sup>Both PI and ICRM can be construed as both incremental conditional regret matching algorithms and as power iteration methods. The difference between these algorithms is merely the definition of the matrix  $P_t$ , and who named them, not what they are named for per se.

Young did not propose this algorithm in the first place). The proof in Hart and Mas-Colell [HMC00] relies on a technical lemma, which states that  $\|q_t P_t^z - q_t P_t^{z-1}\|_1$ , for some  $z > 0$ , is small, whenever all the entries on the main diagonal of  $P_t$  are at least some uniform constant. With our new definition of  $P_t$ , this condition does not hold. Instead, our result relies on a generalization of this lemma in which we pose weaker conditions that guarantee the same conclusion. Specifically, we require only that the trace of  $P_t$  be at least  $n - 1$ . Our lemma may be of independent interest.

Hence, we have succeeded at defining a simple and intuitive,  $O(n^2)$  per-round online learning algorithm that achieves no internal regret against an adaptive adversary. However, it is not enough to look only at the per-round run time of an online learning algorithm. One must also consider the algorithm’s convergence rate. It turns out that the convergence rates of PI, ICRM, and HM are all slower than desired (their regret bounds are  $O(\sqrt{nt}^{-1/10})$ ), whereas FV’s regret bound is  $O(\sqrt{n/t})$  (see, for example, Greenwald, *et al.* [GLM06]). This observation motivates our second algorithm.

As our second contribution, we analyze an alternative algorithm, one which is multithreaded. Again, the basic idea is straightforward: one thread plays the game, taking as its mixed action the most-recently computed fixed point, while the other thread computes a new fixed point. Whenever a new fixed point becomes available, the first thread updates its mixed action accordingly. This second algorithm, which we call MT, for multithreaded, exhibits a trade-off between its run time per round and its convergence rate. If  $p$  is an upper bound on the number of rounds it takes to compute a fixed point, MT’s regret is bounded by  $O(\sqrt{np/t})$ . Observe that this regret bound is a function of  $t/p$ , the number of fixed points computed so far. If  $p$  is small, so that many fixed points have been computed so far, then the run time per round is high, but the regret is low; on the other hand, if  $p$  is large, so that only very few fixed points have been computed so far, then the run time per round is low, but the regret is high.

This paper is organized as follows. In Section 2, we define online decision problems and no-regret learning precisely. In Section 3, we define the HM, ICRM, and PI algorithms, and report their regret bounds. In Section 4, we introduce our second algorithm, MT, and report its regret bound. In Section 5, we prove a straightforward lemma that we use in the analysis of all algorithms. In Section 6, we analyze MT. In Section 7, we analyze PI. In Section 8, we present some preliminary simulation experiments involving PI, HM, and MT. In Section 9, we describe some interesting future directions.

## 2 Formalism

An online decision problem (ODP) is parameterized by a *reward system*  $(A, \mathcal{R})$ , where  $A$  is a set of pure actions and  $\mathcal{R}$  is a set of rewards. Given a reward system  $(A, \mathcal{R})$ , we let  $\Pi \equiv \mathcal{R}^A$  denote the set of possible reward vectors.

**Definition 1** Given a reward system  $(A, \mathcal{R})$ , an online decision problem can be described by a sequence of reward functions  $\langle \tilde{\pi}_t \rangle_{t=1}^\infty$ , where  $\tilde{\pi}_t \in (A^{t-1} \mapsto \Pi)$ .

Given an ODP  $\langle \tilde{\pi}_t \rangle_{t=1}^\infty$ , the particular history  $H_t = (\langle a_\tau \rangle_{\tau=1}^t, \langle \pi_\tau \rangle_{\tau=1}^t)$  corresponds to the agent playing  $a_\tau$  and observing reward vector  $\pi_\tau \equiv \tilde{\pi}_\tau(a_1, \dots, a_{\tau-1})$  at all times  $\tau = 1, \dots, t$ .

In this paper, we restrict our attention to bounded, real-valued reward systems; as such, we assume WLOG that  $\mathcal{R} = [0, 1]$ . We also assume the agent’s pure action set is finite; specifically, we let  $|A| = n$ . Still, we allow agents to play mixed actions. That is, an agent can play a probability distribution over its pure actions. We denote by  $\Delta(A)$  the set of mixed actions: i.e., the set of all probability distributions over  $A$ .

An *online learning algorithm* is a sequence of functions  $\langle \hat{q}_t \rangle_{t=1}^\infty$ , where  $\hat{q}_t : H_{t-1} \rightarrow \Delta(A)$  so that  $\hat{q}_t(h) \in \Delta(A)$  represents the agent’s mixed action at time  $t \geq 1$ , after having observed history  $h \in H_{t-1}$ . When the history  $h$  is clear from context, we abbreviate  $\hat{q}_t(h)$  by  $q_t$ . For a given history of length  $t$ , let  $\hat{q}_t$  be the degenerate probability distribution corresponding to the action actually played at time  $t$ : i.e., for all  $1 \leq i \leq n$ ,  $(\hat{q}_t)_i = \mathbb{1}(a_t = i)$ .<sup>5</sup> Clearly,  $\hat{q}_t$  is a random variable.

We are interested in measuring an agent’s regret in an ODP for playing as prescribed by some online learning algorithm rather than playing otherwise. We parameterize this notion of “otherwise” by considering a variety of other ways that the agent could have played. For example, it could have played any single action  $a$  all the time; or, it could have played  $a'$  every time it actually played  $a$ . In either case, we arrive at an alternative sequence of play by applying some transformation to each action in the agent’s actual sequence of play, and then we measure the difference in rewards obtained by the two sequences, in the worst case. That is the agent’s regret.

A transformation of the sort used in the first example above—a constant transformation that maps every action  $a'$  in the actual sequence of play to a fixed, alternative action  $a$ —is called an *external* transformation. We denote by  $\Phi_{\text{EXT}}$  the set of all external transformations, one per action  $a \in A$ . Many efficient algorithms, with both fast run time per round and fast convergence rates, are known to minimize regret with respect to  $\Phi_{\text{EXT}}$  (e.g., [LW94, FS97, HMC01]). Here, we are interested in transformations of the second type, which are called *internal* transformations. These transformations can be described by the following set of  $n$ -dimensional matrices:

$$\Phi_{\text{INT}} = \{ \phi^{(a,b)} : a \neq b, 1 \leq a, b \leq n \}$$

where

$$(\phi^{(a,b)})_{ij} = \begin{cases} 1 & \text{if } i \neq a \wedge i = j \\ 1 & \text{if } i = a \wedge j = b \\ 0 & \text{otherwise} \end{cases}$$

<sup>5</sup>For predicate  $p$ ,  $\mathbb{1}(p) = \begin{cases} 1 & \text{if } p \\ 0 & \text{otherwise} \end{cases}$ .

For example, if  $|A| = 4$ , then applying the following transformation to a pure action  $a$  yields the third action if  $a$  is the second action, and  $a$  otherwise:

$$\phi^{(2,3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\Phi_{\text{EXT}}$  and  $\Phi_{\text{INT}}$  are the two best-known examples of transformation sets. More generally, a transformation  $\phi$  can be any linear function from  $\Delta(A) \rightarrow \Delta(A)$ . In the definitions that follow, we express reward vectors  $\pi$  as column vectors, mixed actions  $q$  as row vectors, and transformations  $\phi$  as  $n$ -dimensional matrices.

If, at time  $\tau$ , an agent plays mixed action  $q_\tau$  in an ODP with reward vector  $\pi_\tau$ , the agent’s *instantaneous regret*  $(r_\tau)_\phi$  with respect to a transformation  $\phi$  is the difference between the rewards it could have obtained by playing  $q_\tau \phi$  and the rewards it actually obtained by playing  $q_\tau$ : i.e.,

$$(r_\tau)_\phi = q_\tau \phi \pi_\tau - q_\tau \pi_\tau \quad (1)$$

The agent’s *cumulative regret vector*  $(R_t)$  through time  $t$  is then computed in the obvious way: for  $\phi \in \Phi$ ,

$$(R_t)_\phi = \sum_{\tau=1}^t (r_\tau)_\phi \quad (2)$$

One can also define *pure action* variants of the instantaneous and cumulative regret vectors, as follows:

$$(\hat{r}_\tau)_\phi = \hat{q}_\tau \phi \pi_\tau - \hat{q}_\tau \pi_\tau \quad (3)$$

and

$$(\hat{R}_t)_\phi = \sum_{\tau=1}^t (\hat{r}_\tau)_\phi \quad (4)$$

One can bound either the expected pure action regret or the (mixed action) regret. To avoid unilluminating complications, we focus on the latter in this work.

Our objective in this work is to establish sublinear bounds on the average internal-regret vector of various online learning algorithms. Equipped with such bounds, we can then go on to claim that our algorithms exhibit no internal regret by applying standard techniques such as the Hoeffding-Azuma lemma (see, for example, Cesa-Bianchi and Lugosi [CBL06]). Note that we cannot establish our results for general  $\Phi$ . We defer further discussion of this point until Section 9, where we provide a simple counterexample.

For completeness, here is the formal definition of no- $\Phi$ -regret learning:

**Definition 2** Given a finite set of transformations  $\Phi$ , an online learning algorithm  $\langle \hat{q}_t \rangle_{t=1}^\infty$  is said to exhibit **no- $\Phi$ -regret** if for all  $\epsilon > 0$  there exists  $t_0 \geq 0$  such that for any ODP  $\langle \tilde{\pi}_t \rangle_{t=1}^\infty$ ,

$$\Pr \left[ \exists t > t_0 \text{ s.t. } \max_{\phi \in \Phi} \frac{1}{t} \hat{R}_t^\phi \geq \epsilon \right] < \epsilon \quad (5)$$

The relevant probability space in the above definition is the natural one that arises when considering a particular ODP  $\langle \tilde{\pi}_t \rangle_{t=1}^\infty$  together with an online learning algorithm  $\langle \hat{q}_t \rangle_{t=1}^\infty$ . The universe consists of infinite sequences of pure actions  $\langle a_\tau \rangle_{\tau=1}^\infty$  and the measure is defined by the learning algorithm.

We close this section with some notation that appears in later sections:

- We let  $a \bullet b = a^T b$  denote the dot product of column vectors  $a$  and  $b$ .
- For vector  $v \in \mathbb{R}^n$ , we let  $v^+$  denote the component-wise max of  $v$  and the zero vector: i.e.,  $(v^+)_i = \max(v_i, 0)$ .

### 3 Algorithms

We begin this section by describing HM, the simple adaptive procedure due to Hart and Mas-Colell [HMC00] that exhibits no internal regret against a nearly oblivious adversary, as well as ICRM, a variant of HM due to Young [You04] that exhibits no internal regret against an adaptive adversary. We then go on to present a simple variant of these algorithms, which we call PI, for power iteration, for which we establish the stronger of these two guarantees.

**Definition 3** Define the  $n$ -dimensional matrix

$$N_t = \sum_{\phi \in \Phi_{INT}} (R_t^+)_{\phi} \phi$$

and the scalar

$$D_t = \sum_{\phi \in \Phi_{INT}} (R_t^+)_{\phi}$$

At a high-level, HM (Algorithm 1) and ICRM (not shown) operate in much the same way: at each time step  $t$ , an action is played and a reward is earned; then, the regret matrix  $P_t$  is computed in terms of  $N_t$  and  $D_t$ , based on which a new action is derived. But the algorithms differ in an important way: specifically, they differ in their “state” (i.e., what they store from one round to the next). In HM, the state is a pure action, so that during each round, the next pure action is computed based on the current pure action. In ICRM, the state is a mixed action.

Like Young’s algorithm, the state in our algorithm, PI (Algorithm 2), is a mixed action. But, our algorithm differs from both of the others in our choice of the matrix regret  $P_t$ . In PI,  $P_t = N_t/D_t$ , which is the same matrix as in Greenwald *et al.* [GJMar], for example. Intuitively,  $N_t/D_t$  is a convex combination of the transformations in  $\Phi_{INT}$ , with each  $\phi \in \Phi_{INT}$  weighted by the amount of regret the learner experienced for not having transformed its play as prescribed. In HM and ICRM,  $P_t$  is a convex combination of  $N_t/D_t$  and the identity matrix. This convex combination depends on a parameter  $\mu$ , which is an upper bound on the regret per round; typically,  $\mu = 2n$ .

---

#### Algorithm 1 HM [HMC00]

---

Initialize  $a_1$  to be an arbitrary pure action.

During each round  $t = 1, 2, 3, \dots$ :

1. Play the pure action  $a_t$ .
  2. For all  $j$ ,  
let  $(\hat{q}_t)_j = \mathbb{1}(a_t = j)$ .
  3. Observe rewards  $\pi_t$ .
  4. Update the regret vector  $\hat{R}_t$ .
  5. Let the regret matrix  $\hat{P}_t = \frac{\hat{N}_t + (\mu t - \hat{D}_t)I}{\mu t}$ .
  6. Sample a pure action  $a_{t+1}$  from  $\hat{q}_t \hat{P}_t$ .
- 

---

#### Algorithm 2 Power Iteration

---

Initialize  $q_1$  to be an arbitrary mixed action.

During each round  $t = 1, 2, 3, \dots$ :

1. Sample a pure action  $a_t$  from  $q_t$ .
  2. Play the pure action  $a_t$ .
  3. Observe rewards  $\pi_t$ .
  4. Update the regret vector  $R_t$ .
  5. Define the regret matrix  $P_t = \frac{N_t}{D_t}$ .
  6. Set the mixed action  $q_{t+1} \leftarrow q_t P_t$ .
- 

HM has a per-round run time linear in the number of pure actions because it updates one row of the regret matrix during each round, namely that row corresponding to the action played. ICRM and PI both have per-round run times dominated by the matrix-vector multiplication in Step 6, and are hence quadratic in the number of pure actions.

We analyze PI (Algorithm 2) in this paper, and obtain the following result:

**Theorem 4** *PI (Algorithm 2) exhibits no internal regret against an adaptive adversary. Specifically, the bound on its average regret is as follows: for all times  $t$ ,*

$$\left\| \frac{R_t^+}{t} \right\|_{\infty} \leq O(\sqrt{nt}^{-1/10})$$

A slight variant of our analysis shows that ICRM has the same bound as PI. Algorithm 1 was previously analyzed by Cahn [Cah04], who showed that if the adversary is nearly oblivious, then HM exhibits NIR. One can combine ideas from Hart and Mas-Colell [HMC00] and our analysis of PI to show that against an oblivious adversary, the bound on HM’s average regret is as

follows: for all times  $t$ ,

$$\mathbb{E} \left[ \left\| \frac{\hat{R}_t^+}{t} \right\|_\infty \right] \leq O(\sqrt{nt}^{-1/10})$$

Theorem 4 implies, by the Proposition at the bottom of page 1133 in Hart and Mas-Colell [HMC00], that like HM and ICRM, PI also converges to the set of correlated equilibria in self-play.

Note that it is also possible to define a variant of PI with  $P_t = N_t/D_t$ , which like HM, uses the agent's pure action as its state. We conjecture that this algorithm would exhibit no internal regret against an oblivious (or nearly oblivious) adversary, but do not analyze it because it has no obvious advantages over HM or PI.

## 4 Multithreaded algorithm

The ICRM and PI algorithms have better per-round run times than standard NIR learning algorithms, but their convergence rates are far worse. Moreover, these algorithms are inflexible: they cannot expend additional effort during a round to improve their convergence rates. In this section, we present a parameterized, multithreaded algorithm (MT) that smoothly trades off between per-round run time and regret.

The idea underlying MT is simply to spread the computation of a fixed point over many time steps, and in the mean time to play the most recent fixed point computed so far. This idea is formalized in Algorithm 3, in which there are two threads. One thread plays the game, taking as its mixed action the most-recently computed fixed point; the other thread works towards computing a new fixed point.

**Theorem 5** *Let  $p \geq 1$  be an upper bound on how many time steps it takes to compute a fixed point. MT (Algorithm 3) has per-round run time  $O(LS(n)/p + \log n + \rho)$  and regret bound  $O(\sqrt{np}/t)$ , where  $LS(n)$  required solve a linear system of equations expressed as an  $n$ -dimensional matrix and  $\rho$  is the usually negligible run time required to maintain the regret vector (see Section 6). More precisely, for all times  $t$ ,*

$$\left\| \frac{R_t^+}{t} \right\|_\infty \leq \sqrt{\frac{(n-1)(4p-3)}{t}}$$

Suppose you are playing a game every minute and you have just barely enough computational resources to find a fixed point in the time allotted with a standard NIR learning algorithm ( $p = 1$ ). Further, suppose that it takes 1 day of playing for your regret to fall below a desired threshold. Now, suppose the game changes and you now have to make a move every second. If you set  $p = 60$ , and continue to compute 1 fixed point per minute, this will require  $4 \cdot 60 - 3 \approx 4 \cdot 60$  times more rounds to achieve the same level of regret. But each round is 60 times faster, so the wall-clock time for the same level of regret has increased by a factor of about 4, to a bit under 4 days.

With one extreme parameter setting, namely  $p = 1$ , MT is just like a standard NIR learning algorithm, and

---

**Algorithm 3** Multithreaded no-internal-regret learning algorithm.

---

Initialize  $R, N, D$  to zero.

**First thread:** During each round  $t = 1, 2, 3, \dots$ :

- Wait until it is time to take an action.
- Get the most up-to-date fixed point computed by the other thread. Call it  $q_t$ . (If no fixed point has been computed yet, initialize  $q_t$  arbitrarily.)
- Sample a pure action  $a_t$  from  $q_t$ .
- Play the pure action  $a_t$ .
- Observe rewards  $\pi_t$ .
- Update the regret vector  $R_t$ .

**Second thread:** Repeat forever:

- Wait until the other thread updates the regret vector  $R_\tau$  for any  $\tau > 0$ .
  - Get a copy of  $R_\tau$  from the other thread.
  - Compute  $N_\tau$  and  $D_\tau$ .
  - Compute a fixed point of  $N_\tau/D_\tau$ .
  - Pass this fixed point to the other thread.
- 

hence has run time  $O(n^3)$  per round and regret bound  $O(\sqrt{n}/t)$ . With another extreme parameter setting, namely  $p = n^3$ , MT has run time  $O(\log n)$  per round (as long as regret can be calculated quickly; see the end of Section 6) and regret bound  $O(n^2/\sqrt{t})$ . The intermediate parameter setting  $p = n$  yields an  $O(n^2)$  run time per round and an  $O(n/\sqrt{t})$  regret bound. This algorithm, therefore, dominates both PI and ICRM, achieving the same run time per round, but a better regret bound, for all values of  $t \geq n^{5/4}$ .

## 5 General Analysis

In this section, we derive a key lemma that is used in both our analyses. Specifically, we bound the  $L_2$ -norm of the regret vector at time  $t$  in terms of two summations from time  $\tau = 1$  to  $t$ . Each term in the first bounds how close the mixed action played at time  $\tau$  is to being a fixed point of the regret matrix at some previous time  $\tau - w(\tau)$ . Each term in the second bounds the regret that could ensue because the mixed action played at time  $\tau$  is out of date.

**Lemma 6** *For any online learning algorithm and any function  $w(\cdot) > 0$ , we have the following inequality: for*

all times  $t > 0$ ,

$$\begin{aligned} \|R_t^+\|_2^2 &\leq 2 \sum_{\tau=1}^t q_\tau (N_{\tau-w(\tau)} - D_{\tau-w(\tau)} I) \pi_t \\ &\quad + (n-1) \sum_{\tau=1}^t (2w(\tau) - 1) \end{aligned}$$

where  $q_t$  is the mixed action at time  $t$ , and  $I$  is the identity matrix.

We prove this lemma using two preliminary lemmas. The first involves simple algebra.

**Lemma 7** For any two vectors  $a, b \in \mathbb{R}^d$ , with  $d \geq 1$ , we have the following inequality:

$$\|(a+b)^+\|_2^2 \leq \|a^+\|_2^2 + 2a^+ \bullet b + \|b\|_2^2 \quad (6)$$

**Proof:** Both  $\|\cdot\|_2^2$  and dot products are additive component-wise, so it suffices to assume  $a, b$  are real numbers.

If  $a+b \leq 0$  then  $|a^+|^2 + 2a^+b + b^2 = (a^+ + b)^2 \geq 0 = |(a+b)^+|^2$ .

If  $a+b > 0$  then  $a^+ + b \geq a+b = (a+b)^+ > 0$ . Thus  $(a^+ + b)^2 \geq |(a+b)^+|^2$ . ■

**Lemma 8** For any learning algorithm and any  $t > \tau \geq 0$ , we have the following equality:

$$\begin{aligned} r_t \bullet R_\tau^+ &= q_t (N_\tau - D_\tau I) \pi_t \\ &= D_\tau q_t \left( \frac{N_\tau}{D_\tau} - I \right) \pi_t \end{aligned}$$

**Proof:** Standard no-regret arguments about the fixed point (e.g., Theorem 5 in [GLM06]). ■

Note that if  $q_t$  is a fixed point of  $N_\tau/D_\tau$ , as in is in FV and MT for appropriate choices of  $\tau$ , then  $r_t \bullet R_\tau^+ = 0$ . For example, in the traditional algorithm FV,  $r_t \bullet R_{t-1}^+ = 0$ .

**Proof:** [Proof of Lemma 6] Fix a  $\tau \in \{1, \dots, t\}$ . By definition,  $R_\tau^+ = R_{\tau-1}^+ + r_\tau$ . Hence, by applying Lemma 7, we obtain a linear approximation of  $\|R_\tau^+\|_2^2 - \|R_{\tau-1}^+\|_2^2$  with an error term:

$$\begin{aligned} \|R_\tau^+\|_2^2 &\leq \|R_{\tau-1}^+\|_2^2 + 2r_\tau \bullet R_{\tau-1}^+ + \|r_\tau\|_2^2 \\ &= \|R_{\tau-1}^+\|_2^2 + 2r_\tau \bullet R_{\tau-w(\tau)}^+ \\ &\quad + 2r_\tau \bullet (R_{\tau-1}^+ - R_{\tau-w(\tau)}^+) + \|r_\tau\|_2^2 \\ &\leq \|R_{\tau-1}^+\|_2^2 + 2r_\tau \bullet R_{\tau-w(\tau)}^+ \\ &\quad + 2(w(\tau) - 1)(n-1) + (n-1) \\ &= \|R_{\tau-1}^+\|_2^2 + 2r_\tau \bullet R_{\tau-w(\tau)}^+ + (2w(\tau) - 1)(n-1) \end{aligned} \quad (7)$$

The second inequality follows from the fact that  $\|r_\tau\|_2^2 \leq (n-1)$ .

Now if we apply Lemma 8 and sum over time, this yields:

$$\begin{aligned} &\sum_{\tau=1}^t \left( \|R_\tau^+\|_2^2 - \|R_{\tau-1}^+\|_2^2 \right) \\ &\leq 2 \sum_{\tau=1}^t q_\tau (N_{\tau-w(\tau)} - D_{\tau-w(\tau)} I) \pi_\tau \\ &\quad + (n-1) \sum_{\tau=1}^t (2w(\tau) - 1) \end{aligned} \quad (8)$$

The summation on the left hand side of this equation collapses to  $\|R_t^+\|_2^2 - \|R_0^+\|_2^2 = \|R_t^+\|_2^2$ , and the lemma is proved. ■

## 6 Analysis of MT

Equipped with Lemma 6, the proof of Theorem 5 is quite simple.

**Proof:** [Proof of Theorem 5] For general  $p$ , the fixed points may be based on out-of-date regret vectors, but they are never very out of date. Once the fixed point is computed, it is based on data that is  $p$  rounds out of date. That fixed point is then used for another  $p$  rounds while a replacement is computed. Overall, the fixed point played at time  $t$  can be based on a regret vector no more than  $2p$  rounds old. More precisely, the  $\tau$  such that  $R_\tau$  is used to compute  $q_t$  satisfies  $t - (2p-1) \leq \tau \leq t-1$ .

Now apply Lemma 6 letting  $w(\tau)$  be the age of the regret vector used by the second thread in calculating  $q_\tau$ . Since  $q_\tau$  is a fixed point of  $N_{\tau-w(\tau)}/D_{\tau-w(\tau)}$ , it follows that  $q_\tau(N_{\tau-w(\tau)} - D_{\tau-w(\tau)}I) = 0$ . Thus,

$$\begin{aligned} \|R_t^+\|_2^2 &\leq 2 \sum_{\tau=1}^t 0 + (n-1) \sum_{\tau=1}^t (2(2p-1) - 1) \\ &= (n-1)t(4p-3) \end{aligned}$$

Therefore,

$$\left\| \frac{R_t^+}{t} \right\|_\infty \leq \left\| \frac{R_t^+}{t} \right\|_2 \leq \sqrt{\frac{(n-1)(4p-3)}{t}}$$

and the theorem is proved. ■

A naïve computation of the regret vector would limit the per-round run time of PI to  $\Omega(n^2)$ . For applications where  $p$  is  $O(n)$  (or less), this is not a bottleneck, because in that case the  $O(n^3/p)$  bound on the run time of the fixed point computation is larger than the  $O(n^2)$  run time of the regret vector updates.

If the ODP is a repeated game where the opponents have  $O(n)$  joint actions, an agent can simply record the opponents' actions each round in constant time, and then update the regret vector right before solving for a fixed point; this update takes time  $O(n^3)$ . In this case, if  $p = n^3$ , then MT's per-round run time is  $O(\log n)$ .

For general ODPs, where the reward structure may change arbitrarily from one round to the next, keeping track of regret in time  $o(n^2)$  per round seems to

require random sampling (i.e., bandit techniques; see, for example, Auer *et al.* [ACBFS02]). We leave further investigation of this issue to future work.

Choosing a random action from a probability distribution using a binary search requires  $\Theta(\log n)$  time, so ODPs that require extremely quick decisions cannot be handled without further innovation.

## 7 Analysis of PI

In this section, we analyze PI. By construction,  $q_\tau$  is not a fixed point but only an approximate fixed point, so  $q_\tau(N_{\tau-w(\tau)} - D_{\tau-w(\tau)}I) \neq 0$ . Instead, we will show the following:

**Lemma 9** *For all times  $\tau > 0$  and  $0 < w(\tau) < \tau$ ,*  
 $\|q_\tau(N_{\tau-w(\tau)} - D_{\tau-w(\tau)}I)\|_1 = O\left(\frac{n\tau}{\sqrt{w(\tau)}} + n(w(\tau))^2\right)$

Deferring the proof of Lemma 9, we first show how to use Lemmas 6 (choose  $w(\tau) = \tau^{2/5}$ ) and 9, to analyze PI:

$$\begin{aligned} \|R_t^+\|_2^2 &\leq 2 \sum_{\tau=1}^t q_\tau (N_{\tau-w(\tau)} - D_{\tau-w(\tau)}I) \pi_\tau \\ &\quad + (n-1) \sum_{\tau=1}^t (2\tau^{2/5} - 1) \\ &= \sum_{\tau=1}^t O\left(\frac{n\tau}{\sqrt{\tau^{2/5}}} + n(\tau^{2/5})^2\right) + (n-1)O(t^{7/5}) \\ &= O(nt^{9/5}) + (n-1)O(t^{7/5}) \\ &= O(nt^{9/5}) \end{aligned}$$

Taking square roots and dividing by  $t$  proves Theorem 4:

$$\left\| \frac{R_t^+}{t} \right\|_2 \leq O(\sqrt{nt}^{-1/10})$$

It remains to prove Lemma 9. For the remainder of this section, we use the shorthands  $W \equiv w(\tau)$  and  $t = \tau - w(\tau)$ .

We begin to analyze  $q_\tau(N_t - D_tI)$  by rewriting this expression as the sum of two terms. The first, which would be zero if power iteration converged in  $W$  steps, is provably small. The second measures how the matrices  $P_T$  change over time; if all the  $P_T$ 's were equal, this term would be zero. Noting that  $q_\tau = q_t \left(\prod_{T=t}^{\tau-1} P_T\right)$ , where each  $P_T = N_T/D_T$ , we derive the two terms as follows:

$$\begin{aligned} q_\tau(N_t - D_tI) &= D_t q_t \left(\prod_{T=t}^{\tau-1} P_T\right) (P_t - I) \\ &= D_t q_t P_t^W (P_t - I) + \\ &\quad D_t q_t \left(\left[\prod_{T=t}^{\tau-1} P_T\right] - P_t^W\right) (P_t - I) \\ &= D_t q_t (P_t^{W+1} - P_t^W) + \\ &\quad D_t \left(\left[\prod_{T=t}^{\tau-1} P_T\right] - P_t^W\right) (P_t - I) \end{aligned} \quad (9)$$

We will bound the two terms in Equation 9 in turn. Beginning with the first, the quantity  $q_t P_t^W$  can be interpreted as the distribution of a Markov chain with transition matrix  $P_t$  and initial distribution  $q_t$  after  $W$  time steps. Most Markov chains converge to a stationary distribution, so it is intuitively plausible that the related quantity  $q_t (P_t^{W+1} - P_t^W)$  is small. The following lemma, which verifies this intuition, is a strengthening of statement M7 in Hart and Mas-Colell [HMC00]. Our lemma is stronger because our premises are weaker. Whereas their lemma requires that all the entries on the main diagonal of  $P_t$  be at least some uniform constant, ours requires only that the sum of  $P_t$ 's diagonal entries (i.e., its trace) be at least  $n-1$ . The latter of these two conditions (only) is satisfied by PI's choice of  $P_t$ , because each  $P_t$  is a convex combination of internal regret transformations/matrices, each of which has trace  $n-1$ .

**Lemma 10** *For all  $z > 0$ , if  $P$  is  $n$ -dimensional stochastic matrix that is close to the identity matrix in the sense that  $\sum_{i=1}^n P_{ii} \geq n-1$ , then  $\|q(P^z - P^{z-1})\|_1 = O(1/\sqrt{z})$  for all  $n$ -dimensional vectors  $q$  with  $\|q\|_1 = 1$ .*

**Proof:** See Appendix. ■

Now, we can easily bound  $D_t = D_{\tau-W}$  by  $(n-1)(\tau-W) \leq n\tau$ , so the first term in Equation 9 is bounded above by  $O(n\tau/\sqrt{W})$ . The following lemma bounds the second term in Equation 9:

**Lemma 11** *For all times  $\tau > 0$  and  $0 < w(\tau) < \tau$ ,*

$$\left\| q_t \left(\left[\prod_{T=t}^{\tau-1} P_T\right] - P_t^W\right) (P_t - I) \right\|_1 = O(nW^2/D_t)$$

The proof of this lemma makes use of the following definition and related fact: the induced  $L_1$ -norm of a matrix  $M$  is given by

$$\|M\|_1 = \max_{v \neq 0} \frac{\|vM\|_1}{\|v\|_1}$$

and for any  $n$ -dimensional vector  $v$  and matrix  $M$ ,

$$\|vM\|_1 \leq \|v\|_1 \|M\|_1 \quad (10)$$

**Proof:** Since  $\|P_t - I\|_1 \leq \|P_t\|_1 + \|I\|_1 = 2$ , it follows that

$$\begin{aligned} & \left\| q_t \left( \prod_{s=0}^{W-1} P_{t+s} - P_t^W \right) (P_t - I) \right\|_1 \\ & \leq 2 \left\| q_t \left( \prod_{s=0}^{W-1} P_{t+s} - P_t^W \right) \right\|_1 \end{aligned}$$

Hence, it suffices to bound  $\left\| q_t \left( \prod_{s=0}^{W-1} P_{t+s} - P_t^W \right) \right\|_1$ . To do so, we first note that

$$\begin{aligned} & \prod_{s=0}^{W-1} P_{t+s} - P_t^W \\ & = \sum_{s=0}^{W-1} \left( \prod_{u=0}^s P_{t+u} P_t^{W-s-1} - \prod_{u=0}^{s-1} P_{t+u} P_t^{W-s} \right) \\ & = \sum_{s=0}^{W-1} \left( \prod_{u=0}^{s-1} P_{t+u} (P_{t+s} - P_t) P_t^{W-s-1} \right) \quad (11) \end{aligned}$$

Next, we multiply both sides of Equation 11 by  $q_t$  and take the  $L_1$ -norm. Then, we apply Equation 10 and the facts that  $\|q_t\|_1 = 1$  and  $\|P_{t+s}\|_1 = 1$ , for all  $s = 0, \dots, W-1$ , to obtain the following:

$$\begin{aligned} & \left\| q_t \left( \prod_{s=0}^{W-1} P_{t+s} - P_t^W \right) \right\|_1 \\ & = \left\| \sum_{s=0}^{W-1} q_t \left( \prod_{u=0}^{s-1} P_{t+u} \right) (P_{t+s} - P_t) P_t^{W-s-1} \right\|_1 \\ & \leq \sum_{s=0}^{W-1} \left\| q_t \left( \prod_{u=0}^{s-1} P_{t+u} \right) (P_{t+s} - P_t) P_t^{W-s-1} \right\|_1 \\ & \leq \sum_{s=0}^{W-1} \|q_t\|_1 \left( \prod_{u=0}^{s-1} \|P_{t+u}\|_1 \right) \|P_{t+s} - P_t\|_1 \|P_t\|_1^{W-s-1} \\ & = \sum_{s=0}^{W-1} \|P_{t+s} - P_t\|_1 \quad (12) \end{aligned}$$

The first inequality in the above derivation follows from the triangle inequality. The second follows from the fact that the norm of a product is bounded above by the product of the norms. To understand the final quantity (Equation 12) intuitively, consider two coupled Markov chains, one of which uses  $P_t$  as its transition matrix, and the other of which uses  $P_{t+s}$ . These Markov chains lead to different distributions to the extent that they have different transition matrices.

Since  $P_{t+s} = N_{t+s}/D_{t+s}$ , it follows that:

$$\begin{aligned} & \|P_{t+s} - P_t\|_1 \\ & = \left\| \frac{N_{t+s}}{D_{t+s}} - \frac{N_t}{D_t} \right\|_1 \\ & \leq \left\| \frac{N_{t+s}}{D_{t+s}} - \frac{N_{t+s}}{D_t} \right\|_1 + \frac{\|N_{t+s} - N_t\|_1}{D_t} \\ & = \|N_{t+s}\|_1 \frac{|D_{t+s} - D_t|}{(D_{t+s}D_t)} + \frac{\|N_{t+s} - N_t\|_1}{D_t} \end{aligned}$$

The inequality in this derivation follows from the triangle inequality.

Only  $n-1$  of the  $n(n-1)$  internal transformations affect any particular action and rewards are between 0 and 1, so  $|D_{t+1} - D_t|$  is bounded by  $n-1$ . The induced  $L_1$ -norm of a matrix is the maximum row sum, after taking the absolute value of each entry; hence,  $\|N_{t+1} - N_t\|_1$  is bounded by  $n-1$ . Further,  $\|N_{t+s}\|_1 \leq D_{t+s}$ ,  $|D_{t+s} - D_t| \leq s(n-1)$ , and  $\|N_{t+s} - N_t\|_1 \leq s(n-1)$ , so we conclude that  $\|P_{t+s} - P_t\|_1 \leq 2s(n-1)/D_t$ . Summing over  $s$  from 0 to  $W-1$  yields the desired  $O(nW^2/D_t)$ .  $\blacksquare$

## 8 Experiments

We ran some simple experiments on the repeated Shapley game to see whether the theoretical bounds we derived match what is observed in practice. An instance of the internal regret-matching (IRM) algorithm<sup>6</sup> of Greenwald *et al.* [GLM06] was played against PI, HM with  $\mu = 5$ , and MT with  $p = 10$ . Our results are plotted in Figures 1, 2, 3 and 4 (the fourth figure summarizes all our results).

Each experiment was repeated 50 times, and each ensuing data series is plotted with two lines, delimiting the 95% confidence interval. The “true” line corresponding to infinitely many runs probably lies somewhere between the two plotted lines. Note the logarithmic scales of the axes, so powers such as  $1/\sqrt{t}$  appear as straight lines.

What we observe is twofold: (i) PI does much better in practice than it does in theory, achieving better performance than HM and MT (see Figure 4); and (ii) MT does substantially worse than IRM, with the ratio similar to the  $\sqrt{4(10) - 3} \approx 6$  predicted by theory.

## 9 Discussion

Standard no-internal-regret (NIR) algorithms rely on a fixed point computation, and hence typically require  $O(n^3)$  run time per round of learning. The main contribution of this paper is a novel NIR algorithm, which is a simple and straightforward variant of a standard NIR algorithm, namely that in Greenwald [GJMar]. Rather than compute a fixed point every round, our algorithm relies on power iteration to estimate a fixed point, and hence runs in  $O(n^2)$  time per round.

One obvious question that comes to mind is: can power iteration be used in algorithms that minimize  $\Phi$ -regret, for arbitrary  $\Phi$ ? The answer to this question is no, in general. For example, consider an ODP with two actions, and only one action transformation  $\phi$ , which swaps the two actions: i.e.,

$$\phi = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

A standard  $\Phi$ -regret-minimizing algorithm would play the fixed-point of this matrix, which is uniform randomization. However, PI would learn a predictable sequence

<sup>6</sup>This algorithm is a close cousin of FV, and has the same regret bound.

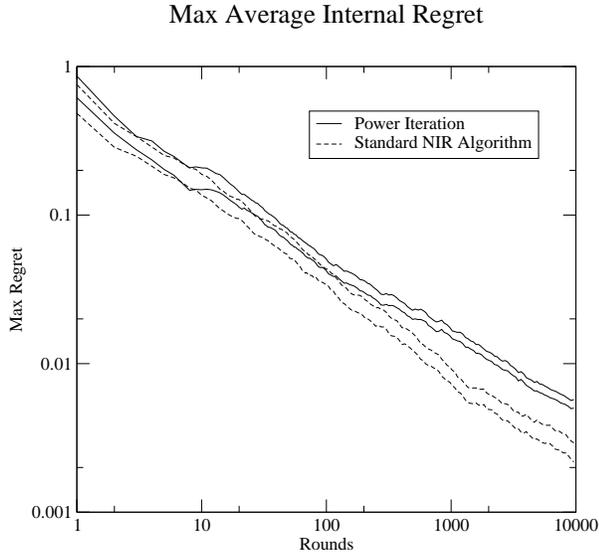


Figure 1: IRM and PI playing Shapley. 95% confidence interval of average of 50 runs shown.

of mixed actions, namely  $q, 1 - q, q, 1 - q, \dots$ . Since an adversary could easily exploit this alternating sequence of plays, the idea does not immediately apply to arbitrary  $\Phi$ . The part of the proof that is specific to  $\Phi_{\text{INT}}$  is  $P_t$  having trace  $n - 1$ , allowing us to use Lemma 10.

Another related question is: can power iteration be used in other NIR algorithms? For example, Cesa-Bianchi and Lugosi [CBL03] and Greenwald *et al.* [GLM06] present a class of NIR algorithms, each one of which is based on a potential function. Similarly, Blum and Mansour [BM05] present a method of constructing NIR learners from no-external-regret (NER) learners. We conjecture that the power iteration idea could be applied to any of these NIR algorithms, but we have not yet thoroughly explored this question.

Our admittedly limited experimental investigations reveal that perhaps PI's convergence rate in practice is not as bad as the theory predicts, but further study is certainly warranted. Another interesting question along the same lines is: would another iterative linear solving method, specifically one that is more sophisticated than power iteration, such as biconjugate gradient, yield better results, either in theory or in practice?

## Acknowledgments

We are grateful to Dean Foster for originally suggesting that we try out power iteration in our experiments with regret-minimizing algorithms. We are also grateful to Casey Marks for providing much of the code for our experiments and to Yuval Peres for assistance simplifying the proof of Lemma 10. This research was supported in

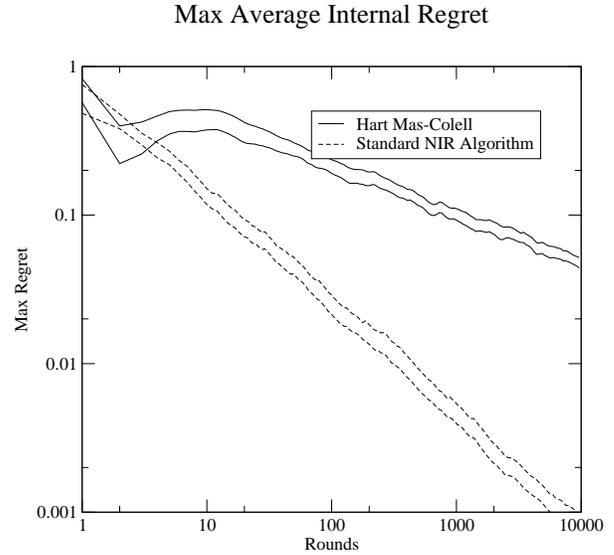


Figure 2: IRM and HM with  $\mu = 5$  playing Shapley. 95% confidence interval of average of 50 runs shown.

part by the Sloan Foundation.

## References

- [ACBFS02] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The nonstochastic multiarmed bandit problem. *Siam J. of Computing*, 32(1):48–77, 2002.
- [BM05] A. Blum and Y. Mansour. From external to internal regret. In *Proceedings of the 2005 Computational Learning Theory Conference*, pages 621–636, June 2005.
- [Cah04] A. Cahn. General procedures leading to correlated equilibria. *International Journal of Game Theory*, 33(1):21–40, December 2004.
- [CBL03] N. Cesa-Bianchi and G. Lugosi. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51(3):239–261, 2003.
- [CBL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [CLRS01] Cormen, Leiserson, Rivest, and Stein. *Introduction to Algorithms*, chapter 28, pages 757–758. MIT Press, 2nd edition, 2001.
- [CW87] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. In *STOC '87: Proceedings of the nineteenth annual ACM Symposium on Theory of Computing*, pages 1–6. ACM Press, New York, NY USA, 1987.

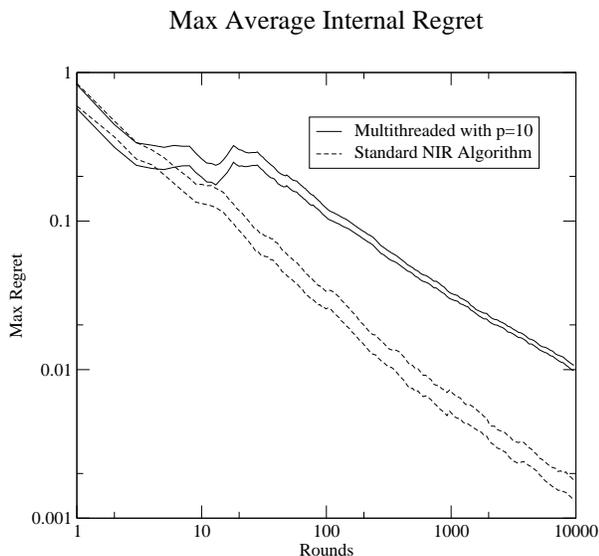


Figure 3: IRM and MT with  $p = 10$  playing Shapley. 95% confidence interval of average of 50 runs shown.

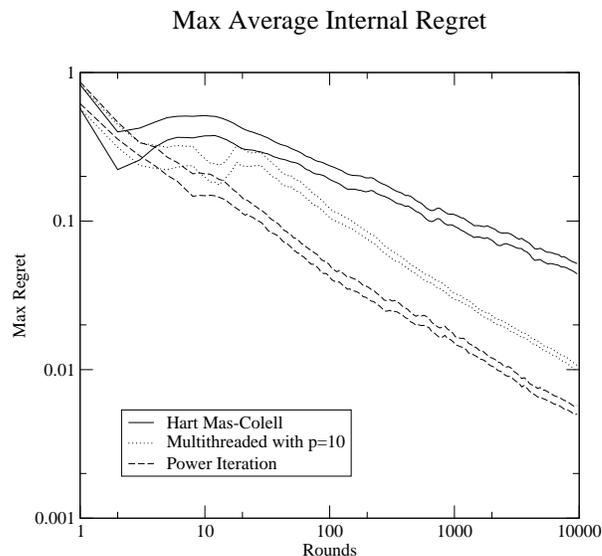


Figure 4: Summary of Figures 1, 2 and 3.

- [FS97] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.
- [FV97] D. Foster and R. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21:40–55, 1997.
- [GJMar] A. Greenwald, A. Jafari, and C. Marks. A general class of no-regret algorithms and game-theoretic equilibria. In Amitabha Gupta, Johan van Benthem, and Eric Pacuit, editors, *Logic at the Crossroads: An Interdisciplinary View*, volume 2. Allied Publishers, To Appear.
- [GLM06] A. Greenwald, Z. Li, and C. Marks. Bounds for regret-matching algorithms. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, 2006.
- [HMC00] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150, 2000.
- [HMC01] S. Hart and A. Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 98(1):26–54, 2001.
- [Lin92] Torgny Lindvall. *Lectures on the Coupling Method*, chapter II.12, pages 41–47. Wiley, 1992.
- [LW94] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212 – 261, 1994.

- [Str69] Volker Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969.
- [You04] P. Young. *Strategic Learning and its Limits*. Oxford University Press, Oxford, 2004.

## A Proof of technical Lemma 10

*Note: in this proof, we use  $N$ ,  $M$  and  $t$  for meanings unrelated to those in the main body of the paper. Don't be confused.*

Let  $M$  be an  $n$  by  $n$  matrix which is row-stochastic (vectors should be multiplied on the left as in  $qM$ ) and has trace at least  $n - 1$ . We want to show:

$$\max_{\|q\|_1=1} \|q(M^W - M^{W-1})\|_1 = O(1/\sqrt{W}).$$

For any two probability measures  $\mu$  and  $\nu$  on probability space  $\Omega$ , their total variation distance is defined to be

$$\|\mu - \nu\|_{TV} = \frac{1}{2} \sum_{a \in \Omega} |\mu(a) - \nu(a)|. \quad (13)$$

We denote by  $\mu(X)$  the distribution of random variable  $X$ . Let  $Q(t)$  denote the state of a Markov chain with transition matrix  $M$  and initial distribution  $q$  after  $t$  time steps. Our desired conclusion can be recast in Markov chain language as

$$\|\mu(Q(W)) - \mu(Q(W-1))\|_{TV} = O(1/\sqrt{W})$$

for all initial distributions  $q$ .

We know that  $\sum_i m_{i,i} \geq n - 1$  where  $m_{i,j}$  is the element of the  $i$ th row and  $j$ th column in matrix  $M$ . All  $m_{i,i}$  are at most 1, so there can be at most one state,

call it  $p$ , satisfying  $m_{p,p} < 1/2$ . If no such state exists the Lemma was already shown as step M7 of [HMC00], so assume it exists.<sup>7</sup> We define a new matrix  $N$  as the unique solution to  $m_{i,j} = \frac{1}{2}(n_{i,j} + \delta_{i,j})$  if  $i \neq p$ , and  $m_{p,j} = n_{p,j}$  otherwise, where  $\delta_{i,j} = 1$  if  $i = j$  and 0 otherwise.<sup>8</sup> It is easy to check  $N$  is row stochastic and is therefore the transition matrix of some Markov Chain. For simplicity we denote by  $N$  the Markov Chain with transition matrix  $N$  and initial distribution  $q$ . Let  $B(0), B(1), \dots$  be the random walk associated with Markov Chain  $N$ .

One can easily create a random walk of the Markov chain  $M$  from the walk of  $N$  as follows. If the current state is the special state  $p$ , do what the  $N$ -chain did. Otherwise, flip a coin, following  $N$  with probability 50% and remaining at the same state otherwise. Formally, define index  $I_t$  inductively by  $I_0 = 0$  and  $I_t = I_{t-1} + a$  symmetric Bernoulli distribution, if  $B(I_{t-1}) \neq p$ , and  $I_t = I_{t-1} + 1$  otherwise. Then it can be shown that  $Q(t) = B(I_t)$  has distribution  $qM^t$ . We need to show the distributions of  $B(I_t)$  and  $B(I_{t-1})$  are close to each other.

For  $i \geq 0$ , define  $X_i = \#\{t \geq 0 : I_t = i\}$ . Intuitively,  $X_i$  is the number of steps the  $M$  chain takes while the  $N$  chain is in state  $B(i)$ . Our proof hinges on the easily seen fact that the  $X_i$ s are *independent* random variables. If  $B(i) = p$ ,  $X_i = 1$  and if  $B(i) \neq p$ ,  $X_i$  is geometrically distributed with mean 2.<sup>9</sup> Let  $T_i = \sum_{j=0}^{i-1} X_j$ . One can see that  $I_W = \max\{i : T_i \leq W\}$ .

In order to prove the conclusion we need the following two lemmas.

**Lemma 12** *Let  $f(y, z)$  be a function,  $Y, Y'$  and  $Z$  be random variables with  $Z$  pairwise independent of  $Y$  and  $Y'$ . Let  $I = f(Y, Z)$  and  $I' = f(Y', Z)$  be random variables. Then*

$$\|\mu(I) - \mu(I')\|_{TV} \leq \|\mu(Y) - \mu(Y')\|. \quad (14)$$

**Proof:** The total variation distance between  $\mu(X)$  and  $\mu(X')$  is equal to minimum possible probability that  $X \neq X'$  over couplings of  $X$  and  $X'$ . A coupling of  $Y$  and  $Y'$  can be extended into a coupling of  $I$  and  $I'$  trivially. ■

**Lemma 13** *Let  $S_k = \sum_{i=1}^k X_i$  where  $X_i$ 's are independent identically distributed random variables with geometric distributions with mean 2, then  $\|\mu(S_k) - \mu(1 + S_k)\|_{TV} = O(n^{-1/2})$ .*

**Proof:** Equation II.12.4 of [Lin92]. ■

The probability that  $I_W < W/3$  is bounded by the probability that the sum of  $W$  Bernoulli random variables with mean 1/2 is less than  $W/3$ . A standard

<sup>7</sup>One can see that the current proof also holds if no such state exists.

<sup>8</sup>I.e.,  $n_{i,j} = 2m_{i,j} - \delta_{i,j}$  if  $i \neq p$ , and  $n_{p,j} = m_{p,j}$  otherwise.

<sup>9</sup>Note that our “geometrically distributed” variables have support  $1, 2, \dots$ , so  $\Pr(X_i = k) = (1/2)^k$  for  $k \geq 1$ .

Chernoff bound therefore shows that the event  $E$  that  $I_W < W/3$  is exponentially unlikely. Condition on the path  $B(0), B(1), \dots$  and event  $E$  not happening. We can therefore write  $I_W = \max\{i : \sum_{j=W/3+1}^i X_j \leq W - T_{W/3}\}$  and  $I_{W-1} = \max\{i : \sum_{j=W/3+1}^i X_j \geq W - 1 - T_{W/3}\}$ . Now by Lemma 12, associating  $Z$  with the vector-valued random variable  $\{X_i\}_{i=W/3+1}^\infty$ ,  $Y$  with  $W - T_{W/3}$  and  $Y'$  with  $W - 1 - T_{W/3}$ , we see that it suffices to bound the total variation distance between  $T_{W/3}$  and  $T_{W/3} + 1$ .

Define  $k = \#\{0 \leq i \leq W/3 : B(i) \neq b\}$ . Every visit to  $b$  is followed by a visit to another state with probability at least 1/2, so with exponentially high probability over the choices of the  $B(i)$ ,  $k \geq W/12$ . Condition on the event  $k \geq W/12$ . By definition  $T_{W/3} = (W/3 - k) + \sum_{i:B(i) \neq b} X_i$ , so it suffices to analyze the variation distance between the sum of  $k \geq W/12$  geometric random variables and the same shifted by 1. By Lemma 13, this is  $O(1/\sqrt{W})$ .

Therefore we obtain

$$\|\mu(I_W) - \mu(I_{W-1})\|_{TV} = O(W^{-1/2}) \quad (15)$$

Back to  $M$  chain  $B(I_W)$  we could have

$$\begin{aligned} \Pr(B(I_W) = i) &= \sum_a \Pr(B(I_W) = i | I_W = a) \Pr(I_W = a) \\ &= \sum_a \Pr(B(a) = i) \Pr(I_W = a) \end{aligned}$$

and similarly

$$\Pr(B(I_{W-1}) = i) = \sum_a \Pr(B(a) = i) \Pr(I_{W-1} = a)$$

Thus

$$\begin{aligned} &\|\mu(B(I_W)) - \mu(B(I_{W-1}))\|_{TV} \\ &= \frac{1}{2} \sum_i |\Pr(B(I_W) = i) - \Pr(B(I_{W-1}) = i)| \\ &\leq \frac{1}{2} \sum_i \sum_a \Pr(B(a) = i) |\Pr(I_W = a) - \Pr(I_{W-1} = a)| \\ &= \frac{1}{2} \sum_a |\Pr(I_W = a) - \Pr(I_{W-1} = a)| \\ &= \|\mu(I_W) - \mu(I_{W-1})\|_{TV} \\ &= O(W^{-1/2}) \end{aligned}$$

This concludes the proof.

We remark that this lemma can also be proved in a more self-contained manner via a Markov chain coupling. The motivating story follows.

Charlie and Eve are walking drunkenly between the  $n$  neighborhood bars. If Charlie is in a good bar, each time step he first flips a coin to decide whether or not he should leave that bar. If he decides to leave, he then makes a probabilistic transition to some bar (perhaps the same one). If Charlie is in a bad bar, he always leaves. Eve starts one time step later than Charlie at the same initial bar. Eve makes her decision to leave or not

independently of Charlie, but reuses Charlie's choices of where to go next. However, if Eve ever catches up with Charlie, she switches to just following him around. A natural question to ask is how likely Eve and Charlie are to be at the same bar after  $t$  time steps? Note that if you look at Eve's motions and ignore Charlie's, she behaves exactly like Charlie does.

The connection to the present lemma is that Charlie's distribution corresponds to  $M^t$  and Eve's to  $M^{t-1}$ . Standard arguments relating total variation distance to couplings show that if Eve and Charlie usually finish at the same bar, their probability distributions must be quite similar.

---

# Linear Algorithms for Online Multitask Classification

---

Giovanni Cavallanti\*

Nicolò Cesa-Bianchi†

Claudio Gentile‡

## Abstract

We design and analyze interacting online algorithms for multitask classification that perform better than independent learners whenever the tasks are related in a certain sense. We formalize task relatedness in different ways, and derive formal guarantees on the performance advantage provided by interaction. Our online analysis gives new stimulating insights into previously known co-regularization techniques, such as the multitask kernels and the margin correlation analysis for multiview learning. In the last part we apply our approach to spectral co-regularization: we introduce a natural matrix extension of the quasi-additive algorithm for classification and prove bounds depending on certain unitarily invariant norms of the matrix of task coefficients.

## 1 Introduction

A fundamental and fascinating problem in learning theory is the study of learning algorithms that influence each other. Although much is known about the behavior of individual strategies that learn a classification or regression task from examples, our understanding of interacting learning systems is still fairly limited. In this paper, we investigate this problem from the specific viewpoint of *multitask learning*, where each one of  $K > 1$  learners has to solve a different task (typically,  $K$  classification or  $K$  regression tasks). In particular, we focus on multitask binary classification, where learners are *online* linear classifiers (such as the Perceptron algorithm). Our goal is to design online interacting algorithms that perform better than independent learners whenever the tasks are related in a certain sense. We formalize task relatedness in different ways, and derive formal guarantees on the performance advantage provided by interaction.

Our analysis builds on ideas that have been developed in the context of statistical learning. In the statistical analysis of multitask learning (e.g., [2, 3, 4, 11, 24, 26]) the starting point is a regularized empirical loss functional or Tikhonov

functional —see, e.g., [10]. In the presence of several tasks, this functional is extended to allow for co-regularization among tasks. Roughly speaking, the co-regularization term forces the set of predictive functions for the  $K$  tasks to lie “close” to each other.

This co-regularization term is typically a squared norm in some Hilbert space of functions. We follow the approach pioneered by [11], where the  $K$  estimated solutions are linear functions parametrized by  $\mathbf{u} = (\mathbf{u}_1, \dots, \mathbf{u}_K) \in \mathbb{R}^{Kd}$  and the co-regularization is  $\mathbf{u}^\top A \mathbf{u}$ , where  $A$  is a positive definite matrix enforcing certain relations among tasks. The key observation in [11] is the following. Assume the instances of the multitask problem are of the form  $(\mathbf{x}_t, i_t)$ , where  $\mathbf{x}_t \in \mathbb{R}^d$  is an attribute vector and  $i_t \in \{1, \dots, K\}$  indicates the task  $\mathbf{x}_t$  refers to. Then one can reduce the  $K$  learning problems in  $\mathbb{R}^d$  to a single problem in  $\mathbb{R}^{Kd}$  by choosing a suitable embedding of the pairs  $(\mathbf{x}_t, i_t)$  into a common RKHS space  $\mathbb{R}^{Kd}$  with inner product  $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^\top A \mathbf{v}$ . This reduction allows us to solve a multitask learning problem by running any kernel-based single-task learning algorithm with the “multitask kernel” defined above. We build on this result by considering a natural online protocol for multitask linear classification. Within this protocol we analyze the performance of the Perceptron algorithm and some of its variants when run with a multitask kernel. Because such kernels are linear, we are not restricted to using kernel-based algorithms for efficiency reasons.

In Section 3 we consider the kernel Perceptron algorithm, and derive mistake bounds for the multitask kernels proposed in [11]. This reveals new insights into the role played by the regularizer matrix  $A$ . First, we see that the update in the kernel space defined by  $A$  factorizes in the “shared update” of  $K$  interacting Perceptrons each running in  $\mathbb{R}^d$ , thus providing a basic example of interactive online learning. Second, we exploit the simplicity of the mistake bound analysis to precisely quantify the performance advantage brought by the multitask approach over  $K$  independent online algorithms. In particular, in Subsections 3.2 and 3.3 we give examples where the mistake bound is used to guide the design of  $A$ . The first part of the paper is concluded with Sections 4 and 5, where we show multitask versions and mistake bound analyses for the second-order Perceptron algorithm of [7] and for the  $p$ -norm algorithm of [13, 14].

In the remaining sections of the paper, we depart from the approach of [11] to investigate the power of online learning when other forms of co-regularization are used. In Sec-

---

\*DSI, Università di Milano, Italy.

†DSI, Università di Milano, Italy.

‡DICOM, Università dell’Insubria, Italy. This is the corresponding author. Email: [claudio.gentile@uninsubria.it](mailto:claudio.gentile@uninsubria.it)

tion 6 we consider the case when instances belong to a space that is different for each task, and the similarity among tasks is measured by comparing their margin sequences (see, e.g., [6, 27]). We introduce and analyze a new multitask variant of the second-order Perceptron algorithm. The mistake bound that we prove is a margin-based version of the bound shown in Subsection 3.2 for the multitask Perceptron. Finally, in Section 7 we consider spectral co-regularization [4] for online multiview learning. Here diversity is penalized using a norm function defined on the  $d \times K$  matrix  $U = [\mathbf{u}_1, \dots, \mathbf{u}_K]$  of view vectors. In the spirit of [19], we interpret this penalization function as a potential defined over arbitrary matrices. We then define a natural extension of the quasi-additive algorithm of [14, 20] to a certain class of matrix norms, and provide a mistake bound analysis depending on the singular values of  $U$ . The results we obtain are similar to those in [28, 29, 30], though we are able to overcome some of the difficulties encountered therein via a careful study of matrix differentials.

In the next section, we introduce the basic online multitask protocol and define the multitask Perceptron algorithm. In order to keep the presentation as simple as possible, and to elucidate the interactive character of the updates, we delay the introduction of kernels until the proof of the mistake bound.

In our initial online protocol, at each time step the multitask learner receives a pair  $(\mathbf{x}_t, i_t)$ , where  $i_t$  is the task index for time  $t$  and  $\mathbf{x}_t$  is the instance for task  $i_t$ . Note that we view multitask learning as a sequential problem where at each time step the learner works on a single adversarially chosen task, rather than working simultaneously on all tasks (a similar protocol was investigated in [1] in the context of prediction with expert advice). One of the advantages of this approach is that, in most cases, the cost of running our multitask algorithms has a mild dependence on the number  $K$  of tasks.

We also remark that linear algorithms for online multitask learning have been studied in [9]. However, these results are sharply different from ours, as they do not depend on task relatedness.

## 2 Learning protocol and notation

There are  $K$  binary classification tasks indexed by  $1, \dots, K$ . At each time step  $t = 1, 2, \dots$  the learner receives a task index  $i_t \in \{1, \dots, K\}$  and the corresponding instance vector<sup>1</sup>  $\mathbf{x}_t \in \mathbb{R}^d$  (which we henceforth assume to be normalized,  $\|\mathbf{x}_t\| = 1$ ). Based on this information, the learner outputs a binary prediction  $\hat{y}_t \in \{-1, 1\}$  and then observes the correct label  $y_t \in \{-1, 1\}$  for task  $i_t$ . As in the standard worst-case online learning model, no assumptions are made on the mechanism generating the sequence  $(\mathbf{x}_t, y_t)_{t \geq 1}$ . Moreover, similarly to [1], the sequence of tasks  $i_t$  is also generated in an adversarial manner.

We compare the learner’s performance to that of a reference predictor that is allowed to use a different linear classifier for each of the  $K$  tasks. In particular, we compare the

<sup>1</sup>Throughout this paper all vectors are assumed to be column vectors.

learner’s mistake count to

$$\inf_{\mathbf{u}_1, \dots, \mathbf{u}_K \in \mathbb{R}^d} \sum_t \ell_t(\mathbf{u}_{i_t}) \quad (1)$$

where  $\ell_t(\mathbf{u}_{i_t}) = [1 - y_t \mathbf{u}_{i_t}^\top \mathbf{x}_t]_+$  is the hinge loss of the reference linear classifier (or *task vector*)  $\mathbf{u}_{i_t}$  at time  $t$ . Our goal is to design algorithms that make fewer mistakes than  $K$  independent learners when the tasks are related, and do not perform much worse than that when the tasks are completely unrelated. In the first part of the paper we use Euclidean distance to measure task relatedness. We say that the  $K$  tasks are related if there exist reference task vectors  $\mathbf{u}_1, \dots, \mathbf{u}_K \in \mathbb{R}^d$  having small pairwise distances  $\|\mathbf{u}_i - \mathbf{u}_j\|$ , and achieving a small cumulative hinge loss in the sense of (1). More general notions of relatedness are investigated in later sections.

## 3 The multitask Perceptron algorithm

We first introduce a simple multitask version of the Perceptron algorithm. This algorithm keeps a weight vector for each task and updates all weight vectors at each mistake using the Perceptron rule with different learning rates. More precisely, let  $\mathbf{w}_{i,t-1}$  be the weight vector associated with task  $i$  at the beginning of time step  $t$ . If we are forced (by the adversary) to predict on task  $i_t$ , and our prediction happens to be wrong, we update  $\mathbf{w}_{i_t,t-1}$  through the standard additive rule  $\mathbf{w}_{i_t,t} = \mathbf{w}_{i_t,t-1} + \eta y_t \mathbf{x}_t$  (where  $\eta > 0$  is a constant learning rate) but, at the same time, we perform a “half-update” on the remaining  $K - 1$  Perceptrons, i.e., we set  $\mathbf{w}_{j,t} = \mathbf{w}_{j,t-1} + \frac{\eta}{2} y_t \mathbf{x}_t$  for each  $j \neq i_t$ . This rule is based on the simple observation that, in the presence of related tasks, any update step that is good for one Perceptron should also be good for the others. Clearly, this rule keeps the weight vectors  $\mathbf{w}_{j,t}, j = 1, \dots, K$ , always close to each other.

The above algorithm is a special case of the *multitask Perceptron algorithm* described below. This more general algorithm updates each weight vector  $\mathbf{w}_{j,t}$  through a learning rate which is an arbitrary positive definite function of the pair  $(j, i_t)$ . These learning rates are defined by a  $K \times K$  interaction matrix  $A$ .

The pseudocode for the multitask Perceptron algorithm using a generic interaction matrix  $A$  is given in Figure 1. At the beginning of each time step, the counter  $s$  stores the mistakes made so far (plus one). The (column) vector  $\phi_t \in \mathbb{R}^{Kd}$  denotes the *multitask instance* defined by

$$\phi_t^\top = \left( \underbrace{0, \dots, 0}_{d(i_t - 1) \text{ times}} \quad \mathbf{x}_t^\top \quad \underbrace{0, \dots, 0}_{d(K - i_t) \text{ times}} \right) \quad (2)$$

where  $\mathbf{x}_t \in \mathbb{R}^d$  is the instance vector for the current task  $i_t$ . (Note that  $\|\phi_t\| = 1$  since the instances  $\mathbf{x}_t$  are normalized.) The weights of the  $K$  Perceptrons are maintained in a compound vector  $\mathbf{w}_s^\top = (\mathbf{w}_{1,s}^\top, \dots, \mathbf{w}_{K,s}^\top)$ , with  $\mathbf{w}_{j,s} \in \mathbb{R}^d$  for all  $j$ . The algorithm predicts  $y_t$  through the sign  $\hat{y}_t$  of the  $i_t$ -th Perceptron’s margin  $\mathbf{w}_{s-1}^\top \phi_t = \mathbf{w}_{i_t, s-1}^\top \mathbf{x}_t$ . Then, if prediction and true label disagree, the update rule becomes  $\mathbf{w}_s = \mathbf{w}_{s-1} + y_t (A \otimes I_d)^{-1} \phi_t$ , where  $\otimes$  denotes the Kro-

**Parameters:** Positive definite  $K \times K$  interaction matrix  $A$ .

**Initialization:**  $\mathbf{w}_0 = \mathbf{0} \in \mathbb{R}^{Kd}$ ,  $s = 1$ .

At each time  $t = 1, 2, \dots$  do the following:

1. Observe task number  $i_t \in \{1, \dots, K\}$  and the corresponding instance vector  $\mathbf{x}_t \in \mathbb{R}^d$ ;
2. Build the associated multitask instance  $\phi_t \in \mathbb{R}^{Kd}$ ;
3. Predict  $\hat{y}_t = \text{SGN}(\mathbf{w}_{s-1}^\top \phi_t) \in \{-1, +1\}$ ;
4. Get label  $y_t \in \{-1, +1\}$ ;
5. If  $\hat{y}_t \neq y_t$  then update:

$$\begin{aligned} \mathbf{w}_s &= \mathbf{w}_{s-1} + y_t (A \otimes I_d)^{-1} \phi_t \\ s &\leftarrow s + 1. \end{aligned}$$

Figure 1: The multitask Perceptron algorithm.

necker product between matrices<sup>2</sup> and  $I_d$  is the  $d \times d$  identity matrix. Since  $(A \otimes I_d)^{-1} = A^{-1} \otimes I_d$ , the above update is equivalent to the  $K$  task updates

$$\mathbf{w}_{j,s} \leftarrow \mathbf{w}_{j,s-1} + y_t A_{j,i_t}^{-1} \mathbf{x}_t \quad j = 1, \dots, K.$$

The algorithm is mistake driven, hence  $\mathbf{w}_{t-1}$  is updated (and is  $s$  increased) only when  $\hat{y}_t \neq y_t$ .

### 3.1 Pairwise distance interaction matrix

We now analyze the choice of  $A$  that corresponds to the updates  $\mathbf{w}_{i_t,s} \leftarrow \mathbf{w}_{i_t,s-1} + \eta y_t \mathbf{x}_t$  and  $\mathbf{w}_{j,s} \leftarrow \mathbf{w}_{j,s-1} + \frac{\eta}{2} y_t \mathbf{x}_t$  for  $j \neq i_t$  with  $\eta = 2/(K+1)$ . As it can be easily verified, this choice is given by

$$A = \begin{bmatrix} K & -1 & \dots & -1 \\ -1 & K & \dots & -1 \\ \dots & \dots & \dots & \dots \\ -1 & \dots & \dots & K \end{bmatrix} \quad (3)$$

with

$$A^{-1} = \frac{1}{K+1} \begin{bmatrix} 2 & 1 & \dots & 1 \\ 1 & 2 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & \dots & \dots & 2 \end{bmatrix}.$$

In order to keep in with the notation just introduced, we equivalently specify an online multitask problem by the sequence  $(\phi_1, y_1), (\phi_2, y_2), \dots \in \mathbb{R}^{dK} \times \{-1, 1\}$  of multitask examples, where  $\phi_t$  is the multitask instance defined in (2). Moreover, given a sequence of multitask examples and reference task vectors  $\mathbf{u}_1, \dots, \mathbf{u}_K \in \mathbb{R}^d$ , we introduce the ‘‘compound’’ reference task vector  $\mathbf{u}^\top = (\mathbf{u}_1^\top, \dots, \mathbf{u}_K^\top) \in \mathbb{R}^{Kd}$  and write

$$\ell_t(\mathbf{u}) \stackrel{\text{def}}{=} [1 - y_t \mathbf{u}^\top \phi_t]_+ = [1 - y_t \mathbf{u}_{i_t}^\top \mathbf{x}_t]_+ = \ell_t(\mathbf{u}_{i_t}).$$

Finally, we use  $A_\otimes$  as a shorthand for  $A \otimes I_d$ , where  $d$  is understood from the context. We have the following result.

<sup>2</sup>The Kronecker or direct product between two matrices  $A = [a_{i,j}]$  and  $B$  of dimension  $m \times n$  and  $q \times r$ , respectively, is the block matrix of dimension  $m q \times n r$  whose block on row  $i$  and column  $j$  is the  $q \times r$  matrix  $a_{i,j} B$ .

**Theorem 1** *The number of mistakes  $m$  made by the multitask Perceptron algorithm in Figure 1, run with interaction matrix (3) on any finite multitask sequence  $(\phi_1, y_1), (\phi_2, y_2), \dots \in \mathbb{R}^{Kd} \times \{-1, 1\}$ , satisfies*

$$m \leq \inf_{\mathbf{u} \in \mathbb{R}^{Kd}} \left( \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u}) + \frac{2(\mathbf{u}^\top A_\otimes \mathbf{u})}{K+1} + \sqrt{\frac{2(\mathbf{u}^\top A_\otimes \mathbf{u})}{K+1} \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u})} \right),$$

where  $\mathcal{M}$  is the set of mistaken trial indices, and

$$\mathbf{u}^\top A_\otimes \mathbf{u} = \sum_{i=1}^K \|\mathbf{u}_i\|^2 + \sum_{1 \leq i < j \leq K} \|\mathbf{u}_i - \mathbf{u}_j\|^2.$$

**Remark** Note that when all tasks are equal, that is when  $\mathbf{u}_1 = \dots = \mathbf{u}_K$ , the bound of Theorem 1 becomes the standard Perceptron mistake bound (see, e.g., [7]). In the general case of distinct  $\mathbf{u}_i$  we have

$$\frac{\mathbf{u}^\top A_\otimes \mathbf{u}}{K+1} < \sum_{i=1}^K \|\mathbf{u}_i\|^2 - \frac{1}{K+1} \sum_{1 \leq i, j \leq K} \mathbf{u}_i^\top \mathbf{u}_j.$$

The sum of squares  $\sum_i \|\mathbf{u}_i\|^2$  is the mistake bound one can prove when learning  $K$  independent Perceptrons (under linear separability assumptions). On the other hand, highly correlated reference task vectors (i.e., large inner products  $\mathbf{u}_i^\top \mathbf{u}_j$ ) imply a large negative second term in the right-hand side of the above expression.

Theorem 1 is immediately proven by using the fact that the multitask Perceptron is a specific instance of the kernel Perceptron algorithm [12] using the linear kernel introduced in [11] (see also [15]). As mentioned in the introduction, this kernel is defined as follows: for any positive definite  $K \times K$  interaction matrix  $A$  introduce the  $Kd$ -dimensional reproducing kernel Hilbert space  $(\mathbb{R}^{Kd}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$  with inner product  $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{H}} = \mathbf{u}^\top (A \otimes I_d) \mathbf{v}$ . Then define the kernel feature map  $\psi : \mathbb{R}^d \times \{1, \dots, K\} \rightarrow \mathcal{H}$  such that

$$\psi(\mathbf{x}_t, i_t) = (A \otimes I_d)^{-1} \phi_t. \quad (4)$$

The kernel used by the multitask Perceptron is thus defined by

$$\begin{aligned} K((\mathbf{x}_s, i_s), (\mathbf{x}_t, i_t)) &= \langle \psi(\mathbf{x}_s, i_s), \psi(\mathbf{x}_t, i_t) \rangle_{\mathcal{H}} \\ &= \phi_s^\top (A \otimes I_d)^{-1} \phi_t. \end{aligned} \quad (5)$$

**Proof of Theorem 1:** We use the following version of the kernel Perceptron bound (see, e.g., [7]),

$$\begin{aligned} m &\leq \sum_t \ell_t(f) + \|h\|_{\mathcal{H}}^2 \left( \max_t \|\psi(\mathbf{x}_t, i_t)\|_{\mathcal{H}}^2 \right) \\ &\quad + \|h\|_{\mathcal{H}} \sqrt{\left( \max_t \|\psi(\mathbf{x}_t, i_t)\|_{\mathcal{H}}^2 \right) \sum_t \ell_t(h)} \end{aligned}$$

where  $h$  is any function in the RKHS  $\mathcal{H}$  induced by the kernel. Let  $A_\otimes = A \otimes I_d$ . For the kernel (5) we have  $\|\mathbf{u}\|_{\mathcal{H}}^2 = \mathbf{u}^\top A_\otimes \mathbf{u}$  and  $\|\psi(\mathbf{x}_t, i_t)\|_{\mathcal{H}}^2 = \phi_t^\top A_\otimes^{-1} A_\otimes^{-1} \phi_t = \phi_t^\top A_\otimes^{-1} \phi_t = A_{i_t, i_t}^{-1}$ . Observing that  $A_{i_t, i_t}^{-1} = 2/(K+1)$  for the matrix  $A^{-1}$  defined in (3) concludes the proof. ■

### 3.2 A more general interaction matrix

In this section we slightly generalize the analysis of the previous section and consider an update rule of the form

$$\mathbf{w}_{j,s} = \mathbf{w}_{j,s-1} + \begin{cases} \frac{b+K}{(1+b)K} y_t \mathbf{x}_t & \text{if } j = i_t, \\ \frac{b}{(1+b)K} y_t \mathbf{x}_t & \text{otherwise,} \end{cases}$$

where  $b$  is a nonnegative parameter. The corresponding interaction matrix is given by

$$A = \frac{1}{K} \begin{bmatrix} a & -b & \dots & -b \\ -b & a & \dots & -b \\ \dots & \dots & \dots & \dots \\ -b & \dots & \dots & a \end{bmatrix}. \quad (6)$$

with  $a = K + b(K - 1)$ . It is immediate to see that the previous case (3) is recovered by choosing  $b = K$ . The inverse of (6) is

$$A^{-1} = \frac{1}{(1+b)K} \begin{bmatrix} b+K & b & \dots & b \\ b & b+K & \dots & b \\ \dots & \dots & \dots & \dots \\ b & \dots & \dots & b+K \end{bmatrix}.$$

When (6) is used in the multitask Perceptron algorithm, the proof of Theorem 1 can be adapted to prove the following result.

**Corollary 2** *The number of mistakes  $m$  made by the multitask Perceptron algorithm in Figure 1, run with interaction matrix (6) on any finite multitask sequence  $(\phi_1, y_1), (\phi_2, y_2), \dots \in \mathbb{R}^{Kd} \times \{-1, 1\}$ , satisfies*

$$m \leq \left( \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u}) + \frac{(b+K)}{(1+b)K} (\mathbf{u}^\top A_{\otimes} \mathbf{u}) \right) + \sqrt{\frac{(b+K)}{(1+b)K} (\mathbf{u}^\top A_{\otimes} \mathbf{u}) \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u})}$$

for any  $\mathbf{u} \in \mathbb{R}^{Kd}$ , where

$$\mathbf{u}^\top A_{\otimes} \mathbf{u} = \sum_{i=1}^K \|\mathbf{u}_i\|^2 + bK \text{VAR}[\mathbf{u}], \quad (7)$$

being  $\text{VAR}[\mathbf{u}] = \frac{1}{K} \sum_{i=1}^K \|\mathbf{u}_i - \bar{\mathbf{u}}\|^2$  the ‘‘variance’’, of the task vectors, and  $\bar{\mathbf{u}}$  the centroid  $(\mathbf{u}_1 + \dots + \mathbf{u}_K)/K$ .

It is interesting to investigate how the above bound depends on the trade-off parameter  $b$ . The optimal value of  $b$  (requiring prior knowledge about the distribution of  $\mathbf{u}_1, \dots, \mathbf{u}_K$ ) is

$$b = \max \left\{ 0, \sqrt{\frac{K-1}{K} \frac{\|\bar{\mathbf{u}}\|^2}{\text{VAR}[\mathbf{u}]} - 1} \right\}.$$

Thus  $b$  grows large as the reference task vectors  $\mathbf{u}_i$  get close to their centroid  $\bar{\mathbf{u}}$  (i.e., as all  $\mathbf{u}_i$  get close to each other). Substituting this choice of  $b$  gives

$$\frac{(b+K)}{(1+b)K} (\mathbf{u}^\top A_{\otimes} \mathbf{u})$$

$$= \begin{cases} \|\mathbf{u}_1\|^2 + \dots + \|\mathbf{u}_K\|^2 & \text{if } b = 0, \\ \left( \|\bar{\mathbf{u}}\| + \sqrt{K-1} \sqrt{\text{VAR}[\mathbf{u}]} \right)^2 & \text{otherwise.} \end{cases}$$

When the variance  $\text{VAR}[\mathbf{u}]$  is large (compared to the squared centroid norm  $\|\bar{\mathbf{u}}\|^2$ ), then the optimal tuning of  $b$  is zero and the interaction matrix becomes the identity matrix, which amounts to running  $K$  independent Perceptron algorithms. On the other hand, when the optimal tuning of  $b$  is nonzero we learn  $K$  reference vectors, achieving a mistake bound equal to that of learning a *single* vector whose length is  $\|\bar{\mathbf{u}}\|$  plus  $\sqrt{K-1}$  times the standard deviation  $\sqrt{\text{VAR}[\mathbf{u}]}$ .

At the other extreme, if the variance  $\text{VAR}[\mathbf{u}]$  is zero (namely, when all tasks coincide) then the optimal  $b$  grows unbounded, and the quadratic term  $\frac{(b+K)}{(1+b)K} (\mathbf{u}^\top A_{\otimes} \mathbf{u})$  tends to the average square norm  $\frac{1}{K} \sum_{i=1}^K \|\mathbf{u}_i\|^2$ . In this case the multitask algorithm becomes essentially equivalent to an algorithm that, before learning starts, chooses one task at random and keeps referring all instance vectors  $\mathbf{x}_t$  to that task (somehow implementing the fact that now the information conveyed by  $i_t$  can be disregarded).

### 3.3 Encoding prior knowledge

We could also pick the interaction matrix  $A$  so to encode prior knowledge about tasks. For instance, suppose we know that only certain pairs of tasks are potentially related. We represent this knowledge in a standard way through an undirected graph  $G = (V, E)$ , where two vertices  $i$  and  $j$  are connected by an edge if and only if we believe task  $i$  and task  $j$  are related. A natural choice for  $A$  is then  $A = I + L$ , where  $L = [L_{i,j}]_{i,j=1}^K$  is the Laplacian of  $G$ , defined as

$$L_{i,j} = \begin{cases} d_i & \text{if } i = j, \\ -1 & \text{if } (i, j) \in E, \\ 0 & \text{otherwise,} \end{cases}$$

where  $d_i$  is the degree (number of incoming edges) of node  $i$ . If we now follow the proof of Theorem 1, which holds for any positive definite matrix  $A$ , we obtain the following result.

**Corollary 3** *The number of mistakes  $m$  made by the multitask Perceptron algorithm in Figure 1, run with interaction matrix  $I + L$  on any finite multitask sequence  $(\phi_1, y_1), (\phi_2, y_2), \dots \in \mathbb{R}^{Kd} \times \{-1, 1\}$ , satisfies*

$$m \leq \inf_{\mathbf{u} \in \mathbb{R}^{Kd}} \left( \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u}) + c_G \mathbf{u}^\top (I + L)_{\otimes} \mathbf{u} + \sqrt{c_G \mathbf{u}^\top (I + L)_{\otimes} \mathbf{u} \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u})} \right)$$

where

$$\mathbf{u}^\top (I + L)_{\otimes} \mathbf{u} = \sum_{i=1}^K \|\mathbf{u}_i\|^2 + \sum_{(i,j) \in E} \|\mathbf{u}_i - \mathbf{u}_j\|^2 \quad (8)$$

and  $c_G = \max_{i=1, \dots, K} \sum_{j=1}^K \frac{v_{j,i}^2}{1+\lambda_j}$ . Here  $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_K$  are the eigenvalues of the positive semidef-

inite matrix  $L$ , and  $v_{j,i}$  denotes the  $i$ -th component<sup>3</sup> of the eigenvector  $\mathbf{v}_j$  of  $L$  associated with eigenvalue  $\lambda_j$ .

**Proof:** Following the proof of Theorem 1, we just need to bound

$$\max_{i=1,\dots,K} A_{i,i}^{-1} = \max_{i=1,\dots,K} (I + L)_{i,i}^{-1}.$$

If  $\mathbf{v}_1, \dots, \mathbf{v}_K$  are the eigenvectors of  $L$ , then

$$(I + L)^{-1} = \sum_{j=1}^K \frac{\mathbf{v}_j \mathbf{v}_j^\top}{1 + \lambda_j}$$

which concludes the proof.  $\blacksquare$

Ideally, we would like to have  $c_G = \mathcal{O}(1/K)$ . Clearly enough, if  $G$  is the clique on  $K$  vertices we expect to exactly recover the bound of Theorem 1. In fact, we can easily verify that the eigenvector  $\mathbf{v}_1$  associated with the zero eigenvalue  $\lambda_1$  is  $(K^{-1/2}, \dots, K^{-1/2})$ . Moreover, it is well known that all the remaining eigenvalues are equal to  $K$  (see, e.g., [16]). Therefore  $c_G = \frac{1}{K} + (1 - \frac{1}{K}) \frac{1}{K+1} = \frac{2}{K+1}$ . In the case of more general graphs  $G$ , we can bound  $c_G$  in terms of the smallest nonzero eigenvalue  $\lambda_2$ ,

$$c_G \leq \frac{1}{K} + \left(1 - \frac{1}{K}\right) \frac{1}{1 + \lambda_2}.$$

The value of  $\lambda_2$ , known as the algebraic connectivity of  $G$ , is 0 only when the graph is disconnected.  $\lambda_2$  is known for certain families of graphs. For instance, if  $G$  is a complete bipartite graph (i.e., if tasks can be divided in two disjoint subsets  $T_1$  and  $T_2$  such that every task in  $T_1$  is related to every task in  $T_2$  and for both  $i = 1, 2$  no two tasks in  $T_i$  are related), then it is known that  $\lambda_2 = \min\{|T_1|, |T_2|\}$ . We refer the reader to, e.g., [16] for further examples.

The advantage of using a graph  $G$  with significantly fewer edges than the clique is that the sum of pairwise distances in (8) will contain less than  $K(K-1)$  terms. On the other hand, this reduction is balanced by a larger coefficient  $c_G$  in front of  $\mathbf{u}^\top (I + L)_{\otimes} \mathbf{u}$ . This coefficient, in general, is related to the total number of edges in the graph (observe that the trace of  $L$  is exactly twice this total number).

## 4 The second-order extension

In this section we consider the second-order kernel Perceptron algorithm of [7] with the multitask kernel (5). The algorithm, which is described in Figure 2, maintains in its internal state a matrix  $S$  (initialized to the empty matrix) and a multitask Perceptron weight vector  $\mathbf{v}$  (initialized to the zero vector). Just like in Figure 1, we use the subscript  $s$  to denote the current number of mistakes plus one. Note that we have exploited the linearity of the kernel (5) to simplify the description of the algorithm. In particular, letting  $A_{\otimes} = A \otimes I_d$ , we have repeatedly used the fact that

$$\begin{aligned} \langle \psi(\mathbf{x}_s, i_s), \psi(\mathbf{x}_t, i_t) \rangle_{\mathcal{H}} &= \phi_s^\top A_{\otimes}^{-1} \phi_t \\ &= \left( A_{\otimes}^{-1/2} \phi_s \right)^\top \left( A_{\otimes}^{-1/2} \phi_t \right) \\ &= \tilde{\phi}_s^\top \tilde{\phi}_t, \end{aligned}$$

<sup>3</sup>Note that the orthonormality of the eigenvectors imply  $v_{1,i}^2 + \dots + v_{K,i}^2 = 1$  for all  $i$ .

**Parameters:** Positive definite  $K \times K$  interaction matrix  $A$ .  
**Initialization:**  $S_0 = \emptyset$ ,  $\mathbf{v}_0 = \mathbf{0} \in \mathbb{R}^{Kd}$ ,  $s = 1$ .

At each time  $t = 1, 2, \dots$  do the following:

1. Observe task number  $i_t \in \{1, \dots, K\}$  and the corresponding instance vector  $\mathbf{x}_t \in \mathbb{R}^d$ ;
2. Build the associated multitask instance  $\phi_t \in \mathbb{R}^{Kd}$  and compute  $\tilde{\phi}_t = (A \otimes I_d)^{-1/2} \phi_t$ ;
3. Predict  $\hat{y}_t = \text{SGN}(\mathbf{w}_{s-1}^\top \tilde{\phi}_t) \in \{-1, +1\}$ , where  $\mathbf{w}_{s-1} = \left( I + S_{s-1} S_{s-1}^\top + \tilde{\phi}_t \tilde{\phi}_t^\top \right)^{-1} \mathbf{v}_{s-1}$ ;
4. Get the label  $y_t \in \{-1, 1\}$ ;
5. If  $\hat{y}_t \neq y_t$  then update:
 
$$\mathbf{v}_s = \mathbf{v}_{s-1} + y_t \tilde{\phi}_t, \quad S_s = \left[ S_{s-1} \mid \tilde{\phi}_t \right], \quad s \leftarrow s+1.$$

Figure 2: The second-order multitask Perceptron algorithm.

where  $\psi$  is the kernel feature map (4). The algorithm computes a tentative (inverse) matrix

$$\left( I + S_{s-1} S_{s-1}^\top + \tilde{\phi}_t \tilde{\phi}_t^\top \right)^{-1}.$$

Such a matrix is combined with the current Perceptron vector  $\mathbf{v}_{s-1}$  to predict the label  $y_t$ . If prediction  $\hat{y}_t$  and label  $y_t$  disagree both  $\mathbf{v}$  and  $S$  get updated (no update takes place otherwise). In particular, the new matrix  $S_s$  is augmented by padding with the current vector  $\tilde{\phi}_t$ . Since supports are shared, the computational cost of an update is not significantly larger than that for learning a single-task (see Section 4.1).

**Theorem 4** *The number of mistakes  $m$  made by the second-order multitask Perceptron of Figure 2, run with any positive definite interaction matrix  $A$ , on any finite multitask sequence  $(\phi_1, y_1), (\phi_2, y_2), \dots \in \mathbb{R}^{Kd} \times \{-1, 1\}$ , satisfies, for all  $\mathbf{u} \in \mathbb{R}^{Kd}$ ,*

$$\begin{aligned} m &\leq \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u}) \\ &+ \sqrt{\left( \mathbf{u}^\top (A \otimes I_d) \mathbf{u} + \sum_{t \in \mathcal{M}} (\mathbf{u}_{i_t}^\top \mathbf{x}_t)^2 \right) \sum_{j=1}^m \ln(1 + \lambda_j)} \end{aligned}$$

where  $\mathcal{M}$  is the sequence of mistaken trial indices and  $\lambda_1, \dots, \lambda_m$  are the eigenvalues of the  $m \times m$  matrix of elements  $\mathbf{x}_s^\top A_{i_s, i_t}^{-1} \mathbf{x}_t$ , where  $s, t \in \mathcal{M}$ .

**Proof:** Recall the mistake bound for the second-order kernel Perceptron algorithm [7]:

$$m \leq \sqrt{\left( \|h\|_{\mathcal{H}}^2 + \sum_{t \in \mathcal{M}} h(\mathbf{x}_t)^2 \right) \sum_{j=1}^m \ln(1 + \lambda_j)}$$

where  $\lambda_1, \dots, \lambda_m$  are the eigenvalues of the  $m \times m$  kernel Gram (sub)matrix including only time steps in  $\mathcal{M}$ . When the

kernel is (5) with feature map  $f$  we have  $\|\mathbf{u}\|_{\mathcal{H}}^2 = \mathbf{u}^\top A_{\otimes} \mathbf{u}^\top$  and  $\langle \mathbf{u}^\top, \psi(\mathbf{x}_t, i_t) \rangle^2 = (\mathbf{u}^\top A_{\otimes} A_{\otimes}^{-1} \phi_t)^2 = (\mathbf{u}_{i_t}^\top \mathbf{x}_t)^2$ . Finally, the kernel Gram matrix is  $K(\psi(\mathbf{x}_s, i_s), \psi(\mathbf{x}_t, i_t)) = \phi_s^\top A_{\otimes}^{-1} \phi_t = \mathbf{x}_s^\top A_{i_s, i_t}^{-1} \mathbf{x}_t$ . This concludes the proof. ■

Again, this bound should be compared to the one obtained when learning  $K$  independent tasks. As in the first-order algorithm, we have the complexity term  $\mathbf{u}^\top (A \otimes I_d) \mathbf{u}$ . In this case, however, the interaction matrix  $A$  also plays a role in the scale of the eigenvalues of the resulting multitask Gram matrix. Roughly speaking, we gain a factor  $K$  from  $\mathbf{u}^\top A_{\otimes}^{-1} \mathbf{u}$  (according to the arguments in Section 3). In addition, however, we gain a further factor  $K$ , since the trace of the multitask Gram matrix  $[\phi_s^\top A_{\otimes}^{-1} \phi_t]_{s,t \in \mathcal{M}} = [\mathbf{x}_s^\top A_{i_s, i_t}^{-1} \mathbf{x}_t]_{s,t \in \mathcal{M}}$  is about  $1/K$  times the trace of the original Gram matrix  $[\mathbf{x}_s^\top \mathbf{x}_t]_{s,t \in \mathcal{M}}$ . Since both factors are under the square root, the resulting gain over the  $K$  independent task bound is about  $K$ .

#### 4.1 Implementation in dual variables

It is easy to see that the second-order multitask Perceptron can be run in dual variables by maintaining  $K$  classifiers that share the same set of support vectors. This allows an efficient implementation that does not impose any significant overhead with respect to the corresponding single-task version. Specifically, given some interaction matrix  $A$ , the margin at time  $t$  is computed as (see [7, Theorem 3.3])

$$\begin{aligned} \mathbf{w}_{s-1}^\top \tilde{\phi}_t &= \mathbf{v}_{s-1}^\top \left( I + S_{s-1} S_{s-1}^\top + \tilde{\phi}_t \tilde{\phi}_t^\top \right)^{-1} \tilde{\phi}_t \\ &= \mathbf{y}_s^\top \left( I + S_s^\top S_s \right)^{-1} S_s^\top \tilde{\phi}_t, \end{aligned} \quad (9)$$

where  $\mathbf{y}_s$  is the  $s$ -dimensional vector whose first  $s-1$  components are the labels  $y_i$  where the algorithm has made a mistake up to time  $t-1$ , and the last component is 0.

First, note that replacing  $I + S_s^\top S_s$  with  $I + S_{s-1}^\top S_{s-1}$  in (9) does not change the sign of the prediction. The margin at time  $t$  can then be computed by calculating the scalar product between  $S_s^\top \tilde{\phi}_t$  and  $\mathbf{y}_s^\top (I + S_{s-1}^\top S_{s-1})^{-1}$ . Now, each entry of the vector  $S_s^\top \tilde{\phi}_t$  is of the form  $A_{j, i_t}^{-1} \mathbf{x}_j^\top \mathbf{x}_t$ , and thus computing  $S_s^\top \tilde{\phi}_t$  requires  $\mathcal{O}(s)$  inner products so that, overall, the prediction step requires  $\mathcal{O}(s)$  scalar multiplications and  $\mathcal{O}(s)$  inner products (independent of the number of tasks  $K$ ).

On the other hand, the update step involves the computation of the vector  $\mathbf{y}_s^\top (I + S_s^\top S_s)^{-1}$ . For the matrix update we can write

$$I_s + S_s^\top S_s = \begin{bmatrix} I_{s-1} + S_{s-1}^\top S_{s-1} & S_{s-1}^\top \tilde{\phi}_t \\ \tilde{\phi}_t^\top S_{s-1} & 1 + \tilde{\phi}_t^\top \tilde{\phi}_t \end{bmatrix}.$$

Using standard facts about the inverse of partitioned matrices (e.g., [17, Ch. 0]), one can see that the inverse of matrix  $I_s + S_s^\top S_s$  can be computed from the inverse of  $I_{s-1} + S_{s-1}^\top S_{s-1}$  with  $\mathcal{O}(s)$  extra inner products (again, independent of  $K$ ) and  $\mathcal{O}(s^2)$  additional scalar multiplications.

## 5 The $p$ -norm extension

We now extend our multitask results to the  $p$ -norm Perceptron algorithm of [14, 13]. As before, when the tasks are all equal we want to recover the bound of the single-task algorithm, and when the task vectors are different we want the mistake bound to increase according to a function that penalizes task diversity according to their  $p$ -norm distance.

We develop our  $p$ -norm multitask analysis for the specific choices of  $p = 2 \ln d$  (or  $p = 2 \ln K$  when  $d \leq K$ ) and for the pairwise distance matrix (3). It is well known that for  $p = 2 \ln d$  the mistake bound of the single-task  $p$ -norm Perceptron is essentially equivalent to the one of the zero-threshold Winnow algorithm of [22]. We now see that this property is preserved in the multitask extension.

We start with the following slightly more general algorithm based on arbitrary norms. Later, we specialize it to  $p$ -norms. The *quasi-additive multitask algorithm* of [20, 14] is defined for any norm  $\|\cdot\|$  over  $\mathbb{R}^{Kd}$ . Initially,  $\mathbf{w}_0 = \mathbf{0} \in \mathbb{R}^{Kd}$ . If  $s-1$  mistakes have been made in the first  $t-1$  time steps, then the prediction at time  $t$  is  $\text{SGN}(\mathbf{w}_{s-1}^\top \phi_t)$ . If a mistake occurs at time  $t$ , then  $\mathbf{w}_{s-1}$  is updated with the rule  $\mathbf{w}_s = \nabla_{\frac{1}{2}} \|\mathbf{v}_s\|^2$ , where the *primal weight*  $\mathbf{v}_s \in \mathbb{R}^{Kd}$  is updated using the multitask Perceptron rule,  $\mathbf{v}_0 = \mathbf{0} \in \mathbb{R}^{Kd}$  and  $\mathbf{v}_s = \mathbf{v}_{s-1} + y_t A_{\otimes}^{-1} \phi_t$  for an arbitrary positive definite interaction matrix  $A$ .

This can be analyzed using the following technique (see [8] for details). Let  $\mathbf{v}_m$  be the primal weight after any number  $m$  of mistakes. Then, by Taylor expanding  $\frac{1}{2} \|\mathbf{v}_s\|^2$  around  $\mathbf{v}_{s-1}$  for each  $s = 1, \dots, m$ , and using the fact  $y_t \mathbf{w}_{s-1}^\top \phi_t \leq 0$  whenever a mistake occurs at step  $t$ , we get

$$\frac{1}{2} \|\mathbf{v}_m\|^2 \leq \sum_{s=1}^m D(\mathbf{v}_s \| \mathbf{v}_{s-1}) \quad (10)$$

$D(\mathbf{v}_s \| \mathbf{v}_{s-1}) = \frac{1}{2} (\|\mathbf{v}_s\|^2 - \|\mathbf{v}_{s-1}\|^2) - y_t \mathbf{w}_{s-1}^\top \mathbf{x}_t$  is a so-called Bregman divergence; i.e., the error term in the first-order Taylor expansion of  $\frac{1}{2} \|\cdot\|^2$  around vector  $\mathbf{v}_{s-1}$ , at vector  $\mathbf{v}_s$ .

Fix any  $\mathbf{u} \in \mathbb{R}^{Kd}$ . Using the convex inequality for norms  $\mathbf{u}^\top \mathbf{v} \leq \|\mathbf{u}\| \|\mathbf{v}\|_*$  where  $\|\cdot\|_*$  is the dual norm of  $\|\cdot\|$  (see, e.g., [25, page 131]), and using the fact  $\mathbf{u}^\top A_{\otimes} \mathbf{v}_s = \mathbf{u}^\top A_{\otimes} \mathbf{v}_{s-1} + y_t \mathbf{u}^\top \phi_t \geq \mathbf{u}^\top A_{\otimes} \mathbf{v}_{s-1} + 1 - \ell_t(\mathbf{u})$ , one then obtains

$$\|\mathbf{v}_m\| \geq \frac{\mathbf{u}^\top A_{\otimes} \mathbf{v}_m}{\|A_{\otimes} \mathbf{u}\|_*} \geq \frac{m - \sum_t \ell_t(\mathbf{u})}{\|A_{\otimes} \mathbf{u}\|_*}. \quad (11)$$

Combining (10) with (11) and solving for  $m$  gives

$$m \leq \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u}) + \|A_{\otimes} \mathbf{u}\|_* \sqrt{2 \sum_{s=1}^m D(\mathbf{v}_s \| \mathbf{v}_{s-1})}. \quad (12)$$

We obtain our multitask version of the  $p$ -norm Perceptron when  $\|\mathbf{u}\| = \|\mathbf{u}\|_p = (|u_1|^p + |u_2|^p + \dots)^{1/p}$ . In particular, we focus our analysis on the choice  $p = 2 \ln \max\{K, d\}$ , which gives mistake bounds in the dual norms  $\|\mathbf{u}\|_1$  and  $\|\mathbf{x}_t\|_{\infty}$ , and on the pairwise distance matrix (3).

Using the analysis in [8] we obtain, for  $t_s = t$ ,

$$D(\mathbf{v}_s \| \mathbf{v}_{s-1}) \leq \frac{p-1}{2} \|A_{\otimes}^{-1} \phi_t\|_p^2 = \frac{p-1}{2} \|\mathbf{x}_t\|_p^2 \|A_{\downarrow i_t}^{-1}\|_p^2$$

where  $A_{\downarrow i_t}^{-1}$  is the  $i_t$ -th column of  $A^{-1}$ . If we now use  $p = 2 \ln \max\{K, d\}$ , then  $\|\mathbf{x}_t\|_p^2 \leq e \|\mathbf{x}_t\|_\infty^2$  and

$$\left\|A_{\downarrow i_t}^{-1}\right\|_p^2 \leq e \left\|A_{\downarrow i_t}^{-1}\right\|_\infty^2 = e (A_{i_t, i_t}^{-1})^2 = \frac{4e}{(K+1)^2}.$$

We now turn to the computation of the dual norm  $\|A_{\otimes} \mathbf{u}\|_q$ , where  $q = p/(p-1)$  is the dual coefficient of  $p$ . We find that

$$\|A_{\otimes} \mathbf{u}\|_q^2 \leq \|A_{\otimes} \mathbf{u}\|_1^2 = \left( \sum_{i=1}^K \|\mathbf{u}_i + \sum_{j \neq i} (\mathbf{u}_i - \mathbf{u}_j)\|_1 \right)^2.$$

Plugging back into (12) gives the following theorem.

**Theorem 5** *The number of mistakes  $m$  made by the  $p$ -norm multitask Perceptron, run with the pairwise distance matrix (3) and  $p = 2 \ln \max\{K, d\}$ , on any finite multitask sequence  $(\phi_1, y_1), (\phi_2, y_2), \dots \in \mathbb{R}^{Kd} \times \{-1, 1\}$ , satisfies, for all  $\mathbf{u} \in \mathbb{R}^{Kd}$ ,*

$$m \leq \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u}) + H + \sqrt{2H \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u})}$$

where  $H$  is equal to

$$\frac{4e^2 \ln \max\{K, d\}}{(K+1)^2} X_\infty^2 \left( \sum_{i=1}^K \|\mathbf{u}_i + \sum_{j \neq i} (\mathbf{u}_i - \mathbf{u}_j)\|_1 \right)^2.$$

and  $X_\infty = \max_{t \in \mathcal{M}} \|\mathbf{x}_t\|_\infty$ .

**Remark** When all tasks are equal,  $\mathbf{u}_1 = \dots = \mathbf{u}_K$ , the coefficient  $H$  in the bound of Theorem 5 becomes

$$(4e^2 \ln \max\{K, d\}) \left( \max_t \|\mathbf{x}_t\|_\infty \right)^2 \|\mathbf{u}_i\|_1^2.$$

If  $K \leq d$  this bound is equivalent (apart from constant factors) to the mistake bound for the single-task zero-threshold Winnow algorithm of [22].

## 6 Learning tasks in heterogeneous spaces

In this section we slightly deviate from the approach followed so far. We consider the case when the  $K$  task vectors  $\mathbf{u}_i$  may live in different spaces:  $\mathbf{u}_i \in \mathbb{R}^{d_i}$ ,  $i = 1, \dots, K$ . This is a plausible assumption when attributes associated with different tasks have a completely different meaning. In such a case, the correlation among tasks is naturally measured through the task margins  $\mathbf{u}_i^\top \mathbf{x}$  (or *views*)—see, e.g., the previous work of [6, 27] for a similar approach in the context of semi-supervised learning. In order to allow for the views to interact in a meaningful way, we slightly modify the learning protocol of Section 2. We now assume that, at each time  $t$ , we receive the adversarial choice of task  $i_t$  together with *all* instance vectors  $\mathbf{x}_{i,t} \in \mathbb{R}^{d_i}$  for  $i = 1, \dots, K$ . The co-regularization terms motivating our algorithm are proportional to the distance between the margin  $\mathbf{u}_{i_t}^\top \mathbf{x}_{i_t, t}$  of task  $i_t$  and the average margin  $\frac{1}{K} \sum_{j=1}^K \mathbf{u}_j^\top \mathbf{x}_{j, t}$  of all tasks:

$$bK \left( \mathbf{u}_{i_t}^\top \mathbf{x}_{i_t, t} - \frac{1}{K} \sum_{j=1}^K \mathbf{u}_j^\top \mathbf{x}_{j, t} \right)^2, \quad (13)$$

where  $b$  is a positive constant. We add the above terms up to time  $t$ , resulting in a cumulative regularization term

$$bK \sum_{s \in \mathcal{M}_t} \left( \mathbf{u}_{i_s}^\top \mathbf{x}_{i_s, s} - \frac{1}{K} \sum_{j=1}^K \mathbf{u}_j^\top \mathbf{x}_{j, s} \right)^2,$$

where  $\mathcal{M}_t$  is the set of mistaken trials up to time  $t$ . Thus, in trial  $t$  our prior knowledge about task relatedness is encoded as a (positive) correlation among the  $K$  margin sequences  $(\mathbf{u}_j^\top \mathbf{x}_{j, 1}, \mathbf{u}_j^\top \mathbf{x}_{j, 2}, \dots, \mathbf{u}_j^\top \mathbf{x}_{j, t})$ ,  $j = 1, \dots, K$ .

The algorithm of Figure 3 is a natural multitask predictor operating with the above view-based regularization criterion. We call this algorithm the multiview-based multitask Perceptron algorithm, or MMPERC for brevity. MMPERC can be viewed as a variant of the second-order Perceptron using the cumulative covariance matrix of the past margin vectors in order to suitably transform instances.

MMPERC has a constant tradeoff parameter  $b > 0$  (playing the same role as the one in Section 3.2), and maintains in its internal state a multitask matrix  $A$  (initialized to the identity matrix  $I$ ) and a Perceptron multitask weight vector  $\mathbf{v}$  (initialized to the zero vector). The subscript  $s$  plays the same role as in the previous algorithms. Unlike the algorithms in previous sections, MMPERC observes the task number  $i_t$  and the multitask instance  $\Phi_t^\top = (\mathbf{x}_{1, t}^\top, \mathbf{x}_{2, t}^\top, \dots, \mathbf{x}_{K, t}^\top)$  made up of the instance vectors of all tasks. Then MMPERC computes a tentative matrix  $A'_t$  to be used for prediction. Matrix  $A'_t$  is obtained by adding the rank-one positive semidefinite matrix  $M_t$  to the previous matrix  $A_{s-1}$ . Here  $\phi_{j, t}$  is the  $(d_1 + \dots + d_K)$ -dimensional vector

$$\phi_{j, t}^\top = \left( \underbrace{0, \dots, 0}_{d_1 + \dots + d_{j-1} \text{ times}} \quad \mathbf{x}_{j, t}^\top \quad \underbrace{0, \dots, 0}_{d_{j+1} + \dots + d_K \text{ times}} \right)$$

for  $j = 1, \dots, K$ . Observe that  $M_t$  has been set so as to make the quadratic form  $\mathbf{u}^\top M_t \mathbf{u}$  coincide with the regularization term (13). Similarly to the algorithms of previous sections, the tentative matrix  $A'_t$  and the current Perceptron vector  $\mathbf{v}_{s-1}$  are used for predicting the true label  $y_t$ . If prediction  $\hat{y}_t$  and label  $y_t$  disagree both  $\mathbf{v}$  and  $A$  get updated. In particular,  $A_s$  is set to the tentative matrix  $A'_t$ .

In this protocol we call example the triple  $(i_t, \Phi_t, y_t)$ . Like the results contained in the previous sections, our analysis will provide a multitask bound on the number of prediction mistakes which is comparable to the one obtained by a single task plus a penalization term due to task relatedness. However, though this algorithm is a second-order prediction method, we only give a first-order analysis that disregards the eigenstructure of the data. This is due to the technical difficulty of handling a time-varying matrix  $A$  that in trial  $t$  includes all instance vectors  $\mathbf{x}_{1, t}, \dots, \mathbf{x}_{K, t}$ .

**Theorem 6** *The number of mistakes  $m$  made by the algorithm in Figure 3, run on any multitask sequence  $(i_1, \Phi_1, y_1), (i_2, \Phi_2, y_2), \dots$  satisfies, for all  $\mathbf{u}^\top =$*

**Parameters:**  $b > 0$ .

**Initialization:**  $A_0 = I$ ,  $\mathbf{v}_0 = \mathbf{0} \in \mathbb{R}^{d_1 + \dots + d_K}$ ,  $s = 1$ .

At each time  $t = 1, 2, \dots$  do the following:

1. Observe task number  $i_t \in \{1, \dots, K\}$ ;

2. Observe multitask instance vector

$$\Phi_t^\top = (\mathbf{x}_{1,t}^\top, \dots, \mathbf{x}_{K,t}^\top) \in \mathbb{R}^{d_1 + \dots + d_K};$$

3. Build the associated multitask instance  $\phi_{i_t,t}$ ;

4. Set  $A'_t = A_{s-1} + M_t$  where

$$M_t = bK \left( \phi_{i_t,t} - \frac{\Phi_t}{K} \right) \left( \phi_{i_t,t} - \frac{\Phi_t}{K} \right)^\top;$$

5. Predict  $\hat{y}_t = \text{SGN}(\mathbf{v}_{s-1}^\top (A'_t)^{-1} \phi_{i_t,t}) \in \{-1, +1\}$ ;

6. Get label  $y_t \in \{-1, +1\}$ ;

7. If  $\hat{y}_t \neq y_t$  then update:

$$\mathbf{v}_s = \mathbf{v}_{s-1} + y_t \phi_{i_t,t}, \quad A_s = A'_t, \quad s \leftarrow s + 1.$$

Figure 3: The multiview-based multitask Perceptron algorithm (MMPERC).

$$(\mathbf{u}_1^\top, \dots, \mathbf{u}_K^\top),$$

$$m \leq \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u}) + \frac{K(b+1) - b}{bK(K-1) + K} (\mathbf{u}^\top A_m \mathbf{u}) + \sqrt{\frac{K(b+1) - b}{bK(K-1) + K} (\mathbf{u}^\top A_m \mathbf{u}) \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u})},$$

where

$$\begin{aligned} & \mathbf{u}^\top A_m \mathbf{u} \\ &= \sum_{i=1}^K \|\mathbf{u}_i\|^2 + bK \sum_{t \in \mathcal{M}} \left( \mathbf{u}_{i_t}^\top \mathbf{x}_{i_t,t} - \frac{1}{K} \sum_{j=1}^K \mathbf{u}_j^\top \mathbf{x}_{j,t} \right)^2, \end{aligned}$$

being  $\mathcal{M}$  the set of mistaken trials.

**Remark** It is the factor

$$\frac{K(b+1) - b}{bK(K-1) + K} (\mathbf{u}^\top A_m \mathbf{u}) \quad (14)$$

that quantifies the relatedness among tasks (the leading constant  $bK$  in the second term of  $\mathbf{u}^\top A_m \mathbf{u}$  is needed for scaling purposes). Note that the notion of relatedness provided by  $\mathbf{u}^\top A_m \mathbf{u}$  is analogous to the one used in Section 3.2. As suggested in Section 3.3, other measures of similarity are possible.

The parameter  $b$  allows for a limited trade-off between  $\sum_{i=1}^K \|\mathbf{u}_i\|^2$  and the cumulative ‘‘margin deviation’’

$$\sum_{t \in \mathcal{M}} \left( \mathbf{u}_{i_t}^\top \mathbf{x}_{i_t,t} - \frac{1}{K} \sum_{j=1}^K \mathbf{u}_j^\top \mathbf{x}_{j,t} \right)^2. \quad (15)$$

In particular, setting  $b = 0$  corresponds to running  $K$  independent (first-order) Perceptron algorithms (no task relatedness), while letting  $b$  go to infinity is optimal only when each

one of the margin deviation terms in (15) is zero (maximal task relatedness). Notice that setting  $b = 1$  gives

$$(14) = \frac{2K-1}{K^2} \sum_{i=1}^K \|\mathbf{u}_i\|^2 + \frac{2K-1}{K} \sum_{t \in \mathcal{M}} \left( \mathbf{u}_{i_t}^\top \mathbf{x}_{i_t,t} - \frac{1}{K} \sum_{j=1}^K \mathbf{u}_j^\top \mathbf{x}_{j,t} \right)^2$$

yielding a gain of  $K$  whenever the  $K$  tasks are significantly related, as measured by (15).

**Proof of Theorem 6:** Although the general structure of the analysis is based on the second-order Perceptron proof, we use the special properties of  $I + M_t$  in order to compute the contribution of  $K$  and  $b$  to the final bound.

Let  $t = t_s$  be the time step when the  $s$ -th mistake occurs. We write

$$\begin{aligned} \mathbf{v}_s^\top A_s^{-1} \mathbf{v}_s &= (\mathbf{v}_{s-1} + y_t \phi_{i_t,t})^\top A_s^{-1} (\mathbf{v}_{s-1} + y_t \phi_{i_t,t}) \\ &\quad (\text{from the update rule in Figure 3}) \\ &= \mathbf{v}_{s-1}^\top A_s^{-1} \mathbf{v}_{s-1} + 2y_t \mathbf{v}_{s-1}^\top A_s^{-1} \phi_{i_t,t} \\ &\quad + \phi_{i_t,t}^\top A_s^{-1} \phi_{i_t,t} \\ &\leq \mathbf{v}_{s-1}^\top A_s^{-1} \mathbf{v}_{s-1} + \phi_{i_t,t}^\top A_s^{-1} \phi_{i_t,t} \\ &\leq \mathbf{v}_{s-1}^\top A_{s-1}^{-1} \mathbf{v}_{s-1} + \phi_{i_t,t}^\top (I + M_t)^{-1} \phi_{i_t,t}. \end{aligned} \quad (16)$$

In order to prove the first inequality, note that on the  $s$ -th mistaken trial  $A_s = A'_t$  and  $y_t \mathbf{v}_{s-1}^\top (A'_t)^{-1} \phi_{i_t,t} \leq 0$ . In order to prove the second inequality note that  $A_s - A_{s-1}$  and  $A_s - (I + M_t)$  are both positive semidefinite.

We now focus on computing the quadratic form  $\phi_{i_t,t}^\top (I + M_t)^{-1} \phi_{i_t,t}$ . Recall that  $\Phi_t = \sum_{j=1}^K \phi_{j,t}$  is the sum of the orthonormal vectors  $\phi_{1,t}, \dots, \phi_{i_t,t}, \dots, \phi_{K,t}$ . Thus, from the very definition of  $M_t$  in Figure 3 it is easy to verify that

$$(I + M_t) \phi_{i_t,t} = (b(K-1) + 1) \phi_{i_t,t} - \frac{b(K-1)}{K} \Phi_t. \quad (17)$$

Also, since  $M_t \Phi_t = 0$ , we have  $(I + M_t) \Phi_t = \Phi_t$ , and thus  $(I + M_t)^{-1} \Phi_t = \Phi_t$ . Hence (17) allows us to write

$$\phi_{i_t,t} = (b(K-1) + 1) (I + M_t)^{-1} \phi_{i_t,t} - \frac{b(K-1)}{K} \Phi_t.$$

Taking the inner product of both sides with  $\phi_{i_t,t}$ , and solving for  $\phi_{i_t,t}^\top (I + M_t)^{-1} \phi_{i_t,t}$  yields

$$\phi_{i_t,t}^\top (I + M_t)^{-1} \phi_{i_t,t} = \frac{K(b+1) - b}{bK(K-1) + K}.$$

Substituting this into (16), recalling that  $\mathbf{v}_0 = \mathbf{0}$ , and summing over  $s = 1, \dots, m$  we obtain

$$\mathbf{v}_m^\top A_m^{-1} \mathbf{v}_m \leq m \frac{K(b+1) - b}{bK(K-1) + K}.$$

A lower bound on the left-hand side can be obtained in the standard way (details omitted),

$$\sqrt{\mathbf{v}_m^\top A_m^{-1} \mathbf{v}_m} \geq \frac{m - \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u})}{\|A_m^{1/2} \mathbf{u}\|}.$$

Thus we get  $\mathbf{v}_m^\top A_m^{-1} \mathbf{v}_m \geq \frac{(m - \sum_{t \in \mathcal{M}} \ell_t(\mathbf{u}))^2}{\mathbf{u}^\top A_m \mathbf{u}}$ . Finally, recall that by construction

$$\begin{aligned} & \mathbf{u}^\top A_m \mathbf{u} \\ &= \sum_{i=1}^K \|\mathbf{u}_i\|^2 + bK \sum_{t \in \mathcal{M}} \left( \mathbf{u}_{i_t}^\top \mathbf{x}_{i_t, t} - \frac{1}{K} \sum_{j=1}^K \mathbf{u}_j^\top \mathbf{x}_{j, t} \right)^2. \end{aligned}$$

Putting together and solving for  $m$  gives the desired bound. ■

## 6.1 Implementation in dual variables

Like the second-order multitask Perceptron algorithm, also MMPERC can be formulated in dual variables. Due to the need to handle  $K$  instance vectors at a time, the implementation we sketch below has an extra linear dependence on  $K$ , as compared to the one in Section 4.1.

Let  $t = t_s$  be the trial when the  $s$ -th mistake occurs,  $\mathbf{z}_r$  be the vector  $\mathbf{z}_r = \sqrt{bK}(\phi_{i_r, r} - \Phi_r/K)$ ,  $S_s$  be the matrix whose columns are the vectors  $\phi_{i_r, r}$  corresponding to mistaken time steps  $r$  up to time  $t$ , and  $Z_s$  be the matrix whose columns are the vectors  $\mathbf{z}_r$  corresponding to mistaken time steps  $r$  up to time  $t$ . It is easy to verify that  $A_s = I + Z_s Z_s^\top$  and  $\mathbf{v}_{s-1} = S_s \mathbf{y}_s$  where  $\mathbf{y}_s$  is as in Section 4.1. From the inversion formula (e.g., [17, Ch. 0])

$$(I + Z_s Z_s^\top)^{-1} = I - Z_s (I + Z_s^\top Z_s)^{-1} Z_s^\top$$

we see that the margin  $\mathbf{v}_{s-1}^\top (A_t')^{-1} \phi_{i_t, t}$  in Figure 3 can be computed as

$$\begin{aligned} & \mathbf{v}_{s-1}^\top (A_t')^{-1} \phi_{i_t, t} \\ &= \mathbf{y}_s^\top S_s^\top \phi_{i_t, t} - \mathbf{y}_s^\top S_s^\top Z_s (I + Z_s^\top Z_s)^{-1} Z_s^\top \phi_{i_t, t}. \end{aligned}$$

Calculating the vectors  $S_s^\top \phi_{i_t, t}$  and  $Z_s^\top \phi_{i_t, t}$  in the above expression takes  $\mathcal{O}(s)$  inner products while other  $\mathcal{O}(s)$  inner products are required to incrementally compute  $S_s^\top Z_s$  from  $S_{s-1}^\top Z_{s-1}$ . Finally, when calculating the inverse  $(I + Z_s^\top Z_s)^{-1}$  we exploit the same updating scheme of Section 4.1,

$$I_s + Z_s^\top Z_s = \begin{bmatrix} I_{s-1} + Z_{s-1}^\top Z_{s-1} & Z_{s-1}^\top \mathbf{z}_s \\ \mathbf{z}_s^\top Z_{s-1} & 1 + \mathbf{z}_s^\top \mathbf{z}_s \end{bmatrix},$$

where  $Z_{s-1}^\top \mathbf{z}_s$  and  $\mathbf{z}_s^\top \mathbf{z}_s$  require  $\mathcal{O}(Ks)$  and  $\mathcal{O}(K)$  inner products, respectively. Hence  $(I_s + Z_s^\top Z_s)^{-1}$  can be computed from  $(I_{s-1} + Z_{s-1}^\top Z_{s-1})^{-1}$  with  $\mathcal{O}(Ks)$  extra inner products and  $\mathcal{O}(s^2)$  additional scalar multiplications.

## 7 Spectral co-regularization

An extreme case of multitask learning is the *multiview* setting, where all tasks share the same label. In the multiview protocol, at each time step  $t$  the learner receives  $K$  instances  $\mathbf{x}_{1,t}, \dots, \mathbf{x}_{K,t} \in \mathbb{R}^d$ , predicts with  $\hat{y}_t \in \{-1, 1\}$ , and then receives the correct binary label  $y_t$ , which—unlike the general multitask case—is the same for all instances. What distinguishes multiview learning from a standard online binary classification task, defined on instances of the form  $\mathbf{x}_t = (\mathbf{x}_{1,t}, \dots, \mathbf{x}_{K,t})$ , is that in multiview one postulates the existence of  $K$  vectors  $\mathbf{u}_1, \dots, \mathbf{u}_K$  such that each

$\mathbf{u}_i$  is a good linear classifier for the corresponding sequence  $(\mathbf{x}_{i,1}, y_1), (\mathbf{x}_{i,2}, y_2), \dots$  of examples. In this respect a natural baseline for online multiview learning is the algorithm that chooses a random index  $i \in \{1, \dots, K\}$  and then runs a Perceptron algorithm on the sequence of examples  $(\mathbf{x}_{i,t}, y_t)$  for  $t \geq 1$ . Equivalently, we may think of running  $K$  Perceptrons in parallel and then average their mistakes (see the remark after Theorem 10).

In this section we design multiview learning algorithms that in certain cases are able to perform significantly better than the above baseline. In order to do so, we arrange the  $K$   $d$ -dimensional instances  $\mathbf{x}_{1,t}, \dots, \mathbf{x}_{K,t}$  in a  $d \times K$  instance matrix  $X_t$  and penalize diversity among the reference vectors  $\mathbf{u}_1, \dots, \mathbf{u}_K$  using a matrix norm of the  $d \times K$  matrix  $U = [\mathbf{u}_1, \dots, \mathbf{u}_K]$ .

We focus our attention on matrix norms that are unitarily invariant. Such norms are of the form  $\|U\|_f = f(\sigma_U)$ , where  $\sigma_U = (\sigma_1, \dots, \sigma_r)$  is the vector of the ordered singular values  $\sigma_1 \geq \dots \geq \sigma_r \geq 0$  of  $U$  and  $f : \mathbb{R}^r \rightarrow \mathbb{R}$ , with  $r = \min\{K, d\}$ , is an absolutely symmetric function—that is,  $f$  is invariant under coordinate permutations and sign-changes.

Matrix norms of this form control the distribution of the singular values of  $U$ , thus acting as spectral co-regularizers for the reference vectors (see, e.g., [4] for very recent developments on this subject). Known examples are the Schatten  $p$ -norms,  $\|U\|_{s_p} \stackrel{\text{def}}{=} \|\sigma_U\|_p$ . For instance, the Schatten 2-norm is the Frobenius norm. For  $p = 1$  the Schatten  $p$ -norm becomes the trace norm, a good proxy for the rank of  $U$ , since  $\|U\|_{s_1} = \|\sigma_U\|_1 \approx \|\sigma_U\|_0 = \text{rank of } U$ .

In order to obtain a multiview bound that depends on  $\|\sigma_U\|_p$ , we extend the dual norm analysis of Section 5 to matrices. We start by defining the matrix version of the quasi-additive algorithm of [14, 20]. We remark that matrix versions of the EG algorithm and the Winnow algorithm (related to specific instances of the quasi-additive algorithm) have been proposed and analyzed in [28, 29, 30]. When dealing with the trace norm regularizer, their algorithms could be specialized to our multiview framework to obtain mistake bounds comparable to ours. See the brief discussion at the end of this section.

The quasi-additive matrix algorithm maintains a  $d \times K$  matrix  $W$ . Initially,  $W_0$  is the zero matrix. If  $s - 1$  mistakes have been made in the first  $t - 1$  time steps, then the prediction at time  $t$  is  $\text{SGN}(\langle W_{s-1}, X_t \rangle)$ , where  $X_t$  is the  $d \times K$  matrix  $[\mathbf{x}_{1,t}, \dots, \mathbf{x}_{K,t}]$  in which  $\mathbf{x}_{i,t}$  is the instance vector associated with the  $i$ -th view at time  $t$ , and  $\langle W_{s-1}, X_t \rangle$  is the standard matrix inner product  $\langle W_{s-1}, X_t \rangle = \text{TR}(W_{s-1}^\top X_t)$ .

If a mistake occurs at time  $t$ , then  $W_{s-1}$  is updated with  $W_s = \nabla_{\frac{1}{2}} \frac{1}{2} \|V_s\|_f^2$  where, in turn, the  $d \times K$  matrix  $V_s$  is updated using a matrix Perceptron rule,  $V_s = V_{s-1} + y_t X_t$ .

A useful property of norms  $\|U\|_f = f(\sigma_U)$  is that their duals are easily computed.

**Theorem 7 [21, Theorem 2.4]** If  $f$  is absolutely symmetric and  $\|U\|_f = f(\sigma_U)$ , then  $\|U\|_{f^*} = f^*(\sigma_U)$  where  $f^*$  is the convex dual of  $f$ .

In the case of Schatten  $p$ -norms, we have that the dual of vector norm  $\|\cdot\|_p$  is vector norm  $\|\cdot\|_q$ , where  $q = p/(p - 1)$

is the dual coefficient of  $p$ .

An important feature of the quasi-additive algorithms for vectors is that the mapping  $\mu : \mathbf{v} \mapsto \mu(\mathbf{v}) = \nabla \frac{1}{2} \|\mathbf{v}\|^2$  is invertible whenever the vector norm  $\|\cdot\|$  satisfies certain regularity properties (see, e.g., [8, page 294]). We call such norms *Legendre*. Hence, we “do not lose information” when the primal weight vector  $\mathbf{v}$  is mapped to the weight vector  $\mathbf{w} = \mu(\mathbf{v})$  used for prediction. In particular, we always have that  $\mu^{-1}(\mathbf{w}) = \nabla \frac{1}{2} \|\mathbf{w}\|_*^2$ , where  $\|\cdot\|_*$  is the dual norm of a Legendre norm  $\|\cdot\|$  (see, e.g., [8, Lemma 11.5]). This property is preserved when the algorithm is applied to matrices. This is shown by the following result where, without loss of generality, we prove the property for  $f(\sigma_U)$  rather than  $\frac{1}{2}f(\sigma_U)^2$ . (In fact, if  $f$  is Legendre, then  $\frac{1}{2}f^2$  is also Legendre).

**Theorem 8** *Let  $f$  be a Legendre function. If  $\|U\|_f = f(\sigma_U)$  then  $(\nabla \|\cdot\|_f)^{-1} = \nabla \|\cdot\|_{f^*}$ .*

The following result will be useful.

**Theorem 9 [21, Theorem 3.1]** *Let  $U \text{DIAG}[\sigma_A] V^\top$  be an SVD decomposition of a matrix  $A$ . If  $\|\cdot\|_f$  is a matrix norm such that  $\|A\|_f = f(\sigma_A)$  for  $f$  Legendre, then  $\nabla f(\sigma_A) = \sigma_{\nabla \|A\|_f}$ . Moreover,  $\nabla \|A\|_f = U \text{DIAG}[\nabla f(\sigma_A)] V^\top$ .*

**Proof of Theorem 8:** If  $A = U \text{DIAG}[\sigma_A] V^\top$ , then by Theorem 9

$$\nabla \|A\|_f = U \text{DIAG}[\nabla f(\sigma_A)] V^\top = U \text{DIAG}[\sigma_{\nabla \|A\|_f}] V^\top.$$

Therefore, using Theorem 9,

$$\begin{aligned} \nabla \left\| (\nabla \|A\|_f) \right\|_{f^*} &= U \text{DIAG} \left[ \nabla f^*(\nabla f(\sigma_A)) \right] V^\top \\ &= U \text{DIAG}[\sigma_A] V^\top \quad (f \text{ is Legendre}) \\ &= A \end{aligned}$$

concluding the proof.  $\blacksquare$

We now develop a general analysis of the quasi-additive matrix algorithms, and then specialize it (in Theorem 10 below) to a multiview algorithm operating with a Schatten  $p$ -norm regularizer.

We start by adapting the dual norm proof of Section 5 to an arbitrary matrix norm  $\|A\|_f = f(\sigma_A)$ , where  $f$  is Legendre. Let  $V_m$  be the primal weight matrix after any number  $m$  of mistakes. By Taylor expanding  $\frac{1}{2} \|V_s\|_f^2$  around  $V_{s-1}$  for each  $s = 1, \dots, m$ , and using  $y_t \langle W_{s-1}, X_t \rangle \leq 0$ , we get

$$\frac{1}{2} \|V_m\|_f^2 \leq \sum_{s=1}^m D(V_s \| V_{s-1})$$

where  $D(V_s \| V_{s-1})$  is the matrix Bregman divergence  $\frac{1}{2} (\|V_s\|_f^2 - \|V_{s-1}\|_f^2) - y_t \langle W_{s-1}, X_t \rangle$ .

Fix any  $d \times K$  matrix  $U$ . First, we derive a matrix version of the convex inequality for vector norms. We use a classical result by von Neumann (see, e.g., [18, p. 182]) stating that  $\langle V, U \rangle \leq \sigma_V^\top \sigma_U$  for any two  $d \times K$  matrices  $U$  and  $V$ . We have

$$\begin{aligned} \|V_m\|_f \|U\|_{f^*} &= f(\sigma_{V_m}) f^*(\sigma_U) \quad (\text{by Theorem 7}) \\ &\geq \sigma_{V_m}^\top \sigma_U \quad (\text{by the convex ineq. for norms}) \\ &\geq \langle V_m, U \rangle \quad (\text{by von Neumann's ineq.}). \end{aligned}$$

In addition, we have  $\langle U, V_s \rangle = \langle U, V_{s-1} \rangle + y_t \langle U, X_t \rangle$ . Thus we obtain

$$\|V_m\| \geq \frac{\langle U, V_m \rangle}{\|U\|_{f^*}} \geq \frac{Km - \sum_t \ell_t(U)}{\|U\|_{f^*}}$$

where  $\ell_t(U) \stackrel{\text{def}}{=} \sum_{i=1}^K [1 - y_t \mathbf{u}_i^\top \mathbf{x}_{i,t}]_+ = \sum_{i=1}^K \ell_t(\mathbf{u}_i)$ . Solving for  $m$  gives

$$m \leq \frac{1}{K} \sum_{t \in \mathcal{M}} \ell_t(U) + \frac{\|U\|_{f^*}}{K} \sqrt{2 \sum_{s=1}^m D(V_s \| V_{s-1})}. \quad (18)$$

Equation (18) is our general starting point for analyzing multiview algorithms working under spectral co-regularization. The analysis reduces to bounding from above the second-order term  $D(\cdot \| \cdot)$  of the specific matrix norm  $\|\cdot\|_f$  under consideration.

For the rest of this section we focus on the Schatten  $2p$ -norm  $\|V\|_{s_{2p}} = \|\sigma_V\|_{2p}$ , where  $V$  is a generic  $d \times K$  matrix, and  $p$  is a positive integer (thus  $2p$  is an even number  $\geq 2$ ).

Note that, in general,  $\|V\|_{s_{2p}}^2 = \text{TR}((V^\top V)^p)^{1/p}$ .

In order to prove our main result, stated below, we use some facts from differential matrix calculus. A standard reference on this subject is [23], to which the reader is referred.

**Theorem 10** *The number of mistakes  $m$  made by the  $2p$ -norm matrix Perceptron, run on any sequence  $(X_1, y_1), (X_2, y_2), \dots$  satisfies, for any  $d \times K$  matrix  $U$ ,*

$$\begin{aligned} m \leq \frac{1}{K} \sum_{t \in \mathcal{M}} \ell_t(U) + (2p-1) &\left( \frac{X_{s_{2p}} \|U\|_{s_{2q}}}{K} \right)^2 \\ &+ \frac{X_{s_{2p}} \|U\|_{s_{2q}}}{K} \sqrt{\frac{2p-1}{K} \sum_{t \in \mathcal{M}} \ell_t(U)} \end{aligned}$$

where  $X_{s_{2p}} = \max_{t \in \mathcal{M}} \|X_t\|_{s_{2p}}$ ,  $\|U\|_{s_{2q}}$  is the Schatten  $2q$ -norm of  $U$ , with  $2q = \frac{2p}{2p-1}$ , and  $\mathcal{M}$  is the set of mistaken trial indices.

**Remark** Similarly to the vector case, when the parameter  $p$  is chosen to be logarithmic in  $r = \min\{d, K\}$ , the  $p$ -norm matrix Perceptron penalizes diversity using the trace norm of  $U$ . If the vectors  $\mathbf{u}_i$  span a subspace of size  $\ll K$ , and instances tend to have  $K$  nonzero singular values of roughly the same magnitude, then  $\|U\|_{s_{2q}} \approx \|U\|_{s_2}$  while  $X_{s_{2p}}^2 \approx X_{s_\infty}^2 \approx X_{s_2}^2 / K$ . Hence this choice of  $p$  leads (at least in the linearly separable case) to a factor  $K$  improvement over the bound achieved by the matrix algorithm based on the Frobenius norm ( $p = 1$  in Theorem 10), which amounts to running  $K$  independent Perceptrons in parallel and then average their mistakes.

The following trace inequality is our main technical lemma.

**Lemma 11** *Let  $A, B$  be positive semidefinite matrices, of size  $d \times d$  and  $K \times K$  respectively, with the same nonzero eigenvalues. Let  $X$  be an arbitrary real matrix of size  $d \times K$ . Then, for any pair on nonnegative exponents  $l, g \geq 0$ , we have  $\text{TR}(X^\top A^l X B^g) \leq (\text{TR}(X^\top X)^p)^{1/p} (\text{TR} A^{(l+g)q})^{1/q}$  where  $1/p + 1/q = 1, p \geq 1$ .*

**Proof of Lemma 11:** We first consider the case  $l \leq g$ . By the Cauchy-Schwartz and Holder's inequalities applied to traces [23, Ch.11] we have

$$\begin{aligned} \text{TR}(X^\top A^l X B^g) &= \text{TR}(B^{(g-l)/2} X^\top A^l X B^{(g+l)/2}) \quad (19) \\ &\leq \text{TR}(X^\top A^{2l} X B^{g-l})^{1/2} \text{TR}(X^\top X B^{g+l})^{1/2} \\ &\leq \text{TR}(X^\top A^{2l} X B^{g-l})^{1/2} T_p(X^\top X)^{1/2} T_q(B^{g+l})^{1/2} \end{aligned}$$

where we used the shorthand  $T_r(Z) = (\text{TR}Z^r)^{1/r}$ . In the case when  $l > g$  we can simply swap the matrices  $X^\top A^l$  and  $X B^g$  and reduce to the previous case.

We now recursively apply the above argument to the left-hand side of (19). Recalling that  $T_q(A) = T_q(B)$  and  $T_p(X^\top X) = T_p(X X^\top)$ , after  $n$  steps we obtain

$$\begin{aligned} \text{TR}(X^\top A^l X B^g) &\leq (\text{TR}(X^\top A^{l'} X B^{g'}))^{1/2^n} \times \\ &\quad \times T_p(X^\top X)^{\sum_{i=1}^n (1/2)^i} T_q(B^{g+l})^{\sum_{i=1}^n (1/2)^i} \end{aligned}$$

for some pair of exponents  $l', g' \geq 0$  such that  $l' + g' = l + g$ . Since for any such pair  $l', g'$ , we have  $\text{TR}(X^\top A^{l'} X A^{g'}) < \infty$ , we can take the limit as  $n \rightarrow \infty$ . Recalling that  $\sum_{i=1}^{\infty} (1/2)^i = 1$  completes the proof. ■

**Proof of Theorem 10:** We set for brevity  $G : \mathbb{R}^{d \times K} \rightarrow \mathbb{R}$ ,

$$G(V) = \frac{1}{2} \text{TR}((V^\top V)^p)^{1/p} = \frac{1}{2} \|V\|_{s_{2p}}^2.$$

Thus in our case

$$D(V_s \| V_{s-1}) = G(V_{s-1} + y_t X_t) - G(V_{s-1}) - y_t \langle W_{s-1}, X_t \rangle.$$

Since  $G(V)$  is twice<sup>4</sup> continuously differentiable, by the mean-value theorem we can write

$$D(V_s \| V_{s-1}) = \frac{1}{2} \text{VEC}(X_t)^\top H_G(\xi) \text{VEC}(X_t), \quad (20)$$

where  $\text{VEC}(X)$  is the standard columnwise vectorization of a matrix  $X$ ,  $H_G$  denotes the Hessian matrix of (matrix) function  $G$  and  $\xi$  is some matrix on the line connecting  $V_{s-1}$  to  $V_s$ . Using the rules of matrix differentiation, the gradient  $\nabla G$  of  $G$  is  $\nabla G(V) = c(V) \text{VEC}(D)^\top$  where we set for brevity  $D = V B^{p-1}$ ,  $c(V) = \text{TR}(B^p)^{1/p-1}$ , with  $B = V^\top V$ . Taking the second derivative  $H_G = \nabla \nabla G$  gives  $H_G(V) = \text{VEC}(D) \nabla(c(V)) + c(V) \nabla(D)$ . Now, recalling the definition of  $c(V)$ , it is not hard to show that  $\text{VEC}(D) \nabla(c(V))$  is the  $Kd \times Kd$  matrix

$$2(1-p) \text{TR}(B^p)^{1/p-2} \text{VEC}(D) \text{VEC}(D)^\top.$$

Since  $p \geq 1$  this matrix is negative semidefinite, and we can disregard it when bounding from the above the quadratic form (20). Thus we continue by considering only the second term  $c(V) \nabla(D)$  of the Hessian matrix. We have

$$\nabla(D) = (B^{p-1} \otimes I_d) + (I_k \otimes V) \nabla(B^{p-1}),$$

<sup>4</sup>In fact  $G$  is  $C^\infty$  everywhere but (possibly) in zero, since  $\text{TR}((V^\top V)^p)$  is just a polynomial function of the entries of  $V$ . Moreover  $\text{TR}((V^\top V)^p) = 0$  if and only if  $V$  is the zero matrix.

where

$$\nabla(B^{p-1}) = \left( \sum_{\ell=0}^{p-2} B^\ell \otimes B^{p-2-\ell} \right) (I_{K^2} + T_K) (I_k \otimes V^\top),$$

and  $T_K$  is the  $K^2 \times K^2$  commutation matrix such that  $T_K \text{VEC}(M) = \text{VEC}(M^\top)$  for any  $K \times K$  matrix  $M$ . Putting together

$$\begin{aligned} (20) &\leq \frac{c(V)}{2} \text{VEC}(X_t)^\top (B^{p-1} \otimes I_d) \text{VEC}(X_t) \\ &\quad + \frac{c(V)}{2} \text{VEC}(X_t)^\top (I_k \otimes V) \Sigma \times \\ &\quad \times (I_{K^2} + T_K) (I_k \otimes V^\top) \text{VEC}(X_t), \quad (21) \end{aligned}$$

where we used the shorthand  $\Sigma = \sum_{\ell=0}^{p-2} B^\ell \otimes B^{p-2-\ell}$ . We now bound the two terms in the right-hand side of (21). By well-known relationships between Kronecker products and the  $\text{VEC}$  operator (see [23, Ch. 3]) we can write

$$\begin{aligned} &\frac{c(V)}{2} \text{VEC}(X_t)^\top (B^{p-1} \otimes I_d) \text{VEC}(X_t) \\ &= \frac{c(V)}{2} \text{TR}(X_t^\top X_t B^{p-1}) \leq \frac{1}{2} (\text{TR}(X_t^\top X_t))^p, \end{aligned}$$

independent of  $V$ . The majorization follows from Holder's inequality applied to the positive semidefinite matrices  $X_t^\top X_t$  and  $B^{p-1}$ . Moreover, it is easy to see that the symmetric matrices  $\Sigma$  and  $T_K$  commute, thereby sharing the same eigenspace. Hence,  $\Sigma (I_{K^2} + T_K) \preceq 2\Sigma$ , and we can bound from above the second term in (21) by

$$c(V) \text{VEC}(X_t)^\top \sum_{\ell=0}^{p-2} B^\ell \otimes A^{p-1-\ell} \text{VEC}(X_t),$$

where we set  $A = V V^\top$ . Again, [23, Ch. 3] allows us to rewrite this quadratic form as the sum of traces

$$c(V) \sum_{\ell=0}^{p-2} \text{TR}(X_t^\top A^{p-1-\ell} X_t B^\ell).$$

Since  $A$  and  $B$  have the same nonzero eigenvalues, we can apply Lemma 11 to each term and put together as in (21). After simplifying we get

$$(20) \leq \frac{1}{2} (2p-1) (\text{TR}(X_t^\top X_t))^p = \frac{1}{2} (2p-1) \|X_t\|_{s_{2p}}^2.$$

The desired bound is then obtained by plugging back into (18), solving the resulting inequality for  $m$ , and overapproximating. ■

The result of Theorem 10 is similar to those obtained in [28, 29, 30]. However, unlike these previous results, our matrix algorithm has no learning rate to tune (a property inherited from the vector  $p$ -norm Perceptron of [13]) and works for arbitrary nonsquare matrices  $U$ . We also observe that the prediction  $\hat{y}_t = \text{SGN}(\text{TR}(W_{s-1}^\top X_t))$  of the  $p$ -norm matrix Perceptron reduces to computing the sign of  $\text{TR}((V_{s-1}^\top V_{s-1})^{p-1} V_{s-1}^\top X_t)$  (recall the expression for  $\nabla G$  calculated in the proof of Theorem 10). Since matrix  $V_s$  is updated additively, it is clear that both  $V_{s-1}^\top V_{s-1}$  and  $V_{s-1}^\top X_t$  do depend on instance vectors  $x_{i,t}$  only through inner products. This allows us to turn our  $p$ -norm matrix Perceptron into a kernel-based algorithm, and repeat the analysis given here using a standard RKHS formalism.

## 8 Conclusions and ongoing research

In this work we have studied the problem of learning multiple tasks online using various approaches to formalize the notion of task relatedness.

Our results can be extended in different directions. First, in Sections 3.2 and 6 it might be interesting to devise methods for dynamically adapting the  $b$  parameter as new data are revealed. Second, the mistakes of the second-order algorithm MMPERC have been bounded using a first-order analysis. A more refined analysis should reveal in the bound an explicit dependence on the spectral properties of the data. It is also worth noting that the significance of the mistake bound obtained for MMPERC relies on the fact that the algorithm assumes the tasks to be different, although somewhat related. In the case when the  $K$  observed instances share the same label at each time step (like in multiview learning), we could not devise an algorithm with a significant advantage over the following trivial baseline: run  $K$  Perceptrons in parallel and use the sum of margins to predict. Third, it would be interesting to study the problem of learning multiple tasks when  $K$  predictions have to be output in each step. In this case the main difficulty appears to be the control of the interaction among instances at each time step. Fourth, it would be also interesting to prove *lower* bounds on the number of mistakes, as a function of task relatedness. Finally, since multitask learning problems arise naturally in a variety of settings, spanning from biology to news processing, we plan to complement the theoretical analysis presented in this paper with experimental results, so as to evaluate the empirical performance of our algorithms in real-case scenarios.

**Acknowledgments.** Thanks to Sham Kakade, Massi Pontil, and Francis Bach for useful discussions. We also thank the COLT 2008 reviewers for their comments. This work was supported in part by the PASCAL2 Network of Excellence under EC grant no. 216886. This publication only reflects the authors' views.

## References

- [1] J. ABERNETHY, P.L. BARTLETT & A. RAKHLIN, Multitask learning with expert advice, Proc. 20th COLT, pp. 484–498, Springer, 2007.
- [2] R.K. ANDO & T. ZHANG, A framework for learning predictive structures from multiple tasks and unlabeled data, *JMLR*, 6, pp. 1817–1853, MIT Press, 2005.
- [3] A. ARGYRIOU, T. EVGENIOU & M. PONTIL Multi-Task feature learning, NIPS 19, pp. 41–48, MIT Press, 2007.
- [4] A. ARGYRIOU, C.A. MICCHELLI, M. PONTIL & Y. YING, A spectral regularization framework for multi-task structure learning, NIPS 20, MIT Press, 2008.
- [5] K. AZOURY AND M. WARMUTH, Relative loss bounds for on-line density estimation with the exponential family of distributions, *Machine Learning*, 43, pp. 211–246, 2001.
- [6] U. BREFELD, T. GAERTNER, T. SCHEFFER, & S. WROBEL, Efficient co-regularised least squares regression, Proc. 23rd ICML, 2006.
- [7] N. CESA-BIANCHI, A. CONCONI & C. GENTILE, A second-order Perceptron algorithm, *SIAM Journal on Computing*, 34/3, pp. 640–668, 2005.
- [8] N. CESA-BIANCHI & G. LUGOSI, *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [9] O. DEKEL, P.M. LONG & Y. SINGER, Online learning of multiple tasks with a shared loss, *JMLR*, 8, pp. 2233–2264, 2007.
- [10] T. EVGENIOU, M. PONTIL & T. POGGIO, Regularization networks and Support Vector Machines, *Advances in Computational Mathematics*, 13/1, pp. 1–50, Springer, 2000.
- [11] T. EVGENIOU, C. MICCHELLI & M. PONTIL, Learning Multiple tasks with kernel methods, *JMLR*, 6, pp. 615–637, MIT Press, 2005.
- [12] Y. FREUND & R.E. SCHAPIRE, Large margin classification using the Perceptron algorithm. *Machine Learning*, 37:3, pp. 277–296, 1999.
- [13] C. GENTILE, The robustness of the  $p$ -norm algorithms, *Machine Learning*, 53, pp. 265–299, 2003.
- [14] A. GROVE, N. LITTLESTONE & D. SCHUURMANS, General convergence results for linear discriminant updates, *Machine Learning*, 43, pp. 173–210, 2001.
- [15] M. HERBSTER, M. PONTIL & L. WAINER, Online learning over graphs, Proc. 22nd ICML, pp. 305–312, ACM Press, 2005.
- [16] L. HOGBEN, *Handbook of Linear Algebra*, Discrete Mathematics and Its Applications, 39, CRC Press, 2006.
- [17] R.A. HORN & C.R. JOHNSON, *Matrix Analysis*. Cambridge University Press, 1985.
- [18] R.A. HORN & C.R. JOHNSON, *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [19] A. JAGOTA & M.K. WARMUTH, Continuous and discrete time nonlinear gradient descent: relative loss bounds and convergence, Electr. Proc. 5th International Symposium on Artificial Intelligence and Mathematics, 1998. Electronic, <http://rutcor.rutgers.edu/~amai>.
- [20] J. KIVINEN & M. WARMUTH, Relative loss bounds for multidimensional regression problems, *Machine Learning*, 45, pp. 301–329, 2001.
- [21] A.S. LEWIS, The convex analysis of unitarily invariant matrix functions, *Journal of Convex Analysis*, 2, pp. 173–183, 1995.
- [22] N. LITTLESTONE, *Mistake bounds and logarithmic linear-threshold learning algorithms*. Ph.D. Thesis, University of California at Santa Cruz, 1989.
- [23] J.R. MAGNUS, & H. NEUDECKER, *Matrix Differential Calculus with Applications in Statistics and Econometrics, revised edition*. John Wiley, 1999.
- [24] A. MAURER, Bounds for linear multi-task learning, *JMLR*, 7, pp. 117–139, MIT Press, 2006.
- [25] R.T. ROCKAFELLAR, *Convex Analysis*. Princeton University Press, 1970.
- [26] D. ROSENBERG & P. BARTLETT, Rademacher complexity of co-regularized kernel classes, Proc. Artificial Intelligence and Statistics, 2007.
- [27] V. SINDHWANI, P. NIYOGI & M. BELKIN, A co-regularized approach to semi-supervised learning. Proc. ICML Workshop on Learning with Multiple Views, 2005.
- [28] K. TSUDA, G. RAETSCH & M.K. WARMUTH, Matrix exponentiated gradient updates for on-line learning and Bregman projection, *JMLR*, 6, pp. 995–1018, 2005.
- [29] M.K. WARMUTH & D. KUZMIN, Online variance minimization, Proc. 19th COLT, Springer, 2006.
- [30] M.K. WARMUTH, Winnowing subspaces. Proc. 24th ICML, pp. 999–1006, ACM Press, 2007.

---

# Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization

---

**Jacob Abernethy**  
Computer Science Division  
UC Berkeley  
jake@cs.berkeley.edu

**Elad Hazan**  
IBM Almaden  
hazan@us.ibm.com

**Alexander Rakhlin**  
Computer Science Division  
UC Berkeley  
rakhlin@cs.berkeley.edu

## Abstract

We introduce an *efficient* algorithm for the problem of online linear optimization in the bandit setting which achieves the optimal  $O^*(\sqrt{T})$  regret. The setting is a natural generalization of the non-stochastic multi-armed bandit problem, and the existence of an efficient optimal algorithm has been posed as an open problem in a number of recent papers. We show how the difficulties encountered by previous approaches are overcome by the use of a self-concordant potential function. Our approach presents a novel connection between online learning and interior point methods.

## 1 Introduction

One's ability to learn and make decisions rests heavily on the availability of feedback. Indeed, an agent may only improve itself when it can reflect on the outcomes of its own taken actions. In many environments feedback is readily available: a gambler, for example, can observe entirely the outcome of a horse race regardless of where he placed his bet. But such perspective is not always available in hindsight. When the same gambler chooses his route to travel to the race track, perhaps at a busy hour, he will likely never learn the outcome of possible alternatives. When betting on horses, the gambler has thus the benefit (or perhaps the detriment) to muse "*I should have done...*", yet when betting on traffic he can only think "*the result was...*".

This problem of sequential decision making was stated by Robbins [18] in 1952 and was later termed "the multi-armed bandit problem". The name inherits from the model whereby, on each of a sequence of rounds, a gambler must pull the arm on one of several slot machines ("one-armed bandits") that each returns a reward chosen stochastically from a fixed distribution. Of course, an ideal strategy would simply be to pull the arm of the machine with the greatest rewards. However, as the gambler does not know the best arm a priori, his goal is then to maximize the reward of his strategy relative to reward he would receive had he known the optimal arm. This problem has gained much interest over the past 20 years in a number of fields, as it presents a very natural model of an agent seeking to simultaneously explore the world while exploiting high-reward actions.

As early as 1990 [8, 13] the sequential decision problem was studied under *adversarial* assumptions, where we assume the environment may even try to hurt the learner. The multi-armed bandit problem was brought into the adversarial learning model in 2002 by Auer et al [1], who showed that one may obtain nontrivial guarantees on the gambler's performance relative to the best arm *even when the arm values are chosen by an adversary!* In particular, Auer et al [1] showed that the gambler's *regret*, i.e. the difference between the gain of the best arm minus the gain of the gambler, can be bounded by  $O(\sqrt{NT})$  where  $N$  is the number of bandit arms, and  $T$  is the length of the game. In comparison, for the game where the gambler is given full information about alternative arms (such as the horse racing example mentioned above), it is possible to obtain  $O(\sqrt{T \log N})$ , which scales better in  $N$  but identically in  $T$ .

One natural and well studied problem, which escapes the Auer et al result, is that of "online shortest path", considered in [11, 20] among others. In this problem the decision set is exponentially large (i.e., the set of all paths in a given graph), and the straightforward reduction of modeling each path as an arm for the multi-armed bandit problem suffers from both efficiency issues as well as regret exponential in the description length of the graph. To cope with these issues, several authors [2, 9, 14] have recently proposed a very natural generalization of the multi-armed bandit problem to the field of Convex Optimization, and we will call this "bandit linear optimization". In this setting we imagine that, on each round  $t$ , an adversary chooses some linear function  $f_t(\cdot)$  which is not revealed to the player. The player then chooses a point  $\mathbf{x}_t$  within some given convex set<sup>1</sup>  $\mathcal{K} \subset \mathbb{R}^n$ . The player then suffers  $f_t(\mathbf{x}_t)$  and this quantity is revealed to him. This process continues for  $T$  rounds, and at the end the learner's payoff is his *regret*:

$$R_T = \sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^T f_t(\mathbf{x}^*).$$

Online linear optimization has been often considered, yet primarily in the full-information setting where the learner sees all of  $f_t(\cdot)$  rather than just  $f_t(\mathbf{x}_t)$ . In the full-information model, it has been known for some time that the optimal regret bound is  $O(\sqrt{T})$ , and it had been conjectured that the

<sup>1</sup>In the case of online shortest path, the convex set can be represented as a set of vectors in  $\mathbb{R}^{|\mathcal{E}|}$ . Hence, the dependence on number of paths in the graph can be circumvented.

same should hold for the bandit setting as well. Nevertheless, several initially proposed algorithms were shown only to obtain bounds with  $O(T^{3/4})$  (e.g. [14, 9]) or  $O(T^{2/3})$  (e.g. [2, 7]). Only recently was this conjecture proven to be true by Dani et al. [6], who provided an algorithm with  $O(\text{poly}(n)\sqrt{T})$  regret. However, their proposed method, which deploys a clever reduction to the multi-armed bandit algorithm of Auer et al [1], is not efficient.

We propose an algorithm for online linear bandit optimization that is the first, we believe, to be both computationally efficient and achieve a  $O(\text{poly}(n)\sqrt{T})$  regret bound. Moreover, with a thorough analysis we aim to shed light on the difficulties in obtaining such an algorithm. Our technique provides a curious link between the notion of Bregman divergences, which have often been used for constructing and analyzing online learning algorithms, and self-concordant barriers, which are of great importance in the study of interior point methods in convex optimization. A rather surprising consequence is that divergence functions, which are widely used as a regularization tool in online learning, provide the right perspective for the problem of managing uncertainty given limited feedback. To our knowledge, this is the first time such connections have been made.

## 2 Notation and Motivation

Let  $\mathcal{K} \subset \mathbb{R}^n$  be a compact closed convex set. For two vectors  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ , we denote their dot product as  $\mathbf{x}^\top \mathbf{y}$ . We write  $A \succeq B$  if  $(A - B)$  is positive semi-definite. Let

$$D_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) := \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y}) - \nabla \mathcal{R}(\mathbf{y})^\top (\mathbf{x} - \mathbf{y})$$

be the *Bregman divergence* between  $\mathbf{x}$  and  $\mathbf{y}$  with respect to a convex differentiable  $\mathcal{R}$ .

Define the Minkowsky function (see page 34 of [16] for details) on  $\mathcal{K}$ , parametrized by a pole  $\mathbf{y}_t$  as

$$\pi_{\mathbf{y}_t}(\mathbf{x}_t) = \inf\{t \geq 0 : \mathbf{y}_t + t^{-1}(\mathbf{x}_t - \mathbf{y}_t) \in \mathcal{K}\}.$$

We define a scaled version of  $\mathcal{K}$  by

$$\mathcal{K}_\delta = \{\mathbf{u} : \pi_{\mathbf{x}_1}(\mathbf{u}) \leq (1 + \delta)^{-1}\}$$

for  $\delta > 0$ . Here  $\mathbf{x}_1$  is a ‘‘center’’ of  $\mathcal{K}$  defined in the later sections. We assume that  $\mathcal{K}$  is not ‘‘flat’’ and so  $\mathbf{x}_1$  is a constant distance away from the boundary.

In the rest of the section we describe the rich body of previous work which led to our result. The reader familiar with online optimization in the full and partial information settings can skip directly to the next section.

The *online linear optimization* problem is defined as the following repeated game between the learner (player) and the environment (adversary).

At each time step  $t = 1$  to  $T$ ,

- Player chooses  $\mathbf{x}_t \in \mathcal{K}$
- Adversary independently chooses  $\mathbf{f}_t \in \mathbb{R}^n$
- Player suffers loss  $\mathbf{f}_t^\top \mathbf{x}_t$  and observes feedback  $\mathfrak{S}$

The goal of the Player is not simply to minimize his total loss  $\sum_{t=1}^T \mathbf{f}_t^\top \mathbf{x}_t$ , for an adversary could simply choose  $\mathbf{f}_t$

to be as large as possible at every point in  $\mathcal{K}$ . Rather, the Player’s goal is to minimize his *regret*  $R_T$  defined as

$$R_T := \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{x}_t - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{x}^*.$$

When the objective is his regret, the Player is not competing against arbitrary strategies, he need only perform well relative to the total loss of the single best fixed point in  $\mathcal{K}$ .

We distinguish the *full-information* and *bandit* versions of the above problem. The full-information version, the Player may observe the entire function  $\mathbf{f}_t$  as his feedback  $\mathfrak{S}$  and can exploit this in making his decisions. In this paper we study the more challenging bandit setting, where the feedback  $\mathfrak{S}$  provided to the player on round  $t$  is only the scalar value  $\mathbf{f}_t^\top \mathbf{x}_t$ . This is significantly less information for the Player: instead of observing the entire function  $\mathbf{f}_t$ , he may only witness the value of  $\mathbf{f}_t$  at a single point.

### 2.1 Algorithms Based on Full Information

All previous work on bandit online learning, including the present one, relies heavily on techniques developed in the full-information setting and we now give a brief overview of some well-known approaches.

Follow The Leader (FTL) is perhaps the simplest online learning strategy one might think of: the player simply uses the heuristic ‘‘select the best choice thus far’’. For the online optimization task we study, this can be written as

$$\mathbf{x}_{t+1} := \arg \min_{\mathbf{x} \in \mathcal{K}} \sum_{s=1}^t \mathbf{f}_s^\top \mathbf{x}. \quad (1)$$

For certain types of problems, applying FTL does guarantee low regret. Unfortunately, when the loss functions  $\mathbf{f}_t$  are linear on the input space it can be shown that FTL will suffer regret that grows linearly in  $T$ . A natural approach<sup>2</sup>, and more well-known within statistical learning, is to *regularize* the optimization problem (1). That is, an appropriate regularization function  $\mathcal{R}(\mathbf{x})$  and a trade-off parameter  $\lambda$  are selected, and the prediction is obtained as

$$\mathbf{x}_{t+1} := \arg \min_{\mathbf{x} \in \mathcal{K}} \left[ \sum_{s=1}^t \mathbf{f}_s^\top \mathbf{x} + \lambda \mathcal{R}(\mathbf{x}) \right]. \quad (2)$$

We call the above approach Follow The Regularized Leader (FTRL). An alternative way to view this exact algorithm is by sequential updates, which capture the difference between consecutive solutions for FTRL. Given that  $\mathcal{R}$  is convex and differentiable, the general form of this update is

$$\bar{\mathbf{x}}_{t+1} = \nabla \mathcal{R}^*(\nabla \mathcal{R}(\bar{\mathbf{x}}_t) - \eta \mathbf{f}_t), \quad (3)$$

followed by a projection onto  $\mathcal{K}$  with respect to the divergence  $D_{\mathcal{R}}$ :

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{u} \in \mathcal{K}} D_{\mathcal{R}}(\mathbf{u}, \bar{\mathbf{x}}_{t+1}).$$

Here  $\mathcal{R}^*$  is the Fenchel dual function and  $\eta$  is a parameter. This procedure is known as the mirror descent (e.g. [5]).

<sup>2</sup>In the context of classification, this approach has been formulated and analyzed by Shalev-Shwartz and Singer [19].

Applying the above rule we see that the well known Online Gradient Descent algorithm [21, 10] is derived<sup>3</sup> by choosing the regularizer to be the squared Euclidean norm. Similarly, the Exponentiated Gradient [12] algorithm is obtained with the entropy function as the regularizer.

This unified view of various well-known algorithms as solutions to regularization problems gives us an important degree of freedom of choosing the regularizer. Indeed, we will choose a regularizer for our problem that possesses key properties needed for the regret to scale as  $O(\sqrt{T})$ . In Section 4, we give a bound on the regret for (2) with any regularizer  $\mathcal{R}$  and in Section 5 we will discuss the specific  $\mathcal{R}$  used in this paper.

## 2.2 The Dilemma of Bandit Optimization

Effectively all previous algorithms for the Bandit setting have utilized a reduction to the full-information setting in one way or another. This is reasonable: any algorithm that aimed for low-regret in the bandit setting would necessarily have to achieve low regret given full information. Furthermore, as the full-information online learning setting is relatively well-understood, it is natural to exploit such techniques for this more challenging problem.

The crucial reduction that has been utilized by several authors [1, 2, 6, 7, 14] is the following. First choose some full-information online learning algorithm  $\mathcal{A}$ .  $\mathcal{A}$  will receive input vectors  $\mathbf{f}_1, \dots, \mathbf{f}_t$ , corresponding to previously observed functions, and will return some point  $\mathbf{x}_{t+1} \in \mathcal{K}$  to predict. On every round  $t$ , do one *or both* of the following:

- Query  $\mathcal{A}$  for its prediction  $\mathbf{x}_t$  and either predict  $\mathbf{x}_t$  exactly or in expectation.
- Construct some random estimate  $\tilde{\mathbf{f}}_t$  in such a way that  $\mathbb{E}\tilde{\mathbf{f}}_t = \mathbf{f}_t$ , and input  $\tilde{\mathbf{f}}_t$  into  $\mathcal{A}$  as though it had been observed on this round

The key idea here is simple: so long as we are roughly predicting  $\mathbf{x}_t$  per advice of  $\mathcal{A}$ , and so long as we are “guessing”  $\mathbf{f}_t$  (i.e. so that the estimates  $\tilde{\mathbf{f}}_t$  is correct in expectation), then we can guarantee low regret. This approach is validated in Lemma 3 which shows that, as long as  $\mathcal{A}$  performs well against the random estimates  $\tilde{\mathbf{f}}_t$  in expectation, then we will also do well against the true functions  $\mathbf{f}_1, \dots, \mathbf{f}_T$ .

This observation is quite reassuring yet unfortunately does not address a significant obstacle: *how can we simultaneously estimate  $\tilde{\mathbf{f}}_t$  and predict  $\mathbf{x}_t$  when only one query is allowed?* The algorithm faces an inherent dilemma: whether to follow the advice of  $\mathcal{A}$  of predicting  $\mathbf{x}_t$ , or to try to estimate  $\mathbf{f}_t$  by sampling in a wide region around  $\mathcal{K}$ , possibly hurting its performance on the given round. This exploration-exploitation trade-off is the primary source of difficulty in obtaining  $O(\sqrt{T})$  guarantees on the regret.

Roughly two categories of approaches have been suggested to perform both exploration and exploitation:

1. **Alternating Explore/Exploit:** Flip an  $\epsilon$ -biased coin to determine whether to explore or exploit. On explore

<sup>3</sup>Strictly speaking, this equivalence is true if the updates are applied to unprojected versions of  $\mathbf{x}_t$ .

rounds, sample uniformly on some wide region around  $\mathcal{K}$  and estimate  $\mathbf{f}_t$  accordingly, and input this into  $\mathcal{A}$ . On exploit rounds, query  $\mathcal{A}$  for  $\mathbf{x}_t$  and predict this.

2. **Simultaneous Explore/Exploit:** Query  $\mathcal{A}$  for  $\mathbf{x}_t$  and construct a random vector  $X_t$  such that  $\mathbb{E}X_t = \mathbf{x}_t$ . Construct  $\tilde{\mathbf{f}}_t$  randomly based on the outcome of  $X_t$  and the learned value  $\mathbf{f}_t^T X_t$ .

The methods of [14, 2, 7] fit within in the first category but, unfortunately, fail to obtain the desired  $O(\text{poly}(n)\sqrt{T})$  regret. This is not surprising: it has been suggested by [7] that  $\Omega(T^{2/3})$  regret is unavoidable by any algorithm in which the observation  $\mathbf{f}_t^T \mathbf{x}_t$  is ignored on rounds pledged for exploitation. Algorithms falling into the second category, such as those of [1, 6, 9], are more sophisticated and help to motivate our results. We review these methods below.

## 2.3 Methods For Simultaneous Exploration and Exploitation

On first glance, it is rather surprising that one can perform the task of predicting some  $\mathbf{x}_t$  (in expectation) while, simultaneously, finding an unbiased estimate of  $\mathbf{f}_t$ . To get a feel for how this can be done, we briefly review the methods of [1] and [9] below.

The work of **Auer et al [1]** is not, strictly speaking, concerned with a general bandit optimization problem but instead the more simple “Multi-armed bandit” problem. The authors consider the problem of sequentially choosing one of  $N$  “arms” each of which contains a hidden loss where the learner may only see the loss of his chosen arm. The regret, in this case, is the learner’s loss minus the smallest cumulative loss over all arms. This multi-armed bandit problem can indeed be cast as a bandit optimization problem: let  $\mathcal{K}$  be the  $N$ -simplex (convex hull of  $\{\mathbf{e}_1, \dots, \mathbf{e}_N\}$ ), let  $\mathbf{f}_t$  be identically the vector of hidden losses on the set of arms, and note that  $\min_{\mathbf{x} \in \mathcal{K}} \sum \mathbf{f}_s^T \mathbf{x} = \min_i \sum \mathbf{f}_s[i]$ .

The algorithm of [1], EXP3, utilizes EG (mentioned earlier) as its black box full-information algorithm  $\mathcal{A}$ . First, a point  $\mathbf{x}_t \in \mathcal{K}$  is returned by  $\mathcal{A}$ . The hypothesis  $\mathbf{x}_t$  is then *biased* slightly:

$$\mathbf{x}_t \leftarrow (1 - \gamma)\mathbf{x}_t + \gamma \left\langle \frac{1}{n}, \dots, \frac{1}{n} \right\rangle.$$

We describe the need for this bias in Section 2.4. EXP3 then randomly chooses one of the corners of  $\mathcal{K}$  according to the distribution  $\mathbf{x}_t$  and uses this as its prediction. More precisely, a basis vector  $\mathbf{e}_i$  is sampled with probability  $\mathbf{x}_t[i]$  and clearly  $\mathbb{E}_{I \sim \mathbf{x}_t} \mathbf{e}_I = \mathbf{x}_t$ . Once we observe  $\mathbf{f}_t^T \mathbf{e}_i = \mathbf{f}_t[i]$ , the estimate is constructed as follows:

$$\tilde{\mathbf{f}}_t := \left( \frac{\mathbf{f}_t[i]}{\mathbf{x}_t[i]} \right) \mathbf{e}_i.$$

It is very easy to check that  $\mathbb{E}\tilde{\mathbf{f}}_t = \mathbf{f}_t$ .

**Flaxman et al [9]** developed a bandit optimization algorithm that used OGD as the full-information subroutine  $\mathcal{A}$ . Their approach uses a quite different method of performing exploration and exploitation. On each round, the algorithm queries  $\mathcal{A}$  for a hypothesis  $\mathbf{x}_t$  and, as in [1], this hypothesis is biased slightly:

$$\mathbf{x}_t \leftarrow (1 - \gamma)\mathbf{x}_t + \gamma \mathbf{u}$$

where  $\mathbf{u}$  is some “center” vector of the set  $\mathcal{K}$ . Similarly to EXP3, the algorithm doesn’t actually predict  $\mathbf{x}_t$ . The algorithm determines the distance  $r$  to the boundary of the set, and a vector  $r\mathbf{v}$  is sampled uniformly at random from a sphere of radius  $r$ . The prediction is  $\mathbf{y}_t := \mathbf{x}_t + r\mathbf{v}$  and indeed  $\mathbb{E}\mathbf{y}_t = \mathbf{x}_t + r\mathbb{E}\mathbf{v} = \mathbf{x}_t$  as desired. The algorithm predicts  $\mathbf{y}_t$ , receives feedback  $\mathbf{f}_t^\top \mathbf{y}_t$ , and function  $\mathbf{f}_t$  is estimated as

$$\tilde{\mathbf{f}}_t := \frac{\mathbf{f}_t^\top \mathbf{y}_t}{r} \mathbf{v}.$$

It is, again, easy to check that this provides an unbiased estimate of  $\mathbf{f}_t$ .

## 2.4 The Curse of High Variance and the Blessing of Regularization

Upon inspecting the definitions of  $\tilde{\mathbf{f}}_t$  in the method of Auer et al and Flaxman et al it becomes apparent that the estimates are inversely proportional to the distance of  $\mathbf{x}_t$  to the boundary. This implies high variance of the estimated functions. At first glance, this seems to be a disaster. Indeed, most full-information algorithms scale linearly with the magnitude of the functions played by the environment. Let us take a closer look at how exactly this leads to the suboptimality of the algorithm of Flaxman et al.

The bound on the expected regret of OGD on  $\tilde{\mathbf{f}}_t$ ’s involves terms  $\mathbb{E}\|\tilde{\mathbf{f}}_t\|^2$  (see proof of Lemma 2), which scale as the inverse of the squared distance to the boundary. Biasing of  $\mathbf{x}_t$  away from the boundary leads to an upper bound on this quantity of the order  $\gamma^{-2}$ . Unfortunately,  $\gamma$  cannot be taken to be large. Indeed, the optimal point  $\mathbf{x}^*$ , chosen in hindsight, lies on the boundary of the set, as the cost functions are linear. Thus, stepping away from the boundary comes at a cost of potentially losing  $O(\gamma T)$  over the course of the game. Since the goal is to obtain an  $O(\sqrt{T})$  bound on the regret,  $\gamma = O(T^{-1/2})$  is the most that can be tolerated. Biasing away from the boundary does reduce the variance of the estimates somewhat; unfortunately, it is not the panacea. To terminate the discussion on the method of Flaxman et al, we state the dependence of the regret bound on the learning rate  $\eta$  and the biasing parameter  $\gamma$ :

$$R_T = O(\eta^{-1} + \gamma^{-2}\eta T + \gamma T).$$

The first term is due to the distance between the initial choice and the comparator; the second is the problematic  $\mathbb{E}\|\tilde{\mathbf{f}}_t\|^2$  term summed over time; and the last term is due to stepping away from the boundary. The best choice of the parameters leads to the unsatisfying  $O(T^{3/4})$  bound.

From the above discussion it is clear that the problematic term is  $\mathbb{E}\|\tilde{\mathbf{f}}_t\|^2 = O(1/r^2)$ , owing its high magnitude to its inverse dependence on the squared distance to the boundary. A similar dependence occurs in the estimate of Auer et al, though the non-uniform sampling from the basis implies an  $O(1/\mathbf{x}_t[i])$  magnitude. One can ask whether this inverse dependence on the distance is an artifact of these algorithms and can be avoided. In fact, it is possible to prove that this is intrinsic to the problem if we require that  $\tilde{\mathbf{f}}_t$  be unbiased and  $\mathbf{x}_t$  be the center of the sampling distribution.

Does this result imply that no  $O(\sqrt{T})$  bound on the regret is possible? Fortunately, no. If we restrict our search

to a regularization algorithm of the type (2), the expected regret can be proved to be *equal* to an expression involving  $\mathbb{E}D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1})$  terms. For  $\mathcal{R}(\mathbf{x}) \propto \|\mathbf{x}\|^2$  we indeed recover (modulo projections) the method of Flaxman et al with its insurmountable hurdle of  $\mathbb{E}\|\tilde{\mathbf{f}}_t\|^2$ . Fortunately, other choices of  $\mathcal{R}$  have better behavior. Here, the formulation of the regularized minimization (2) as a dual-space mirror descent comes to the rescue.

In the space of gradients (the dual space), the step-wise updates (3) for Follow The Regularized Leader are  $\eta\tilde{\mathbf{f}}_t$  no matter what  $\mathcal{R}$  we choose. It is a known fact (e.g. [5]) that the divergence in the original space between  $\mathbf{x}_t$  and  $\mathbf{x}_{t+1}$  is equal to the divergence between the corresponding gradients with respect to the dual potential  $\mathcal{R}^*$ . It is, therefore, not surprising that the dual divergence can be tuned to be small even if  $\|\tilde{\mathbf{f}}_t\|$  is very large. Having small divergence corresponds to the requirement that  $\mathcal{R}^*$  be “flat” whenever  $\|\tilde{\mathbf{f}}_t\|$  is large, i.e. when  $\mathbf{x}_t$  is close to the boundary. Flatness in the dual space corresponds to large curvature in the primal. This motivates the use of a potential function  $\mathcal{R}$  which becomes more and more curved at the boundary of the set  $\mathcal{K}$ . In a nutshell, this is the Blessing of Regularization which allows us to obtain an efficient optimal algorithm which was escaping all previous attempts.

Recall that the method of Auer et al attains the optimal  $O(\sqrt{T})$  rate *but only when  $\mathcal{K}$  is the simplex*. If our intuition about the importance of regularization is sound, we should find that the method uses a potential which curves at the edges of the simplex. One can see that the exponential weights (more generally, EG) used by Auer et al corresponds to regularization with  $\mathcal{R}$  being the entropy function  $\mathcal{R}(\mathbf{x}) = \sum_{i=1}^n \mathbf{x}[i] \log \mathbf{x}[i]$ . Taking the second derivative, we see that, indeed, the curvature increases as  $1/\mathbf{x}[i]$  as  $\mathbf{x}$  gets closer to the boundary. For the present paper, we will actually choose a regularizer that curves as inverse *squared* distance to the boundary. The reader can probably guess that such a regularizer should be defined, roughly, as the log-distance to the boundary.

While for simple convex bodies, such as sphere, existence of a function behaving like log-distance to the boundary seems plausible, a similar statement for general convex sets  $\mathcal{K}$  seems very complex. Luckily, this very question has been studied in the theory of Interior Point Methods, and existence and construction of such functions, called *self-concordant barriers*, is well-established.

## 3 Main Result

We first state our main result: an algorithm for online linear optimization in the bandit setting for an arbitrary compact convex set  $\mathcal{K}$ . The analysis of this algorithm has a number of facets and we discuss these individually throughout the remainder of this paper. In Section 4 we describe the regularization framework in detail and show how the regret can be computed in terms of Bregman divergences. In Section 5 we review the theory of self-concordant functions and state two important properties of such functions. In Section 6 we highlight several key elements of the proof of our regret bound. In Section 7 we show how this algorithm can be used for one interesting case, namely the bandit version of the Online

---

**Algorithm 1** Bandit Online Linear Optimization
 

---

- 1: Input:  $\eta > 0$ ,  $\vartheta$ -self-concordant  $\mathcal{R}$
- 2: Let  $\mathbf{x}_1 = \arg \min_{\mathbf{x} \in \mathcal{K}} [\mathcal{R}(\mathbf{x})]$ .
- 3: **for**  $t = 1$  to  $T$  **do**
- 4: Let  $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$  and  $\{\lambda_1, \dots, \lambda_n\}$  be the set of eigenvectors and eigenvalues of  $\nabla^2 \mathcal{R}(\mathbf{x}_t)$ .
- 5: Choose  $i_t$  uniformly at random from  $\{1, \dots, n\}$  and  $\varepsilon_t = \pm 1$  with probability  $1/2$ .
- 6: Predict  $\mathbf{y}_t = \mathbf{x}_t + \varepsilon_t \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}$ .
- 7: Observe the gain  $\mathbf{f}_t^\top \mathbf{y}_t \in \mathbb{R}$ .
- 8: Define  $\tilde{\mathbf{f}}_t := n(\mathbf{f}_t^\top \mathbf{y}_t) \varepsilon_t \lambda_{i_t}^{1/2} \cdot \mathbf{e}_{i_t}$ .
- 9: Update

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{x} \in \mathcal{K}} \left[ \eta \sum_{s=1}^t \tilde{\mathbf{f}}_s^\top \mathbf{x} + \mathcal{R}(\mathbf{x}) \right].$$

10: **end for**

---

Shortest Path problem. The precise analysis of our algorithm is given in Section 8. Finally, in Section 9 we spell out how to implement the algorithm with only one iteration of the Damped Newton method per time step.

The following theorem is the main result of this paper (see Section 5 for the definition of  $\vartheta$ -self-concordant barrier).

**Theorem 1** Let  $\mathcal{K}$  be a convex set and  $\mathcal{R}$  be a  $\vartheta$ -self-concordant barrier on  $\mathcal{K}$ . Let  $\mathbf{u}$  be any vector in  $\mathcal{K}' = \mathcal{K}_{1/\sqrt{T}}$ . Suppose we have the property that  $|\mathbf{f}_t^\top \mathbf{x}| \leq 1$  for any  $\mathbf{x} \in \mathcal{K}$ . Setting  $\eta = \frac{\sqrt{\vartheta \log T}}{4n\sqrt{T}}$ , the regret of Algorithm 1 is bounded as

$$\mathbb{E} \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{y}_t \leq \min_{\mathbf{u} \in \mathcal{K}'} \mathbb{E} \left( \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{u} \right) + 16n\sqrt{\vartheta T \log T}$$

whenever  $T > 8\vartheta \log T$ .

The expected regret over the original set  $\mathcal{K}$  is within an additive  $O(\sqrt{nT})$  factor from the above guarantee, as implied by Lemma 8 in the Appendix.

## 4 Regularization Algorithms and Bregman Divergences

As our algorithm is clearly based on a regularization framework, we now state a general result for the performance of any algorithm minimizing the regularized empirical loss. We call this method Follow the Regularized Leader, and we defer the proof of the regret bound to the Appendix. A similar analysis for convex loss functions can be found in [5], Chapter 11. We remark that the use of Bregman divergences in the context of online learning goes back at least to Kivinen and Warmuth [12].

Let  $\tilde{\mathbf{f}}_1, \dots, \tilde{\mathbf{f}}_T \in \mathbb{R}^n$  be any sequence of vectors. Suppose  $\mathbf{x}_{t+1}$  is obtained as

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{x} \in \mathcal{K}} \underbrace{\left[ \eta \sum_{s=1}^t \tilde{\mathbf{f}}_s^\top \mathbf{x} + \mathcal{R}(\mathbf{x}) \right]}_{\Phi_t(\mathbf{x})} \quad (4)$$

for some strictly-convex differentiable function  $\mathcal{R}$ . We denote  $\Phi_0(x) = \mathcal{R}(x)$  and  $\Phi_t = \Phi_{t-1} + \eta \tilde{\mathbf{f}}_t$ .

We will assume  $\nabla \mathcal{R}$  approaches infinity at the boundary of  $\mathcal{K}$  so that the unconstrained minimization problem will have a unique solution within  $\mathcal{K}$ . We have the following bound on the performance of such an algorithm.

**Lemma 2** For any  $\mathbf{u} \in \mathcal{K}$ , the algorithm defined by (4) enjoys the following regret guarantee

$$\begin{aligned} \eta \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u}) &\leq D_{\mathcal{R}}(\mathbf{u}, \mathbf{x}_1) + \sum_{t=1}^T D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) \\ &\leq D_{\mathcal{R}}(\mathbf{u}, \mathbf{x}_1) + \eta \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \end{aligned}$$

for any sequence  $\{\tilde{\mathbf{f}}_t\}_{t=1}^T$ .

In addition, we state a useful result that bounds the true regret based on the regret against the estimated functions  $\tilde{\mathbf{f}}_t$ .

**Lemma 3** Suppose that, for  $t = 1, \dots, T$ ,  $\tilde{\mathbf{f}}_t$  is such that  $\mathbb{E} \tilde{\mathbf{f}}_t = \mathbf{f}_t$  and  $\mathbf{y}_t$  is such that  $\mathbb{E} \mathbf{y}_t = \mathbf{x}_t$ . Suppose that we have the following regret bound:

$$\sum_{t=1}^T \tilde{\mathbf{f}}_t^\top \mathbf{x}_t \leq \min_{\mathbf{u} \in \mathcal{K}'} \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top \mathbf{u} + C_T.$$

Then the expected regret satisfies

$$\mathbb{E} \left( \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{y}_t \right) \leq \min_{\mathbf{u} \in \mathcal{K}'} \mathbb{E} \left( \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{u} \right) + C_T.$$

## 5 Self-concordant Functions and the Dikin ellipsoid

Interior-point methods are arguably one of the greatest achievements in the field of Convex Optimization in the past two decades. These iterative polynomial-time algorithms for Convex Optimization find the solution by adding a barrier function to the objective and solving the unconstrained minimization problem. The rough idea is to gradually reduce the weight of the barrier function as one approaches the solution. The construction of barrier functions for general convex sets has been studied extensively, and we refer the reader to [16, 4] for a thorough treatment on the subject. To be more precise, most of the results of this section can be found in [15], page 22-23, as well as in the aforementioned texts.

### 5.1 Definitions and Properties

**Definition 4** A self-concordant function  $\mathcal{R} : \text{int } \mathcal{K} \rightarrow \mathbb{R}$  is a  $C^3$  convex function such that

$$|D^3 \mathcal{R}(\mathbf{x})[\mathbf{h}, \mathbf{h}, \mathbf{h}]| \leq 2(D^2 \mathcal{R}(\mathbf{x})[\mathbf{h}, \mathbf{h}])^{3/2}.$$

Here, the third-order differential is defined as

$$\begin{aligned} D^3 \mathcal{R}(\mathbf{x})[\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3] &:= \\ &\frac{\partial^3}{\partial t_1 \partial t_2 \partial t_3} \Big|_{t_1=t_2=t_3=0} \mathcal{R}(\mathbf{x} + t_1 \mathbf{h}_1 + t_2 \mathbf{h}_2 + t_3 \mathbf{h}_3). \end{aligned}$$

We will further assume that the function approaches infinity for any sequence of points approaching the boundary of  $\mathcal{K}$ . An additional requirement leads to the notion of a self-concordant *barrier*.

**Definition 5** A  $\vartheta$ -self-concordant barrier  $\mathcal{R}$  is a self-concordant function with

$$|D\mathcal{R}(\mathbf{x})[\mathbf{h}]| \leq \vartheta^{1/2} [D^2\mathcal{R}(\mathbf{x})[\mathbf{h}, \mathbf{h}]]^{1/2}.$$

The generality of interior-point methods comes from the fact that any arbitrary  $n$ -dimensional closed convex set admits an  $O(n)$ -self-concordant barrier [16]. Hence, throughout this paper,  $\vartheta = O(n)$ , but can even be independent of the dimension, as for the sphere.

We note that some of the results of this paper, such as the Dikin ellipsoid, rely on  $\mathcal{R}$  being a self-concordant function, while others necessarily require the barrier property. We therefore assume from the outset that  $\mathcal{R}$  is a self-concordant barrier.

Since  $\mathcal{K}$  is compact, we can assume that  $\mathcal{R}$  is non-degenerate. For a given  $\mathbf{x} \in \mathcal{K}$ , define

$$\langle \mathbf{g}, \mathbf{h} \rangle_{\mathbf{x}} = \mathbf{g}^\top \nabla^2 \mathcal{R}(\mathbf{x}) \mathbf{h} \quad \text{and} \quad \|\mathbf{h}\|_{\mathbf{x}} = (\langle \mathbf{h}, \mathbf{h} \rangle_{\mathbf{x}})^{-1/2}.$$

This inner product defines the local Euclidean structure at  $\mathbf{x}$ . Nondegeneracy of  $\mathcal{R}$  implies that the above norm is indeed a norm, not a seminorm.

It is natural to talk about a ball with respect to the above norm. Define the open *Dikin ellipsoid* of radius  $r$  centered at  $\mathbf{x}$  as the set

$$W_r(\mathbf{x}) = \{\mathbf{y} \in \mathcal{K} : \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} < r\}.$$

The following facts about the Dikin ellipsoid are central to the results of this paper (we refer to [15], page 23 for proofs). The first non-trivial fact is that  $W_1(\mathbf{x}) \subseteq \mathcal{K}$  for any  $\mathbf{x} \in \mathcal{K}$ . In other words, the inverse Hessian of the self-concordant function  $\mathcal{R}$  stretches the space in such a way that the eigenvectors fall in the set  $\mathcal{K}$ . This is crucial for our sampling procedure. Indeed, our method (Algorithm 1) samples  $\mathbf{y}_t$  from the Dikin ellipsoid centered at  $\mathbf{x}_t$ . Since  $W_1(\mathbf{x}_t)$  is contained in  $\mathcal{K}$ , the sampling procedure is legal.

The second fact is that within the Dikin ellipsoid, that is for  $\|\mathbf{h}\|_{\mathbf{x}} < 1$ , the Hessians of  $\mathcal{R}$  are “almost proportional” to the Hessian of  $\mathcal{R}$  at the center of the ellipsoid :

$$(1 - \|\mathbf{h}\|_{\mathbf{x}})^2 \nabla^2 \mathcal{R}(\mathbf{x}) \preceq \nabla^2 \mathcal{R}(\mathbf{x} + \mathbf{h}) \preceq (1 + \|\mathbf{h}\|_{\mathbf{x}})^2 \nabla^2 \mathcal{R}(\mathbf{x}). \quad (5)$$

This gives us the crucial control of the Hessians for second-order approximations. Finally, if  $\|\mathbf{h}\|_{\mathbf{x}} < 1$  (i.e.  $\mathbf{x} + \mathbf{h}$  is in the unit Dikin ellipsoid), then for any  $\mathbf{z}$ ,

$$|\mathbf{z}^\top (\nabla \mathcal{R}(\mathbf{x} + \mathbf{h}) - \nabla \mathcal{R}(\mathbf{x}))| \leq \frac{\|\mathbf{h}\|_{\mathbf{x}}}{1 - \|\mathbf{h}\|_{\mathbf{x}}} \|\mathbf{z}\|_{\mathbf{x}}. \quad (6)$$

Assuming that  $\mathcal{R}$  is a  $\vartheta$ -self-concordant barrier, we have (see page 34 of [16])

$$\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1) \leq \vartheta \ln \frac{1}{1 - \pi_{\mathbf{x}_1}(\mathbf{u})}.$$

For any  $\mathbf{u} \in \mathcal{K}_\delta$ ,  $\pi_{\mathbf{x}_1}(\mathbf{u}) \leq (1 + \delta)^{-1}$  by definition, implying that  $(1 - \pi_{\mathbf{x}_1}(\mathbf{u}))^{-1} \leq \frac{1 + \delta}{\delta}$ . We conclude that

$$\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1) \leq \vartheta \ln(\sqrt{T} + 1) \leq 2\vartheta \log T \quad (7)$$

for  $\mathbf{u} \in \mathcal{K}_{1/\sqrt{T}}$ .

## 5.2 Examples of Self-Concordant Functions

A nice fact about self-concordant barriers is that  $\mathcal{R}_1 + \mathcal{R}_2$  is  $\vartheta_1 + \vartheta_2$ -self-concordant for  $\vartheta_1$ -self-concordant  $\mathcal{R}_1$  and  $\vartheta_2$ -self-concordant  $\mathcal{R}_2$ . For linear constraints  $\mathbf{a}^\top \mathbf{x}_t \leq b$ , the barrier  $-\ln(b - \mathbf{a}^\top \mathbf{x}_t)$  is 1-self-concordant. Hence, for a polyhedron defined by  $m$  constraints, the corresponding barrier is  $m$ -self-concordant. Thus, for the  $n$ -dimensional simplex or a cube,  $\theta = n$ , leading to  $n^{3/2}$  dependence on the dimension in the main result.

For the  $n$ -dimensional ball,

$$\mathcal{B}_n = \{\mathbf{x} \in \mathbb{R}^n, \sum_i x_i^2 \leq 1\},$$

the barrier function  $\mathcal{R}(\mathbf{x}) = -\log(1 - \|\mathbf{x}\|^2)$  is 1-self-concordant. This, somewhat surprisingly, leads to the linear dependence of the regret bound on the dimension  $n$ , as  $\vartheta = 1$ .

## 6 Sketch of Proof

We have now presented all necessary tools to prove Theorem 1: regret in terms of Bregman divergences, self-concordant barriers and the Dikin ellipsoid. While we provide a complete proof in Section 8 here we sketch the key elements of the analysis of our algorithm.

As we tried to motivate in the end of Section 2, any method that can simultaneously (a) predict  $\mathbf{x}_t$  in expectation and (b) obtain an unbiased one-sample estimate of  $\tilde{\mathbf{f}}_t$  will necessarily suffer from high variance when  $\mathbf{x}_t$  is close to the boundary of the set  $\mathcal{K}$ . As we have hinted previously, we would like our regularizer  $\mathcal{R}$  to control the variance. Yet the problem is even more subtle than this:  $\mathbf{x}_t$  may be close to the boundary in one dimension while have plenty of space in another, which in turn suggests that  $\tilde{\mathbf{f}}_t$  need only have high variance in certain directions.

Quite amazingly, the self-concordant function  $\mathcal{R}$  gives us a handle on two key issues. The Dikin ellipsoid, defined in terms  $\nabla^2 \mathcal{R}(\mathbf{x}_t)$ , gives us exactly a rough approximation to the available “space” around  $\mathbf{x}_t$ . At the same time,  $\nabla^2 \mathcal{R}(\mathbf{x}_t)^{-1}$  annihilates  $\tilde{\mathbf{f}}_t$  in exactly the directions in which it is large. This is absolutely necessary for bounding the regret, as we discuss next.

Lemma 2 implies that regret scales with the cumulative divergence  $\eta^{-1} \sum_t D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1})$  and thus we must have that  $\mathbb{E} D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) = O(\eta^2)$  on average to obtain a regret bound of  $O(\sqrt{T})$ . Analyzing the divergence requires some care and so we provide only a rough sketch here (with more in Section 8). If  $\mathcal{R}$  were exactly quadratic then the divergence is

$$D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) := \eta^2 \tilde{\mathbf{f}}_t^\top (\nabla^2 \mathcal{R}(\mathbf{x}_t))^{-1} \tilde{\mathbf{f}}_t. \quad (8)$$

Even when  $\mathcal{R}$  is not quadratic, however, (8) still provides a decent approximation to the divergence and, given certain regularity conditions on  $\mathcal{R}$ , it is enough to bound the quadratic form  $\tilde{\mathbf{f}}_t^\top (\nabla^2 \mathcal{R}(\mathbf{x}_t))^{-1} \tilde{\mathbf{f}}_t$ .

The precise interaction between the Dikin ellipsoid, the estimates  $\tilde{\mathbf{f}}_t$ , and the divergence  $D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1})$  is as follows. Assume we are at the point  $\mathbf{x}_t$  and we have computed the unit eigenvectors  $\mathbf{e}_1, \dots, \mathbf{e}_n$  and corresponding eigenvalues  $\lambda_1, \dots, \lambda_n$  of  $\nabla^2 \mathcal{R}(\mathbf{x}_t)$ . Properties of self-concordant functions ensure that the Dikin ellipsoid around  $\mathbf{x}_t$  is contained

within  $\mathcal{K}$  and thus, in particular, so are the points  $\mathbf{x}_t \pm \lambda_i^{-1/2} \mathbf{e}_i$  for each  $i$ . Assuming the point  $\mathbf{y}_t := \mathbf{x}_t + \lambda_j^{-1/2} \mathbf{e}_j$  was sampled and we received the value  $\mathbf{f}_t^\top \mathbf{y}_t$ , we then construct the estimate

$$\tilde{\mathbf{f}}_t := n \sqrt{\lambda_j} (\mathbf{f}_t^\top \mathbf{y}_t) \mathbf{e}_j.$$

Notice it is crucial that we scale by  $\sqrt{\lambda_j}$ , the *inverse*  $\ell_2$  distance between  $\mathbf{x}_t$  and  $\mathbf{y}_t$ , to ensure that  $\tilde{\mathbf{f}}_t$  is unbiased. On the other hand, we see that the divergence is approximately computed as

$$\begin{aligned} D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) &\approx \eta^2 \tilde{\mathbf{f}}_t^\top \nabla^2 \mathcal{R}^{-1} \tilde{\mathbf{f}}_t \\ &= \eta^2 n^2 (\mathbf{f}_t^\top \mathbf{y}_t)^2 \lambda_j (\mathbf{e}_j^\top \nabla^2 \mathcal{R}^{-1} \mathbf{e}_j) \\ &= \eta^2 n^2 (\mathbf{f}_t^\top \mathbf{y}_t)^2. \end{aligned}$$

As an interesting and important aside, a *necessary* requirement of the above analysis is that we construct our estimates  $\tilde{\mathbf{f}}_t$  from the eigendirections  $\mathbf{e}_j$ . To see this, imagine that one eigenvalue  $\lambda_1$  is very large, while another,  $\lambda_2$  small. This corresponds to a thin and long Dikin ellipsoid, which would occur near a flat boundary. Suppose that instead of eigen-directions, we sample at an angle between them. With the thin ellipsoid the sampled points are still close in  $\ell_2$  distance, implying that  $\tilde{\mathbf{f}}_t$  will be large in both eigen-directions. However, the inverse Hessian will only annihilate one of these directions.

## 7 Application to the online shortest path problem

Because of its appealing structure, the online shortest path problem is one of the best studied problems in online optimization. Takimoto and Warmuth [20], and later Kalai and Vempala [11], gave efficient algorithms for the full information setting. Awerbuch and Kleinberg [2] were the first to give an efficient algorithm with  $O(T^{2/3})$  regret in the partial information (bandit) setting. The recent work of Dani et al [6] implies a  $O(m^{3/2} \sqrt{T})$ -regret algorithm, where  $m = |E|$  is the number of edges in the graph.

Turning to Algorithm 1, we notice that whenever  $\mathcal{K}$  is defined by linear constraints,  $\mathcal{R}$  is defined in a straightforward way (see Section 5.2). As we show below, the online shortest path is an optimization problem on such a set, and we obtain an efficient  $O(m^{3/2} \sqrt{T})$ -regret algorithm.

Formally, the *bandit shortest path* problem is defined as the following repeated game:

Given a directed graph  $G = (V, E)$  and a source-sink pair  $s, t \in V$ , at each time step  $t = 1$  to  $T$ ,

- Player chooses a path  $p_t \in \mathcal{P}_{s,t}$ , where  $\mathcal{P}_{s,t} \subseteq \{E\}^{|V|}$  is the set of all  $s, t$ -paths in the graph
- Adversary independently chooses weights on the edges of the graph  $\mathbf{f}_t \in \mathbb{R}^m$
- Player suffers and observes loss, which is the weighted length of the chosen path  $\sum_{e \in p_t} \mathbf{f}_t(e)$

The problem is transformed into an instance of bandit linear optimization by associating each path with a vector  $\mathbf{x} \in \{0, 1\}^{|E|}$ , where  $\mathbf{x}(i)$  indicates the presence of the  $i$ th edge. The loss is then defined through the dot product  $\mathbf{f}^\top \mathbf{x}$ .

Define the set  $\mathcal{K}$  as the convex hull of the set of paths. It is well-known that this set is the set of flows in the graph and can be defined using  $O(m)$  constraints: positivity constraints and conservation of in-flow and out-flow for every vertex other than source/sink (which have unit out-flow and in-flow, respectively).

Theorem 1 implies that Algorithm 1 attains  $O(m^{3/2} \sqrt{T})$  regret for the bandit linear optimization problem over this set  $\mathcal{K}$ . However, an astute reader would notice that with this definition of  $\mathcal{K}$ , the algorithm produces a flow  $\mathbf{y}_t \in \mathcal{K}$ , not necessarily a path, at each round. The loss suffered by the online player is  $\mathbf{f}_t^\top \mathbf{y}_t$  and the game is specified differently from the bandit shortest path.

However, it is easy to convert this flow algorithm into a randomized online shortest path algorithm: according to the standard flow decomposition theorem (see e.g. [17]), a given flow in the graph can be decomposed into a distribution over at most  $m + 1$  paths in polynomial time. Hence, given a flow  $\mathbf{y}_t \in \mathcal{K}$ , one can obtain an unbiased estimator for  $\mathbf{f}_t^\top \mathbf{y}_t$  by choosing a path according to the distribution of the decomposition, and estimating  $\mathbf{f}_t^\top \mathbf{y}_t$  by the length of this path. In fact, we have the following general statement.

**Proposition 1** *Suppose that, having computed  $\mathbf{y}_t$  in step (1) of Algorithm 1, we predict a random  $\tilde{\mathbf{y}}_t \in \mathcal{K}$  such that  $\mathbb{E} \tilde{\mathbf{y}}_t = \mathbf{y}_t$ , and in step (1) observe  $\mathbf{f}_t^\top \tilde{\mathbf{y}}_t$ . If we use this observed value instead of  $\mathbf{f}_t^\top \mathbf{y}_t$  in step (1), the expected regret of the modified algorithm is the same as that of Algorithm 1.*

The proposition implies that the modified algorithm attains low regret for games defined over discrete sets of possible predictions for the player. This is achieved by working with the *convex hull* of the discrete set while predicting in the original set. In particular, the modification allows us to predict a legal path while the algorithm works with the set of flows.

The proof of Proposition 1 is straightforward: following closely the proof of Theorem 1, we observe that the value  $\mathbf{f}_t^\top \mathbf{y}_t$  is used in only two places. The first is in Equation (9), where it is upper-bounded by 1, and the second is in the proof of the fact that  $\tilde{\mathbf{f}}_t$  is unbiased.

## 8 Proof of the regret bound

### 8.1 Unbiasedness

First, we show that  $\mathbb{E} \tilde{\mathbf{f}}_t = \mathbf{f}_t$ . Condition on the choice  $i_t$  and average over the choice of  $\varepsilon_t$ :

$$\begin{aligned} \mathbb{E}_{\varepsilon_t} \tilde{\mathbf{f}}_t &= \frac{1}{2} n \left( \mathbf{f}_t \cdot (\mathbf{x}_t + \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}) \right) \lambda_{i_t}^{1/2} \cdot \mathbf{e}_{i_t} \\ &\quad - \frac{1}{2} n \left( \mathbf{f}_t \cdot (\mathbf{x}_t - \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}) \right) \lambda_{i_t}^{1/2} \cdot \mathbf{e}_{i_t} \\ &= n (\mathbf{f}_t^\top \mathbf{e}_{i_t}) \mathbf{e}_{i_t}. \end{aligned}$$

Hence,

$$\mathbb{E} \tilde{\mathbf{f}}_t = n (\mathbb{E}_{i_t} \mathbf{e}_{i_t} \mathbf{e}_{i_t}^\top) \mathbf{f}_t = \mathbf{f}_t.$$

Furthermore,  $\mathbb{E} \mathbf{y}_t = \mathbf{x}_t$ .

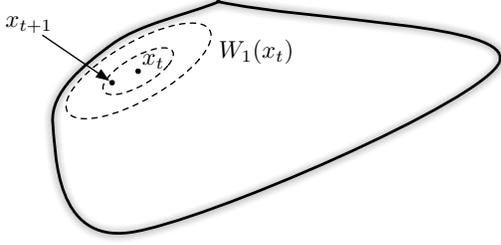


Figure 1: The Dikin ellipsoid  $W_1(\mathbf{x}_t)$  at  $\mathbf{x}_t$ . The next minimum is guaranteed to lie in its scaled version  $W_{4n\eta}(\mathbf{x}_t)$ .

## 8.2 Closeness of the next minimum

We now use the properties of the Dikin ellipsoids mentioned in the previous section.

**Lemma 6** *The next minimizer  $\mathbf{x}_{t+1}$  is “close” to  $\mathbf{x}_t$ :*

$$\mathbf{x}_{t+1} \in W_{4n\eta}(\mathbf{x}_t).$$

**Proof:**

Recall that

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{x} \in \mathcal{K}} \Phi_t(\mathbf{x}) \quad \text{and} \quad \mathbf{x}_t = \arg \min_{\mathbf{x} \in \mathcal{K}} \Phi_{t-1}(\mathbf{x})$$

where  $\Phi_t(\mathbf{x}) = \eta \sum_{s=1}^t \tilde{\mathbf{f}}_s^\top \mathbf{x} + \mathcal{R}(\mathbf{x})$ . Since  $\nabla \Phi_{t-1}(\mathbf{x}_t) = 0$ , we conclude that  $\nabla \Phi_t(\mathbf{x}_t) = \eta \tilde{\mathbf{f}}_t$ .

Consider any point in  $\mathbf{z} \in W_{\frac{1}{2}}(\mathbf{x}_t)$ . It can be written as  $\mathbf{z} = \mathbf{x}_t + \alpha \mathbf{u}$  for some vector  $\mathbf{u}$  such that  $\|\mathbf{u}\|_{\mathbf{x}_t} = 1$  and  $\alpha \in (-\frac{1}{2}, \frac{1}{2})$ . Expanding,

$$\begin{aligned} \Phi_t(\mathbf{z}) &= \Phi_t(\mathbf{x}_t + \alpha \mathbf{u}) \\ &= \Phi_t(\mathbf{x}_t) + \alpha \nabla \Phi_t(\mathbf{x}_t)^\top \mathbf{u} + \alpha^2 \frac{1}{2} \mathbf{u}^\top \nabla^2 \Phi_t(\xi) \mathbf{u} \\ &= \Phi_t(\mathbf{x}_t) + \alpha \eta \tilde{\mathbf{f}}_t^\top \mathbf{u} + \alpha^2 \frac{1}{2} \mathbf{u}^\top \nabla^2 \Phi_t(\xi) \mathbf{u} \end{aligned}$$

for some  $\xi$  on the path between  $\mathbf{x}_t$  and  $\mathbf{x}_t + \alpha \mathbf{u}$ .

Let us check where the optimum of the RHS is obtained. Setting the derivative with respect to  $\alpha$  to zero, we obtain

$$|\alpha^*| = \frac{\eta |\tilde{\mathbf{f}}_t^\top \mathbf{u}|}{\mathbf{u}^\top \nabla^2 \Phi_t(\xi) \mathbf{u}} = \frac{\eta |\tilde{\mathbf{f}}_t^\top \mathbf{u}|}{\mathbf{u}^\top \nabla^2 \mathcal{R}(\xi) \mathbf{u}}.$$

The fact that  $\xi$  is on the line  $\mathbf{x}_t$  to  $\mathbf{x}_t + \alpha \mathbf{u}$  implies that  $\|\xi - \mathbf{x}_t\|_{\mathbf{x}_t} \leq \|\alpha \mathbf{u}\|_{\mathbf{x}_t} < \frac{1}{2}$ . Hence, by Eq (5),

$$\nabla^2 \mathcal{R}(\xi) \succeq (1 - \|\xi - \mathbf{x}_t\|_{\mathbf{x}_t})^2 \nabla^2 \mathcal{R}(\mathbf{x}_t) \succ \frac{1}{4} \nabla^2 \mathcal{R}(\mathbf{x}_t).$$

Thus  $\mathbf{u}^\top \nabla^2 \mathcal{R}(\xi) \mathbf{u} > \frac{1}{4} \|\mathbf{u}\|_{\mathbf{x}_t} = \frac{1}{4}$ , and hence

$$|\alpha^*| < 4\eta |\tilde{\mathbf{f}}_t^\top \mathbf{u}|.$$

Recall that  $\tilde{\mathbf{f}}_t = n(\mathbf{f}_t \cdot \mathbf{y}_t) \varepsilon_t \lambda_{i_t}^{1/2} \cdot \mathbf{e}_{i_t}$  and so  $\tilde{\mathbf{f}}_t^\top \mathbf{u}$  is maximized/minimized when  $\mathbf{u}$  is a unit (with respect to  $\|\cdot\|_{\mathbf{x}_t}$ ) vector in the direction of  $\mathbf{e}_{i_t}$ , i.e.  $\mathbf{u} = \pm \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}$ . We conclude that

$$|\tilde{\mathbf{f}}_t^\top \mathbf{u}| \leq n |\mathbf{f}_t \cdot \mathbf{y}_t| \leq n \quad (9)$$

and

$$|\alpha^*| < 4n\eta < \frac{1}{2}$$

by our choice of  $\eta$  and  $T$ . We conclude that the local optimum  $\arg \min_{\mathbf{z} \in W_{\frac{1}{2}}(\mathbf{x}_t)} \Phi_t(\mathbf{z})$  is strictly inside  $W_{4n\eta}(\mathbf{x}_t)$ , and since  $\Phi_t$  is convex, the global optimum is

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{z} \in \mathcal{K}} \Phi_t(\mathbf{z}) \in W_{4n\eta}(\mathbf{x}_t). \quad \blacksquare$$

## 8.3 Proof of Theorem 1

We are now ready to prove the regret bound for Algorithm 1. Since  $\mathbf{x}_{t+1} \in W_{4n\eta}(\mathbf{x}_t)$ , we invoke Eq (6) at  $\mathbf{x} = \mathbf{x}_t$  and  $\mathbf{z} = \mathbf{h} = \mathbf{x}_{t+1} - \mathbf{x}_t$ :

$$\|\mathbf{h}^\top (\nabla \mathcal{R}(\mathbf{x}_{t+1}) - \nabla \mathcal{R}(\mathbf{x}_t))\| \leq \frac{\|\mathbf{h}\|_{\mathbf{x}_t}^2}{1 - \|\mathbf{h}\|_{\mathbf{x}_t}}.$$

Observe that  $\mathbf{x}_{t+1} \in W_{4n\eta}(\mathbf{x}_t)$  implies  $\|\mathbf{h}\|_{\mathbf{x}_t} < 4n\eta$ .

The proof of Lemma 2 (Equation (12) in the Appendix) reveals that

$$\nabla \mathcal{R}(\mathbf{x}_t) - \nabla \mathcal{R}(\mathbf{x}_{t+1}) = \eta \tilde{\mathbf{f}}_t.$$

We have

$$\begin{aligned} \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) &= \eta^{-1} \mathbf{h}^\top (\nabla \mathcal{R}(\mathbf{x}_{t+1}) - \nabla \mathcal{R}(\mathbf{x}_t)) \\ &\leq \eta^{-1} \frac{\|\mathbf{h}\|_{\mathbf{x}_t}^2}{1 - \|\mathbf{h}\|_{\mathbf{x}_t}} \\ &\leq \frac{16n^2\eta}{1 - 4n\eta} \\ &\leq 32n^2\eta. \end{aligned} \quad (10)$$

By Lemma 2, for any  $\mathbf{u} \in \mathcal{K}_{1/\sqrt{T}}$

$$\begin{aligned} \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u}) &\leq \eta^{-1} D_{\mathcal{R}}(\mathbf{u}, \mathbf{x}_1) + \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\leq \eta^{-1} D_{\mathcal{R}}(\mathbf{u}, \mathbf{x}_1) + 32n^2\eta T \\ &= \eta^{-1} (\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)) + 32n^2\eta T \\ &\leq \frac{1}{\eta} (2\vartheta \log T) + 32n^2\eta T, \end{aligned}$$

where the first equality follows since  $\nabla \mathcal{R}(\mathbf{x}_1) = 0$ , by the choice of  $\mathbf{x}_1$ ; the last inequality follows from Equation (7).

Balancing with  $\eta = \frac{\sqrt{\vartheta \log T}}{4n\sqrt{T}}$ , we get

$$\sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u}) \leq 16n\sqrt{\vartheta T \log T}.$$

for any  $\mathbf{u}$  in the scaled set  $\mathcal{K}'$ . Using Lemma 3, which we prove below, we obtain the statement of Theorem 1.

## 8.4 Expected Regret

Note that it is not  $\tilde{\mathbf{f}}_t^\top \mathbf{x}_t$  that the algorithm should be incurring, but rather  $\mathbf{f}_t^\top \mathbf{y}_t$ . However, it is easy to see that these are equal in expectation.

**Proof:**[Lemma 3] Let  $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot | i_1, \dots, i_{t-1}, \varepsilon_1, \dots, \varepsilon_{t-1}]$  denote the conditional expectation. Note that

$$\mathbb{E}_t \tilde{\mathbf{f}}_t^\top \mathbf{x}_t = \mathbf{f}_t^\top \mathbf{x}_t = \mathbb{E}_t \mathbf{f}_t^\top \mathbf{y}_t.$$

Taking expectations on both sides of the bound for  $\tilde{\mathbf{f}}_t$ 's,

$$\begin{aligned} \mathbb{E} \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top \mathbf{x}_t &\leq \mathbb{E} \min_{\mathbf{u} \in \mathcal{K}'} \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top \mathbf{u} + C_T \\ &\leq \min_{\mathbf{u} \in \mathcal{K}'} \mathbb{E} \left( \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top \mathbf{u} \right) + C_T \\ &= \min_{\mathbf{u} \in \mathcal{K}'} \mathbb{E} \left( \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{u} \right) + C_T. \end{aligned}$$

■

In the case of an oblivious adversary,

$$\min_{\mathbf{u} \in \mathcal{K}'} \mathbb{E} \left( \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{u} \right) = \min_{\mathbf{u} \in \mathcal{K}'} \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{u}.$$

However, if the adversary is not oblivious,  $\mathbf{f}_t$  depends on the random choices at time steps  $1, \dots, t-1$ . Of course, it is desirable to obtain a stronger bound on the regret

$$\mathbb{E} \left[ \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{y}_t - \min_{\mathbf{u} \in \mathcal{K}'} \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{u} \right] = O(\sqrt{T}),$$

which allows the optimal  $\mathbf{u}$  to depend on the randomness of the player<sup>4</sup>. Obtaining guarantees for adaptive adversaries is another dimension of the bandit optimization problem and is beyond the scope of the present paper.

Auer et al [1] provide a clever modification of their EXP3 algorithm which leads to high-probability bounds on the regret, thus guaranteeing low regret against an adaptive adversary. The modification is based on the idea of adding confidence intervals to the losses. The same idea has been employed in the work of [3] (note that [3] is submitted concurrently with this paper) for the bandit optimization over arbitrary convex sets. While the work of [3] does succeed in obtaining a high-probability bound, the algorithm is based on the inefficient method of Dani et al [6], which is a reduction to the algorithm of Auer et al.

## 9 Efficient Implementation

In this section we describe how to efficiently implement Algorithm 1. Recall that in each iteration our algorithm requires the eigen-decomposition of the Hessian in order to derive the unbiased estimator, which takes  $O(n^3)$  time. This is coupled with a convex minimization problem in order to compute  $\mathbf{x}_t$ , which seems to be the most time consuming operation in the entire algorithm.

The message of this section is that the computation of  $\mathbf{x}_t$  given the previous iterate  $\mathbf{x}_{t-1}$  takes essentially only **one iteration of the Damped Newton method**. More precisely, instead of using  $\mathbf{x}_t$  as defined in Algorithm 1, it suffices

<sup>4</sup>It is known that the optimal strategy for the adversary does not need any randomization beyond the player's choices.

to maintain a sequence of points  $\{\mathbf{z}_t\}$ , such that  $\mathbf{z}_t$  is obtained from  $\mathbf{z}_{t-1}$  by only one iteration of the Damped Newton method. The sequence of points  $\{\mathbf{z}_t\}$  are shown to be sufficiently close to  $\{\tilde{\mathbf{x}}_t\}$ , which enjoy the same guarantee as the sequence of  $\{\mathbf{x}_t\}$  defined by Algorithm 1.

A single iteration of the Damped Newton method requires matrix inversion. However, since we have the eigen-decomposition ready made, as it was required for the unbiased estimator, we can produce the inverse and the Newton direction in  $O(n^2)$  time. Thus, the most time-consuming part of the algorithm is the eigen-decomposition of the Hessian, and the total running time is  $O(n^3)$  per iteration.

Before we begin, we require a few more facts from the theory of interior point methods, taken from [15].

Let  $\Psi$  be a non-degenerate self-concordant barrier on domain  $\mathcal{K}$ , for any  $\mathbf{x} \in \mathcal{K}$  define the Newton direction as

$$e(\Psi, \mathbf{x}) = -[\nabla^2 \Psi(\mathbf{x})]^{-1} \nabla \Psi(\mathbf{x})$$

and let the Newton decrement be

$$\lambda(\Psi, \mathbf{x}) = \sqrt{\nabla \Psi(\mathbf{x})^\top [\nabla^2 \Psi(\mathbf{x})]^{-1} \nabla \Psi(\mathbf{x})}.$$

The *Damped Newton iteration* for a given  $\mathbf{x} \in \mathcal{K}$  is

$$DN(\Psi, \mathbf{x}) = \mathbf{x} - \frac{1}{1 + \lambda(\Psi, \mathbf{x})} e(\Psi, \mathbf{x}).$$

The following facts can be found in [15]:

- A:  $DN(\Psi, \mathbf{x}) \in \mathcal{K}$ .<sup>5</sup>
- B:  $\lambda(\Psi, DN(\Psi, \mathbf{x})) \leq 2\lambda(\Psi, \mathbf{x})^2$ .
- C:  $\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \frac{\lambda(\Psi, \mathbf{x})}{1 - \lambda(\Psi, \mathbf{x})}$ .
- D:  $\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}} \leq \frac{\lambda(\Psi, \mathbf{x})}{1 - 2\lambda(\Psi, \mathbf{x})}$ .

Here  $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{K}} \Psi(\mathbf{x})$ .

---

### Algorithm 2 Efficient Implementation

---

- 1: Input:  $\eta > 0$ ,  $\vartheta$ -self-concordant  $\mathcal{R}$ .
- 2: Let  $\mathbf{z}_1 = \arg \min_{\mathbf{x} \in \mathcal{K}} \mathcal{R}(\mathbf{x})$ .
- 3: **for**  $t = 1$  to  $T$  **do**
- 4: Let  $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$  and  $\{\lambda_1, \dots, \lambda_n\}$  be the set of eigenvectors and eigenvalues of  $\nabla^2 \mathcal{R}(\mathbf{z}_t)$ .
- 5: Choose  $i_t$  uniformly at random from  $\{1, \dots, n\}$  and  $\varepsilon_t = \pm 1$  with probability  $1/2$ .
- 6: Predict  $\mathbf{y}_t = \mathbf{z}_t + \varepsilon_t \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}$ .
- 7: Observe the gain  $\mathbf{f}_t^\top \mathbf{y}_t \in \mathbb{R}$ .
- 8: Define  $\hat{\mathbf{f}}_t := n(\mathbf{f}_t^\top \mathbf{y}_t) \varepsilon_t \lambda_{i_t}^{1/2} \cdot \mathbf{e}_{i_t}$ .
- 9: Update

$$\mathbf{z}_{t+1} = \mathbf{z}_t - \frac{1}{1 + \lambda(\Psi_t, \mathbf{z}_t)} e(\Psi_t, \mathbf{z}_t),$$

where

$$\Psi_t(\mathbf{z}) \equiv \eta \sum_{s=1}^t \hat{\mathbf{f}}_s^\top \mathbf{z} + \mathcal{R}(\mathbf{z}).$$

- 10: **end for**
- 

<sup>5</sup>This follows easily since the Newton increment is in the Dikin ellipsoid  $\frac{1}{1 + \lambda(\Psi, \mathbf{x})} e(\Psi, \mathbf{x}) \in W_1(\mathbf{x})$ .

The functions  $\hat{\mathbf{f}}_t$  computed by the above algorithm are unbiased estimates of  $\mathbf{f}_t$  constructed by sampling eigenvectors of  $\nabla^2 \mathcal{R}(\mathbf{z}_t)$ . Define the Follow The Regularized Leader solutions

$$\hat{\mathbf{x}}_{t+1} \equiv \arg \min_{\mathbf{x} \in \mathcal{K}} \Psi_t(\mathbf{x}),$$

on the new functions  $\hat{\mathbf{f}}_t$ 's. The sequence  $\{\hat{\mathbf{x}}_t, \hat{\mathbf{f}}_t\}$  is different from the sequence  $\{\mathbf{x}_t, \tilde{\mathbf{f}}_t\}$  generated by Algorithm 1. However, the same regret bound can be proved for the new algorithm. The only difference from the proof for Algorithm 1 is in the fact that  $\hat{\mathbf{f}}_t$ 's are estimated using the Hessian at  $\mathbf{z}_t$ , not  $\hat{\mathbf{x}}_t$ . However, as we show next,  $\mathbf{z}_t$  is very close to  $\hat{\mathbf{x}}_t$ , and therefore the Hessians are within a factor of 2 by Equation (5), leading to a slightly worse constant for the regret.

**Lemma 7** *It holds that for all  $t$ ,*

$$\lambda^2(\Psi_t, \mathbf{z}_t) \leq 4n^2\eta^2$$

**Proof:** The proof is by induction on  $t$ . For  $t = 1$  the result is true because  $\mathbf{x}_1$  is chosen to minimize  $\mathcal{R}$ . Suppose the statement holds for  $t - 1$ . By definition,

$$\begin{aligned} \lambda^2(\Psi_t, \mathbf{z}_t) &= \nabla \Psi_t(\mathbf{z}_t) [\nabla^2 \Psi_t(\mathbf{z}_t)]^{-1} \nabla \Psi_t(\mathbf{z}_t) \\ &= \nabla \Psi_t(\mathbf{z}_t) [\nabla^2 \mathcal{R}(\mathbf{z}_t)]^{-1} \nabla \Psi_t(\mathbf{z}_t). \end{aligned}$$

Note that

$$\nabla \Psi_t(\mathbf{z}_t) = \nabla \Psi_{t-1}(\mathbf{z}_t) + \eta \hat{\mathbf{f}}_t^\top.$$

Using  $(x + y)^\top A(x + y) \leq 2x^\top A x + 2y^\top A y$  we obtain

$$\begin{aligned} \frac{1}{2} \lambda^2(\Psi_t, \mathbf{z}_t) &\leq \nabla \Psi_{t-1}(\mathbf{z}_t) [\nabla^2 \mathcal{R}(\mathbf{z}_t)]^{-1} \nabla \Psi_{t-1}(\mathbf{z}_t) \\ &\quad + \eta^2 \hat{\mathbf{f}}_t^\top [\nabla^2 \mathcal{R}(\mathbf{z}_t)]^{-1} \hat{\mathbf{f}}_t \\ &= \lambda^2(\Psi_{t-1}, \mathbf{z}_t) + \eta^2 \hat{\mathbf{f}}_t^\top [\nabla^2 \mathcal{R}(\mathbf{z}_t)]^{-1} \hat{\mathbf{f}}_t. \end{aligned}$$

The first term can be bounded by fact (B) and using the induction hypothesis,

$$\lambda^2(\Psi_{t-1}, \mathbf{z}_t) \leq 4\lambda^4(\Psi_{t-1}, \mathbf{z}_{t-1}) \leq 64n^4\eta^4. \quad (11)$$

As for the second term,

$$\hat{\mathbf{f}}_t^\top [\nabla^2 \mathcal{R}(\mathbf{z}_t)]^{-1} \hat{\mathbf{f}}_t \leq n^2$$

because of the way  $\hat{\mathbf{f}}_t$  is defined and since  $|\mathbf{f}_t^\top \mathbf{y}_t| \leq 1$  by assumption. Combining the results,

$$\lambda^2(\Psi_t, \mathbf{z}_t) \leq 128n^4\eta^4 + 2n^2\eta^2 \leq 4n^2\eta^2$$

using the definition of  $\eta$  of Theorem 1 and large enough  $T$ . This proves the induction step.  $\blacksquare$

Note that Equation (11) with the choice of  $\eta$  and large enough  $T$  implies  $\lambda^2(\Psi_{t-1}, \mathbf{z}_t) \ll \frac{1}{2}$ . Using this together with the above Lemma and facts (B) and (C), we conclude that

$$\|\mathbf{z}_t - \hat{\mathbf{x}}_t\|_{\hat{\mathbf{x}}_t} \leq 2\lambda(\Psi_{t-1}, \mathbf{z}_t) \leq 4\lambda(\Psi_{t-1}, \mathbf{z}_{t-1})^2 \leq 16n^2\eta^2$$

We observe that  $\hat{\mathbf{x}}_t$  and  $\mathbf{z}_t$  are very close in the local distance. This implies closeness in  $L_2$  distance as well. Indeed, square roots of inverse eigenvalues  $\lambda_i^{-1/2}$ , being the distances from  $\hat{\mathbf{x}}_t$  to the corresponding radii of the Dikin ellipsoid, can be

at most the  $D$ . Thus,  $\nabla^2 \mathcal{R} \geq D^2 I$  and thus  $\|\mathbf{z}_t - \hat{\mathbf{x}}_t\|_2 \leq D^{-1} \|\mathbf{z}_t - \hat{\mathbf{x}}_t\|_{\hat{\mathbf{x}}_t} \leq 16D^{-1}n^2\eta^2$ .

As we proved, it requires only one Damped Newton update to maintain the sequence  $\mathbf{z}_t$ , which are  $O(1/T)$  close to  $\hat{\mathbf{x}}_t$ . Hence,

$$\sum_{t=1}^T |\mathbf{f}_t^\top(\mathbf{z}_t - \hat{\mathbf{x}}_t)| \leq \sum_{t=1}^T \|\mathbf{f}_t\| \|\mathbf{z}_t - \hat{\mathbf{x}}_t\| = O(1).$$

Therefore, for any  $\mathbf{u} \in \mathcal{K}$

$$\begin{aligned} \mathbb{E} \sum_{t=1}^T \mathbf{f}_t^\top(\mathbf{y}_t - \mathbf{u}) &= \mathbb{E} \sum_{t=1}^T \hat{\mathbf{f}}_t^\top(\mathbf{z}_t - \mathbf{u}) \\ &= \mathbb{E} \sum_{t=1}^T \hat{\mathbf{f}}_t^\top(\hat{\mathbf{x}}_t - \mathbf{u}) + \mathbb{E} \sum_{t=1}^T \hat{\mathbf{f}}_t^\top(\mathbf{z}_t - \hat{\mathbf{x}}_t) \\ &= \mathbb{E} \sum_{t=1}^T \hat{\mathbf{f}}_t^\top(\hat{\mathbf{x}}_t - \mathbf{u}) + \mathbb{E} \sum_{t=1}^T \mathbf{f}_t^\top(\mathbf{z}_t - \hat{\mathbf{x}}_t) \\ &= \mathbb{E} \sum_{t=1}^T \hat{\mathbf{f}}_t^\top(\hat{\mathbf{x}}_t - \mathbf{u}) + O(1) \end{aligned}$$

A slight modification of the proofs of Section 8 leads to a  $O(\sqrt{T})$  bound on the expected regret of the sequence  $\{\hat{\mathbf{x}}_t\}$ .

#### Acknowledgments.

We would like to thank Peter Bartlett for numerous illuminating discussions. We gratefully acknowledge the support of DARPA under grant FA8750-05-2-0249 and NSF under grant DMS-0707060.

#### References

- [1] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2003.
- [2] Baruch Awerbuch and Robert D. Kleinberg. Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 45–53, New York, NY, USA, 2004. ACM.
- [3] P. Bartlett, V. Dani, T. Hayes, S. Kakade, A. Rakhlin, and A. Tewari. High-probability bounds for the regret of bandit online linear optimization, 2008. In submission to COLT 2008.
- [4] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, volume 2 of *MPS/SIAM Series on Optimization*. SIAM, Philadelphia, 2001.
- [5] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [6] Varsha Dani, Thomas Hayes, and Sham Kakade. The price of bandit information for online optimization. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.

- [7] Varsha Dani and Thomas P. Hayes. Robbing the bandit: less regret in online geometric optimization against an adaptive adversary. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 937–943, New York, NY, USA, 2006. ACM.
- [8] Meir Feder, Neri Merhav, and Michael Gutman. Correction to 'universal prediction of individual sequences' (jul 92 1258-1270). *IEEE Transactions on Information Theory*, 40(1):285, 1994.
- [9] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.
- [10] D. P. Helmbold, J. Kivinen, and M. K. Warmuth. Relative loss bounds for single neurons. *IEEE Transactions on Neural Networks*, 10(6):1291–1304, November 1999.
- [11] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.
- [12] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Inf. Comput.*, 132(1):1–63, 1997.
- [13] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.
- [14] H. Brendan McMahan and Avrim Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *COLT*, pages 109–123, 2004.
- [15] A.S. Nemirovskii. Interior point polynomial time methods in convex programming, 2004. Lecture Notes.
- [16] Y. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.
- [17] Satish Rao. Lecture notes: Cs 270, graduate algorithms. 2006.
- [18] Herbert Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 1952.
- [19] Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Mach. Learn.*, 69(2-3):115–142, 2007.
- [20] Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. *J. Mach. Learn. Res.*, 4:773–818, 2003.
- [21] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

## A Proofs

**Proof:** [Lemma 2]

Since the argmin is in the set,  $\nabla\Phi_{t-1}(\mathbf{x}_t) = 0$  and

$$D_{\Phi_{t-1}}(\mathbf{u}, \mathbf{x}_t) = \Phi_{t-1}(\mathbf{u}) - \Phi_{t-1}(\mathbf{x}_t).$$

Moreover,

$$\Phi_t(\mathbf{u}) = \Phi_{t-1}(\mathbf{u}) + \eta \tilde{\mathbf{f}}_t^\top \mathbf{u}.$$

Combining the above,

$$\eta \tilde{\mathbf{f}}_t^\top \mathbf{u} = D_{\Phi_t}(\mathbf{u}, \mathbf{x}_{t+1}) + \Phi_t(\mathbf{x}_{t+1}) - \Phi_{t-1}(\mathbf{u})$$

and

$$\eta \tilde{\mathbf{f}}_t^\top \mathbf{x}_t = D_{\Phi_t}(\mathbf{x}_t, \mathbf{x}_{t+1}) + \Phi_t(\mathbf{x}_{t+1}) - \Phi_{t-1}(\mathbf{x}_t).$$

Thus,

$$\eta \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u}) = D_{\Phi_t}(\mathbf{x}_t, \mathbf{x}_{t+1}) + D_{\Phi_{t-1}}(\mathbf{u}, \mathbf{x}_t) - D_{\Phi_t}(\mathbf{u}, \mathbf{x}_{t+1}).$$

Summing over  $t = 1 \dots T$ ,

$$\begin{aligned} \eta \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u}) &= D_{\Phi_0}(\mathbf{u}, \mathbf{x}_1) - D_{\Phi_T}(\mathbf{u}, \mathbf{x}_{T+1}) \\ &\quad + \sum_{t=1}^T D_{\Phi_t}(\mathbf{x}_t, \mathbf{x}_{t+1}) \\ &\leq D_{\Phi_0}(\mathbf{u}, \mathbf{x}_1) + \sum_{t=1}^T D_{\Phi_t}(\mathbf{x}_t, \mathbf{x}_{t+1}) \end{aligned}$$

By definition,  $\mathbf{x}_t$  satisfies  $\sum_{s=1}^{t-1} \tilde{\mathbf{f}}_s + \nabla \mathcal{R}(\mathbf{x}_t) = 0$  and  $\mathbf{x}_{t+1}$  satisfies  $\sum_{s=1}^t \tilde{\mathbf{f}}_s + \nabla \mathcal{R}(\mathbf{x}_{t+1}) = 0$ . Subtracting,

$$\nabla \mathcal{R}(\mathbf{x}_t) - \nabla \mathcal{R}(\mathbf{x}_{t+1}) = \eta \tilde{\mathbf{f}}_t. \quad (12)$$

Now we realize that

$$\begin{aligned} D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) &\leq D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) + D_{\mathcal{R}}(\mathbf{x}_{t+1}, \mathbf{x}_t) \\ &= -\nabla \mathcal{R}(\mathbf{x}_{t+1})(\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\quad - \nabla \mathcal{R}(\mathbf{x}_t)(\mathbf{x}_{t+1} - \mathbf{x}_t) \\ &= \eta \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}). \end{aligned}$$

■

**Lemma 8** For any point  $\mathbf{x} \in \mathcal{K}$ , it holds that

$$\min_{\mathbf{y} \in \mathcal{K}_\delta} \|\mathbf{x} - \mathbf{y}\| \leq \delta.$$

**Proof:** Consider the point on the segment  $[\mathbf{x}_1, \mathbf{x}]$  which intersects the boundary of  $\mathcal{K}_\delta$ , denote it  $\mathbf{z}$ . By definition, we have

$$\frac{\|\mathbf{z} - \mathbf{x}_1\|}{\|\mathbf{x} - \mathbf{x}_1\|} = \frac{1}{1 + \delta}.$$

As  $\mathbf{x}, \mathbf{x}_1, \mathbf{z}$  are on the same line

$$\|\mathbf{z} - \mathbf{x}\| = \|\mathbf{x} - \mathbf{x}_1\| - \|\mathbf{z} - \mathbf{x}_1\| = \|\mathbf{x} - \mathbf{x}_1\| \cdot \left(1 - \frac{1}{1 + \delta}\right) \leq \delta.$$

The last inequality holds by our assumption that the diameter of  $\mathcal{K}$  is bounded by one. The lemma follows. ■



---

# Combining Expert Advice Efficiently

---

Wouter M. Koolen and Steven de Rooij

Centrum voor Wiskunde en Informatica (CWI)

Kruislaan 413, P.O. Box 94079

1090 GB Amsterdam, The Netherlands

{W.M.Koolen-Wijkstra,S.de.Rooij}@cwi.nl

## Abstract

We show how models for prediction with expert advice can be defined concisely and clearly using hidden Markov models (HMMs); standard HMM algorithms can then be used to efficiently calculate how the expert predictions should be weighted according to the model. We cast many existing models as HMMs and recover the best known running times in each case. We also describe two new models: the switch distribution, which was recently developed to improve Bayesian/Minimum Description Length model selection, and a new generalisation of the fixed share algorithm based on run-length coding. We give loss bounds for all models and shed new light on the relationships between them.

## 1 Introduction

We cannot predict exactly how complicated processes such as the weather, the stock market, social interactions and so on, will develop into the future. Nevertheless, people do make weather forecasts and buy shares all the time. Such predictions can be based on formal models, or on human expertise or intuition. An investment company may even want to choose between portfolios on the basis of a combination of these kinds of predictors. In such scenarios, predictors typically cannot be considered “true”. Thus, we may well end up in a position where we have a whole collection of prediction strategies, or *experts*, each of whom has *some* insight into *some* aspects of the process of interest. We address the question how a given set of experts can be combined into a single predictive strategy that is as good as, or if possible even better than, the best individual expert.

The setup is as follows. Let  $\Xi$  be a finite set of experts. Each expert  $\xi \in \Xi$  issues a distribution  $P_\xi(\mathbf{x}_{n+1}|x^n)$  on the next outcome  $\mathbf{x}_{n+1}$  given the previous observations  $x^n := x_1, \dots, x_n$ . Here, each outcome  $x_i$  is an element of some countable space  $\mathcal{X}$ , and random variables are written in bold face. The probability that an expert assigns to a sequence of outcomes is given by the chain rule:  $P_\xi(x^n) = P_\xi(x_1) \cdot P_\xi(x_2|x_1) \cdot \dots \cdot P_\xi(x_n|x^{n-1})$ .

A standard Bayesian approach to combine the expert predictions is to define a prior  $w$  on the experts  $\Xi$  which induces a joint distribution with mass function  $P(x^n, \xi) =$

$w(\xi)P_\xi(x^n)$ . Inference is then based on this joint distribution. We can compute, for example: (a) the *marginal probability* of the data  $P(x^n) = \sum_{\xi \in \Xi} w(\xi)P_\xi(x^n)$ , (b) the *predictive distribution* on the next outcome  $P(\mathbf{x}_{n+1}|x^n) = P(x^n, \mathbf{x}_{n+1})/P(x^n)$ , which defines a prediction strategy that combines those of the individual experts, or (c) the *posterior distribution* on the experts  $P(\xi|x^n) = P_\xi(x^n)w(\xi)/P(x^n)$ , which tells us how the experts’ predictions should be weighted. This simple probabilistic approach has the advantage that it is computationally easy: predicting  $n$  outcomes using  $|\Xi|$  experts requires only  $O(n \cdot |\Xi|)$  time. Additionally, this Bayesian strategy guarantees that the overall probability of the data is only a factor  $w(\hat{\xi})$  smaller than the probability of the data according to the best available expert  $\hat{\xi}$ . On the flip side, with this strategy we never do any *better* than  $\hat{\xi}$  either: we have  $P_{\hat{\xi}}(x^n) \geq P(x^n) \geq P_\xi(x^n)w(\hat{\xi})$ , which means that potentially valuable insights from the other experts are not used to our advantage!

More sophisticated combinations of prediction strategies can be found in the literature under various headings, including (Bayesian) statistics, source coding and universal prediction. In the latter the experts’ predictions are not necessarily probabilistic, and scored using an arbitrary loss function. In this paper we consider only logarithmic loss, although our results can probably be generalised to the framework described in, e.g. [12].

The three main contributions of this paper are the following. First, we introduce prior distributions on *sequences* of experts, which allows unified description of many existing models. Second, we show how HMMs can be used as an intuitive graphical language to describe such priors and obtain computationally efficient prediction strategies. Third, we use this new approach to describe and analyse several important existing models, as well as one recent and one completely new model for expert tracking.

### 1.1 Overview

In §2 we develop a new, more general framework for combining expert predictions, where we consider the possibility that the *optimal* weights used to mix the expert predictions may *vary over time*, i.e. as the sample size increases. We stick to Bayesian methodology, but we define the prior distribution as a probability measure on *sequences of experts* rather than on experts. The prior probability of a sequence

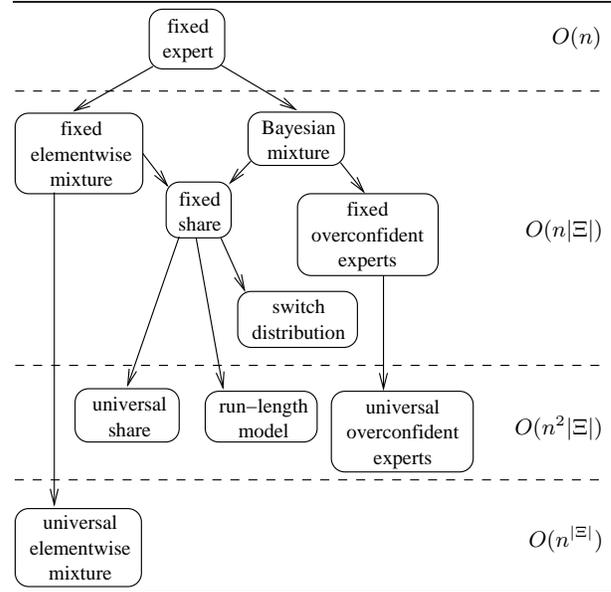
$\xi_1, \xi_2, \dots$  is the probability that we rely on expert  $\xi_1$ 's prediction of the first outcome and expert  $\xi_2$ 's prediction of the second outcome, etc. This allows for the expression of more sophisticated models for the combination of expert predictions. For example, the nature of the data generating process may evolve over time; consequently different experts may be better during different periods of time. It is also possible that not the data generating process, but the experts themselves change as more and more outcomes are being observed: they may learn from past mistakes, possibly at different rates, or they may have occasional bad days, etc. In both situations we may hope to benefit from more sophisticated modelling.

Of course, not all models for combining expert predictions are computationally feasible. §3 describes a methodology for the specification of models that allow efficient evaluation. We achieve this by using hidden Markov models (HMMs) on two levels. On the first level, we use an HMM as a formal specification of a distribution on sequences of *experts* as defined in §2. We introduce a graphical language to conveniently represent its structure. These graphs help to understand and compare existing models and to design new ones. We then modify this first HMM to construct a second HMM that specifies the distribution on sequences of *outcomes*. Subsequently, we can use the standard dynamic programming algorithms for HMMs (forward, backward and Viterbi) on both levels to efficiently calculate most relevant quantities, most importantly the marginal probability of the observed outcomes  $P(x^n)$  and posterior weights on the next expert given the previous observations  $P(\xi_{n+1}|x^n)$ .

It turns out that many existing models for prediction with expert advice can be specified as HMMs. We provide an overview in §4 by giving the graphical representations of the HMMs corresponding to the following three models. First, universal elementwise mixtures (sometimes called mixture models) that learn the optimal mixture parameter from data. Second, Herbster and Warmuth's fixed share algorithm for tracking the best expert [4, 5]. Third, universal share, which was introduced by Volf and Willems as *the switching method* [11] and later independently proposed by Bousquet [1]. Here the goal is to learn the optimal fixed-share parameter from data. We render each model as a prior on sequences of experts by giving its HMM. The size of the HMM immediately determines the required running time for the forward algorithm. The generalisation relationships between these models as well as their running times are displayed in Figure 1. In each case this running time coincides with that of the best known algorithm. We also give a loss bound for each model, relating the loss of the model to the loss of the best competitor among a set of alternatives in the worst case. Such loss bounds can help select between different models for specific prediction tasks.

Besides the models found in the literature, Figure 1 also includes two new generalisations of fixed share: the switch distribution and the run-length model. These models are the subject of §5. The switch distribution was introduced in [10] as a practical means of improving Bayes/Minimum Description Length prediction to achieve the optimal rate of convergence in nonparametric settings. Here we give the concrete HMM that allows for its linear time computation. The run-length model is based on a distribution on the number of

**Figure 1** Expert sequence priors: generalisation relationships and run time



successive outcomes that are typically well-predicted by the same expert. Run-length codes are typically applied directly to the data, but in our novel application they define the prior on expert sequences instead. Again, we provide the graphical representation of their defining HMMs as well as loss bounds. We conclude by comparing the two models.

## 2 Expert Sequence Priors

In this section we explain how expert tracking can be described in probability theory using expert sequence priors (ES-priors). These ES-priors are distributions on the space of infinite sequences of experts that are used to express regularities in the development of the relative quality of the experts' predictions. As illustrations we render Bayesian mixtures and elementwise mixtures as ES-priors. In the next section we show how ES-priors can be implemented efficiently by hidden Markov models.

**Notation** We denote by  $\mathbb{N}$  the natural numbers including zero, and by  $\mathbb{Z}_+$  the natural numbers excluding zero. Let  $Q$  be a set. We denote the cardinality of  $Q$  by  $|Q|$ . For any natural number  $n$ , we let the variable  $q^n$  range over the  $n$ -fold Cartesian product  $Q^n$ , and we write  $q^n = \langle q_1, \dots, q_n \rangle$ . We also let  $q^\omega$  range over  $Q^\omega$  — the set of infinite sequences over  $Q$  — and write  $q^\omega = \langle q_1, \dots \rangle$ . We read the statement  $q^\lambda \in Q^{\leq \omega}$  to first bind  $\lambda \leq \omega$  and subsequently  $q^\lambda \in Q^\lambda$ . If  $q^\lambda$  is a sequence, and  $\kappa \leq \lambda$ , then we denote by  $q^\kappa$  the prefix of  $q^\lambda$  of length  $\kappa$ .

**Forecasting System** Let  $\mathcal{X}$  be a countable outcome space. We use the notation  $\mathcal{X}^*$  for the set of all finite sequences over  $\mathcal{X}$  and let  $\Delta(\mathcal{X})$  denote the set of all probability mass functions on  $\mathcal{X}$ . A (*prequential*)  $\mathcal{X}$ -forecasting system (PFS) is a function  $P : \mathcal{X}^* \rightarrow \Delta(\mathcal{X})$  that maps sequences of previous observations to a predictive distribution on the next outcome.

Prequential forecasting systems were introduced by Dawid in [2].

**Distributions** We also use probability measures on spaces of infinite sequences. In such a space, a basic event is the set of all continuations of a given prefix. We identify such events with their prefix. Thus a distribution on  $\mathcal{X}^\omega$  is defined by a function  $P : \mathcal{X}^* \rightarrow [0, 1]$  that satisfies  $P(\epsilon) = 1$ , where  $\epsilon$  is the empty sequence, and for all  $n \geq 0$ , all  $x^n \in \mathcal{X}^n$  we have  $\sum_{x \in \mathcal{X}} P(x_1, \dots, x_n, x) = P(x^n)$ . We identify  $P$  with the distribution it defines. We write  $P(x^n|x^m)$  for  $P(x^n)/P(x^m)$  if  $0 \leq m \leq n$ .

Note that forecasting systems continue to make predictions even after they have assigned probability 0 to a previous outcome, while distributions' predictions become undefined. Nonetheless we use the same notation: we write  $P(x_{n+1}|x^n)$  for the probability that a forecasting system  $P$  assigns to the  $n + 1$ st outcome given the first  $n$  outcomes, as if  $P$  were a distribution.

**ES-Priors** The slogan of this paper is *we do not understand the data*. Instead of modelling the data, we work with experts. We assume that there is a fixed set of experts  $\Xi$ , and that each expert  $\xi \in \Xi$  predicts using a forecasting system  $P_\xi$ .

We are interested in switching between different forecasting systems at different sample sizes. For a sequence of experts with prefix  $\xi^n$ , the combined forecast, where expert  $\xi_i$  predicts the  $i$ th outcome, is denoted

$$P_{\xi^n}(x^n) := \prod_{i=1}^n P_{\xi_i}(x_i|x^{i-1}).$$

Adopting Bayesian methodology, we impose a prior  $\pi$  on infinite sequences of experts; this prior is called an *expert sequence prior* (ES-prior). Inference is then based on the distribution on the joint space  $(\mathcal{X} \times \Xi)^\omega$ , called the *ES-joint*, which is defined as follows:

$$P(\langle \xi_1, x_1 \rangle, \dots, \langle \xi_n, x_n \rangle) := \pi(\xi^n) P_{\xi^n}(x^n). \quad (1)$$

We adopt shorthand notation for events: we write  $P(S)$ , where  $S$  is a subsequence of  $\xi^n$  and/or of  $x^n$ , for the probability under  $P$  of the set of sequences of pairs which match  $S$  exactly. For example, the marginal probability of a sequence of outcomes is:

$$P(x^n) = \sum_{\xi^n \in \Xi^n} P(\xi^n, x^n). \quad (2)$$

Compare this to the usual Bayesian statistics, where a model class  $\{P_\theta \mid \theta \in \Theta\}$  is also endowed with a prior distribution  $w$  on  $\Theta$ . Then, after observing outcomes  $x^n$ , inference is based on the posterior  $P(\theta|x^n)$  on the parameter, which is never actually observed. Our approach is exactly the same, but we always consider  $\Theta = \Xi^\omega$ . Thus as usual our predictions are based on the posterior  $P(\xi^\omega|x^n)$ . However, since the predictive distribution of  $x_{n+1}$  only depends on  $\xi_{n+1}$  (and  $x^n$ ) we always marginalise as follows:

$$P(\xi_{n+1}|x^n) = \frac{\sum_{\xi^n} P(\xi^n, x^n) \cdot \pi(\xi_{n+1}|\xi^n)}{\sum_{\xi^n} P(\xi^n, x^n)}. \quad (3)$$

At each moment in time we predict the data using the posterior, which is a mixture over our experts' predictions. Ideally, the ES-prior  $\pi$  should be chosen such that the posterior coincides with the optimal mixture weights of the experts at each sample size. The traditional interpretation of our ES-prior as a representation of belief about an unknown "true" expert sequence is tenuous, as normally experts do not generate data, they only predict it. Moreover, by mixing different expert sequences, it is often possible to predict significantly better than by using any single sequence of experts, a feature that is crucial to the performance of many of the models that will be described below and in §4. In the remainder of this paper we motivate ES-priors by giving performance guarantees in the form of bounds on running time and loss.

## 2.1 Examples

We now show how two ubiquitous models can be rendered as ES-priors.

**Example 2.1.1** (Bayesian Mixtures). Let  $\Xi$  be a set of experts, and let  $P_\xi$  be a PFS for each  $\xi \in \Xi$ . Suppose that we do not know which expert will make the best predictions. Following the usual Bayesian methodology, we combine their predictions by conceiving a prior  $w$  on  $\Xi$ , which (depending on the adhered philosophy) may or may not be interpreted as an expression of one's beliefs in this respect. Then the standard Bayesian mixture  $P_{\text{bayes}}$  is given by

$$P_{\text{bayes}}(x^n) = \sum_{\xi \in \Xi} P_\xi(x^n) w(\xi). \quad (4)$$

Recall that  $P_\xi(x^n)$  means  $\prod_{i=1}^n P_\xi(x_i|x^{i-1})$ . The Bayesian mixture is not an ES-joint, but it can easily be transformed into one by using the ES-prior that assigns probability  $w(\xi)$  to the identically- $\xi$  sequence for each  $\xi \in \Xi$ :

$$\pi_{\text{bayes}}(\xi^n) = \begin{cases} w(k) & \text{if } \xi_i = k \text{ for all } i = 1, \dots, n, \\ 0 & \text{o.w.} \end{cases}$$

We will use the adjective "Bayesian" generously throughout this paper, but when we write *the standard Bayesian ES-prior* this always refers to  $\pi_{\text{bayes}}$ .  $\diamond$

**Example 2.1.2** (Elementwise Mixtures). The *elementwise mixture*<sup>1</sup> is formed from some mixture weights  $\alpha \in \Delta(\Xi)$  by

$$P_{\text{mix}, \alpha}(x^n) := \prod_{i=1}^n \left( \sum_{\xi \in \Xi} P_\xi(x_i|x^{i-1}) \alpha(\xi) \right).$$

In the preceding definition, it may seem that elementwise mixtures do not fit in the framework of ES-priors. But we

<sup>1</sup>These mixtures are sometimes just called mixtures, or predictive mixtures. We use the term elementwise mixtures both for descriptive clarity and to avoid confusion with Bayesian mixtures.

can rewrite this definition in the required form as follows:

$$\begin{aligned}
P_{\text{mix},\alpha}(x^n) &= \prod_{i=1}^n \sum_{\xi \in \Xi} P_{\xi}(x_i | x^{i-1}) \alpha(\xi) \\
&= \sum_{\xi^n \in \Xi^n} \prod_{i=1}^n P_{\xi_i}(x_i | x^{i-1}) \alpha(\xi_i) \quad (5a) \\
&= \sum_{\xi^n} P_{\xi^n}(x^n) \pi_{\text{mix},\alpha}(\xi^n),
\end{aligned}$$

which is the ES-joint based on the prior

$$\pi_{\text{mix},\alpha}(\xi^n) := \prod_{i=1}^n \alpha(\xi_i). \quad (5b)$$

Thus, the ES-prior for elementwise mixtures is just the product distribution of  $\alpha$ .  $\diamond$

We mentioned above that ES-priors cannot be interpreted as expressions of belief about individual expert sequences. This is a prime example, as the ES-prior is crafted such that its posterior  $\pi_{\text{mix},\alpha}(\xi_{n+1} | \xi^n)$  exactly coincides with the desired *mixture* of experts.

### 3 Expert Tracking using HMMs

We explained in the previous section how expert tracking can be implemented using expert sequence priors. In this section we specify ES-priors using hidden Markov models (HMMs). The advantage of using HMMs is that the complexity of the resulting expert tracking procedure can be read off directly from the structure of the HMM. We first give a short overview of the particular kind of HMMs that we use throughout this paper. We then show how HMMs can be used to specify ES-priors. As illustrations we render the ES-priors that we obtained for Bayesian mixtures and elementwise mixtures in the previous sections as HMMs. In §4 we provide an overview of ES-priors and their defining HMMs that are found in the literature.

#### 3.1 Hidden Markov Models Overview

Hidden Markov models (HMMs) are a well-known tool for specifying probability distributions on sequences with temporal structure. Furthermore, these distributions are very appealing algorithmically: many important probabilities can be computed efficiently for HMMs. These properties make HMMs ideal models of expert sequences: ES-priors. For an introduction to HMMs, see [9]. We require a slightly more general notion that incorporates silent states and forecasting systems as explained below.

We define our HMMs on a generic set of outcomes  $\mathcal{O}$  to avoid confusion in later sections, where we use HMMs in two different contexts. First in §3.2, we use HMMs to define ES-priors, and instantiate  $\mathcal{O}$  with the set of experts  $\Xi$ . Then in §3.4 we modify the HMM that defines the ES-prior to incorporate the experts' predictions, whereupon  $\mathcal{O}$  is instantiated with the set of observable outcomes  $\mathcal{X}$ .

**Definition 1.** Let  $\mathcal{O}$  be a finite set of outcomes. We call a quintuple

$$\mathbb{A} = \langle Q, Q_p, P_o, P, \langle P_q \rangle_{q \in Q_p} \rangle$$

a *hidden Markov model* on  $\mathcal{O}$  if  $Q$  is a countable set,  $Q_p \subseteq Q$ ,  $P_o \in \Delta(Q)$ ,  $P : Q \rightarrow \Delta(Q)$  and  $P_q$  is an  $\mathcal{O}$ -forecasting system for each  $q \in Q_p$ .

**Terminology and Notation** We call elements of  $Q$  *states*. We call the states in  $Q_p$  *productive* and the other states *silent*. We call  $P_o$  the *initial distribution*, let  $I$  denote its support (i.e.  $I := \{q \in Q \mid P_o(q) > 0\}$ ) and call  $I$  the set of *initial states*. We call  $P$  the *stochastic transition function*. We let  $S_q$  denote the support of  $P(q)$ , and call  $q' \in S_q$  a *direct successor* of  $q$ . We abbreviate  $P(q)(q')$  to  $P(q \rightarrow q')$ . A finite or infinite sequence of states  $q^\lambda \in Q^{\leq \omega}$  is called a *branch* through  $\mathbb{A}$ . A branch  $q^\lambda$  is called a *run* if either  $\lambda = 0$  (so  $q^\lambda = \epsilon$ ), or  $q_1 \in I$  and  $q_{i+1} \in S_{q_i}$  for all  $1 \leq i < \lambda$ . A finite run  $q^n \neq \epsilon$  is called a *run to  $q_n$* . For each branch  $q^\lambda$ , we denote by  $q_p^\lambda$  its subsequence of productive states. We denote the elements of  $q_p^\lambda$  by  $q_1^p, q_2^p$  etc. We call an HMM *continuous* if  $q_p^\omega$  is infinite for each infinite run  $q^\omega$ .

**Restriction** In this paper we will only work with continuous HMMs. This restriction is necessary for the following to be well-defined.

**Definition 2.** An HMM  $\mathbb{A}$  defines the following distribution on sequences of states.  $\pi_{\mathbb{A}}(\epsilon) := 1$ , and for  $\lambda \geq 1$

$$\pi_{\mathbb{A}}(q^\lambda) := P_o(q_1) \prod_{i=1}^{\lambda-1} P(q_i \rightarrow q_{i+1}).$$

Then via the PFSSs,  $\mathbb{A}$  induces the joint distribution  $P_{\mathbb{A}}$  on runs and sequences of outcomes. Let  $o^n \in \mathcal{O}^n$  be a sequence of outcomes and let  $q^\lambda \neq \epsilon$  be a run with at least  $n$  productive states, then

$$P_{\mathbb{A}}(o^n, q^\lambda) := \pi_{\mathbb{A}}(q^\lambda) \prod_{i=1}^n P_{q_i^p}(o_i | o^{i-1}).$$

The value of  $P_{\mathbb{A}}$  at arguments  $o^n, q^\lambda$  that do not fulfil the condition above is determined by the additivity axiom of probability.

**The Forward Algorithm** For a given HMM  $\mathbb{A}$  and data  $o^n$ , the *forward algorithm* (c.f. [9]) computes the marginal probability  $P_{\mathbb{A}}(o^n)$ . The forward algorithm operates by percolating weights along the transitions of the HMM. The running time is proportional to the number of transitions that need to be considered. Details can be found in [6]. In this paper we present all HMMs unfolded, so that each transition needs to be considered exactly once, and hence the running time can be read off easily.

#### 3.2 HMMs as ES-Priors

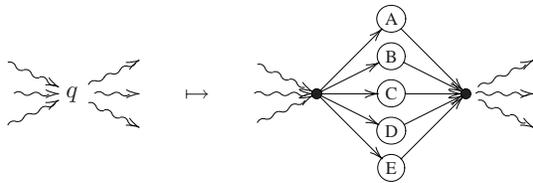
In applications HMMs are often used to model data. This is often useful if there are local correlations between outcomes. A graphical model depicting this approach is displayed in Figure 2a.

In this paper we use HMMs as ES-priors, that is, to specify temporal correlations between the performance of our *experts*. Thus instead of concrete observations our HMMs will “produce” sequences of experts, that are never actually observed. Figure 2b. illustrates this approach.

Using HMMs as priors allows us to use the standard algorithms for HMMs to answer questions about the prior. For example, we can use the forward algorithm to compute the prior probability of the sequence of one hundred experts with expert number one at all odd indices and expert number two at all even indices. However, we are obviously also interested in questions about the data rather than about the prior. In §3.4 we show how joints based on HMM priors (Figure 2c) can be transformed into ordinary HMMs (Figure 2a) with at most a  $|\Xi|$ -fold increase in size, allowing us to use the standard algorithms for HMMs not only for the experts, but for the data as well, with the same increase in complexity. This is the best we can generally hope for, as we now need to integrate over all possible expert sequences instead of considering only a single one. Here we first consider properties of HMMs that represent ES-priors.

**Restriction** HMM priors “generate”, or define the distribution on, sequences of experts. But contrary to the data, which are observed, no concrete sequence of experts is realised. This means that we cannot conveniently condition the distribution on experts in a productive state  $q_n^p$  on the sequence of previously produced experts  $\xi^{n-1}$ . In other words, we can only use an HMM on  $\Xi$  as an ES-prior if the forecasting systems in its states are simply distributions, so that all dependencies between consecutive experts are carried by the state. This is necessary to avoid having to sum over all (exponentially many) possible expert sequences.

**Deterministic** Under the restriction above, but in the presence of silent states, we can make any HMM deterministic in the sense that each forecasting system assigns probability one to a single outcome. We just replace each productive state  $q \in Q_p$  by the following gadget:

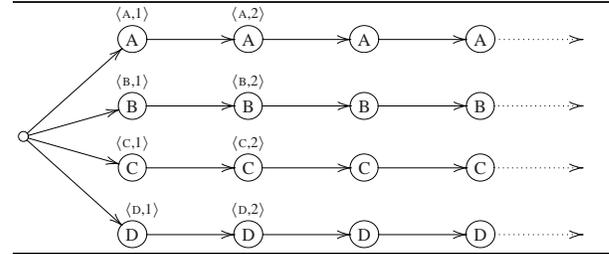


In the left diagram, the state  $q$  has distribution  $P_q$  on outcomes  $\mathcal{O} = \{A, \dots, E\}$ . In the right diagram, the leftmost silent state has transition probability  $P_q(o)$  to a state that deterministically outputs outcome  $o$ . We often make the functional relationship explicit and by calling  $\langle Q, Q_p, P_o, P, \Lambda \rangle$  a *deterministic HMM* on  $\mathcal{O}$  if  $\Lambda : Q_p \rightarrow \mathcal{O}$ . Here we slightly abuse notation; the last component of a (general) HMM assigns a *PFS* to each productive state, while the last component of a deterministic HMM assigns an *outcome* to each productive states.

Sequential prediction using a general HMM or its deterministic counterpart costs the same amount of work: the  $|\mathcal{O}|$ -fold increase in the number of states is compensated by the  $|\mathcal{O}|$ -fold reduction in the number of outcomes that need to be considered per state.

**Diagrams** Deterministic HMMs can be graphically represented by pictures. In general, we draw a node  $N_q$  for each state  $q$ . We draw a small black dot, e.g.  $\bullet$ , for a silent state, and an ellipse labelled  $\Lambda(q)$ , e.g.  $\textcircled{D}$ , for a productive state.

**Figure 3** Standard Bayesian mixture.



We draw an arrow from  $N_q$  to  $N_{q'}$  if  $q'$  is a direct successor of  $q$ . We often reify the initial distribution  $P_o$  by including a virtual node, drawn as an open circle, e.g.  $\circ$ , with an outgoing arrow to  $N_q$  for each initial state  $q \in I$ . The transition probability  $P(q \rightarrow q')$  is not displayed in the graph.

### 3.3 Examples

We are now ready to give the deterministic HMMs that correspond to the ES-priors of our earlier examples from §2.1: Bayesian mixtures and elementwise mixtures with fixed parameters.

**Example 3.3.1** (HMM for Bayesian Mixtures). The Bayesian mixture ES-prior  $\pi_{\text{bayes}}$  as introduced in Example 2.1.1 represents the hypothesis that a single expert predicts best for all sample sizes. A simple deterministic HMM on  $\Xi$  that generates the prior  $\pi_{\text{bayes}}$  is given by  $\mathbb{A}_{\text{bayes}} = \langle Q, Q_p, P_o, P, \Lambda \rangle$ , where

$$Q, Q_p = \Xi \times \mathbb{Z}_+ \quad \Lambda(\xi, n) = \xi \quad P_o(\xi, 1) = w(\xi) \quad (6a)$$

$$P(\langle \xi, n \rangle \rightarrow \langle \xi, n+1 \rangle) = 1 \quad (6b)$$

The diagram of (6) is displayed in Figure 3. From the picture of the HMM it is clear that it computes the Bayesian mixture. Hence, using (4), the loss of the HMM with prior  $w$  is bounded for all data  $x^n$  and all experts  $\xi \in \Xi$  by

$$-\log P_{\mathbb{A}_{\text{bayes}}}(x^n) + \log P_\xi(x^n) \leq -\log w(\xi). \quad (7)$$

In particular this bound holds for  $\hat{\xi} = \text{argmax}_\xi P_\xi(x^n)$ , so we predict as well as the single best expert with *constant* overhead. Also  $P_{\mathbb{A}_{\text{bayes}}}(x^n)$  can obviously be computed in  $O(n|\Xi|)$  using its definition (4). We show in [6] that computing it using the HMM prior above gives the same running time  $O(n|\Xi|)$ , a perfect match.  $\diamond$

**Example 3.3.2** (HMM for Elementwise Mixtures). We now present the deterministic HMM  $\mathbb{A}_{\text{mix}, \alpha}$  that implements the ES-prior  $\pi_{\text{mix}, \alpha}$  of Example 2.1.2. Its diagram is displayed in Figure 4. The HMM has a single silent state per outcome, and its transition probabilities are the mixture weights  $\alpha$ . Formally,  $\mathbb{A}_{\text{mix}, \alpha}$  is given using  $Q = Q_s \cup Q_p$  by

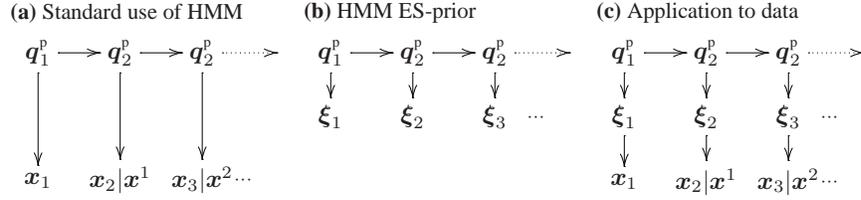
$$Q_s = \{\mathbf{p}\} \times \mathbb{N} \quad P_o(\mathbf{p}, 0) = 1 \quad (8a)$$

$$Q_p = \Xi \times \mathbb{Z}_+ \quad \Lambda(\xi, n) = \xi$$

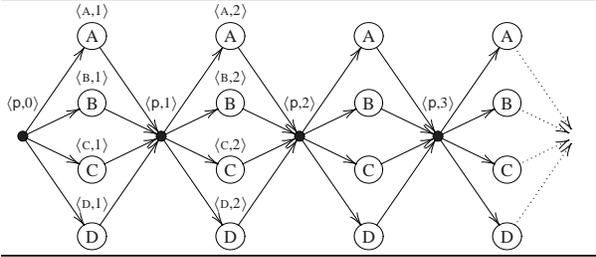
$$P \left( \begin{array}{l} \langle \mathbf{p}, n \rangle \rightarrow \langle \xi, n+1 \rangle \\ \langle \xi, n \rangle \rightarrow \langle \mathbf{p}, n \rangle \end{array} \right) = \left( \begin{array}{l} \alpha(\xi) \\ 1 \end{array} \right) \quad (8b)$$

The vector-style definition of  $P$  is shorthand for one  $P$  per line. We show in [6] that this HMM allows us to compute  $P_{\mathbb{A}_{\text{mix}, \alpha}}(x^n)$  in time  $O(n|\Xi|)$ .  $\diamond$

**Figure 2** HMMs.  $q_i^p$ ,  $\xi_i$  and  $x_i$  are the  $i^{\text{th}}$  productive state, expert and observation.



**Figure 4** Fixed elementwise mixture



### 3.4 The HMM for Data

We obtain our model for the data (Figure 2c) by composing an HMM prior on  $\Xi^\omega$  with a PFS  $P_\xi$  for each expert  $\xi \in \Xi$ . We now show that the resulting marginal distribution on data can be implemented by a single HMM on  $\mathcal{X}$  (Figure 2a) *with the same number of states as the HMM prior*. Let  $P_\xi$  be an  $\mathcal{X}$ -forecasting system for each  $\xi \in \Xi$ , and let the ES-prior  $\pi_{\mathbb{A}}$  be given by the deterministic HMM  $\mathbb{A} = \langle Q, Q_p, P_o, P, \Lambda \rangle$  on  $\Xi$ . Then the marginal distribution of the data (see (1)) is given by

$$P_{\mathbb{A}}(x^n) = \sum_{\xi^n} \pi_{\mathbb{A}}(\xi^n) \prod_{i=1}^n P_{\xi_i}(x_i | x^{i-1}).$$

The HMM  $\mathbb{X} := \langle Q, Q_p, P_o, P, \langle P_{\Lambda(q)} \rangle_{q \in Q_p} \rangle$  on  $\mathcal{X}$  induces the same marginal distribution (see Definition 2). That is,  $P_{\mathbb{X}}(x^n) = P_{\mathbb{A}}(x^n)$ . Moreover,  $\mathbb{X}$  contains only the forecasting systems that also exist in  $\mathbb{A}$  and it retains the structure of  $\mathbb{A}$ . In particular this means that the algorithms for HMMs have the *same* running time on the prior  $\mathbb{A}$  as on the marginal  $\mathbb{X}$ .

## 4 Zoology

Perhaps the simplest way to predict using a number of experts is to pick one of them and mirror her predictions exactly. Beyond this “fixed expert model”, we have considered two methods of combining experts so far, namely taking Bayesian mixtures, and taking elementwise mixtures as described in §3.3. Figure 1 shows these and a number of other, more sophisticated methods that fit in our framework. The arrows indicate which methods are generalised by which other methods. They have been partitioned in groups that can be computed in the same amount of time using HMMs.

We have presented two examples so far, the Bayesian mixture and the elementwise mixture with fixed coefficients (Examples 3.3.1 and 3.3.2). The latter model is parameterised. Choosing a fixed value for the parameter beforehand is often difficult. The first model we discuss learns the optimal parameter value on-line, at the cost of only a small additional loss. We then proceed to discuss a number of important existing expert models.

### 4.1 Universal Elementwise Mixtures

A distribution is “universal” for a family of distributions if it incurs small additional loss compared to the best member of the family. A standard Bayesian mixture constitutes the simplest example. It is universal for the fixed expert model, where the unknown parameter is the used expert. For the uniform prior, the additional loss (7) is at most  $\log|\Xi|$ .

In Example 3.3.2, we described elementwise mixtures with fixed coefficients as ES-priors. Prior knowledge about the mixture coefficients is often unavailable. We now expand this model to learn the optimal mixture coefficients from the data, resulting in a distribution that is universal for the fixed elementwise mixtures. To this end we place a prior distribution  $w$  on the space of mixture weights  $\Delta(\Xi)$ . Using (5) we obtain the following marginal distribution:

$$\begin{aligned} P_{\text{umix}}(x^n) &= \int_{\Delta(\Xi)} P_{\text{mix},\alpha}(x^n) w(\alpha) d\alpha \\ &= \int_{\Delta(\Xi)} \sum_{\xi^n} P_{\xi^n}(x^n) \pi_{\text{mix},\alpha}(\xi^n) w(\alpha) d\alpha \\ &= \sum_{\xi^n} P_{\xi^n}(x^n) \pi_{\text{umix}}(\xi^n), \quad \text{where} \\ \pi_{\text{umix}}(\xi^n) &= \int_{\Delta(\Xi)} \pi_{\text{mix},\alpha}(\xi^n) w(\alpha) d\alpha. \end{aligned} \tag{9}$$

Thus  $P_{\text{umix}}$  is the ES-joint with ES-prior  $\pi_{\text{umix}}$ . This applies more generally: parameters  $\alpha$  can be integrated out of an ES-prior regardless of which experts are used, since the expert predictions  $P_{\xi^n}(x^n)$  do not depend on  $\alpha$ .

We will proceed to calculate a loss bound for the universal elementwise mixture model, showing that it really is universal. After that we will describe how it can be implemented as an HMM.

#### 4.1.1 A Loss Bound

In this section we relate the loss of a universal elementwise mixture with the loss obtained by the maximum likelihood elementwise mixture. While mixture models occur regularly

in the statistical literature, we are not aware of any appearance in universal prediction. Therefore, to the best of our knowledge, the following simple loss bound is new. Our goal is to obtain a bound in terms of properties of the prior. A difficulty here is that there are many expert sequences exhibiting mixture frequencies close to the maximum likelihood mixture weights, so that each individual expert sequence contributes relatively little to the total probability (9). The following theorem is a general tool to deal with such situations.

**Theorem 3.** *Let  $\pi, \rho$  be ES-priors s.t.  $\rho$  is zero whenever  $\pi$  is. Then for all  $x^n$ , reading  $0/0 = 0$ ,*

$$\frac{P_\rho(x^n)}{P_\pi(x^n)} \leq \max_{\xi^n} \frac{\rho(\xi^n)}{\pi(\xi^n)}.$$

*Proof.* Clearly  $P_\rho$  is zero whenever  $P_\pi$  is. Thus

$$\begin{aligned} \frac{P_\rho(x^n)}{P_\pi(x^n)} &= \frac{\sum_{\xi^n} P_\rho(x^n, \xi^n)}{\sum_{\xi^n} P_\pi(x^n, \xi^n)} \leq \max_{\xi^n} \frac{P_\rho(x^n, \xi^n)}{P_\pi(x^n, \xi^n)} \\ &= \max_{\xi^n} \frac{P_{\xi^n}(x^n) \rho(\xi^n)}{P_{\xi^n}(x^n) \pi(\xi^n)} = \max_{\xi^n} \frac{\rho(\xi^n)}{\pi(\xi^n)}. \quad \square \end{aligned}$$

Using this theorem, we obtain a loss bound for universal elementwise mixtures that can be computed prior to observation and without reference to the experts' PFSs.

**Corollary 4.** *Let  $P_{\text{umix}}$  be the universal elementwise mixture model defined using the  $(\frac{1}{2}, \dots, \frac{1}{2})$ -Dirichlet prior (that is, Jeffreys' prior) as the prior  $w(\alpha)$  in (9). Let  $\hat{\alpha}(x^n)$  maximise the likelihood  $P_{\text{mix}, \alpha}(x^n)$  w.r.t.  $\alpha$ . Then for all  $x^n$  the additional loss incurred by the universal elementwise mixture is bounded thus*

$$-\log P_{\text{umix}}(x^n) + \log P_{\text{mix}, \hat{\alpha}(x^n)}(x^n) \leq \frac{|\Xi| - 1}{2} \log \frac{n}{\pi} + c$$

for a fixed constant  $c$ .

*Proof.* By Theorem 3

$$-\log P_{\text{umix}}(x^n) + \log P_{\text{mix}, \hat{\alpha}(x^n)}(x^n) \leq \max_{\xi^n} \left( -\log \pi_{\text{umix}}(\xi^n) + \log \pi_{\text{mix}, \hat{\alpha}(x^n)}(\xi^n) \right). \quad (10)$$

We now bound the right hand side. Let  $\hat{\alpha}(\xi^n)$  maximise  $\pi_{\text{mix}, \alpha}(\xi^n)$  w.r.t.  $\alpha$ . Then for all  $x^n$  and  $\xi^n$

$$\pi_{\text{mix}, \hat{\alpha}(x^n)}(\xi^n) \leq \pi_{\text{mix}, \hat{\alpha}(\xi^n)}(\xi^n). \quad (11)$$

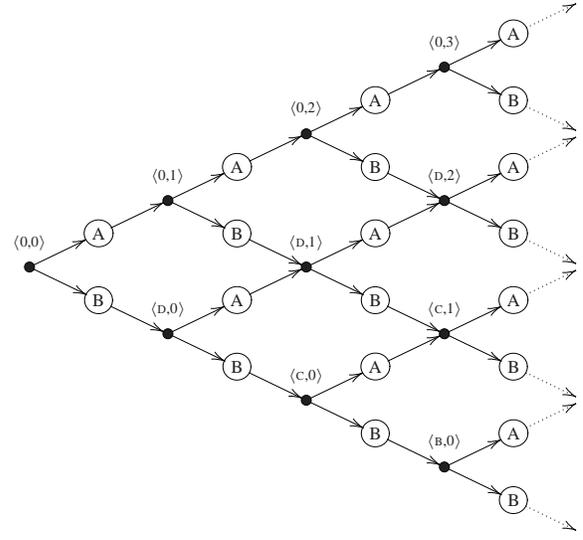
For the  $(\frac{1}{2}, \dots, \frac{1}{2})$ -Dirichlet prior, for all  $\xi^n$

$$-\log \pi_{\text{umix}}(\xi^n) + \log \pi_{\text{mix}, \hat{\alpha}(\xi^n)}(\xi^n) \leq \frac{|\Xi| - 1}{2} \log \frac{n}{\pi} + c$$

for some fixed constant  $c$  (see e.g. [13]) Combination with (11) and (10) completes the proof.  $\square$

Since the overhead incurred as a penalty for not knowing the optimal parameter  $\hat{\alpha}(x^n)$  in advance is only logarithmic in the sample size  $n$ , we find that  $P_{\text{umix}}$  is universal in a strong sense for the fixed elementwise mixtures.

**Figure 5** Universal elementwise mixture (two experts only)



#### 4.1.2 HMM

While universal elementwise mixtures can be described using the ES-prior  $\pi_{\text{umix}}$  defined in (9), unfortunately any HMM that computes it needs a state for each possible count vector, and is therefore huge if the number of experts is large. The HMM  $\mathbb{A}_{\text{umix}}$  for an arbitrary number of experts using the  $(\frac{1}{2}, \dots, \frac{1}{2})$ -Dirichlet prior is given using  $Q = Q_s \cup Q_p$  by

$$\begin{aligned} Q_s &= \mathbb{N}^\Xi \quad Q_p = \mathbb{N}^\Xi \times \Xi \quad P_o(\mathbf{0}) = 1 \quad \Lambda(\vec{n}, \xi) = \xi \\ P \left( \begin{array}{l} \langle \vec{n} \rangle \rightarrow \langle \vec{n}, \xi \rangle \\ \langle \vec{n}, \xi \rangle \rightarrow \langle \vec{n} + \mathbf{1}_\xi \rangle \end{array} \right) &= \left( \begin{array}{c} \frac{1/2 + n_\xi}{|\Xi|/2 + \sum_{\xi} n_\xi} \\ 1 \end{array} \right) \quad (12) \end{aligned}$$

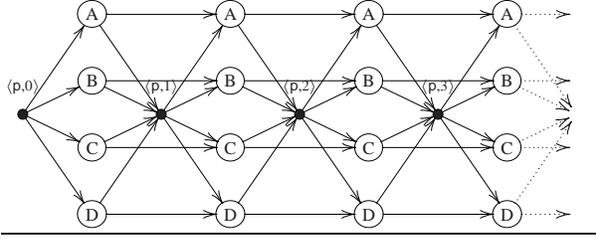
We write  $\mathbb{N}^\Xi$  for the set of assignments of counts to experts;  $\mathbf{0}$  for the all zero assignment, and  $\mathbf{1}_\xi$  marks one count for expert  $\xi$ . We show the diagram of  $\mathbb{A}_{\text{umix}}$  for the practical limit of two experts in Figure 5. In this case, the forward algorithm has running time  $O(n^2)$ . Each productive state in Figure 5 corresponds to a vector of two counts  $(n_1, n_2)$  that sum to the sample size  $n$ , with the interpretation that of the  $n$  experts, the first was used  $n_1$  times while the second was used  $n_2$  times. These counts are a sufficient statistic for the multinomial model class: per (5b) and (9) the probability of the next expert only depends on the counts, and these probabilities are exactly the successor probabilities of the silent states (12).

Other priors on  $\alpha$  are possible. In particular, when all mass is placed on a single value of  $\alpha$ , we retrieve the elementwise mixture with fixed coefficients.

#### 4.2 Fixed Share

The first publication that considers a scenario where the best predicting expert may change with the sample size is Herbster and Warmuth's paper on *tracking the best expert* [4, 5]. They partition the data of size  $n$  into  $m$  segments, where each segment is associated with an expert, and give algorithms to

**Figure 6** Fixed share



predict almost as well as the best *partition* where the best expert is selected per segment. They give two algorithms called fixed share and dynamic share. The second algorithm does not fit in our framework; furthermore its motivation applies only to loss functions other than log-loss. We focus on fixed share, which is in fact identical to our algorithm applied to the HMM depicted in Figure 6, where all arcs *into* the silent states have fixed probability  $\alpha \in [0, 1]$  and all arcs *from* the silent states have some fixed distribution  $w$  on  $\Xi$ .<sup>2</sup> The same algorithm is also described as an instance of the Aggregating Algorithm in [12]. Fixed share reduces to fixed elementwise mixtures by setting  $\alpha = 1$  and to Bayesian mixtures by setting  $\alpha = 0$ . Formally, using  $Q = Q_s \cup Q_p$ :

$$\begin{aligned} Q_s &= \{\mathbf{p}\} \times \mathbb{N} & P_0(\mathbf{p}, 0) &= 1 \\ Q_p &= \Xi \times \mathbb{Z}_+ & \Lambda(\xi, n) &= \xi \end{aligned} \quad (13a)$$

$$P \begin{pmatrix} \langle \mathbf{p}, n \rangle \rightarrow \langle \xi, n+1 \rangle \\ \langle \xi, n \rangle \rightarrow \langle \mathbf{p}, n \rangle \\ \langle \xi, n \rangle \rightarrow \langle \xi, n+1 \rangle \end{pmatrix} = \begin{pmatrix} w(\xi) \\ \alpha \\ 1 - \alpha \end{pmatrix} \quad (13b)$$

Each productive state represents that a particular expert is used at a certain sample size. Once a transition to a silent state is made, all history is forgotten and a new expert is chosen according to  $w$ .<sup>3</sup>

Let  $\hat{L}$  denote the loss achieved by the best partition, with switching rate  $\alpha^* := m/(n-1)$ . Let  $L_{fs,\alpha}$  denote the loss of fixed share with uniform  $w$  and parameter  $\alpha$ . Herbster and Warmuth prove<sup>4</sup>

$$L_{fs,\alpha} - \hat{L} \leq (n-1)H(\alpha^*, \alpha) + (m-1)\log(|\Xi|-1) + \log|\Xi|,$$

which we for brevity loosen slightly to

$$L_{fs,\alpha} - \hat{L} \leq nH(\alpha^*, \alpha) + m \log|\Xi|. \quad (14)$$

Here  $H(\alpha^*, \alpha) = -\alpha^* \log \alpha - (1 - \alpha^*) \log(1 - \alpha)$  is the cross entropy. The best loss guarantee is obtained for  $\alpha = \alpha^*$ , in which case the cross entropy reduces to the binary entropy  $H(\alpha)$ . A drawback of the method is that the optimal

<sup>2</sup>This is actually a slight generalisation: the original algorithm uses a uniform  $w(\xi) = 1/|\Xi|$ .

<sup>3</sup>Contrary to the original fixed share, we allow switching to the same expert. In the HMM framework this is necessary to achieve running-time  $O(n|\Xi|)$ . Under uniform  $w$ , non-reflexive switching with fixed rate  $\alpha$  can be simulated by reflexive switching with fixed rate  $\beta = \frac{\alpha|\Xi|}{|\Xi|-1}$  (provided  $\beta \leq 1$ ). For non-uniform  $w$ , the rate becomes expert-dependent.

<sup>4</sup>This bound can be obtained for the fixed share HMM using the previous footnote.

value of  $\alpha$  has to be known in advance in order to minimise the loss. In Sections §4.3 and §5 we describe a number of generalisations of fixed share that avoid this problem.

### 4.3 Universal Share

Volf and Willems describe universal share (they call it *the switching method*) [11], which is very similar to a probabilistic version of Herbster and Warmuth's fixed share algorithm, except that they put a prior on the unknown parameter, so that their algorithm adaptively learns the optimal value during prediction. In formula:

$$P_{us}(x^n) = \int P_{fs,\alpha}(x^n) w(\alpha) d\alpha.$$

In [1], Bousquet shows that the overhead for not knowing the optimal parameter value is equal to the overhead of estimating a Bernoulli parameter: let  $L_{fs,\alpha}$  be as before, and let  $L_{us} = -\log P_{us}(x^n)$  denote the loss of universal share with Jeffreys' prior  $w(\alpha) = \alpha^{-1/2}(1-\alpha)^{-1/2}/\pi$ . Then

$$L_{us} - \min_{\alpha} L_{fs,\alpha} \leq 1 + \frac{1}{2} \log n. \quad (15)$$

Thus  $P_{us}$  is universal for the model class  $\{P_{fs,\alpha} \mid \alpha \in [0, 1]\}$  that consists of all ES-joints where the ES-priors are distributions with a fixed switching rate.

Universal share requires quadratic running time  $O(n^2|\Xi|)$ , restricting its use to moderately small data sets. In [8], Monteleoni and Jaakkola place a discrete prior on the parameter that divides its mass over  $\sqrt{n}$  well-chosen points, in a setting where the ultimate sample size  $n$  is known beforehand. This way they still manage to achieve (15) up to a constant, while reducing the running time to  $O(n\sqrt{n}|\Xi|)$ .

The HMM for universal share with the  $(\frac{1}{2}, \frac{1}{2})$ -Dirichlet prior on the switching rate  $\alpha$  is displayed in Figure 7. It is formally specified (using  $Q = Q_s \cup Q_p$ ) by:

$$\begin{aligned} Q_s &= \{\mathbf{p}, \mathbf{q}\} \times \{ \langle m, n \rangle \in \mathbb{N}^2 \mid m \leq n \} & \Lambda(\xi, m, n) &= \xi \\ Q_p &= \Xi \times \{ \langle m, n \rangle \in \mathbb{N}^2 \mid m < n \} & P_0(\mathbf{p}, 0, 0) &= 1 \end{aligned}$$

$$P \begin{pmatrix} \langle \mathbf{p}, m, n \rangle \rightarrow \langle \xi, m, n+1 \rangle \\ \langle \mathbf{q}, m, n \rangle \rightarrow \langle \mathbf{p}, m+1, n \rangle \\ \langle \xi, m, n \rangle \rightarrow \langle \mathbf{q}, m, n \rangle \\ \langle \xi, m, n \rangle \rightarrow \langle \xi, m, n+1 \rangle \end{pmatrix} = \begin{pmatrix} w(\xi) \\ 1 \\ (m + \frac{1}{2})/n \\ (n - m - \frac{1}{2})/n \end{pmatrix} \quad (16)$$

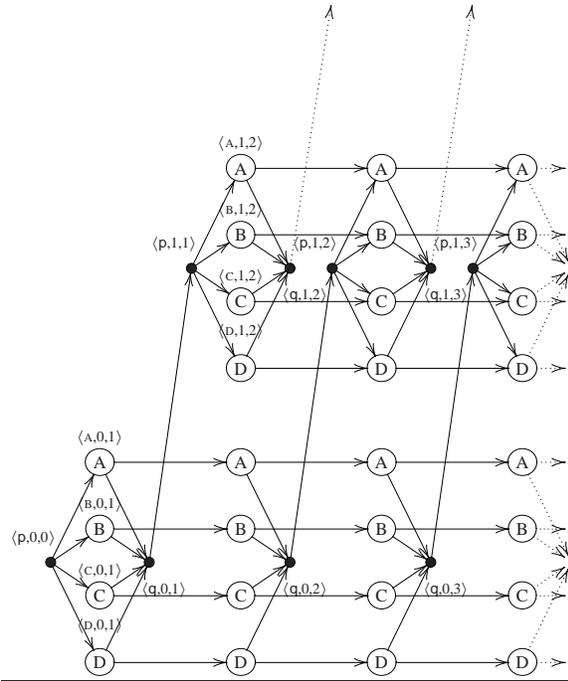
Each productive state  $\langle \xi, n, m \rangle$  represents the fact that at sample size  $n$  expert  $\xi$  is used, while there have been  $m$  switches in the past. Note that the last two lines of (16) are subtly different from the corresponding topmost line of (12). In a sample of size  $n$  there are  $n$  possible positions to use a given expert, while there are only  $n-1$  possible switch positions.

The presence of the switch count in the state is the new ingredient compared to fixed share. It allows us to adapt the switching probability to the data, but it also renders the number of states quadratic. We discuss reducing the number of states without sacrificing much performance in [6].

## 5 New Models to Switch between Experts

So far we have considered two models for switching between experts: fixed share and its generalisation, universal share.

**Figure 7** Universal share



While fixed share is an extremely efficient algorithm, it requires that the frequency of switching between experts is estimated a priori, which can be hard in practice. Moreover, we may have prior knowledge about how the switching probability will change over time, but unless we know the ultimate sample size in advance, we may be forced to accept a linear overhead compared to the best parameter value. Universal share overcomes this problem by marginalising over the unknown parameter, but has quadratic running time.

The first model considered in this section, the switch distribution, avoids both problems. It is parameterless and has essentially the same running time as fixed share. It also achieves a loss bound competitive to that of universal share. Moreover, for a bounded number of switches the bound has even better asymptotics.

The second model is called the run-length model because it uses a run-length code (c.f. [7]) as an ES-prior. This may be useful because, while both fixed and universal share model the distance between switches with a geometric distribution, the real distribution on these distances may be different. This is the case if, for example, the switches are highly clustered. This additional expressive power comes at the cost of quadratic running time, but we discuss a special case where this may be reduced to linear.

We conclude this section with a comparison of the two expert switching models.

### 5.1 Switch Distribution

The switch distribution is a recent model for combining expert predictions. Like fixed share, it is intended for settings where the best predicting expert is expected to change as a function of the sample size, but it has two major innovations.

First, we let the probability of switching to a different expert decrease with the sample size. This allows us to derive a loss bound close to that of the fixed share algorithm, without the need to tune any parameters.<sup>5</sup> Second, the switch distribution has a special provision to ensure that in the case where the number of switches remains bounded, the incurred loss overhead is  $O(1)$ .

The switch distribution was introduced in [10], which addresses a long standing open problem in statistical model class selection known as the “AIC vs BIC dilemma”. Here we disregard such applications and treat the switch distribution like the other models for combining expert predictions. In §5.1.1, we describe an HMM that corresponds to the switch distribution; this illuminates the relationship between the switch distribution and the fixed share algorithm which it in fact generalises. We provide a loss bound for the switch distribution in §5.1.2.

#### 5.1.1 Switch HMM

Let  $\sigma^\omega$  and  $\tau^\omega$  be sequences of distributions on  $\{0, 1\}$  which we call the *switch* and *stabilisation probabilities*. The switch HMM  $\mathbb{A}_{\text{sw}}$ , displayed in Figure 8, is defined below using  $Q = Q_s \cup Q_p$ :

$$Q_s = \{\mathbf{p}, \mathbf{p}_s, \mathbf{p}_u\} \times \mathbb{N} \quad P_0(\mathbf{p}, 0) = 1 \quad \Lambda(\mathbf{s}, \xi, n) = \xi$$

$$Q_p = \{\mathbf{s}, \mathbf{u}\} \times \Xi \times \mathbb{Z}_+ \quad \Lambda(\mathbf{u}, \xi, n) = \xi$$

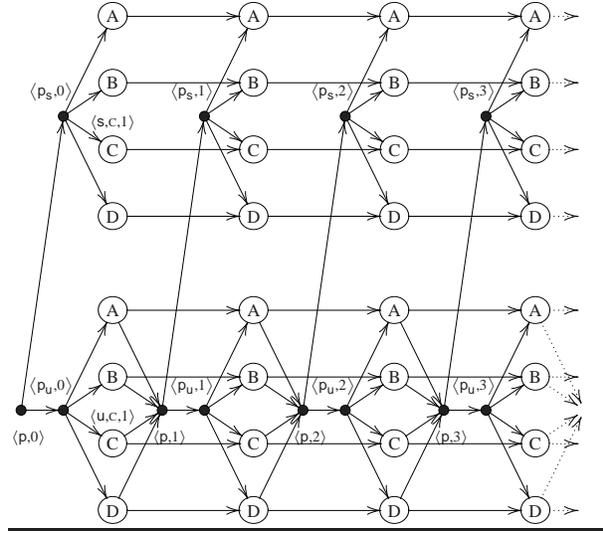
$$P \begin{pmatrix} \langle \mathbf{p}, n \rangle \rightarrow \langle \mathbf{p}_u, n \rangle \\ \langle \mathbf{p}, n \rangle \rightarrow \langle \mathbf{p}_s, n \rangle \\ \langle \mathbf{p}_u, n \rangle \rightarrow \langle \mathbf{u}, \xi, n+1 \rangle \\ \langle \mathbf{p}_s, n \rangle \rightarrow \langle \mathbf{s}, \xi, n+1 \rangle \\ \langle \mathbf{s}, \xi, n \rangle \rightarrow \langle \mathbf{s}, \xi, n+1 \rangle \\ \langle \mathbf{u}, \xi, n \rangle \rightarrow \langle \mathbf{u}, \xi, n+1 \rangle \\ \langle \mathbf{u}, \xi, n \rangle \rightarrow \langle \mathbf{p}, n \rangle \end{pmatrix} = \begin{pmatrix} \tau_n(0) \\ \tau_n(1) \\ w(\xi) \\ w(\xi) \\ 1 \\ \sigma_n(0) \\ \sigma_n(1) \end{pmatrix}$$

This HMM contains two “expert bands”. Consider a productive state  $\langle \mathbf{u}, \xi, n \rangle$  in the bottom band, which we call the *unstable* band, from a generative viewpoint. Two things can happen. With probability  $\sigma_n(0)$  the process continues horizontally to  $\langle \mathbf{u}, \xi, n+1 \rangle$  and the story repeats. We say that *no switch occurs*. With probability  $\sigma_n(1)$  the process continues to the silent state  $\langle \mathbf{p}, n \rangle$  directly to the right. We say that *a switch occurs*. Then a new choice has to be made. With probability  $\tau_n(0)$  the process continues rightward to  $\langle \mathbf{p}_u, n \rangle$  and then branches out to some productive state  $\langle \mathbf{u}, \xi', n+1 \rangle$  (possibly  $\xi = \xi'$ ), and the story repeats. With probability  $\tau_n(1)$  the process continues to  $\langle \mathbf{p}_s, n \rangle$  in the top band, called the *stable* band. Also here it branches out to some productive state  $\langle \mathbf{s}, \xi', n+1 \rangle$ . But from this point onward there are no choices anymore; expert  $\xi'$  is produced forever. We say that the process has *stabilised*.

By choosing  $\tau_n(1) = 0$  and  $\sigma_n(1) = \theta$  for all  $n$  we essentially remove the stable band and arrive at fixed share with parameter  $\theta$ . The presence of the stable band enables us to improve the loss bound of fixed share in the particular

<sup>5</sup>The idea of decreasing the switch probability as  $1/(n+1)$ , which has not previously been published, was independently conceived by Mark Herbster and the authors.

**Figure 8** The switch distribution



case that the number of switches is bounded; in that case, the stable band allows us to remove the dependency of the loss bound on  $n$  altogether. We will use the particular choice  $\tau_n(0) = 1/2$  for all  $n$ , and  $\sigma_n(1) = \pi_\tau(\mathbf{Z} = n | \mathbf{Z} \geq n)$  an arbitrary distribution  $\pi_\tau$  on  $\mathbb{N}$ . This allows us to relate the switch HMM to the parametric representation that we present next.

### 5.1.2 A Loss Bound

We derive a loss bound of the same type as the bound for the fixed share algorithm (see §4.2). We need the following lemma, that is proven in [6].

**Lemma 5.** Fix an expert sequence  $\xi^n$ . Let  $m$  denote the number of blocks in  $\xi^n$ , where the blocks are the maximal subsequences containing only a single expert. Let  $1 = t_1 < t_2 < \dots < t_m \leq n$  be the indices where the blocks start. Then

$$\pi_{\text{sw}}(\xi^n) \geq 2^{-m} w(\xi_1) \prod_{i=2}^m w(\xi_{t_i}) \pi_\tau(\mathbf{Z} = t_i | \mathbf{Z} > t_{i-1}).$$

**Theorem 6.** Fix data  $x^n$ . Let  $\xi^n$  maximise the likelihood  $P_{\xi^n}(x^n)$  among all expert sequences with  $m$  blocks. Let  $t_m$  be the index of the first element of the last block in  $\xi^n$ . Let  $\pi_\tau(n) = 1/(n(n-1))$  and  $w$  be uniform. Then the loss overhead  $-\log P_{\text{sw}}(x^n) + \log P_{\xi^n}(x^n)$  of the switch distribution is bounded by

$$m + m \log |\Xi| + \log \binom{t_m}{m} + \log(m!).$$

*Proof.* We have

$$\begin{aligned} & -\log P_{\text{sw}}(x^n) + \log P_{\xi^n}(x^n) \leq -\log \pi_{\text{sw}}(\xi^n) \\ & \leq -\log \left( 2^{-m} w(\xi_1) \prod_{i=2}^m \pi_\tau(t_i | t_i > t_{i-1}) w(\xi_{t_i}) \right) \\ & = m + m \log |\Xi| - \sum_{i=2}^m \log \pi_\tau(t_i | t_i > t_{i-1}). \end{aligned} \quad (17)$$

The prior  $\pi_\tau$  may be written  $\pi_\tau(n) = \frac{1}{n-1} - \frac{1}{n}$ , so that

$$\pi_\tau(t_i | t_i > t_{i-1}) = \frac{1/(t_i(t_i-1))}{\sum_{n>t_{i-1}} (\frac{1}{n-1} - \frac{1}{n})} = \frac{t_{i-1}}{t_i(t_i-1)}.$$

If we substitute this in the last term of (17), the sum telescopes and we are left with

$$\underbrace{-\log(t_1)}_{=0} + \log(t_m) + \sum_{i=2}^m \log(t_i - 1). \quad (18)$$

If we fix  $t_m$ , this expression is maximised if  $t_2, \dots, t_{m-1}$  take on the values  $t_m - m + 2, \dots, t_m - 1$ , so that (18) becomes

$$\sum_{i=t_m-m+1}^{t_m} \log i = \log \left( \frac{t_m!}{(t_m - m)!} \right) = \log \binom{t_m}{m} + \log(m!).$$

The theorem follows using this upper bound.  $\square$

Note that this loss bound is a function of the index of the last switch  $t_m$  rather than of the sample size  $n$ ; this means that in the important scenario where the number of switches remains bounded in  $n$ , the loss compared to the best partition is  $O(1)$ .

The bound compares quite favourably with the loss bound for the fixed share algorithm (see §4.2). We can tighten our bound slightly by using the fact that we allow switches to the same expert, as also remarked in Footnote 3 on page 8. For brevity we do not pursue this here, but the difference is exactly that between (14) and the original bound for the fixed share algorithm.

We now investigate how much worse the above guarantees are compared to (14). The overhead of fixed share is bounded from above by  $nH(\alpha) + m \log(|\Xi|)$ . We first underestimate this worst-case loss by substituting the optimal value  $\alpha = m/n$ , and rewrite

$$nH(\alpha) \geq nH(m/n) \geq \log \binom{n}{m}.$$

Second we overestimate the loss of the switch distribution by substituting the worst case  $t_m = n$ . We then find the maximal difference between the two bounds to be

$$\begin{aligned} & \left( m + m \log |\Xi| + \log \binom{n}{m} + \log(m!) \right) - \\ & \left( \log \binom{n}{m} + m \log |\Xi| \right) \\ & = m + \log(m!) \leq m + m \log m. \end{aligned} \quad (19)$$

Thus using the switch distribution instead of fixed share lowers the guarantee by at most  $m + m \log m$  bits, which is significant only if the number of switches is relatively large. On the flip side, using the switch distribution does not require any prior knowledge about the data (i.e. the maximum likelihood switching rate). This is a big advantage in a setting where we desire to maintain the bound sequentially. This is impossible with the fixed share algorithm in case the optimal value of  $\alpha$  varies with  $n$ .

## 5.2 Run-length Model

Run-length codes have been used extensively in the context of data compression, see e.g. [7]. Rather than applying run length codes directly to the observations, we reinterpret the corresponding probability distributions as ES-priors, because they may constitute good models for the distances between consecutive switches.

The run length model is especially useful if the switches are clustered, in the sense that some blocks in the expert sequence contain relatively few switches, while other blocks contain many. The fixed share algorithm remains oblivious to such properties, as its predictions of the expert sequence are based on a Bernoulli model: the probability of switching remains the same, regardless of the index of the previous switch. Essentially the same limitation also applies to the universal share algorithm, whose switching probability normally converges as the sample size increases. The switch distribution is efficient when the switches are clustered toward the beginning of the sample: its switching probability decreases in the sample size. However, this may be unrealistic and may introduce a new unnecessary loss overhead.

The run-length model is based on the assumption that the *intervals* between successive switches are independently distributed according to some distribution  $\pi_\tau$ . After the universal share model and the switch distribution, this is a third generalisation of the fixed share algorithm, which is recovered by taking a geometric distribution for  $\pi_\tau$ . As may be deduced from the defining HMM, which is given below, we require quadratic running time  $O(n^2|\Xi|)$  to evaluate the run-length model in general.

### 5.2.1 Run-length HMM

Let  $\mathbb{S} := \{(m, n) \in \mathbb{N}^2 \mid m < n\}$ , and let  $\pi_\tau$  be a distribution on  $\mathbb{Z}_+$ . The specification of the run-length HMM is given using  $Q = Q_s \cup Q_p$  by:

$$\begin{aligned} Q_s &= \{\mathbf{q}\} \times \mathbb{S} \cup \{\mathbf{p}\} \times \mathbb{N} & \Lambda(\xi, m, n) &= \xi \\ Q_p &= \Xi \times \mathbb{S} & P_o(\mathbf{p}, 0) &= 1 \\ P \left( \begin{array}{l} \langle \mathbf{p}, n \rangle \rightarrow \langle \xi, n, n+1 \rangle \\ \langle \xi, m, n \rangle \rightarrow \langle \xi, m, n+1 \rangle \\ \langle \xi, m, n \rangle \rightarrow \langle \mathbf{q}, m, n \rangle \\ \langle \mathbf{q}, m, n \rangle \rightarrow \langle \mathbf{p}, n \rangle \end{array} \right) &= \left( \begin{array}{l} w(\xi) \\ \pi_\tau(\mathbf{Z} > n \mid \mathbf{Z} \geq n) \\ \pi_\tau(\mathbf{Z} = n \mid \mathbf{Z} \geq n) \\ 1 \end{array} \right) \end{aligned}$$

### 5.2.2 A Loss Bound

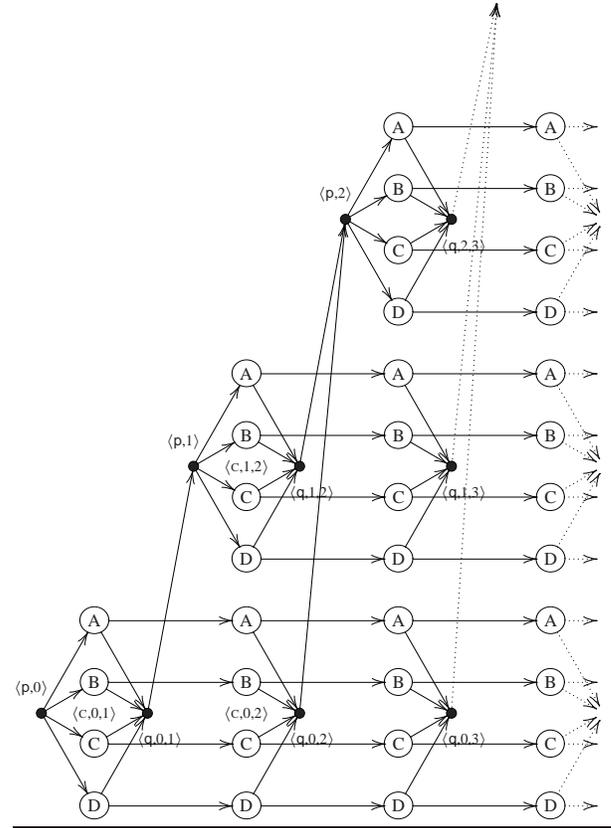
Fix an expert sequence  $\xi^n$  with  $m$  blocks. For  $i = 1, \dots, m$ , let  $\delta_i$  and  $k_i$  denote the length and expert of block  $i$ . From the definition of the HMM above, we obtain that  $\pi_{\mathfrak{H}}(\xi^n)$  equals

$$\sum_{i=1}^m -\log w(k_i) + \sum_{i=1}^{m-1} -\log \pi_\tau(\mathbf{Z} = \delta_i) - \log \pi_\tau(\mathbf{Z} \geq \delta_m).$$

**Theorem 7.** Fix data  $x^n$ . Let  $\xi^n$  maximise the likelihood  $P_{\xi^n}(x^n)$  among all expert sequences with  $m$  blocks. Let  $w$  be the uniform distribution on experts, and let  $\pi_\tau$  be log-convex. Then the loss overhead is bounded thus

$$-\log P_{\mathfrak{H}}(x^n) + \log P_{\xi^n}(x^n) \leq m \left( \log |\Xi| - \log \pi_\tau \left( \frac{n}{m} \right) \right).$$

Figure 9 The run-length model



*Proof.* Let  $\delta_i$  denote the length of block  $i$ . We overestimate

$$\begin{aligned} -\log P_{\mathfrak{H}}(x^n) + \log P_{\xi^n}(x^n) &\leq -\log \pi_{\mathfrak{H}}(\xi^n) \\ &= m \log |\Xi| + \sum_{i=1}^{m-1} -\log \pi_\tau(\mathbf{Z} = \delta_i) - \log \pi_\tau(\mathbf{Z} \geq \delta_m) \\ &\leq m \log |\Xi| + \sum_{i=1}^m -\log \pi_\tau(\delta_i). \end{aligned} \quad (20)$$

Since  $-\log \pi_\tau$  is concave, by Jensen's inequality we have

$$\sum_{i=1}^m \frac{-\log \pi_\tau(\delta_i)}{m} \leq -\log \pi_\tau \left( \frac{\sum_{i=1}^m \delta_i}{m} \right) = -\log \pi_\tau \left( \frac{n}{m} \right).$$

In other words, the block lengths  $\delta_i$  are all equal in the worst case. Plugging this into (20) we obtain the theorem.  $\square$

### 5.2.3 Finite Support

We have seen that the run-length model reduces to fixed share if the prior on switch distances  $\pi_\tau$  is geometric, so that it can be evaluated in linear time in that case. We also obtain a linear time algorithm when  $\pi_\tau$  has finite support, because then only a constant number of states can receive positive weight at any sample size. For this reason it can be advantageous to choose a  $\pi_\tau$  with finite support, even if one expects that arbitrarily long distances between consecutive switches may

occur. Expert sequences with such longer distances between switches can still be represented with a truncated  $\pi_\tau$  using a sequence of switches from and to the same expert. This way, long runs of the same expert receive exponentially small, but positive, probability.

### 5.3 Comparison

We have discussed two models for switching: the recent switch distribution and the new run-length model. It is natural to wonder which model to apply. One possibility is to compare asymptotic loss bounds. To compare the bounds given by Theorems 6 and 7, we substitute  $t_m + 1 = n$  in the bound for the switch distribution, and use a prior  $\pi_\tau$  for the run-length model that satisfies  $-\log \pi_\tau(n) \leq \log n + 2 \log \log(n + 1) + 3$  (for instance an Elias code [3]). The next step is to determine which bound is better depending on how fast  $m$  grows as a function of  $n$ . It only makes sense to consider  $m$  non-decreasing in  $n$ .

**Theorem 8.** *The loss bound of the switch distribution (with  $t_n = n$ ) is asymptotically lower than that of the run-length model (with  $\pi_\tau$  as above) if  $m = o((\log n)^2)$ , and asymptotically higher if  $m = \Omega((\log n)^2)$ .<sup>6</sup>*

*Proof sketch.* After eliminating terms common to both loss bounds, it remains to compare

$$m + m \log m \quad \text{to} \quad 2m \log \log \left( \frac{n}{m} + 1 \right) + 3.$$

If  $m$  is bounded, the left hand side is clearly lower for sufficiently large  $n$ . Otherwise we may divide by  $m$ , exponentiate, simplify, and compare

$$m \quad \text{to} \quad (\log n - \log m)^2,$$

from which the theorem follows directly.  $\square$

For finite samples, the switch distribution can be used in case the switches are expected to occur early on average, or if the running time is paramount. Otherwise the run-length model is preferable.

## 6 Conclusion

In prediction with expert advice, the goal is to formulate prediction strategies that perform as well as the best possible expert (combination). Expert predictions can be combined by taking a weighted mixture at every sample size. The best weights generally evolve over time. In this paper we introduced expert sequence priors (ES-priors), which are probability distributions over infinite sequences of experts, to model the trajectory followed by the optimal mixture weights. Prediction with expert advice then amounts to marginalising the joint distribution constructed from the chosen ES-prior and the experts' predictions.

We employed hidden Markov models (HMMs) to specify ES-priors. HMMs' explicit notion of current state and state-to-state evolution naturally fit the temporal correlations we seek to model. For reasons of efficiency we use HMMs with

silent states. The standard algorithms for HMMs (Forward, Backward, Viterbi and Baum-Welch) can be used to answer questions about the ES-prior as well as the induced distribution on data. The running time of the forward algorithm can be read off directly from the graphical representation of the HMM.

Our approach allows unification of many existing expert models, including mixture models and fixed share. We gave their defining HMMs and recovered the best known running times. We also introduced two new parameterless generalisations of fixed share. The first, called the switch distribution, was recently introduced to improve model selection performance. We rendered it as a small HMM, which shows how it can be evaluated in linear time. The second, called the run-length model, uses a run-length code in a novel way, namely as an ES-prior. This model has quadratic running time. We compared the loss bounds of the two models asymptotically, and showed that the run-length model is preferred if the number of switches grows like  $(\log n)^2$  or faster, while the switch distribution is preferred if it grows slower. We provided graphical representations and loss bounds for all considered models.

## Acknowledgements

Peter Grünwald's and Tim van Erven's suggestions significantly improved the quality of this paper. Thanks also go to Mark Herbster for a fruitful and enjoyable afternoon exchanging ideas, which has certainly influenced the shape of this paper.

## References

- [1] O. Bousquet. A note on parameter tuning for on-line shifting algorithms. Technical report, Max Planck Institute for Biological Cybernetics, 2003.
- [2] A. P. Dawid. Statistical theory: The prequential approach. *Journal of the Royal Statistical Society, Series A*, 147, Part 2:278–292, 1984.
- [3] P. Elias. Universal codeword sets and representations of the integers. *IEEE Transactions on Information Theory*, 21(2):194–203, 1975.
- [4] M. Herbster and M. K. Warmuth. Tracking the best expert. In *Proceedings of the 12th Annual Conference on Learning Theory (COLT 1995)*, pages 286–294, 1995.
- [5] M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32:151–178, 1998.
- [6] W. M. Koolen and S. de Rooij. Combining expert advice efficiently. arXiv:0802.2015, Feb 2008.
- [7] A. Moffat. *Compression and Coding Algorithms*. Kluwer Academic Publishers, 2002.
- [8] C. Monteleoni and T. Jaakkola. Online learning of non-stationary sequences. *Advances in Neural Information Processing Systems*, 16, 2003.
- [9] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, issue 2, pages 257–285, 1989.
- [10] T. van Erven, P. D. Grünwald, and S. de Rooij. Catching up faster in Bayesian model selection and model averaging. In *To appear in Advances in Neural Information Processing Systems 20 (NIPS 2007)*, 2008.
- [11] P. Volf and F. Willems. Switching between two universal source coding algorithms. In *Proceedings of the Data Compression Conference, Snowbird, Utah*, pages 491–500, 1998.
- [12] V. Vovk. Derandomizing stochastic prediction strategies. *Machine Learning*, 35:247–282, 1999.
- [13] Q. Xie and A. Barron. Asymptotic minimax regret for data compression, gambling and prediction. *IEEE Transactions on Information Theory*, 46(2):431–445, 2000.

<sup>6</sup>Let  $f, g : \mathbb{N} \rightarrow \mathbb{N}$ . We say  $f = o(g)$  if  $\lim_{n \rightarrow \infty} f(n)/g(n) = 0$ . We say  $f = \Omega(g)$  if  $\exists c > 0 \exists n_0 \forall n \geq n_0 : f(n) \geq cg(n)$ .

---

# Improved Guarantees for Learning via Similarity Functions

---

Maria-Florina Balcan     Avrim Blum

Computer Science Department, Carnegie Mellon University  
{ninamf, avrim}@cs.cmu.edu

Nathan Srebro

Toyota Technological Institute at Chicago  
nati@uchicago.edu

## Abstract

We continue the investigation of natural conditions for a similarity function to allow learning, without requiring the similarity function to be a valid kernel, or referring to an implicit high-dimensional space. We provide a new notion of a “good similarity function” that builds upon the previous definition of Balcan and Blum (2006) but improves on it in two important ways. First, as with the previous definition, any large-margin kernel is also a good similarity function in our sense, but the translation now results in a much milder increase in the labeled sample complexity. Second, we prove that for distribution-specific PAC learning, our new notion is strictly more powerful than the traditional notion of a large-margin kernel. In particular, we show that for any hypothesis class  $C$  there exists a similarity function under our definition allowing learning with  $O(\log |C|)$  labeled examples. However, in a lower bound which may be of independent interest, we show that for any class  $C$  of pairwise uncorrelated functions, there is *no* kernel with margin  $\gamma \geq 8/\sqrt{|C|}$  for all  $f \in C$ , even if one allows average hinge-loss as large as 0.5. Thus, the sample complexity for learning such classes with SVMs is  $\Omega(|C|)$ . This extends work of Ben-David et al. (2003) and Forster and Simon (2006) who give hardness results with comparable margin bounds, but at much lower error rates.

Our new notion of similarity relies upon  $L_1$  regularized learning, and our separation result is related to a separation result between what is learnable with  $L_1$  vs.  $L_2$  regularization.

## 1 Introduction

Kernel functions have become an extremely popular tool in machine learning, with an attractive theory as well (Scholkopf & Smola, 2002; Herbrich, 2002; Shawe-Taylor & Cristianini, 2004; Scholkopf et al., 2004). This theory views a kernel as implicitly mapping data points into a possibly very high dimensional space, and describes a kernel function as being good for a given learning problem if data is separable by a

large margin in that implicit space. However, while quite elegant, this theory does not necessarily correspond to the intuition of a good kernel as a good measure of similarity, and the underlying margin in the implicit space usually is not apparent in “natural” representations of the data. Therefore, it may be difficult for a domain expert to use the theory to help design an appropriate kernel for the learning task at hand. Moreover, the requirement of positive semi-definiteness may rule out the most natural pairwise similarity functions for the given problem domain.

In recent work, Balcan and Blum (2006) developed an alternative, more general theory of learning with pairwise similarity functions that may not necessarily be valid positive semi-definite kernels. Specifically, this work developed sufficient conditions for a similarity function to allow one to learn well) that does not require reference to implicit spaces, and does not require the function to be positive semi-definite (or even symmetric). While this theory provably generalizes the standard theory in that any good kernel function in the usual sense can be shown to also be a good similarity function under this definition, the translation does incur a penalty. Subsequently, Srebro (2007) tightly quantified the gap between the learning guarantees based on kernel-based learning, and those that can be obtained by using the kernel as a similarity function in this way. In particular, Srebro (2007) shows that a kernel of margin  $\gamma$  is guaranteed to be a similarity function of margin  $\Omega(\epsilon\gamma^2)$  at hinge-loss  $\epsilon$ , and furthermore there exist examples for which this is tight. To sum up, while the theory of Balcan and Blum (2006) applies to a wider class of pairwise functions than the standard notion of kernel learning, it might be quantitatively inferior in those cases that both notions apply.

In this work we develop a new notion of a good similarity function that broadens the definition of Balcan and Blum (2006) while still guaranteeing learnability. As with the previous definition, our notion talks in terms of natural similarity-based properties and does not require positive semi-definiteness or reference to implicit spaces. However, our new notion improves on the previous definition in two important respects:

First, our new notion provides a better kernel-to-similarity translation. Any large-margin kernel function is a good similarity function under our definition, and while we still incur some loss in the parameters, this loss is much smaller than under the prior definition, especially in terms of the final labeled sample-complexity bounds. In particular, when using

a valid kernel function as a similarity function, a substantial portion of the previous sample-complexity bound can be transferred over to merely a need for *unlabeled* examples.

Second, we show that our new definition allows for good similarity functions to exist for concept classes for which there is *no* good kernel. In particular, for any concept class  $C$  and sufficiently unconcentrated distribution  $D$ , we show there exists a similarity function under our definition with parameters yielding a labeled sample complexity bound of  $O(\frac{1}{\epsilon} \log |C|)$  to achieve error  $\epsilon$ , matching the ideal sample complexity for a generic hypothesis class. In fact, we also extend this result to classes of finite VC-dimension rather than finite cardinality. In contrast, we show there exist classes  $C$  such that under the uniform distribution over the instance space, there is no kernel with margin  $8/\sqrt{|C|}$  for all  $f \in C$  even if one allows 0.5 average hinge-loss. Thus, the margin-based guarantee on sample complexity for learning such classes with kernels is  $\Omega(|C|)$ . This extends work of Ben-David et al. (2003) and Forster and Simon (2006) who give hardness results with comparable margin bounds, but at much lower error rates. Warmuth and Vishwanathan (2005) provide lower bounds for kernels with similar error rates, but their results hold only for regression (not hinge loss). Note that given access to unlabeled data, any similarity function under the definition of Balcan and Blum (2006) can be converted to a kernel function with approximately the same parameters. Thus, our lower bound for kernel functions applies to that definition as well. These results establish a gap in the representational power of similarity functions under our new definition relative to the representational power of either kernels or similarity functions under the old definition.

Both our new definition and that of Balcan and Blum (2006) are based on the idea of a similarity function being good for a learning problem if there exists a non-negligible subset  $R$  of “reasonable points” such that most examples  $x$  are on average more similar to the reasonable points of their own label than to the reasonable points of the other label. (Formally, the “reasonableness” of an example may be given by a weight between 0 and 1 and viewed as probabilistic or fractional.) However, the previous definition combined the two quantities of interest—the probability mass of reasonable points and the gap in average similarity to reasonable points of each label—into a single margin parameter. The new notion keeps these quantities distinct, which turns out to make a substantial difference both in terms of broadness of applicability and in terms of the labeled sample complexity bounds that result.

Note that we distinguish between labeled and unlabeled sample complexities: while the total number of examples needed depends polynomially on the two quantities of interest, the number of labeled examples depends only logarithmically on the probability mass of the reasonable set and therefore may be much smaller under the new definition. This is especially beneficial in situations in which unlabeled data is plentiful (or the distribution is known and so unlabeled data is free), but labeled data is scarce.

Another way to view the distinction between the two notions of similarity is that we now require good predictions using a weight function with bounded expectation, rather than bounded supremum: compare the old Definition 4 and the

variant of the new definition given as Definition 17. (We do in fact still have a bound on the supremum, but this bound only affects the labeled sampled complexity logarithmically.) In Theorem 19 we make the connection between the two versions of the new definition explicit.

Conditioning on a subset of reasonable points, or equivalently bounding the expectation of the weight function, allows us to base our learnability results on  $L_1$ -regularized linear learning. The actual learning rule we get, given in Equation (4.6), is very similar, and even identical, to learning rules suggested by various authors and commonly used in practice as an alternative to Support Vector Machines (Bennett & Campbell, 2000; Roth, 2001; Guigue et al., 2005; Singer, 2000; Tipping, 2001). Here we give a firm theoretical basis to this learning rule, with explicit learning guarantees, and relate it to simple and intuitive properties of the similarity function or kernel used (see the discussion at the end of Section 4).

**Structure of this paper:** After presenting background on the previous definitions and their relation to kernels in Section 2, we present our new notion of a good similarity function in Section 3. In Section 4 we show that our new broader notions still imply learnability. In Section 5 we give our separation results, showing that our new notion is strictly more general than the notion of a large margin kernel. In Section 6 we show that any large margin kernel is also a good similarity function in our sense, and finally in Section 7 we discuss learning with multiple similarity functions.

## 2 Background and Notation

We consider a learning problem specified as follows. We are given access to labeled examples  $(x, \ell)$  drawn from some distribution  $P$  over  $X \times \{-1, 1\}$ , where  $X$  is an abstract instance space. We will sometimes use  $D$  to denote the distribution over  $x$ , and for simplicity, we will assume a deterministic target function, so that  $(x, \ell) = (x, \ell(x))$ . The goal of a learning algorithm is to produce a classification function  $g : X \rightarrow \{-1, 1\}$  whose error rate  $\Pr_{(x, \ell) \sim P}[g(x) \neq \ell]$  is low. We will consider learning algorithms whose only access to points  $x$  is through a pairwise similarity function  $K(x, x')$  mapping pairs of points to values in the range  $[-1, 1]$ . Specifically,

**Definition 1** A similarity function over  $X$  is any pairwise function  $K : X \times X \rightarrow [-1, 1]$ . We say that  $K$  is a symmetric similarity function if  $K(x, x') = K(x', x)$  for all  $x, x'$ .

Our goal is to describe “goodness” properties that are sufficient for a similarity function to allow one to learn well that ideally are intuitive and subsume the usual notion of good kernel function.

A similarity function  $K$  is a valid kernel function if it is positive-semidefinite, i.e. there exists a function  $\phi$  from the instance space  $X$  into some (implicit) Hilbert “ $\phi$ -space” such that  $K(x, x') = \langle \phi(x), \phi(x') \rangle$ . See, e.g., Smola and Schölkopf (2002) for a discussion on conditions for a mapping being a kernel function. Throughout this work, and without loss of generality, we will only consider kernels such that  $K(x, x) \leq 1$  for all  $x \in X$  (any kernel  $K$  can be converted into this form by, for instance, defining  $\tilde{K}(x, x') =$

$K(x, x')/\sqrt{K(x, x)K(x', x')}$ . We say that  $K$  is  $(\epsilon, \gamma)$ -kernel good for a given learning problem  $P$  if there exists a vector  $\beta$  in the  $\phi$ -space that has error  $\epsilon$  at margin  $\gamma$ ; for simplicity we consider only separators through the origin. Specifically:

**Definition 2**  $K$  is an  $(\epsilon, \gamma)$ -good kernel function if there exists a vector  $\beta$ ,  $\|\beta\| \leq 1$  such that

$$\Pr_{(x, \ell) \sim P} [\ell \langle \phi(x), \beta \rangle \geq \gamma] \geq 1 - \epsilon.$$

We say that  $K$  is  $\gamma$ -kernel good if it is  $(\epsilon, \gamma)$ -kernel good for  $\epsilon = 0$ ; i.e., it has zero error at margin  $\gamma$ .

Given a kernel that is  $(\epsilon, \gamma)$ -kernel-good for some learning problem  $P$ , a predictor with error rate at most  $\epsilon + \epsilon_{\text{acc}}$  can be learned (with high probability) from a sample of  $\tilde{O}((\epsilon + \epsilon_{\text{acc}})/(\gamma^2 \epsilon_{\text{acc}}^2))$  random examples from  $P$  by minimizing the number of margin  $\gamma$  violations on the sample (McAllester, 2003). However, minimizing the number of margin violations on the sample is a difficult optimization problem: it is NP-hard, and even NP-hard to approximate (Arora et al., 1997; Feldman et al., 2006; Guruswami & Raghavendra, 2006). Instead, it is common to minimize the so-called *hinge loss* relative to a margin.

**Definition 3** We say that  $K$  is  $(\epsilon, \gamma)$ -kernel good in hinge-loss if there exists a vector  $\beta$ ,  $\|\beta\| \leq 1$  such that

$$\mathbf{E}_{(x, \ell) \sim P} [1 - \ell \langle \beta, \phi(x) \rangle / \gamma]_+ \leq \epsilon,$$

where  $[1 - z]_+ = \max(1 - z, 0)$  is the hinge loss.

Given a kernel that is  $(\epsilon, \gamma)$ -kernel-good in hinge-loss, a predictor with error rate at most  $\epsilon + \epsilon_{\text{acc}}$  can be efficiently learned from a sample of size  $\tilde{O}((\epsilon + \epsilon_{\text{acc}})/(\gamma^2 \epsilon_{\text{acc}}^2))$  with high probability by minimizing the average hinge loss relative to margin  $\gamma$  on the sample (Bartlett & Mendelson, 2003).

We now present the definition of a good similarity function from (Balcan & Blum, 2006; Srebro, 2007).

**Definition 4 (Previous, Margin Violations)** A pairwise function  $K$  is an  $(\epsilon, \gamma)$ -good similarity function for a learning problem  $P$  if there exists a weighting function  $w : X \rightarrow [0, 1]$  such that at least a  $1 - \epsilon$  probability mass of examples  $(x, \ell)$  satisfy:

$$\mathbf{E}_{(x', \ell') \sim P} [\ell \ell' w(x') K(x, x')] \geq \gamma. \quad (2.1)$$

That is, if the underlying distribution is 50/50 positive and negative, this is saying that the average weighted similarity of an example  $x$  to random examples  $x'$  of its own label should be  $2\gamma$  larger than the average weighted similarity of  $x$  to random examples  $x'$  of the other label.

Balcan and Blum (2006) show how a predictor with error rate at most  $\epsilon + \epsilon_{\text{acc}}$  can be learned from  $\tilde{O}((\epsilon + \epsilon_{\text{acc}})/(\gamma^2 \epsilon_{\text{acc}}^2))$  samples using an  $(\epsilon, \gamma)$ -good similarity function  $K$ : First draw from  $P$  an (unlabeled) sample  $S = \{x'_1, \dots, x'_d\}$  of  $d = (4/\gamma)^2 \ln(4/(\delta \epsilon_{\text{acc}}))$  random “landmarks”, and construct the mapping  $\phi^S : X \rightarrow \mathbb{R}^d$  defined as  $\phi^S_i(x) = \frac{1}{\sqrt{d}} K(x, x'_i)$ ,  $i \in \{1, \dots, d\}$ . With probability at least  $1 - \delta$

<sup>1</sup>The  $\tilde{O}(\cdot)$  notation hides logarithmic factors in the arguments and in the failure probability.

over the random sample  $S$ , the induced distribution  $\phi^S(P)$  in  $\mathbb{R}^d$  has a separator of error at most  $\epsilon + \epsilon_{\text{acc}}/2$  at margin at least  $\gamma/2$ . Now, draw a fresh sample, map it into the transformed space using  $\phi^S$ , and then learn a good linear separator in the transformed space. The total sample complexity is dominated by the  $\tilde{O}((\epsilon + \epsilon_{\text{acc}})d/\epsilon_{\text{acc}}^2) = \tilde{O}((\epsilon + \epsilon_{\text{acc}})/(\gamma^2 \epsilon_{\text{acc}}^2))$  sample complexity of learning in the transformed space, yielding the same overall sample complexity as with an  $(\epsilon, \gamma)$ -good kernel function.

The above bounds refer to learning a linear separator by minimizing the error over the training sample. As mentioned earlier, this minimization problem is NP-hard even to approximate. Again, we can instead consider the hinge-loss rather than the number of margin violations. Balcan and Blum (2006) and Srebro (2007) therefore provide the following hinge-loss version of their definition:

**Definition 5 (Previous, Hinge Loss)** A similarity function  $K$  is an  $(\epsilon, \gamma)$ -good similarity function in hinge loss for a learning problem  $P$  if there exists a weighting function  $w(x') \in [0, 1]$  for all  $x' \in X$  such that

$$\mathbf{E}_{(x, \ell) \sim P} [1 - \ell g(x) / \gamma]_+ \leq \epsilon, \quad (2.2)$$

where  $g(x) = \mathbf{E}_{(x', \ell') \sim P} [\ell' w(x') K(x, x')]$  is the similarity-based prediction made using  $w(\cdot)$ , and recall that  $[1 - z]_+ = \max(0, 1 - z)$  is the hinge-loss.

The same algorithm as above, but now using SVM to minimize hinge-loss in the transformed space, allows one to efficiently use a similarity function satisfying this definition to find a predictor of error  $\epsilon + \epsilon_{\text{acc}}$  using  $\tilde{O}((\epsilon + \epsilon_{\text{acc}})/(\gamma^2 \epsilon_{\text{acc}}^2))$  examples.

### 3 New Notions of Good Similarity Functions

In this section we provide new notions of good similarity functions generalizing Definitions 4 and 5 that we prove have a number of important advantages.

In the definitions of Balcan and Blum (2006), a weight  $w(x') \in [0, 1]$  was used in defining the quantity of interest  $\mathbf{E}_{(x', \ell') \sim P} [\ell' w(x') K(x, x')]$ . Here, it will instead be more convenient to think of  $w$  as the expected value of an indicator random variable  $R(x) \in \{0, 1\}$  where we will view the (probabilistic) set  $\{x : R(x) = 1\}$  as a set of “reasonable points”. Formally, we will then be sampling from the joint distribution  $P(x, \ell(x), R(x)) = P(x, \ell(x))P(R(x)|x)$  but we will sometimes omit the explicit dependence on  $R$  when it is clear from context. Our new definition is now as follows.

**Definition 6 (Main, Margin Violations)** A similarity function  $K$  is an  $(\epsilon, \gamma, \tau)$ -good similarity function for a learning problem  $P$  if there exists a (random) indicator function  $R(x)$  defining a (probabilistic) set of “reasonable points” such that the following conditions hold:

1. A  $1 - \epsilon$  probability mass of examples  $(x, \ell)$  satisfy

$$\mathbf{E}_{(x', \ell') \sim P} [\ell \ell' K(x, x') \mid R(x')] \geq \gamma \quad (3.1)$$

2.  $\Pr_{x'} [R(x')] \geq \tau$ .

If the reasonable set  $R$  is 50/50 positive and negative (i.e.,  $\Pr_{x'}[\ell(x') = 1 | R(x')] = 1/2$ ), we can interpret the condition as stating that most examples  $x$  are on average  $2\gamma$  more similar to random reasonable examples  $x'$  of their own label than to random reasonable examples  $x'$  of the other label. The second condition is that at least a  $\tau$  fraction of the points should be reasonable.

We also consider a hinge-loss version of the definition:

**Definition 7 (Main, Hinge Loss)** *A similarity function  $K$  is an  $(\epsilon, \gamma, \tau)$ -good similarity function in hinge loss for a learning problem  $P$  if there exists a (probabilistic) set  $R$  of “reasonable points” such that the following conditions hold:*

1. We have

$$\mathbf{E}_{(x,\ell)\sim P} \left[ [1 - \ell g(x)/\gamma]_+ \right] \leq \epsilon, \quad (3.2)$$

where  $g(x) = \mathbf{E}_{(x',\ell',R(x'))}[\ell' K(x, x') | R(x')]$ .

2.  $\Pr_{x'}[R(x')] \geq \tau$ .

It is not hard to see that an  $(\epsilon, \gamma)$ -good similarity function under Definitions 4 and 5 is also an  $(\epsilon, \gamma, \gamma)$ -good similarity function under Definitions 6 and 7, respectively. In the reverse direction, an  $(\epsilon, \gamma, \tau)$ -good similarity function under Definitions 6 and 7 is an  $(\epsilon, \gamma\tau)$ -good similarity function under Definitions 4 and 5 (respectively). For formal proofs, see Theorems 23 and 24 in Appendix A.

As we will see, under both old and new definitions, the number of labeled samples required for learning grows as  $1/\gamma^2$ . The key distinction between them is that we introduce a new parameter,  $\tau$ , that primarily affects the number of *unlabeled* examples required. This decoupling of the number of labeled and unlabeled examples enables us to handle a wider variety of situations with an improved labeled sample complexity. In particular, in translating from a kernel to a similarity function, we will find that much of the loss can now be placed into the  $\tau$  parameter.

In the following we prove three types of results about this new notion of similarity. The first is that similarity functions satisfying these conditions are sufficient for learning (in polynomial time in the case of Definition 7), with a sample size of  $O(\frac{1}{\gamma^2} \ln(\frac{1}{\gamma\tau}))$  labeled examples and  $O(\frac{1}{\tau\gamma^2})$  unlabeled examples. This is particularly useful in settings where unlabeled data is plentiful and cheap—such settings are increasingly common in learning applications (Mitchell, 2006; Chapelle et al., 2006)—or for distribution-specific learning where unlabeled data may be viewed as free.

The second main theorem we prove is that *any* class  $C$ , over a sufficiently unconcentrated distribution on examples, has a  $(0, 1, 1/(2|C|))$ -good similarity function (under either definition 6 or 7), whereas there exist classes  $C$  that have no  $(0.5, 8/\sqrt{|C|})$ -good kernel functions in hinge loss. This provides a clear separation between the similarity and kernel notions in terms of the parameters controlling labeled sample complexity. The final main theorem we prove is that any large-margin kernel function also satisfies our similarity definitions, with substantially less loss in the parameters controlling labeled sample complexity compared to the definition of (Balcan & Blum, 2006). For example, if  $K$  is a  $(0, \gamma)$ -good kernel, then it is an  $(\epsilon', \epsilon'\gamma^2)$ -good similarity function under

Definitions 4 and 5, and this is tight (Srebro, 2007), resulting in a sample complexity of  $\tilde{O}(1/(\gamma^4\epsilon^3))$  to achieve error  $\epsilon$ . However, we can show  $K$  is an  $(\epsilon', \gamma^2, \epsilon')$ -good similarity function under the new definition,<sup>2</sup> resulting in a sample complexity of only  $\tilde{O}(1/(\gamma^4\epsilon))$ .

## 4 Good Similarity Functions Allow Learning

The basic approach proposed for learning using a similarity function is similar to that of Balcan and Blum (2006). First, a feature space is constructed, consisting of similarities to randomly chosen landmarks. Then, a linear predictor is sought in this feature space. However, under the previous definitions, we were guaranteed large  $L_2$ -margin in this feature space, whereas under the new definitions we are guaranteed large  $L_1$ -margin in the feature space.

After recalling the notion of an  $L_1$ -margin and its associated learning guarantee, we first establish that, for an  $(\epsilon, \gamma, \tau)$ -good similarity function, the feature map constructed using  $\tilde{O}(1/(\tau\gamma^2))$  landmarks indeed has (with high probability) a large  $L_1$ -margin separator. Using this result, we then obtain a learning guarantee by following the strategy outlined above.

In speaking of  $L_1$ -margin  $\gamma$ , we refer to separation with a margin  $\gamma$  by a unit- $L_1$ -norm linear separator, in a unit- $L_\infty$ -bounded feature space. Formally, let  $\phi : x \mapsto \phi(x)$ ,  $\phi(x) \in \mathbb{R}^d$ , with  $\|\phi(x)\|_\infty \leq 1$  be a mapping of the data to a  $d$ -dimensional feature space. We say that a linear predictor  $\alpha \in \mathbb{R}^d$ , achieves error  $\epsilon$  relative to  $L_1$ -margin  $\gamma$  if  $\Pr_{x,\ell(x)}(\ell(x)\langle \alpha, \phi(x) \rangle \geq \gamma) \geq 1 - \epsilon$  (this is the standard margin constraint) and  $\|\alpha\|_1 = 1$ .

Given a  $d$ -dimensional feature map under which there exists some (unknown) zero-error linear separator with  $L_1$ -margin  $\gamma$ , we can efficiently learn a predictor with error at most  $\epsilon_{\text{acc}}$  using  $O\left(\frac{\log d}{\epsilon_{\text{acc}}\gamma^2}\right)$  examples (with high probability).

This can be done using the Winnow algorithm with a standard online-to-batch conversion (Littlestone, 1989). If we can only guarantee the existence of a separator with error  $\epsilon > 0$  relative to  $L_1$ -margin  $\gamma$ , then a predictor with error  $\epsilon + \epsilon_{\text{acc}}$  can be theoretically learned (with high probability) from a sample of  $\tilde{O}((\log d)/(\gamma^2\epsilon_{\text{acc}}^2))$  examples by minimizing the number of  $L_1$ -margin  $\gamma$  violations on the sample (Zhang, 2002).

We are now ready to state the main result enabling learning using good similarity functions:

**Theorem 8** *Let  $K$  be an  $(\epsilon, \gamma, \tau)$ -good similarity function for a learning problem  $P$ . Let  $S = \{x'_1, x'_2, \dots, x'_d\}$  be a (potentially unlabeled) sample of*

$$d = \frac{2}{\tau} \left( \log(2/\delta) + 8 \frac{\log(2/\delta)}{\gamma^2} \right)$$

*landmarks drawn from  $P$ . Consider the mapping  $\phi^S : X \rightarrow \mathbb{R}^d$  defined as follows:  $\phi^S_i(x) = K(x, x'_i)$ ,  $i \in \{1, \dots, d\}$ . Then, with probability at least  $1 - \delta$  over the random sample  $S$ , the induced distribution  $\phi^S(P)$  in  $\mathbb{R}^d$  has a separator of error at most  $\epsilon + \delta$  relative to  $L_1$  margin at least  $\gamma/2$ .*

<sup>2</sup>Formally, the translation produces an  $(\epsilon', \gamma^2/c, \epsilon')$ -good similarity function for some  $c \leq 1$ . However, smaller values of  $c$  only improve the bounds.

**Proof:** First, note that since  $|K(x, x)| \leq 1$  for all  $x$ , we have  $\|\phi^S(x)\|_\infty \leq 1$ .

Consider the linear separator  $\alpha \in \mathbb{R}^d$ , given by  $\alpha_i = \ell(x'_i)R(x'_i)/d_1$  where  $d_1 = \sum_i R(x'_i)$  is the number of landmarks with  $R(x') = 1$ . This normalization ensures  $\|\alpha\|_1 = 1$ . Note that we take  $R(x'_i)$  to be drawn jointly with  $x'_i$ . If it is random, then it is randomly instantiated to either zero or one.

We have, for any  $x, \ell(x)$ :

$$\ell(x)\langle \alpha, \phi^S(x) \rangle = \frac{\sum_{i=1}^d \ell(x)\ell(x'_i)R(x'_i)K(x, x'_i)}{d_1} \quad (4.1)$$

This is an empirical average of  $d_1$  terms

$$-1 \leq \ell(x)\ell(x')K(x, x') \leq 1$$

for which  $R(x') = 1$ . For any  $x$  we can apply Hoeffding's inequality, and obtain that with probability at least  $1 - \delta^2/2$  over the choice of  $S$ , we have:

$$\ell(x)\langle \alpha, \phi^S(x) \rangle \geq \mathbf{E}_{x'}[K(x, x')\ell(x')\ell(x)|R(x')] - \sqrt{\frac{2 \log(\frac{2}{\delta^2})}{d_1}} \quad (4.2)$$

Since the above holds for any  $x$  with probability at least  $1 - \delta^2/2$  over  $S$ , it also holds with probability at least  $1 - \delta^2/2$  over the choice of  $x$  and  $S$ . We can write this as:

$$\mathbf{E}_{S \sim P^d} \left[ \Pr_{x \sim P}(\text{violation}) \right] \leq \delta^2/2 \quad (4.3)$$

where ‘‘violation’’ refers to violating (4.2). Applying Markov's inequality we get that with probability at least  $1 - \delta/2$  over the choice of  $S$ , at most  $\delta$  fraction of points violate (4.2). Recalling Definition 6, at most an additional  $\epsilon$  fraction of the points violate (3.1). But for the remaining  $1 - \epsilon - \delta$  fraction of the points, for which both (4.2) and (3.1) hold, we have:

$$\ell(x)\langle \alpha, \phi^S(x) \rangle \geq \gamma - \sqrt{\frac{2 \log(\frac{1}{\delta^2})}{d_1}}. \quad (4.4)$$

To bound the second term we need an upper bound on  $d_1$ , the number of reasonable landmarks. The probability of each of the  $d$  landmarks being reasonable is at least  $\tau$  and so the number of reasonable landmarks follows a Binomial distribution, ensuring  $d_1 \geq 8 \log(1/\delta)/\gamma^2$  with probability at least  $1 - \delta/2$ . When this happens, we have  $\sqrt{\frac{2 \log(\frac{1}{\delta^2})}{d_1}} \leq \gamma/2$ . We get then, that with probability at least  $1 - \delta$ , for at least  $1 - \epsilon - \delta$  of the points:

$$\ell(x)\langle \alpha, \phi^S(x) \rangle \geq \gamma/2. \quad (4.5)$$

■

For the realizable ( $\epsilon = 0$ ) case, we obtain:

**Corollary 9** *If  $K$  is an  $(0, \gamma, \tau)$ -good similarity function then with high probability we can efficiently find a predictor with error at most  $\epsilon_{acc}$  from an unlabeled sample of size  $d_u = \tilde{O}\left(\frac{1}{\gamma^2\tau}\right)$  and from a labeled sample of size  $d_l = \tilde{O}\left(\frac{\log d_u}{\gamma^2\epsilon_{acc}}\right)$ .*

**Proof:** We have proved in Theorem 8 that if  $K$  is  $(0, \gamma, \tau)$ -good similarity function, then with high probability there exists a low-error (at most  $\delta$ ) large-margin (at least  $\frac{\gamma}{2}$ ) separator in the transformed space under mapping  $\phi^S$ . Thus, all we need now to learn well is to draw a new fresh sample  $\tilde{S}$ , map it into the transformed space using  $\phi^S$ , and then apply a good algorithm for learning linear separators in the new space that produces a hypothesis of error at most  $\epsilon_{acc}$  with probability at least  $1 - \delta$ . In particular, remember that the vector  $\alpha$  has error at most  $\delta$  at  $L_1$  margin  $\gamma/2$  over  $\phi^S(P)$ , where the mapping  $\phi^S$  produces examples of  $L_\infty$  norm at most 1. In order to enjoy the better learning guarantees of the separable case, we will set  $\delta$  small enough so that no bad points appear in the sample. The Corollary now follows from the  $L_1$ -margin learning guarantee in the separable case, discussed earlier in the Section. ■

For the general ( $\epsilon > 0$ ) case, Theorem 8 implies that by following our two-stage approach, first using  $d_u = \tilde{O}\left(\frac{1}{\gamma^2\tau}\right)$  unlabeled examples as landmarks in order to construct  $\phi^S(\cdot)$ , and then using a fresh sample of size  $d_l = \tilde{O}\left(\frac{1}{\gamma^2\epsilon_{acc}^2} \ln d_u\right)$  to learn a low-error  $L_1$ -margin  $\gamma$  separator in  $\phi^S(\cdot)$ , we have:

**Corollary 10** *If  $K$  is a  $(\epsilon, \gamma, \tau)$ -good similarity function then by minimizing  $L_1$  margin violations we can find a predictor with error at most  $\epsilon_{acc}$  from an unlabeled sample of size  $d_u = \tilde{O}\left(\frac{1}{\gamma^2\tau}\right)$  and from a labeled sample of size  $d_l = \tilde{O}\left(\frac{\log d_u}{\gamma^2\epsilon_{acc}^2}\right)$ .*

The procedure described above, although well defined, involves a difficult optimization problem: minimizing the number of  $L_1$ -margin violations. In order to obtain a computationally tractable procedure, we consider the hinge-loss instead of the margin error. In a feature space with  $\|\phi(x)\|_\infty \leq 1$  as above, we say that a unit- $L_1$ -norm predictor  $\alpha$ ,  $\|\alpha\|_1 = 1$ , has expected hinge-loss  $\mathbf{E}[[1 - \ell(x)\langle \alpha, \phi(x) \rangle / \gamma]_+]$  relative to  $L_1$ -margin  $\gamma$ . Now, if we know there is some (unknown) predictor with hinge-loss  $\epsilon$  relative  $L_1$ -margin  $\gamma$ , then a predictor with error  $\epsilon + \epsilon_{acc}$  can be learned (with high probability) from a sample of  $\tilde{O}(\log d / (\gamma^2 \epsilon_{acc}^2))$  examples by minimizing the empirical average hinge-loss relative to  $L_1$ -margin  $\gamma$  on the sample (Zhang, 2002).

Before proceeding to discussing the optimization problem of minimizing the average hinge-loss relative to a fixed  $L_1$ -margin, let us establish the analogue of Theorem 8 for the hinge-loss:

**Theorem 11** *Let  $K$  be an  $(\epsilon, \gamma, \tau)$ -good similarity function in hinge-loss for a learning problem  $P$ . For any  $\epsilon_1 > 0$  and  $0 < \lambda < \gamma\epsilon_1/4$  let  $S = \{x'_1, x'_2, \dots, x'_d\}$  be a sample of size  $d = \frac{2}{\tau} (\log(2/\delta) + 16 \log(2/\delta) / (\epsilon_1\gamma)^2)$  drawn from  $P$ . With probability at least  $1 - \delta$  over the random sample  $S$ , the induced distribution  $\phi^S(P)$  in  $\mathbb{R}^d$ , for  $\phi^S$  as defined in Theorem 8, has a separator achieving hinge-loss at most  $\epsilon + \epsilon_1$  at margin  $\gamma$ .*

**Proof:** We use the same construction as in Theorem 8. ■

**Corollary 12**  $K$  is an  $(\epsilon, \gamma, \tau)$ -good similarity function in hinge loss then we can efficiently find a predictor with error at most  $\epsilon + \epsilon_{acc}$  from an unlabeled sample of size  $d_u = \tilde{O}\left(\frac{1}{\gamma^2 \epsilon_{acc}^2 \tau}\right)$  and from a labeled sample of size  $d_l = \tilde{O}\left(\frac{\log d_u}{\gamma^2 \epsilon_{acc}^2}\right)$ .

For the hinge-loss, our two stage procedure boils down to solving the following optimization problem w.r.t.  $\alpha$ :

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{d_l} \left[ 1 - \sum_{j=1}^{d_u} \alpha_j \ell(x_i) K(x_i, x'_j) \right]_+ \quad (4.6) \\ \text{s.t.} \quad & \sum_{j=1}^{d_u} |\alpha_j| \leq 1/\gamma \end{aligned}$$

This is a linear program and can thus be solved in polynomial time, establishing the efficiency in Corollary 12.

An optimization problem similar to (4.6), though usually with the same set of points used both as landmarks and as training examples, is actually fairly commonly used as a learning rule in practice (Bennett & Campbell, 2000; Roth, 2001; Guigue et al., 2005). Such a learning rule is typically discussed as an alternative to SVMs. In fact, Tipping (2001) suggest the Relevance Vector Machine (RVM) as a Bayesian alternative to SVMs. The MAP estimate of the RVM is given by an optimization problem similar to (4.6), though with a loss function different from the hinge loss (the hinge-loss cannot be obtained as a log-likelihood). Similarly, Singer (2000) suggests Norm-Penalized Leveraging Procedures as a boosting-like approach that mimics SVMs. Again, although the specific loss functions studied by Singer are different from the hinge-loss, the method (with a norm exponent of 1, as in Singer’s experiments) otherwise corresponds to a coordinate-descent minimization of (4.6). In both cases, no learning guarantees are provided.

The motivation for using (4.6) as an alternative to SVMs is usually that the  $L_1$ -regularization on  $\alpha$  leads to sparsity, and hence to “few support vectors” (although Vincent and Bengio (2002), who also discuss (4.6), argue for more direct ways of obtaining such sparsity), and also that the linear program (4.6) might be easier to solve than the SVM quadratic program. However, we are not aware of a previous discussion on how learning using (4.6) relates to learning using a SVM, or on learning guarantees using (4.6) in terms of properties of the similarity function  $K$ . Guarantees solely in terms of the feature space in which we seek low  $L_1$ -margin ( $\phi^S$  in our notation) are problematic, as this feature space is generated randomly from data.

In fact, in order to enjoy the SVM guarantees while using  $L_1$  regularization to obtain sparsity, some authors suggest regularizing both the  $L_1$  norm  $\|\alpha\|_1$  of the coefficient vector  $\alpha$  (as in (4.6)), and the norm  $\|\beta\|$  of the corresponding predictor  $\beta = \sum_j \alpha_j \phi(x'_j)$  in the Hilbert space implied by  $K$ , where  $K(x, x') = \langle \phi(x), \phi(x') \rangle$ , as when using a SVM with  $K$  as a kernel (Osuna & Girosi, 1999; Gunn & Kandola, 2002).

Here, we provide a natural condition on the similarity function  $K$  (Definition 7), that justifies the learning rule (4.6). Furthermore, we show (in Section 6) than any similarity function that is good as a kernel, and can ensure SVM learning,

is also good as a similarity function and can thus also ensure learning using the learning rule (4.6) (though possibly with some deterioration of the learning guarantees). These arguments can be used to justify (4.6) as an alternative to SVMs.

Before concluding this discussion, we would like to mention that Girosi (1998) previously established a rather different connection between regularizing the  $L_1$  norm  $\|\alpha\|_1$  and regularizing the norm of the corresponding predictor  $\beta$  in the implied Hilbert space. Girosi considered a hard-margin SVR (Support Vector Regression Machine, i.e. requiring each prediction to be within  $(\ell(x) - \epsilon, \ell(x) + \epsilon)$ ), in the noiseless case where the mapping  $x \mapsto \ell(x)$  is in the Hilbert space. In this setting, Girosi showed that a hard-margin SVR is equivalent to minimizing the distance in the implied Hilbert space between the correct mapping  $x \mapsto \ell(x)$  and the predictions  $x \mapsto \sum_j \alpha_j K(x, x'_j)$ , with an  $L_1$  regularization term  $\|\alpha\|_1$ . However, this distance between prediction functions is very different than the objective in (4.6), and again refers back to the implied feature space which we are trying to avoid.

## 5 Separation Results

In this Section, we show an example of a finite concept class for which no kernel yields good learning guarantees when used as a kernel, but for which there does exist a good similarity function yielding the optimal sample complexity. That is, we show that some concept classes cannot be reasonably represented by kernels, but can be reasonably represented by similarity functions.

Specifically, we consider a class  $C$  of  $n$  pairwise uncorrelated functions. This is a finite class of cardinality  $|C| = n$ , and so if the target belongs to  $C$  then  $O(\frac{1}{\epsilon} \log n)$  samples are enough for learning a predictor with error  $\epsilon$ .

Indeed, we show here that for *any* concept class  $C$ , so long as the distribution  $D$  is sufficiently unconcentrated, there exists a similarity function that is  $(0, 1, \frac{1}{2|C|})$ -good under our definition for every  $f \in C$ . This yields a (labeled) sample complexity  $O(\frac{1}{\epsilon} \log |C|)$  to achieve error  $\epsilon$ , matching the ideal sample complexity. In other words, for distribution-specific learning (where unlabeled data may be viewed as free) and finite classes, there is no *intrinsic* loss in sample-complexity incurred by choosing to learn via similarity functions. In fact, we also extend this result to classes of bounded VC-dimension rather than bounded cardinality.

In contrast, we show that if  $C$  is a class of  $n$  functions that are pairwise uncorrelated with respect to distribution  $D$ , then *no* kernel is  $(\epsilon, \gamma)$ -good in hinge-loss for all  $f \in C$  even for  $\epsilon = 0.5$  and  $\gamma = 8/\sqrt{n}$ . This extends work of (Ben-David et al., 2003; Forster & Simon, 2006) who give hardness results with comparable margin bounds, but at a much lower error rate. Thus, this shows there is an intrinsic loss incurred by using kernels together with margin bounds, since this results in a sample complexity bound of at least  $\Omega(|C|)$ , rather than the ideal  $O(\log |C|)$ .

We thus demonstrate a gap between the kind of prior knowledge can be represented with kernels as opposed to general similarity functions and demonstrate that similarity functions are strictly more expressive (up to the degradation in parameters discussed earlier).

**Definition 13** We say that a distribution  $D$  over  $X$  is  $\alpha$ -

unconcentrated if the probability mass on any given  $x \in X$  is at most  $\alpha$ .

**Theorem 14** For any class finite class of functions  $C$  and for any  $1/|C|$ -unconcentrated distribution  $D$  over the instance space  $X$ , there exists a similarity function  $K$  that is a  $(0, 1, \frac{1}{2|C|})$ -good similarity function for all  $f \in C$ .

**Proof:** Let  $C = \{f_1, \dots, f_n\}$ . Now, let us partition  $X$  into  $n$  regions  $R_i$  of at least  $1/(2n)$  probability mass each, which we can do since  $D$  is  $1/n$ -unconcentrated. Finally, define  $K(x, x')$  for  $x'$  in  $R_i$  to be  $f_i(x)f_i(x')$ . We claim that for this similarity function,  $R_i$  is a set of “reasonable points” establishing margin  $\gamma = 1$  for target  $f_i$ . Specifically,

$$\begin{aligned} \mathbf{E}[K(x, x')f_i(x)f_i(x') | x' \in R_i] \\ &= \mathbf{E}[f_i(x)f_i(x')f_i(x)f_i(x')] \\ &= 1. \end{aligned}$$

Since  $\Pr(R_i) \geq \frac{1}{2n}$ , this implies that under distribution  $D$ ,  $K$  is a  $(0, 1, \frac{1}{2n})$ -good similarity function for all  $f_i \in C$ . ■

**Note 1:** We can extend this argument to any class  $C$  of small VC dimension. In particular, for any distribution  $D$ , the class  $C$  has an  $\epsilon$ -cover  $C_\epsilon$  of size  $(1/\epsilon)^{O(d/\epsilon)}$ , where  $d$  is the VC-dimension of  $C$  (Benedek & Itai, 1988). By Theorem 14, we can have a  $(0, 1, 1/|C_\epsilon|)$ -good similarity function for the cover  $C_\epsilon$ , which in turn implies an  $(\epsilon, 1, 1/|C_\epsilon|)$ -good similarity function for the original set (even in hinge loss since  $\gamma = 1$ ). Plugging in our bound on  $|C_\epsilon|$ , we get an  $(\epsilon, 1, \epsilon^{O(d/\epsilon)})$ -good similarity function for  $C$ . Thus, the labeled sample complexity we get for learning with similarity functions is only  $O((d/\epsilon) \log(1/\epsilon))$ , and again there is no *intrinsic* loss in sample complexity bounds due to learning with similarity functions.

**Note 2:** The need for the underlying distribution to be unconcentrated stems from our use of this distribution for both labeled and unlabeled data. We could further extend our definition of “good similarity function” to allow for the unlabeled points  $x'$  to come from some other distribution  $D'$  given *a priori*, such as the uniform distribution over the instance space  $X$ . Now, the expectation over  $x'$  and the probability mass of  $R$  would both be with respect to  $D'$ , and the generic learning algorithm would draw points  $x'_i$  from  $D'$  rather than  $D$ . In this case, we would only need  $D'$  to be unconcentrated, rather than  $D$ .

We now prove our lower bound for margin-based learning with kernels.

**Theorem 15** Let  $C$  be a class of  $n$  pairwise uncorrelated functions over distribution  $D$ . Then, there is no kernel that for all  $f \in C$  is  $(\epsilon, \gamma)$ -good in hinge-loss even for  $\epsilon = 0.5$  and  $\gamma = 8/\sqrt{n}$ .

**Proof:** Let  $C = \{f_1, \dots, f_n\}$ . We begin with the basic Fourier setup (Linial et al., 1989; Mansour, 1994). Given two functions  $f$  and  $g$ , define  $\langle f, g \rangle = \mathbf{E}_x[f(x)g(x)]$  to be their correlation with respect to distribution  $D$ . (This is their inner-product if we view  $f$  as a vector whose  $j$ th coordinate

is  $f(x_j)[D(x_j)]^{1/2}$ ). Because the functions  $f_i \in C$  are pairwise uncorrelated, we have  $\langle f_i, f_j \rangle = 0$  for all  $i \neq j$ , and because the  $f_i$  are boolean functions we have  $\langle f_i, f_i \rangle = 1$  for all  $i$ . Thus they form at least part of an orthonormal basis, and for any hypothesis  $h$  (i.e. any mapping  $X \rightarrow \{\pm 1\}$ ) we have

$$\sum_{f_i \in C} \langle h, f_i \rangle^2 \leq 1.$$

So, this implies

$$\sum_{f_i \in C} |\langle h, f_i \rangle| \leq \sqrt{n}.$$

or equivalently

$$\mathbf{E}_{f_i \in C} |\langle h, f_i \rangle| \leq 1/\sqrt{n}. \quad (5.1)$$

In other words, for any hypothesis  $h$ , if we pick the target at random from  $C$ , the expected magnitude of the correlation between  $h$  and the target is at most  $1/\sqrt{n}$ .

We now consider the implications of having a good kernel. Suppose for contradiction that there exists a kernel  $K$  that is  $(0.5, \gamma)$ -good in hinge loss for every  $f_i \in C$ . What we will show is this implies that for any  $f_i \in C$ , the expected value of  $|\langle h, f_i \rangle|$  for a *random* linear separator  $h$  in the  $\phi$ -space is greater than  $\gamma/8$ . If we can prove this, then we are done because this implies there must *exist* an  $h$  that has  $\mathbf{E}_{f_i \in C} |\langle h, f_i \rangle| > \gamma/8$ , which contradicts equation (5.1) for  $\gamma = 8/\sqrt{n}$ .

So, we just have to prove the statement about random linear separators. Let  $w^*$  denote the vector in the  $\phi$ -space that has hinge-loss at most 0.5 at margin  $\gamma$  for target function  $f_i$ . For any example  $x$ , define  $\gamma_x$  to be the margin of  $\phi(x)$  with respect to  $w^*$ , and define  $\alpha_x = \sin^{-1}(\gamma_x)$  to be the angular margin of  $\phi(x)$  with respect to  $w^*$ .<sup>3</sup> Now, consider choosing a random vector  $h$  in the  $\phi$ -space, where we associate  $h(x) = \text{sign}(h \cdot \phi(x))$ . Since we only care about the absolute value  $|\langle h, f_i \rangle|$ , and since  $\langle -h, f_i \rangle = -\langle h, f_i \rangle$ , it suffices to show that  $\mathbf{E}_h[\langle h, f_i \rangle | h \cdot w^* \geq 0] > \gamma/8$ . We do this as follows.

First, for any example  $x$ , we claim that:

$$\Pr_h[(h(x) \neq f_i(x)) | h \cdot w^* \geq 0] = 1/2 - \alpha_x/\pi. \quad (5.2)$$

This is because we look at the 2-dimensional plane defined by  $\phi(x)$  and  $w^*$ , and consider the half-circle of  $\|h\| = 1$  such that  $h \cdot w^* \geq 0$ , then (5.2) is the portion of the half-circle that labels  $\phi(x)$  incorrectly. Thus, we have:

$$\mathbf{E}_h[\text{err}(h) | h \cdot w^* \geq 0] = \mathbf{E}_x[1/2 - \alpha_x/\pi],$$

and so, using  $\langle h, f_i \rangle = 1 - 2 \text{err}(h)$ , we have:

$$\mathbf{E}_h[\langle h, f_i \rangle | h \cdot w^* \geq 0] = 2\mathbf{E}_x[\alpha_x]/\pi.$$

Finally, we just need to relate angular margin and hinge loss: if  $L_x$  is the hinge-loss of  $\phi(x)$ , then a crude bound on  $\alpha_x$  is

$$\alpha_x \geq \gamma(1 - (\pi/2)L_x).$$

<sup>3</sup>So,  $\alpha_x$  is a bit larger in magnitude than  $\gamma_x$ . This works in our favor when the margin is positive, and we just need to be careful when the margin is negative.

Since we assumed that  $\mathbf{E}_x[L_x] \leq 0.5$ , we have:

$$\mathbf{E}_x[\alpha_x] \geq \gamma(1 - \pi/4).$$

Putting this together we get expected magnitude of correlation of a random halfspace is at least  $2\gamma(1 - \pi/4)/\pi > \gamma/8$  as desired, proving the theorem. ■

An example of a class  $C$  satisfying the above conditions is the class of parity functions over  $\{0, 1\}^{\lg n}$ , which are pairwise uncorrelated with respect to the uniform distribution. Note that the uniform distribution is  $1/|C|$ -unconcentrated, and thus there is a good similarity function. (In particular, one could use  $K(x_i, x_j) = f_j(x_i)f_j(x_j)$ , where  $f_j$  is the parity function associated with indicator vector  $x_j$ .)

We can extend Theorem 15 to classes of large Statistical Query dimension as well. In particular, the SQ-dimension of a class  $C$  with respect to distribution  $D$  is the size  $d$  of the largest set of functions  $\{f_1, f_2, \dots, f_d\} \subseteq C$  such that  $|\langle f_i, f_j \rangle| \leq 1/d^3$  for all  $i \neq j$  (Blum et al., 1994). In this case, we just need to adjust the Fourier analysis part of the argument to handle the fact that the functions may not be completely uncorrelated.

**Theorem 16** *Let  $C$  be a class of functions of SQ-dimension  $d$  with respect to distribution  $D$ . Then, there is no kernel that for all  $f \in C$  is  $(\epsilon, \gamma)$ -good in hinge-loss even for  $\epsilon = 0.5$  and  $\gamma = 16/\sqrt{d}$ .*

**Proof:** Let  $f_1, \dots, f_d$  be  $d$  functions in  $C$  such that  $|\langle f_i, f_j \rangle| \leq 1/d^3$  for all  $i \neq j$ . We can define an orthogonal set of functions  $f'_1, f'_2, \dots, f'_d$  as follows: let  $f'_1 = f_1$ ,  $f'_2 = f_2 - f_1 \langle f_2, f_1 \rangle$ , and in general let  $f'_i$  be the portion of  $f_i$  orthogonal to the space spanned by  $f_1, \dots, f_{i-1}$ . (That is,  $f'_i = f_i - \text{proj}(f_i, \text{span}(f_1, \dots, f_{i-1}))$ , where “proj” is orthogonal projection.) Since the  $f'_i$  are orthogonal and have length at most 1, for any boolean function  $h$  we have  $\sum_i \langle h, f'_i \rangle^2 \leq 1$  and therefore  $\mathbf{E}_i |\langle h, f'_i \rangle| \leq 1/\sqrt{d}$ . Finally, since  $\langle f_i, f_j \rangle \leq 1/d^3$  for all  $i \neq j$ , one can show this implies that  $|f_i - f'_i| \leq 1/d$  for all  $i$ . So,  $\mathbf{E}_i |\langle h, f_i \rangle| \leq 1/\sqrt{d} + 1/d \leq 2/\sqrt{d}$ . The rest of the argument in the proof of Theorem 15 now applies with  $\gamma = 16/\sqrt{d}$ . ■

For example, the class of size- $n$  decision trees over  $\{0, 1\}^n$  has  $n^{\Omega(\log n)}$  pairwise uncorrelated functions over the uniform distribution (in particular, any parity of  $\log n$  variables can be written as an  $n$ -node decision tree). So, this means we cannot have a kernel with margin  $1/\text{poly}(n)$  for all size- $n$  decision trees over  $\{0, 1\}^n$ . However, we can have a similarity function with margin 1, though the  $\tau$  parameter (which controls running time) will be exponentially small.

## 6 Relation between kernels and similarity functions

As is shown in the Appendix (Theorem 25), if a similarity function  $K$  is indeed a kernel, and it is  $(\epsilon, \gamma, \tau)$ -good as a similarity function (possibly in hinge-loss), then it is also  $(\epsilon, \gamma)$ -good as a kernel (respectively, in hinge loss). That is, although the notion of a good similarity function is more widely applicable, for those similarity functions that are positive semidefinite, a good similarity function is also a good

kernel. We now show the converse: if a kernel function is good in the kernel sense, it is also good in the similarity sense, though with some degradation of the margin. This degradation is much smaller than the one incurred previously by Balcan and Blum (2006) and Srebro (2007). Specifically, we can show that if  $K$  is a  $(0, \gamma)$ -good kernel, then  $K$  is  $(\epsilon, \gamma^2, \epsilon)$ -good similarity function for any  $\epsilon$  (formally, it is  $(\epsilon, \gamma^2/c, \epsilon c)$ -good for some  $c \leq 1$ ).

To prove this relationship, we introduce an intermediate notion of a good similarity function.

**Definition 17 (Intermediate, Margin Violations)** *A similarity function  $K$  is a strongly  $(\epsilon, \gamma, M)$ -good similarity function for a learning problem  $P$  if there exists a bounded weighting function  $w$  over  $X$ ,  $w(x') \in [0, M]$  for all  $x' \in X$ ,  $\mathbf{E}[w] \leq 1$  such that at least a  $1 - \epsilon$  probability mass of examples  $x$  satisfy:*

$$\mathbf{E}_{x' \sim P}[\ell(x)\ell(x')w(x')K(x, x')] \geq \gamma. \quad (6.1)$$

**Definition 18 (Intermediate, Hinge Loss)** *A similarity function  $K$  is a strongly  $(\epsilon, \gamma, M)$ -good similarity function in hinge loss for a learning problem  $P$  if there exists a weighting function  $w(x') \in [0, M]$  for all  $x' \in X$ ,  $\mathbf{E}[w] \leq 1$  such that*

$$\mathbf{E}_x \left[ [1 - \ell(x)g(x)/\gamma]_+ \right] \leq \epsilon, \quad (6.2)$$

where  $g(x) = \mathbf{E}_{x' \sim P}[\ell(x')w(x')K(x, x')]$  is the similarity-based prediction made using  $w(\cdot)$ .

These intermediate definitions are closely related to our main similarity function definitions: in particular, if  $K$  is a strongly  $(\epsilon, \gamma, M)$ -good similarity function for a learning problem  $P$ , then it is also an  $(\epsilon, \gamma/c, c/M)$ -good similarity function for some  $\gamma \leq c \leq 1$ .

**Theorem 19** *If  $K$  is a strongly  $(\epsilon, \gamma, M)$ -good similarity function for a learning problem  $P$ , then there exists  $\gamma \leq c \leq 1$  such that  $K$  is a  $(\epsilon, \gamma/c, c/M)$ -good similarity function for  $P$ . If  $K$  is a strongly  $(\epsilon, \gamma, M)$ -good similarity function in hinge loss for  $P$ , then there exists  $\gamma \leq c \leq 1$  such that  $K$  is a  $(\epsilon, \gamma/c, c/M)$ -good similarity function for  $P$ .*

Note that since our guarantees for  $(\epsilon, \gamma, \tau)$ -good similarity functions depend on  $\tau$  only through  $\gamma^2\tau$ , a decrease in  $\tau$  and a proportional increase in  $\gamma$  (as when  $c < 1$  in Theorem 19) only improves the guarantees. However, allowing flexibility in this tradeoff will make the kernel-to-similarity function translation much easier.

**Proof: (of Theorem 19)** First, divide  $w$  by  $M$  to scale its range to  $[0, 1]$ , so  $\mathbf{E}[w] = c/M$  for some  $c \leq 1$  and the margin is now  $\gamma/M$ . Define random indicator  $R(x')$  to equal 1 with probability  $w(x')$  and 0 with probability  $1 - w(x')$ , so we have

$$\tau = \Pr_{x'}[R(x') = 1] = \mathbf{E}[w] = c/M,$$

and we can rewrite (6.1) as

$$\mathbf{E}_{x' \sim P, R}[\ell(x)\ell(x')R(x')K(x, x')] \geq \gamma/M. \quad (6.3)$$

Finally, divide both sides of (6.3) by  $\tau = c/M$ , producing the conditional  $\mathbf{E}_{x'}[\ell(x)\ell(x')K(x, x') \mid R(x')]$  on the LHS

and a margin of  $\gamma/c$  on the RHS. The case of hinge-loss is identical. ■

We will now establish that a similarity function  $K$  that is good as a kernel, is also good as a similarity function in this intermediate sense, and hence, by Theorem 19, also in our original sense. We begin by considering goodness in hinge-loss, and will return to margin violations at the end of the Section.

**Theorem 20** *If  $K$  is  $(\epsilon_0, \gamma)$ -good kernel in hinge loss for learning problem (with deterministic labels), then it is also a strongly  $(\epsilon_0 + \epsilon_1, \frac{\gamma^2}{1+\epsilon_0/2\epsilon_1}, \frac{1}{2\epsilon_1+\epsilon_0})$ -good similarity in hinge loss for the learning problem, for any  $\epsilon_1 > 0$ .*

**Proof:** We initially only consider finite discrete distributions, where:

$$\Pr(x_i, y_i) = p_i \quad (6.4)$$

for  $i = 1 \dots n$ , with  $\sum_{i=1}^n p_i = 1$  and  $x_i \neq x_j$  for  $i \neq j$ .

Let  $K$  be any kernel function that is  $(\epsilon_0, \gamma)$ -kernel good in hinge loss. Let  $\phi$  be the implied feature mapping and denote  $\phi_i = \phi(x_i)$ . Consider the following weighted-SVM quadratic optimization problem with regularization parameter  $C$ :

$$\text{minimize } \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^n p_i [1 - y_i \langle \beta, \phi_i \rangle]_+ \quad (6.5)$$

The dual of this problem, with dual variables  $\alpha_i$ , is:

$$\begin{aligned} \text{maximize } & \sum_i \alpha_i - \frac{1}{2} \sum_{ij} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \text{subject to } & 0 \leq \alpha_i \leq C p_i \end{aligned} \quad (6.6)$$

There is no duality gap, and furthermore the primal optimum  $\beta^*$  can be expressed in terms of the dual optimum  $\alpha^*$ :  $\beta^* = \sum_i \alpha_i^* y_i \phi_i$ .

Since  $K$  is  $(\epsilon_0, \gamma)$ -kernel-good in hinge-loss, there exists a predictor  $\|\beta_0\| = 1$  with average-hinge loss  $\epsilon_0$  relative to margin  $\gamma$ . The primal optimum  $\beta^*$  of (6.5), being the optimum solution, then satisfies:

$$\begin{aligned} \frac{1}{2} \|\beta^*\|^2 + C \sum_i p_i [1 - y_i \langle \beta^*, \phi_i \rangle]_+ &\leq \\ \frac{1}{2} \left\| \frac{1}{\gamma} \beta_0 \right\|^2 + C \sum_i p_i [1 - y_i \langle \frac{1}{\gamma} \beta_0, \phi_i \rangle]_+ & \\ = \frac{1}{2\gamma^2} + C \mathbf{E} \left[ [1 - y \langle \frac{1}{\gamma} \beta_0, \phi(x) \rangle]_+ \right] &= \frac{1}{2\gamma^2} + C\epsilon_0 \end{aligned} \quad (6.7)$$

Since both terms on the left hand side are non-negative, each of them is bounded by the right hand side, and in particular:

$$C \sum_i p_i [1 - y_i \langle \beta^*, \phi_i \rangle]_+ \leq \frac{1}{2\gamma^2} + C\epsilon_0 \quad (6.8)$$

Dividing by  $C$  we get a bound on the average hinge-loss of the predictor  $\beta^*$ , relative to a margin of one:

$$\mathbf{E}[[1 - y \langle \beta^*, \phi(x) \rangle]_+] \leq \frac{1}{2C\gamma^2} + \epsilon_0 \quad (6.9)$$

We now use the fact that  $\beta^*$  can be written as  $\beta^* = \sum_i \alpha_i^* y_i \phi_i$  with  $0 \leq \alpha_i^* \leq C p_i$ . Let us consider the weights

$$w_i = w(x_i) = \alpha_i^* / (A p_i) \leq 1 \quad (6.10)$$

So,  $w_i \leq \frac{C}{A}$  and  $\mathbf{E}[w] = \frac{\sum_i \alpha_i^*}{A}$ . Furthermore, since we have no duality gap we also have

$$\begin{aligned} \sum_i \alpha_i^* - \frac{1}{2} \|\beta^*\|^2 &= \frac{1}{2} \|\beta^*\|^2 + C \sum_i p_i [1 - y_i \langle \beta^*, \phi_i \rangle]_+, \\ \text{so } \sum_i \alpha_i^* &\leq \frac{1}{\gamma^2} + C\epsilon_0. \end{aligned}$$

So, we have for every  $x$ , y:

$$\begin{aligned} y \mathbf{E}_{x', y'} [w(x') y' K(x, x')] &= y \sum_i p_i w(x_i) y_i K(x, x_i) \\ &= y \sum_i p_i \alpha_i^* y_i K(x, x_i) / (A p_i) \\ &= y \sum_i \alpha_i^* y_i \langle \phi_i, \phi(x) \rangle / A \\ &= y \langle \beta^*, \phi(x) \rangle / A \end{aligned}$$

Multiplying by  $A$  and using (6.9):

$$\begin{aligned} \mathbf{E}_{x, y} [[1 - A y \mathbf{E}_{x', y'} [w(x') y' K(x, x')]]_+] & \quad (6.11) \\ = \mathbf{E}_{x, y} [[1 - y \langle \beta^*, \phi(x) \rangle]_+] & \leq \frac{1}{2C\gamma^2} + \epsilon_0 \end{aligned}$$

This holds for any  $A$  and  $C$  such that  $(\frac{1}{\gamma^2} + C\epsilon_0) \frac{1}{A} \leq 1$ , and describes the average hinge-loss relative to margin  $1/A$ . We also have the constraint  $\frac{C}{A} \leq M$ . Choosing  $M = \frac{1}{2\epsilon_1 + \epsilon_0}$ ,  $A = \frac{1 + \epsilon_0/2\epsilon_1}{\gamma^2}$ , we set  $C = 1/(2\epsilon_1\gamma^2)$  and get an average hinge-loss of  $\epsilon_0 + \epsilon_1$ ,

$$\mathbf{E}_{x, y} [[1 - y \mathbf{E}_{x', y'} [w(x') y' K(x, x')]/(2\epsilon_1\gamma^2)]_+] \leq \epsilon_0 + \epsilon_1 \quad (6.12)$$

as desired.

This establishes that if  $K$  is  $(\epsilon_0, \gamma)$ -good kernel in hinge loss then it is also a strongly  $(\epsilon_0 + \epsilon_1, \frac{\gamma^2}{1+\epsilon_0/2\epsilon_1}, \frac{1}{2\epsilon_1+\epsilon_0})$ -good similarity in hinge loss, for any  $\epsilon_1 > 0$ , at least for finite discrete distributions.

To extend the result also to non-discrete distributions, we can consider the variational ‘infinite SVM’ problem and apply the same arguments, as in (Srebro, 2007). ■

We can now use the hinge-loss correspondence to get a similar result for the margin-violation definitions:

**Theorem 21** *If  $K$  is  $(\epsilon_0, \gamma)$ -good kernel for a learning problem (with deterministic labels), then it is also a strongly  $(\epsilon_0 + \epsilon_1, \gamma^2/2, \frac{1}{(1-\epsilon_0)\epsilon_1})$ -good similarity function for the learning problem, for any  $\epsilon_1 > 0$ .*

**Proof:** If  $K$  is  $(0, \gamma)$ -good as a kernel, it is also  $(0, \gamma)$  good as a kernel in hinge loss, and we can apply Theorem 20 to obtain that  $K$  is also  $(\epsilon_0/2, \gamma_1, \tau_1)$ -good, where  $\gamma_1 = \gamma^2$  and  $\tau_1 = 1/\epsilon_1$ . We can then bound the number of margin violations at  $\gamma_2 = \gamma_1/2$  by half the hinge loss at margin  $\gamma_1$  to obtain the desired result.

If  $K$  is only  $(\epsilon, \gamma)$ -good as a kernel, we follow a similar procedure to that described in (Srebro, 2007), and consider a distribution conditioned only on those places where there is no error. Returning to the original distribution, we must scale the weights up by an amount proportional to the probability of the event we conditioned on (i.e. the probability of no margin violation). This yields the desired bound. ■

## 7 Learning with Multiple Similarity Functions

Suppose that rather than having a single similarity function, we were instead given  $n$  functions  $K_1, \dots, K_n$ , and our hope is that some convex combination of them will satisfy Definition 6. Is this sufficient to be able to learn well? (Note that a convex combination of similarity functions is guaranteed to have range  $[-1, 1]$  and so be a legal similarity function.) The following generalization of Theorem 8 shows that this is indeed the case. (The analog of Theorem 11 can be derived similarly.)

**Theorem 22** *Suppose  $K_1, \dots, K_n$  are similarity functions such that some (unknown) convex combination of them is  $(\epsilon, \gamma, \tau)$ -good. For any  $\delta > 0$ , let  $S = \{x'_1, x'_2, \dots, x'_d\}$  be a sample of size  $d = 16 \frac{\log(1/\delta)}{\tau\gamma^2}$  drawn from  $P$ . Consider the mapping  $\phi^S : X \rightarrow \mathbb{R}^{nd}$  defined as follows:  $\phi^S_i(x) = (K_1(x, x'_1), \dots, K_n(x, x'_1), \dots, K_1(x, x'_d), \dots, K_n(x, x'_d))$ .*

*With probability at least  $1 - \delta$  over the random sample  $S$ , the induced distribution  $\phi^S(P)$  in  $R^{nd}$  has a separator of error at most  $\epsilon + \delta$  at  $L_1, L_\infty$  margin at least  $\gamma/2$ .*

**Proof:** Let  $K = \alpha_1 K_1 + \dots + \alpha_n K_n$  be an  $(\epsilon, \gamma, \tau)$ -good convex-combination of the  $K_i$ . By Theorem 8, had we instead performed the mapping:  $\tilde{\phi}^S : X \rightarrow R^d$  defined as

$$\tilde{\phi}^S(x) = (K(x, x'_1), \dots, K(x, x'_d)),$$

then with probability  $1 - \delta$ , the induced distribution  $\tilde{\phi}^S(P)$  in  $R^d$  would have a separator of error at most  $\epsilon + \delta$  at margin at least  $\gamma/2$ . Let  $\hat{\beta}$  be the vector corresponding to such a separator in that space. Now, let us convert  $\hat{\beta}$  into a vector in  $R^{nd}$  by replacing each coordinate  $\hat{\beta}_j$  with the  $n$  values  $(\alpha_1 \hat{\beta}_j, \dots, \alpha_n \hat{\beta}_j)$ . Call the resulting vector  $\tilde{\beta}$ . Notice that by design, for any  $x$  we have  $\langle \tilde{\beta}, \phi^S(x) \rangle = \langle \hat{\beta}, \tilde{\phi}^S(x) \rangle$ . Furthermore,  $\|\tilde{\beta}\|_1 = \|\hat{\beta}\|_1$ . Thus, the vector  $\tilde{\beta}$  under distribution  $\phi^S(P)$  has the same properties as the vector  $\hat{\beta}$  under  $\tilde{\phi}^S(P)$ . This implies the desired result. ■

Note that we get significantly better bounds here than in (Balcan & Blum, 2006), since the margin does not drop by a factor of  $\frac{1}{\sqrt{n}}$ .

## 8 Conclusions

We provide a new notion of a “good similarity function” that we prove is strictly more powerful than the traditional notion of a large-margin kernel. Our new notion relies upon  $L_1$  regularized learning, and our separation result is related to a

separation result between what is learnable with  $L_1$  vs.  $L_2$  regularization. In a lower bound of independent interest, we show that if  $C$  is a class of  $n$  pairwise uncorrelated functions, then *no* kernel is  $(\epsilon, \gamma)$ -good in hinge-loss for all  $f \in C$  even for  $\epsilon = 0.5$  and  $\gamma = 8/\sqrt{n}$ .

It would be interesting to explore whether the lower bound could be extended to cover *margin violations* with a constant error rate  $\epsilon > 0$  rather than only hinge-loss. In addition, it would be particularly interesting to develop even broader natural notions of good similarity functions, that allow for functions that are not positive-semidefinite and yet provide even better kernel-to-similarity translations (e.g., not squaring the margin parameter).

**Acknowledgments:** We would like to thank Manfred Warmuth and Hans-Ulrich Simon for helpful discussions. This work was supported in part by the National Science Foundation under grant CCF-0514922, by an IBM Graduate Fellowship, and by a Google Research Grant.

## References

- Arora, S., Babai, L., Stern, J., & Sweedyk, Z. (1997). The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54, 317 – 331.
- Balcan, M.-F., & Blum, A. (2006). On a theory of learning with similarity functions. *Proceedings of the 23rd International Conference on Machine Learning*.
- Bartlett, P. L., & Mendelson, S. (2003). Rademacher and gaussian complexities: risk bounds and structural results. *J. Mach. Learn. Res.*, 3, 463–482.
- Ben-David, S., Eiron, N., & Simon, H.-U. (2003). Limitations of learning via embeddings in euclidean half-spaces. *The Journal of Machine Learning Research*, 3, 441 – 461.
- Benedek, G., & Itai, A. (1988). Learnability by fixed distributions. *Proc. 1st Workshop Computat. Learning Theory* (pp. 80–90).
- Bennett, K. P., & Campbell, C. (2000). Support vector machines: hype or hallelujah? *SIGKDD Explor. Newsl.*, 2, 1–13.
- Blum, A., Furst, M., Jackson, J., Kearns, M., Mansour, Y., & Rudich, S. (1994). Weakly learning DNF and characterizing statistical query learning using fourier analysis. *Proceedings of the 26th Annual ACM Symposium on Theory of Computing* (pp. 253–262).
- Chapelle, O., Schlkopf, B., & Zien, A. (2006). *Semi-supervised learning*. MIT Press.
- Feldman, V., Gopalan, P., Khot, S., & Ponnuswami, A. (2006). New results for learning noisy parities and half-spaces. *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (pp. 563–574).
- Forster, J., & Simon, H.-U. (2006). On the smallest possible dimension and the largest possible margin of linear arrangements representing given concept classes. *Theoretical Computer Science*, 350, 40–48.

Girosi, F. (1998). An equivalence between sparse approximation and support vector machines. *Neural Comput.*, 10, 1455–1480.

Guigue, V., Rakotomamonjy, A., & Canu, S. (2005). Kernel basis pursuit. *Proceedings of the 16th European Conference on Machine Learning (ECML'05)*. Springer.

Gunn, S. R., & Kandola, J. S. (2002). Structural modelling with sparse kernels. *Mach. Learn.*, 48, 137–163.

Guruswami, V., & Raghavendra, P. (2006). Hardness of learning halfspaces with noise. *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (pp. 543–552).

Herbrich, R. (2002). *Learning kernel classifiers*. MIT Press, Cambridge.

Linial, N., Mansour, Y., & Nisan, N. (1989). Constant depth circuits, fourier transform, and learnability. *Proceedings of the Thirtieth Annual Symposium on Foundations of Computer Science* (pp. 574–579). Research Triangle Park, North Carolina.

Littlestone, N. (1989). From online to batch learning. *Proc. 2nd Annual ACM Conference on Computational Learning Theory* (pp. 269–284).

Mansour, Y. (1994). Learning boolean functions via the fourier transform. In *Theoretical advances in neural computation and learning*, 391–424.

McAllester, D. (2003). Simplified pac-bayesian margin bounds. *Proceedings of the 16th Conference on Computational Learning Theory*.

Mitchell, T. (2006). The discipline of machine learning. *CMU-ML-06 108*.

Osuna, E. E., & Girosi, F. (1999). Reducing the run-time complexity in support vector machines. In *Advances in kernel methods: support vector learning*, 271–283. Cambridge, MA, USA: MIT Press.

Roth, V. (2001). Sparse kernel regressors. *ICANN '01: Proceedings of the International Conference on Artificial Neural Networks* (pp. 339–346). London, UK: Springer-Verlag.

Scholkopf, B., & Smola, A. J. (2002). *Learning with kernels. support vector machines, regularization, optimization, and beyond*. MIT University Press, Cambridge.

Scholkopf, B., Tsuda, K., & Vert, J.-P. (2004). *Kernel methods in computational biology*. MIT Press.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Singer, Y. (2000). Leveraged vector machines. *Advances in Neural International Proceedings System 12*.

Smola, A. J., & Schölkopf, B. (2002). *Learning with kernels*. MIT Press.

Srebro, N. (2007). How Good is a Kernel as a Similarity Function. *COLT*.

Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1, 211–244.

Vincent, P., & Bengio, Y. (2002). Kernel matching pursuit. *Mach. Learn.*, 48, 165–187.

Warmuth, M. K., & Vishwanathan, S. V. N. (2005). Leaving the span. *Proceedings of the Annual Conference on Learning Theory*.

Zhang, T. (2002). Covering number bounds of certain regularized linear function classes. *J. Mach. Learn. Res.*, 2, 527–550.

## A Kernels and Similarity Functions

**Theorem 23** *If  $K$  is an  $(\epsilon, \gamma)$ -good similarity function under Definitions 4 and 5, then  $K$  is also an  $(\epsilon, \gamma, \gamma)$ -good similarity function under Definitions 6 and 7, respectively.*

**Proof:** If we set  $\Pr(R(x) | x) = w(x)$ , we get that in order for any point  $x$  to fulfill equation (2.1), we must have

$$\Pr(R(x)) = \mathbf{E}[w(x)] \geq \mathbf{E}[\ell \ell' w(x') K(x, x')] \geq \gamma.$$

Furthermore, for any  $x, \ell$  for which (2.1) is satisfied, we have

$$\begin{aligned} \mathbf{E}[\ell \ell' K(x, x') | R(x')] &= \mathbf{E}[\ell \ell' K(x, x') w(x')] / \Pr(R(x)) \\ &\geq \mathbf{E}[\ell \ell' K(x, x') w(x')] \geq \gamma \quad (\text{A.1}) \end{aligned}$$

**Theorem 24** *If  $K$  is an  $(\epsilon, \gamma, \tau)$ -good similarity function under Definitions 6 and 7, then  $K$  is an  $(\epsilon, \gamma\tau)$ -good similarity function under Definitions 4 and 5 (respectively).*

**Proof:** Setting  $w(x) = \Pr(R(x) | x)$  we have for any  $x, \ell$  satisfying (3.1) that

$$\begin{aligned} \mathbf{E}[\ell \ell' K(x, x') w(x')] &= \mathbf{E}[\ell \ell' K(x, x') R(x')] = \\ &\mathbf{E}[\ell \ell' K(x, x') | R(x')] \Pr(R(x')) \geq \gamma\tau. \quad (\text{A.2}) \end{aligned}$$

A similar calculation establishes the correspondence for the hinge loss. ■

We show in the following that a kernel good as a similarity function is also good as a kernel.

**Theorem 25** *If  $K$  is a valid kernel function, and is  $(\epsilon, \gamma, \tau)$ -good similarity for some learning problem, then it is also  $(\epsilon, \gamma)$ -kernel-good for the learning problem. If  $K$  is  $(\epsilon, \gamma, \tau)$ -good similarity in hinge loss, then it is also  $(\epsilon, \gamma)$ -kernel-good in hinge loss.*

**Proof:** Consider a similarity function  $K$  that is a valid kernel, i.e.  $K(x, x') = \langle \phi(x), \phi(x') \rangle$  for some mapping  $\phi$  of  $x$  to a Hilbert space  $\mathcal{H}$ . For any input distribution and any probabilistic set of reasonable points  $R$  of the input we will construct a linear predictor  $\beta_w \in \mathcal{H}$ , with  $\|\beta_w\| \leq 1$ , such that similarity-based predictions using  $R$  are the same as the linear predictions made with  $\beta_R$ .

Define the following linear predictor  $\beta_R \in \mathcal{H}$ :

$$\beta_R = \mathbf{E}_{x'}[\ell(x')\phi(x')|R(x')]. \quad (\text{A.3})$$

The predictor  $\beta_w$  has norm at most:

$$\begin{aligned} \|\beta_R\| &= \|\mathbf{E}_{x'}[\ell(x')\phi(x')|R(x')]\| \leq \max_{x'} \|\ell(x')\phi(x')\| \\ &\leq \max \|\phi(x')\| = \max \sqrt{K(x', x')} \leq 1 \end{aligned} \quad (\text{A.4})$$

where the second inequality follows from  $|\ell(x')| \leq 1$ .

The predictions made by  $\beta_R$  are:

$$\begin{aligned} \langle \beta_R, \phi(x) \rangle &= \langle \mathbf{E}_{x'}[\ell(x')\phi(x')|R(x')], \phi(x) \rangle = \\ \mathbf{E}_{x'}[\ell(x')\langle \phi(x'), \phi(x) \rangle | R(x')] &= \mathbf{E}_{x'}[\ell(x')K(x, x') | R(x')] \end{aligned} \quad (\text{A.5})$$

That is, using  $\beta_R$  is the same as using similarity-based prediction with  $R$ . In particular, the margin violation rate, as well as the hinge loss, with respect to any margin  $\gamma$ , is the same for predictions made using either  $R$  or  $\beta_R$ . This is enough to establish Theorem 25: If  $K$  is  $(\epsilon, \gamma)$ -good (perhaps for to the hinge-loss), there exists some valid  $R$  that yields margin violation error rate (resp. hinge loss) at most  $\epsilon$  with respect to margin  $\gamma$ , and so  $\beta_R$  yields the same margin violation (resp. hinge loss) with respect to the same margin, establishing  $K$  is  $(\epsilon, \gamma)$ -kernel-good (resp. for the hinge loss). ■

---

# Geometric & Topological Representations of Maximum Classes with Applications to Sample Compression

---

Benjamin I. P. Rubinstein<sup>1</sup> and J. Hyam Rubinstein<sup>2</sup>

<sup>1</sup> Computer Science Division, University of California, Berkeley, U.S.A.

<sup>2</sup> Department of Mathematics & Statistics, the University of Melbourne, Australia

<sup>1</sup>benr@cs.berkeley.edu, <sup>2</sup>rubin@ms.unimelb.edu.au

## Abstract

We systematically investigate finite maximum classes, which play an important role in machine learning as concept classes meeting Sauer's Lemma with equality. Simple arrangements of hyperplanes in Hyperbolic space are shown to represent maximum classes, generalizing the corresponding Euclidean result. We show that sweeping a generic hyperplane across such arrangements forms an unlabeled compression scheme of size VC dimension and corresponds to a special case of peeling the one-inclusion graph, resolving a conjecture of Kuzmin & Warmuth. A bijection between maximum classes and certain arrangements of Piecewise-Linear (PL) hyperplanes in either a ball or Euclidean space is established. Finally, we show that  $d$ -maximum classes corresponding to PL hyperplane arrangements in  $\mathbb{R}^d$  have cubical complexes homeomorphic to a  $d$ -ball, or equivalently complexes that are manifolds with boundary.

## 1 Introduction

*Maximum* concept classes have the largest cardinality possible for their given VC dimension. Such classes are of particular interest as their special recursive structure underlies all general sample compression schemes known to-date [Flo89, War03, KW07]. It is this structure that admits many elegant geometric and algebraic topological representations upon which this paper focuses.

Littlestone & Warmuth [LW86] introduced the study of *sample compression schemes*, defined as a pair of mappings for given concept class  $C$ : a *compression function* mapping a  $C$ -labeled  $n$ -sample to a subsequence of labeled examples and a *reconstruction function* mapping the subsequence to a concept consistent with the entire  $n$ -sample. A compression scheme of bounded size—the maximum cardinality of the subsequence image—was shown to imply learnability [LW86]. The converse—that classes of VC dimension  $d$  admit compression schemes of size  $d$ —has become one of the oldest unsolved problems actively pursued within learning theory. Recently Kuzmin and Warmuth achieved compression of maximum classes without the use of labels [KW07]. They also conjectured that their elegant Min-

Peeling Algorithm constitutes such an unlabeled  $d$ -compression scheme for  $d$ -maximum classes.

As in our previous work [RBR08], maximum classes can be fruitfully viewed as *cubical complexes*. These are also topological spaces, with each cube equipped with a natural topology of open sets from its standard embedding into Euclidean space. We proved that  $d$ -maximum classes correspond to  *$d$ -contractible complexes*—topological spaces with an identity map homotopic to a constant map—extending the result that 1-maximum classes have trees for one-inclusion graphs. Peeling can be viewed as a special form of contractibility for maximum classes. However, there are many non-maximum contractible cubical complexes that cannot be peeled, which demonstrates that peelability reflects more detailed structure of maximum classes than given by contractibility alone.

In this paper we approach peeling from the direction of simple hyperplane arrangement representations of maximum classes. Kuzmin & Warmuth predicted that  $d$ -maximum classes corresponding to simple linear hyperplane arrangements could be unlabeled  $d$ -compressed by sweeping a generic hyperplane across the arrangement, and that concepts are min-peeled as their corresponding cell is swept away [KW07, Conjecture 1]. We positively resolve the first part of the conjecture and show that sweeping such arrangements corresponds to a new form of *corner-peeling*, which we prove is distinct from min-peeling. While *min-peeling* removes minimum degree concepts from a one-inclusion graph, *corner-peeling* peels vertices that are contained in unique cubes of maximum dimension.

We explore simple hyperplane arrangements in Hyperbolic geometry, which we show correspond to a set of maximum classes, properly containing those represented by simple linear Euclidean arrangements. These classes can again be corner-peeled by sweeping. Citing the proof of existence of maximum unlabeled compression schemes presented in [BDL98], Kuzmin & Warmuth ask whether unlabeled compression schemes for infinite classes such as positive half spaces can be constructed explicitly [KW07]. We present constructions for illustrative but simpler classes, suggesting that there are many interesting infinite maximum classes admitting explicit compression schemes, and under appropriate conditions, sweeping infinite Euclidean and Hyperbolic arrangements corresponds to compression by corner-peeling.

Next we prove that all maximum classes in  $\{0, 1\}^n$  are represented as simple arrangements of Piecewise-Linear (PL)

hyperplanes in the  $n$ -ball. This extends previous work on viewing simple PL hyperplane arrangements as maximum classes [GW94]. The close relationship between such arrangements and their Hyperbolic versions suggests that they could be equivalent. Although PL sweeping does not immediately admit corner-peeling or compression, the PL representation result is used to prove the peeling conjecture [KW07, Conjecture 1] for VC dimension two.

We investigate algebraic topological properties of maximum classes. Most notably we characterize  $d$ -maximum classes, corresponding to simple linear Euclidean arrangements, as cubical complexes homeomorphic to the  $d$ -ball. The result that such classes' boundaries are homeomorphic to the  $(d - 1)$ -sphere begins the study of the boundaries of maximum classes, which are closely related to peeling.

Compressing *maximal classes*—classes which cannot be grown without an increase to their VC dimension—is sufficient for compressing all classes, as embedded classes trivially inherit compression schemes of their super-classes. This reasoning motivates the attempt to embed  $d$ -maximal classes into  $O(d)$ -maximum classes [KW07, Open Problem 3]. We present non-embeddability results following from our earlier counter-examples to Kuzmin & Warmuth's minimum degree conjecture [RBR08] and our new results on corner-peeling.

## 2 Background

### 2.1 Algebraic Topology

**Definition 1** A homeomorphism is a one-to-one and onto map  $f$  between topological spaces such that both  $f$  and  $f^{-1}$  are continuous. Spaces  $X$  and  $Y$  are said to be homeomorphic if there exists a homeomorphism  $f : X \rightarrow Y$ .

**Definition 2** A homotopy is a continuous map  $F : X \times [0, 1] \rightarrow Y$ . The initial map is  $F$  restricted to  $X \times \{0\}$  and the final map is  $F$  restricted to  $X \times \{1\}$ . We say that the initial and final maps are homotopic. A homotopy equivalence between spaces  $X$  and  $Y$  is a pair of maps  $f : X \rightarrow Y$  and  $g : Y \rightarrow X$  such that  $f \circ g$  and  $g \circ f$  are homotopic to the identity maps on  $X$  and  $Y$  respectively. We say that  $X$  and  $Y$  have the same homotopy type if there is a homotopy equivalence between them.

**Definition 3** A cubical complex is a union of solid cubes of the form  $[a_1, b_1] \times \dots \times [a_m, b_m]$ , for bounded  $m \in \mathbb{N}$ , such that the intersection of any two cubes in the complex is either a cubical face of both cubes or the empty-set.

**Definition 4** A contractible cubical complex  $X$  is one which has the same homotopy type as a one point space  $\{p\}$ .  $X$  is contractible if and only if the constant map from  $X$  to  $p$  is a homotopy equivalence.

### 2.2 Concept Classes and their Learnability

A concept class  $C$  on domain  $X$ , is a subset of the power set of set  $X$  or equivalently  $C \subseteq \{0, 1\}^X$ . We primarily consider finite domains and so will write  $C \subseteq \{0, 1\}^n$  in the sequel, where it is understood that  $n = |X|$  and the  $n$  dimensions or colors are identified with an ordering  $\{x_i\}_{i=1}^n = X$ .

The one-inclusion graph  $\mathcal{G}(C)$  of  $C \subseteq \{0, 1\}^n$  is the graph with vertex-set  $C$  and edge-set containing  $\{u, v\} \subseteq C$

iff  $u$  and  $v$  differ on exactly one component [HLW94];  $\mathcal{G}(C)$  forms the basis of a prediction strategy with essentially-optimal worst-case expected risk.  $\mathcal{G}(C)$  can be viewed as a simplicial complex in  $\mathbb{R}^n$  by filling in each face with a product of continuous intervals [RBR08]. Each edge in  $\mathcal{G}(C)$  is labeled by the component on which the two vertices differ.

Probably Approximately Correct learnability of a concept class  $C \subseteq \{0, 1\}^X$  is characterized by the finiteness of the Vapnik-Chervonenkis (VC) dimension of  $C$  [BEHW89]. One key to all such results is Sauer's Lemma.

**Definition 5** The VC-dimension of  $C \subseteq \{0, 1\}^X$  is defined as  $\text{VC}(C) = \sup \left\{ n \mid \exists Y \in \binom{X}{n}, \Pi_Y(C) = \{0, 1\}^n \right\}$  where  $\Pi_Y(C) = \{(c(x_1), \dots, c(x_n)) \mid c \in C\} \subseteq \{0, 1\}^n$  is the projection of  $C$  on sequence  $Y = (x_1, \dots, x_n)$ .

**Lemma 6 (IVC71, Sau72, She72)**  $|C| \leq \sum_{i=1}^{\text{VC}(C)} \binom{n}{i}$  for all  $C \subseteq \{0, 1\}^n$ .

Motivated by maximizing concept class cardinality under a fixed VC-dimension, which is related to constructing general sample compression schemes (see Section 2.3), Welzl defined the following special classes in [Wel87].

**Definition 7** Concept class  $C \subseteq \{0, 1\}^X$  is called maximal if  $\text{VC}(C \cup \{c\}) > \text{VC}(C)$  for all  $c \in \{0, 1\}^X \setminus C$ . Furthermore if  $\Pi_Y(C)$  satisfies Sauer's Lemma with equality for each  $Y \in \binom{X}{n}$ , for every  $n \in \mathbb{N}$ , then  $C$  is termed maximum. If  $C \subseteq \{0, 1\}^n$  then  $C$  is maximum (and hence maximal) if  $C$  meets Sauer's Lemma with equality.

The reduction of  $C \subseteq \{0, 1\}^n$  with respect to  $i \in [n] = \{1, \dots, n\}$  is class  $C^i = \Pi_{[n] \setminus \{i\}}(\{c \in C \mid i \in I_{\mathcal{G}(C)}(c)\})$  where  $I_{\mathcal{G}(C)}(c) \subseteq [n]$  denotes the labels of the edges incident to vertex  $c$ ; the tail is  $\text{tail}_i(C) = \{c \in C \mid i \notin I_{\mathcal{G}(C)}(c)\}$ . Welzl showed that if  $C$  is  $d$ -maximum, then  $\Pi_{[n] \setminus \{i\}}(C)$  and  $C^i$  are maximum of VC-dimensions  $d$  and  $d - 1$  respectively.

The results presented below relate to other geometric and topological representations of maximum classes existing in the literature. Under the guise of 'forbidden labels', Floyd showed in [Flo89] that maximum  $C \subseteq \{0, 1\}^n$  of VC-dim  $d$  is the union of a maximally overlapping  $d$ -complete collection of cubes [RBR08]—defined as a collection of  $d$ -cubes which project onto all  $\binom{n}{d}$  possible sets of  $d$  coordinate directions. (See also [Ney06] for a different proof of this.) It has long been known that VC-1 maximum classes have one-inclusion graphs that are trees [Dud85]; in [RBR08] we extended this result by showing that when viewed as complexes,  $d$ -maximum classes are contractible  $d$ -cubical complexes. Finally the cells of a simple linear arrangement of  $n$  hyperplanes in  $\mathbb{R}^d$  form a VC- $d$  maximum class in the  $n$ -cube [Ede87], but not all finite maximum classes correspond to such Euclidean arrangements [Flo89].

### 2.3 Sample Compression Schemes

Littlestone and Warmuth showed that the existence of a compression scheme of finite size is sufficient for learnability of  $C$ , and conjectured the converse, that  $\text{VC}(C) = d < \infty$  implies a compression scheme of size  $d$  [LW86]. Later

Warmuth weakened the conjectured size to  $O(d)$  [War03]. To-date it is only known that maximum classes can be  $d$ -compressed [Flo89]. Unlabeled compression was first explored in [BDL98]; Kuzmin and Warmuth define such compression as follows, explicitly constructing schemes of size  $d$  for maximum classes [KW07].

**Definition 8** Let  $C$  be a  $d$ -maximum class on a finite domain  $X$ . A representation mapping  $r$  of  $C$  satisfies:

1.  $r$  is a bijection between  $C$  and subsets of  $X$  of size at most  $d$ ; and
2. [non-clashing] :  $c|(r(c) \cup r(c')) \neq c'|(r(c) \cup r(c'))$  for all  $c, c' \in C, c \neq c'$ .

As with all published labeled schemes, known unlabeled compression schemes for maximum classes exploit their special recursive projection-reduction structure and so it is doubtful whether such schemes will generalize. Kuzmin and Warmuth conjectured that their *Min-Peeling* Algorithm constitutes an unlabeled  $d$ -compression scheme for maximum classes; it iteratively removes minimum degree vertices from  $\mathcal{G}(C)$ , representing the corresponding concepts by the remaining incident dimensions in the graph [KW07, Conjecture 2]. The authors also conjecture that sweeping a hyperplane in general position across a simple linear arrangement forms a compression scheme that corresponds to min-peeling the associated maximum class [KW07, Conjecture 1]. Possibly the most promising approach to compressing general classes is via their maximum-embeddings: a class  $C$  embedded in class  $C'$  trivially inherits any compression scheme for  $C'$ , and so an important open problem is to embed maximal classes into maximum classes with at most a linear increase in VC-dimension [KW07, Open Problem 3].

### 3 Preliminaries

#### 3.1 Constructing All Maximum Classes

The aim in this section is to describe an algorithm for constructing all maximum classes of VC dimension  $d$  in the  $n$ -cube. This process can be viewed as the inverse of mapping a maximum class to its  $d$ -maximum projection on  $[n] \setminus \{i\}$  and the corresponding  $(d-1)$ -maximum reduction.

**Definition 9** Let  $C, C' \subseteq \{0, 1\}^n$  be maximum classes of VC-dimensions  $d, d-1$  respectively, so that  $C' \subset C$ , and let  $C_1, C_2 \subset C$  be  $d$ -cubes, i.e.  $d$ -faces of the  $n$ -cube  $\{0, 1\}^n$ .

1.  $C_1, C_2$  are connected if there exists a path in the one-inclusion graph  $\mathcal{G}(C)$  with end-points in  $C_1$  and  $C_2$ ; and
2.  $C_1, C_2$  are said to be  $C'$ -connected if there exists such a connecting path that further does not intersect  $C'$ .

The  $C'$ -connected components of  $C$  are the equivalence classes of the  $d$ -cubes of  $C$  under the  $C'$ -connectedness relation.

The recursive algorithm for constructing all maximum classes of VC-dimension  $d$  in the  $n$ -cube, detailed as Algorithm 1, considers each possible  $d$ -maximum class  $C$  in the  $(n-1)$ -cube and each possible  $(d-1)$ -maximum subclass  $C'$

---

#### Algorithm 1 MAXIMUMCLASSES( $n, d$ )

---

**Given:**  $n \in \mathbb{N}, d \in [n]$

**Returns:** the set of  $d$ -maximum classes in  $\{0, 1\}^n$

1. **if**  $d = 0$  **then return**  $\{\{\mathbf{v}\} \mid \mathbf{v} \in \{0, 1\}^n\}$  ;
  2. **if**  $d = n$  **then return**  $\{0, 1\}^n$  ;
  3.  $\mathcal{M} \leftarrow \emptyset$  ;
  4. **for each**  $C \in \text{MAXIMUMCLASSES}(n-1, d)$ ,  
 $C' \in \text{MAXIMUMCLASSES}(n-1, d-1)$   
s.t.  $C' \subset C$  **do**
  5.  $\{C_1, \dots, C_k\} \leftarrow C'$ -connected components of  $C$  ;  
 $\mathcal{M} \leftarrow \mathcal{M} \cup$   
 $\bigcup_{\mathbf{p} \in \{0, 1\}^k} \left\{ (C' \times \{0, 1\}) \cup \bigcup_{q \in [k]} C_q \times \{p_q\} \right\}$  ;
  6. **done**
  7. **return**  $\mathcal{M}$  ;
- 

of  $C$  as the projection and reduction of a  $d$ -maximum class in the  $n$ -cube, respectively. The algorithm *lifts*  $C$  and  $C'$  to all possible maximum classes in the  $n$ -cube. Then  $C' \times \{0, 1\}$  is contained in each lifted class; so all that remains is to find the tails from the complement of the reduction in the projection. It turns out that each  $C'$ -connected component  $C_i$  of  $C$  can be lifted to either  $C_i \times \{0\}$  or  $C_i \times \{1\}$  arbitrarily and independently of how the other  $C'$ -connected components are lifted. The set of lifts equates to the set of  $d$ -maximum classes in the  $n$ -cube that project-reduce to  $(C, C')$ .

**Lemma 10** MAXIMUMCLASSES( $n, d$ ) (cf. Algorithm 1) returns the set of maximum classes of VC-dimension  $d$  in the  $n$ -cube for all  $n \in \mathbb{N}, d \in [n]$ .

**Proof:** We proceed by induction on  $n$  and  $d$ . The base cases correspond to  $n \in \mathbb{N}, d \in \{0, n\}$  for which all maximum classes, enumerated as singletons in the  $n$ -cube and the  $n$ -cube respectively, are correctly produced by the algorithm. For the inductive step we assume that for  $n \in \mathbb{N}, d \in [n-1]$  all maximum classes of VC-dimension  $d$  and  $d-1$  in the  $(n-1)$ -cube are already known by recursive calls to the algorithm. Given this, we will show that MAXIMUMCLASSES( $n, d$ ) returns only  $d$ -maximum classes in the  $n$ -cube, and that all such classes are produced by the algorithm.

Let classes  $C \in \text{MAXIMUMCLASSES}(n-1, d)$  and  $C' \in \text{MAXIMUMCLASSES}(n-1, d-1)$  be such that  $C' \subset C$ . Then  $C$  is the union of a  $d$ -complete collection and  $C'$  is the union of a  $(d-1)$ -complete collection of cubes that are faces of the cubes of  $C$ . Consider a concept class  $C^*$  formed from  $C$  and  $C'$  by Algorithm 1. The algorithm partitions  $C$  into  $C'$ -connected components  $C_1, \dots, C_k$  each of which is a union of  $d$ -cubes. While  $C'$  is lifted to  $C' \times \{0, 1\}$ , some subset of the components  $\{C_i\}_{i \in S_0}$  are lifted to  $\{C_i \times \{0\}\}_{i \in S_0}$  while the remaining components are lifted to  $\{C_i \times \{1\}\}_{i \notin S_0}$ . By definition  $C^*$  is a  $d$ -complete collection of cubes with cardinality equal to  $\binom{n}{\leq d}$  since  $|C^*| = |C'| + |C|$ , as in [KW07]. So by [RBR08, Theorem 34]  $C^*$  is  $d$ -maximum.

If we now consider any  $d$ -maximum class  $C^* \subseteq \{0, 1\}^n$ , its projection on  $[n] \setminus \{i\}$  is a  $d$ -maximum class  $C \subseteq \{0, 1\}^{n-1}$  and  $C^{*i}$  is the  $(d-1)$ -maximum projection  $C' \subset C$  of all the

$d$ -cubes in  $C^*$  which contain color  $i$ . It is thus clear that  $C^*$  must be obtained by lifting parts of the  $C'$ -connected components of  $C$  to the 1 level and the remainder to the 0 level, and  $C'$  to  $C' \times \{0, 1\}$ . We will now show that if the vertices of each component are not lifted to the same levels, then while the number of vertices in the lift match that of a  $d$ -maximum class in the  $n$ -cube, the number of edges are too few for such a maximum class. Define a lifting operator on  $C$  as  $\ell(v) = \{v\} \times \ell_v$ , where  $\ell_v \subseteq \{0, 1\}$  and

$$|\ell_v| = \begin{cases} 2, & \text{if } v \in C' \\ 1, & \text{if } v \in C \setminus C' \end{cases}.$$

Consider now an edge  $\{u, v\}$  in  $\mathcal{G}(C)$ . By the definition of a  $C'$ -connected component there exists some  $C_j$  such that either  $u, v \in C_j \setminus C'$ ,  $u, v \in C'$  or WLOG  $u \in C_j \setminus C'$ ,  $v \in C'$ . In the first case  $\ell(u) \cup \ell(v)$  is an edge in the lifted graph iff  $\ell_u = \ell_v$ . In the second case  $\ell(u) \cup \ell(v)$  contains four edges and in the last it contains a single edge. Furthermore, it is clear that this accounts for all edges in the lifted graph by considering the projection of an edge in the lifted product. Thus any lift other than those produced by Algorithm 1 induces strictly too few edges for a  $d$ -maximum class in the  $n$ -cube (cf. [KW07, Corollary 7.5]). ■

### 3.2 Corner-Peeling

Kuzmin and Warmuth conjectured in [KW07, Conjecture 2] that their simple *Min-Peeling* procedure is a valid unlabeled compression scheme for maximum classes. Beginning with a concept class  $C_0 = C \subseteq \{0, 1\}^n$ , Min-Peeling operates by iteratively removing a vertex  $v_t$  of minimum-degree in  $\mathcal{G}(C_t)$  to produce the peeled class  $C_{t+1} = C_t \setminus \{v_t\}$ . The concept class corresponding to  $v_t$  is then represented by the dimensions of the edges incident to  $v_t$  in  $\mathcal{G}(C_t)$ ,  $I_{\mathcal{G}(C_t)}(v_t) \subseteq [n]$ . Providing that no-clashing holds for the algorithm, the size of the min-peeling scheme is the largest degree encountered during peeling. Kuzmin and Warmuth predicted that this size is always at most  $d$  for  $d$ -maximum classes. We explore these questions for a related special case of peeling, where we prescribe which vertex to peel at step  $t$  as follows.

**Definition 11** Let  $C \subseteq \{0, 1\}^n$  be a class with  $d = \text{VC}(C)$ . We say that  $C$  can be corner-peeled if there exists an ordering  $v_1, \dots, v_{|C|}$  of the vertices of  $C$  such that, for each  $t \in [|C|]$  where  $C_0 = C$ ,

1.  $v_t \in C_{t-1}$  and  $C_t = C_{t-1} \setminus \{v_t\}$ ;
2. There exists a unique cube  $C'_{t-1}$  of maximum dimension over all cubes in  $C_{t-1}$  containing  $v_t$ ;
3. The neighbors  $\Gamma(v_t)$  of  $v_t$  in  $\mathcal{G}(C_{t-1})$  satisfy  $\Gamma(v_t) \subseteq C'_{t-1}$ ; and
4.  $C_{|C|} = \emptyset$ .

The  $v_t$  are termed the corner vertices of  $C_{t-1}$  respectively.

Note that we do not constrain the cubes  $C'_t$  to be of non-increasing dimension. It turns out that an important property of maximum classes is invariant to this kind of peeling.

**Definition 12** We call a class  $C \subseteq \{0, 1\}^n$  shortest-path closed if for any  $u, v \in C$ ,  $\mathcal{G}(C)$  contains a path connecting  $u, v$  of length  $\|u - v\|_1$ .

**Lemma 13** If  $C \subseteq \{0, 1\}^n$  is shortest-path closed and  $v \in C$  is a corner vertex of  $C$ , then  $C \setminus \{v\}$  is shortest-path closed.

**Proof:** Consider a shortest-path closed  $C \subseteq \{0, 1\}^n$ . Let  $c$  be a corner vertex of  $C$ , and denote the cube of maximum dimension in  $C$ , containing  $c$ , by  $C'$ . Consider  $\{u, v\} \subseteq C \setminus \{c\}$ . By assumption there exists a  $u$ - $v$ -path  $p$  of length  $\|u - v\|_1$  contained in  $C$ . If  $c$  is not in  $p$  then  $p$  is contained in the peeled product  $C \setminus \{c\}$ . If  $c$  is in  $p$  then  $p$  must cross  $C'$  such that there is another path of the same length which avoids  $c$ , and thus  $C \setminus \{c\}$  is shortest-path closed. ■

#### 3.2.1 Corner-Peeling implies Compression

**Theorem 14** If a maximum class  $C$  can be corner-peeled then  $C$  can be  $d$ -unlabeled compressed.

**Proof:** The invariance of the shortest-path closed property under corner-peeling is key. The corner-peeling unlabeled compression scheme represents each  $v_t \in C$  by  $r(v_t) = I_{\mathcal{G}(C_{t-1})}(v_t)$ , the colors of the cube  $C'_{t-1}$  which is deleted from  $C_{t-1}$  when  $v_t$  is corner-peeled. We claim that any two vertices  $v_s, v_t \in C$  have non-clashing representatives. WLOG, suppose that  $s < t$ . The class  $C_{s-1}$  must contain a shortest  $v_s$ - $v_t$ -path  $p$ . Let  $i$  be the color of the single edge contained in  $p$  that is incident to  $v_s$ . Color  $i$  appears once in  $p$ , and is contained in  $r(v_s)$ . This implies that  $v_{s,i} \neq v_{t,i}$  and that  $i \in r(v_s) \cup r(v_t)$ , and so  $v_s | (r(v_s) \cup r(v_t)) \neq v_t | (r(v_s) \cup r(v_t))$ . By construction,  $r(\cdot)$  is a bijection between  $C$  and all subsets of  $[n]$  of cardinality  $\leq \text{VC}(C)$ . ■

If the oriented one-inclusion graph, with each edge directed away from the incident vertex represented by the edge's color, has no cycles, then that representation's compression scheme is termed *acyclic* [Flo89, BDL98, KW07].

**Proposition 15** All corner-peeling unlabeled compression schemes are acyclic.

**Proof:** We follow the proof that the Min-Peeling Algorithm is acyclic [KW07]. Let  $v_1, \dots, v_{|C|}$  be a corner vertex ordering of  $C$ . As a corner vertex  $v_t$  is peeled, its unoriented incident edges are oriented away from  $v_t$ . Thus all edges incident to  $v_1$  are oriented away from  $v_1$  and so the vertex cannot take part in any cycle. For  $t > 1$  assume  $V_t = \{v_s \mid s < t\}$  is disjoint from all cycles. Then  $v_t$  cannot be contained in a cycle, as all incoming edges into  $v_t$  are incident to some vertex in  $V_t$ . Thus the oriented  $\mathcal{G}(C)$  is indeed acyclic. ■

#### 3.3 Boundaries of Maximum Classes

We now turn to the geometric boundaries of maximum classes, which are closely related to corner-peeling.

**Definition 16** The boundary  $\partial C$  of a  $d$ -maximum class  $C$  is defined as all the  $(d - 1)$ -subcubes which are the faces of a single  $d$ -cube in  $C$ .

Maximum classes, when viewed as cubical complexes, are analogous to soap films (an example of a minimal energy surface encountered in nature), which are obtained when a wire frame is dipped into a soap solution. Under this analogy the boundary corresponds to the wire frame and the number

of  $d$ -cubes can be considered the area of the soap film. An important property of the boundary of a maximum class is that all lifted reductions meet the boundary multiple times.

**Theorem 17** *Every  $d$ -maximum class has boundary containing at least two  $(d - 1)$ -cubes of every combination of  $d - 1$  colors, for all  $d > 1$ .*

**Proof:** We use the lifting construction of Section 3.1. Let  $C^* \subseteq \{0, 1\}^n$  be a 2-maximum class and consider color  $i \in [n]$ . Then the reduction  $C^{*i}$  is an unrooted tree with at least two leaves, each of which lifts to an  $i$ -colored edge in  $C^*$ . Since the leaves are of degree 1 in  $C^{*i}$ , the corresponding lifted edges belong to exactly one 2-cube in  $C^*$  and so lie in  $\partial C^*$ . Consider now a  $d$ -maximum class  $C^* \subseteq \{0, 1\}^n$  for  $d > 2$ , and make the inductive assumption that the projection  $C = \Pi_{[n-1]}(C^*)$  has two of each type of  $(d - 1)$ -cube, and that the reduction  $C' = C^{*n}$  has two of each type of  $(d - 2)$ -cube, in their boundaries. Pick  $d - 1$  colors  $I \subseteq [n]$ . If  $n \in I$  then consider two  $(d - 2)$ -cubes colored by  $I \setminus \{x_n\}$  in  $\partial C'$ . By the same argument as in the base case, these lift to two  $I$ -colored cubes in  $\partial C^*$ . If  $n \notin I$  then  $\partial C$  contains two  $I$ -colored  $(d - 1)$ -cubes. For each cube, if the cube is contained in  $C'$  then it has two lifts one of which is contained in  $\partial C^*$ , otherwise its unique lift is contained in  $\partial C^*$ . Therefore  $\partial C^*$  contains at least two  $I$ -colored cubes. ■

Having a large boundary is an important property of maximum classes that does not follow from contractibility.

**Example 18** *Take a 2-simplex with vertices  $A, B, C$ . Glue the edges  $AB$  to  $AC$  to form a cone. Next glue the end loop  $BC$  to the edge  $AB$ . The result is a complex  $D$  with a single vertex, edge and 2-simplex, which is classically known as the dunce hat. It is not hard to verify that  $D$  is contractible, but has no (geometric) boundary.*

Although Theorem 17 will not be explicitly used in the sequel, we return to boundaries of maximum complexes later.

## 4 Euclidean Arrangements

**Definition 19** *A linear arrangement is a collection of  $n \geq d$  oriented hyperplanes in  $\mathbb{R}^d$ . Each region or cell in the complement of the arrangement is naturally associated with a concept in  $\{0, 1\}^n$ ; the side of the  $i^{\text{th}}$  hyperplane on which a cell falls determines the concept's  $i^{\text{th}}$  component. A simple arrangement is one in which any subset of  $d$  planes has a unique point in common and all subsets of  $d + 1$  planes have an empty mutual intersection. Moreover any subset of  $k < d$  planes meet in a plane of dimension  $n - k$ . Such a collection of  $n$  planes is also said to be in general position.*

Many of the familiar operations on concept classes in the  $n$ -cube have elegant analogues on arrangements.

- Projection on  $[n] \setminus \{i\}$  corresponds to removing the  $i^{\text{th}}$  plane;
- The reduction  $C^i$  is the new arrangement given by the intersection of the arrangement with the  $i^{\text{th}}$  plane; and

- The corresponding lifted reduction is the collection of cells in the arrangement that adjoin the  $i^{\text{th}}$  plane.

A  $k$ -cube in the one-inclusion graph corresponds to a collection of  $2^k$  cells, all having a common  $(n - k + 1)$ -face, which is contained in the intersection of  $k$  planes, and an edge corresponds to a pair of cells which have a common face on a single plane. The following result is due to [Ede87].

**Lemma 20** *The concept class  $C \subseteq \{0, 1\}^n$  induced by a simple linear arrangement of  $n$  planes in  $\mathbb{R}^d$  is  $d$ -maximum.*

**Proof:** Note that  $C$  has VC-dimension at most  $d$ , since general position is invariant to projection i.e. no  $d + 1$  planes are shattered. Since  $C$  is the union of a  $d$ -complete collection of cubes (every cell contains  $d$ -intersection points in its boundary) it follows that  $C$  is  $d$ -maximum by [RBR08]. ■

**Corollary 21** *Let  $A$  be a simple linear arrangement of  $n$  hyperplanes in  $\mathbb{R}^d$  with corresponding  $d$ -maximum  $C \subseteq \{0, 1\}^n$ . The intersection of  $A$  with a generic hyperplane corresponds to a  $(d - 1)$ -maximum class  $C' \subseteq C$ . In particular if all  $d$ -intersection points of  $A$  lie to one side of the generic hyperplane, then  $C'$  lies on the boundary of  $C$ ; and  $\partial C$  is the disjoint union of two  $(d - 1)$ -maximum sub-classes.*

**Proof:** The intersection of  $A$  with a generic hyperplane is again a simple arrangement of  $n$  hyperplanes but now in  $\mathbb{R}^{d-1}$ . Hence by Lemma 20  $C'$  is a  $(d - 1)$ -maximum class in the  $n$ -cube.  $C' \subseteq C$  since the adjacency relationships on the cells of the intersection are inherited from those of  $A$ .

Suppose that all  $d$ -intersections in  $A$  lie in one half-space of the generic hyperplane.  $C'$  is the union of a  $(d - 1)$ -complete collection. We claim that each of these  $(d - 1)$ -cubes is a face of exactly one  $d$ -cube in  $C$  and is thus in  $\partial C$ . A  $(d - 1)$ -cube in  $C'$  corresponds to a line in  $A$  where  $d - 1$  planes mutually intersect. The  $(d - 1)$ -cube is a face of a  $d$ -cube in  $C$  iff this line is further intersected by a  $d^{\text{th}}$  plane. This occurs for exactly one plane, which is closest to the generic hyperplane. For once the  $d$ -intersection point is reached, when following along the line away from the generic plane, a new cell is entered. This verifies the second part of the result.

Consider two parallel generic hyperplanes  $h_1, h_2$  such that all  $d$ -intersection points of  $A$  lie in between them. We claim that each  $(d - 1)$ -cube in  $\partial C$  is in exactly one of the concept classes induced by the intersection of  $A$  with  $h_1$  and  $A$  with  $h_2$ . Consider an arbitrary  $(d - 1)$ -cube in  $\partial C$ . As before this cube corresponds to a region of a line formed by a mutual intersection of  $d - 1$  planes. Moreover this region is a ray, with one end-point at a  $d$ -intersection. Because the ray begins at a point between the generic hyperplanes  $h_1, h_2$ , it follows that the ray must cross exactly one of these. ■

**Corollary 22** *Let  $A$  be a simple linear arrangement of  $n$  hyperplanes in  $\mathbb{R}^d$  and let  $C \subseteq \{0, 1\}^n$  be the corresponding  $d$ -maximum class. Then  $C$  considered as a cubical complex is homeomorphic to the  $d$ -ball  $B^d$ ; and  $\partial C$  considered as a  $(d - 1)$ -cubical complex is homeomorphic to the  $(d - 1)$ -sphere  $S^{d-1}$ .*

	$x_1$	$x_2$	$x_3$	$x_4$
$v_0$	0	0	0	0
$v_1$	1	0	0	0
$v_2$	0	1	0	0
$v_3$	0	0	1	0
$v_4$	1	0	1	0
$v_5$	1	1	0	0
$v_6$	0	1	1	0
$v_7$	1	0	0	1
$v_8$	1	1	0	1
$v_9$	0	1	0	1
$v_{10}$	0	1	1	1

Figure 1: A 2-maximum class in  $\{0, 1\}^4$  having a simple linear line arrangement in  $\mathbb{R}^2$ .

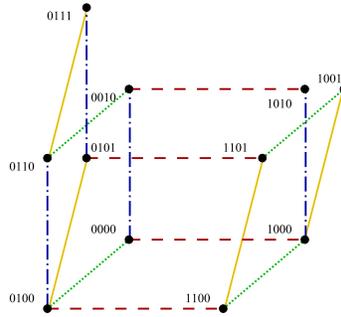


Figure 2: The 2-maximum class in the 4-cube, enumerated in Figure 1.

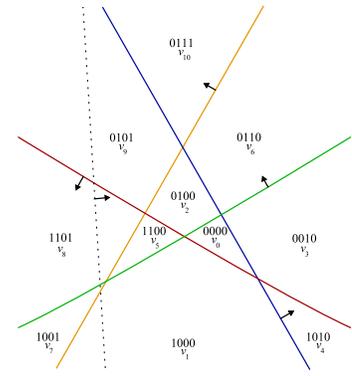


Figure 3: A simple linear line arrangement corresponding to the class in Figure 1, swept by the dashed line. Each cell has a unique vertex.

**Proof:** We construct a Voronoi cell decomposition corresponding to the set of  $d$ -intersection points inside a very large ball in Euclidean space. By induction on  $d$ , we claim that this is a cubical complex and the vertices and edges correspond to the class  $C$ . By induction, on each hyperplane, the induced arrangement has a Voronoi cell decomposition which is a  $(d - 1)$ -cubical complex with edges and vertices matching the one-inclusion graph for the tail of  $C$  corresponding to the label associated with the hyperplane. It is not hard to see that the Voronoi cell defined by a  $d$ -intersection point  $p$  on this hyperplane is a  $d$ -cube. In fact, its  $(d - 1)$ -faces correspond to the Voronoi cells for  $p$ , on each of the  $d$  hyperplanes passing through  $p$ . We also see that this  $d$ -cube has a single vertex in the interior of each of the  $2^d$  cells of the arrangement adjacent to  $p$ . In this way, it follows that the vertices of this Voronoi cell decomposition are in bijective correspondence to the cells of the hyperplane arrangement. Finally the edges of the Voronoi cells pass through the faces in the hyperplanes. So these correspond bijectively to the edges of  $C$ , as there is one edge for each face of the hyperplanes. Using a very large ball, containing all the  $d$ -intersection points, the boundary faces become spherical cells. In fact, these form a spherical Voronoi cell decomposition, so it is easy to replace these by linear ones by taking the convex hull of their vertices. So a piecewise linear cubical complex  $\mathbf{C}$  is constructed, which has one-skeleton (graph consisting of all vertices and edges) isomorphic to the one-inclusion graph for  $C$ .

Finally we want to prove that  $\mathbf{C}$  is homeomorphic to  $B^d$ . This is quite easy by construction. For we see that  $\mathbf{C}$  is obtained by dividing up  $B^d$  into Voronoi cells and replacing the spherical boundary cells by linear ones, using convex hulls of the boundary vertices. This process is clearly given by a homeomorphism by projection. In fact, the homeomorphism preserves the PL-structure so is a PL homeomorphism. ■

The following example demonstrates that not all maximum classes of VC-dimension  $d$  are homeomorphic to the  $d$ -ball. The key to such examples is branching.

**Example 23** A simple linear arrangement in  $\mathbb{R}$  corresponds

to points on the line—cells are simply intervals between these points and so corresponding 1-maximum classes are sticks. Any tree that is not a stick can therefore not be represented as a simple linear arrangement in  $\mathbb{R}$  and is also not homeomorphic to the 1-ball which is simply the interval  $[-1, 1]$ .

As in Kuzmin & Warmuth [KW07], consider a generic hyperplane  $h$  sweeping across a simple linear arrangement  $A$ .  $h$  begins with all  $d$ -intersection points of  $A$  lying in its positive half-space  $\mathcal{H}_+$ . The concept corresponding to cell  $c$  is peeled from  $C$  when  $|\mathcal{H}_+ \cap c| = 1$ , i.e.  $h$  crosses the last  $d$ -intersection point adjoining  $c$ . At any step in the process, the result of peeling  $j$  vertices from  $C$  to reach  $C_j$ , is captured by the arrangement  $\mathcal{H}_+ \cap A$  for the appropriate  $h$ .

**Example 24** Figure 1 enumerates the 11 vertices of a 2-maximum class in the 4-cube. Figures 3 and 2 display a hyperplane arrangement in Euclidean space and its Voronoi cell decomposition, corresponding to this maximum class. In this case, sweeping the vertical dashed line across the arrangement corresponds to a partial corner-peeling of the concept class with peeling sequence  $v_7, v_8, v_5, v_9, v_2, v_0$ . What remains is the 1-maximum stick  $\{v_1, v_3, v_4, v_6, v_{10}\}$ .

Next we resolve the first half of [KW07, Conjecture 1].

**Theorem 25** Any  $d$ -maximum class  $C \subseteq \{0, 1\}^n$  corresponding to a simple linear arrangement  $A$  can be corner-peeled by sweeping  $A$ , and this process is a valid unlabeled compression scheme for  $C$  of size  $d$ .

**Proof:** We must show that as the  $j^{\text{th}}$   $d$ -intersection point  $p_j$  is crossed, there is a corner vertex of  $C_{j-1}$  peeled away. It then follows that sweeping a generic hyperplane  $h$  across  $A$  corresponds to corner-peeling  $C$  to a  $(d - 1)$ -maximum subclass  $C' \subseteq \partial C$  by Corollary 21. Moreover  $C'$  corresponds to a simple linear arrangement of  $n$  hyperplanes in  $\mathbb{R}^{d-1}$ .

We proceed by induction on  $d$ , noting that for  $d = 1$  corner-peeling is trivial. Consider  $h$  as it approaches the  $j^{\text{th}}$   $d$ -intersection point  $p_j$ . The  $d$  planes defining this point intersect  $h$  in a simple arrangement of hyperplanes on  $h$ . There

is a compact cell  $\Delta$  for the arrangement on  $h$ , which is a  $d$ -simplex<sup>1</sup> and shrinks to a point as  $h$  passes through  $p_j$ . We claim that the cell  $c$  for the arrangement  $A$ , whose intersection with  $h$  is  $\Delta$ , is a corner vertex  $v_j$  of  $C_{j-1}$ . Consider the lines formed by intersections of  $d - 1$  of the  $d$  hyperplanes, passing through  $p_j$ . Each is a segment starting at  $p_j$  and ending at  $h$  without passing through any other  $d$ -intersection points. So all faces of hyperplanes adjacent to  $c$  meet  $h$  in faces of  $\Delta$ . Thus, there are no edges in  $C_{j-1}$  starting at the vertex corresponding to  $p_j$ , except for those in the cube  $C'_{j-1}$ . So  $c$  corresponds to a corner vertex  $v_j$  of the  $d$ -cube  $C'_{j-1}$  in  $C_{j-1}$ . Finally, just after the simplex is a point,  $c$  is no longer in  $\mathcal{H}_+$  and so  $v_j$  is corner-peeled from  $C_{j-1}$ .

Theorem 14 completes the proof that this corner-peeling of  $C$  constitutes unlabeled compression. ■

**Corollary 26** *The sequence of cubes  $C'_0, \dots, C'_{|C|}$ , removed when corner-peeling by sweeping simple linear arrangements, is of non-increasing dimension. In fact, there are  $\binom{n}{d}$  cubes of dimension  $d$ , then  $\binom{n}{d-1}$  cubes of dimension  $d - 1$ , etc.*

While corner-peeling and min-peeling share some properties in common, they are distinct procedures.

**Example 27** *Consider sweeping a simple linear arrangement corresponding to a 2-maximum class. After all but one 2-intersection point has been swept, the corresponding corner-peeled class  $C_t$  is the union of a single 2-cube with a 1-maximum stick. Min-peeling applied to  $C_t$  would first peel a leaf, while corner-peeling must begin with the 2-cube.*

The next result follows from our counter-examples to Kuzmin & Warmuth's minimum degree conjecture [RBR08].

**Corollary 28** *There is no constant  $c$  so that all maximal classes of VC dimension  $d$  can be embedded into maximum classes corresponding to simple hyperplane arrangements of dimension  $d + c$ .*

## 5 Hyperbolic Arrangements

We briefly discuss the Klein model of hyperbolic geometry [Rat94, pg. 7]. Consider the open unit ball  $\mathbb{H}^k$  in  $\mathbb{R}^k$ . Geodesics (lines of shortest length in the geometry) are given by intersections of straight lines in  $\mathbb{R}^k$  with the unit ball. Similarly planes of any dimension between 2 and  $k - 1$  are given by intersections of such planes in  $\mathbb{R}^k$  with the unit ball. Note that such planes are completely determined by their spheres of intersection with the unit sphere  $S^{k-1}$ , which is called the ideal boundary of hyperbolic space  $\mathbb{H}^k$ . Note that in the appropriate metric, the ideal boundary consists of points which are infinitely far from all points in the interior of the unit ball.

We can now see immediately that a simple hyperplane arrangement in  $\mathbb{H}^k$  can be described by taking a simple hyperplane arrangement in  $\mathbb{R}^k$  and intersecting it with the unit ball. However we require an important additional property to mimic the Euclidean case. Namely we add the constraint

<sup>1</sup>A topological simplex—the convex hull of affinely independent  $d + 1$  points in  $\mathbb{R}^d$ .

that every subcollection of  $d$  of the hyperplanes in  $\mathbb{H}^k$  has mutual intersection points inside  $\mathbb{H}^k$ , and that no  $(d + 1)$ -intersection point lies in  $\mathbb{H}^k$ . We need this requirement to obtain that the resulting class is maximum.

**Definition 29** *A simple hyperbolic  $d$ -arrangement is a collection of  $n$  hyperplanes in  $\mathbb{H}^k$  with the property that every sub-collection of  $d$  hyperplanes mutually intersect in a  $(k - d)$ -dimensional hyperbolic plane, and that every sub-collection of  $d + 1$  hyperplanes mutually intersect as the empty set.*

**Corollary 30** *The concept class  $C$  corresponding to a simple  $d$ -arrangement of hyperbolic hyperplanes in  $\mathbb{H}^k$  is  $d$ -maximum in the  $k$ -cube.*

**Proof:** The result follows by the same argument as before. Projection cannot shatter any  $(d + 1)$ -cube and the class is a complete union of  $d$ -cubes, so is  $d$ -maximum. ■

The key to why hyperbolic arrangements represent many new maximum classes is that they allow flexibility of choosing  $d$  and  $k$  independently. This is significant because the unit ball can be chosen to miss much of the intersections of the hyperplanes in Euclidean space. Note that the new maximum classes are embedded in maximum classes induced by arrangements of linear hyperplanes in Euclidean space.

A simple example is any 1-maximum class. It is easy to see that this can be realized in the hyperbolic plane by choosing an appropriate family of lines and the unit ball in the appropriate position. In fact, we can choose sets of pairs of points on the unit circle, which will be the intersections with our lines. So long as these pairs of points have the property that the smaller arcs of the circle between them are disjoint, the lines will not cross inside the disk and the desired 1-maximum class will be represented.

Corner-peeling maximum classes represented by hyperbolic hyperplane arrangements proceeds by sweeping, just as in the Euclidean case. Note first that intersections of the hyperplanes of the arrangement with the moving hyperplane appear precisely when there is a first intersection at the ideal boundary. Thus it is necessary to slightly perturb the collection of hyperplanes to ensure that only one new intersection with the moving hyperplane occurs at any time. Note also that new intersections of the sweeping hyperplane with the various lower dimensional planes of intersection between the hyperplanes appear similarly at the ideal boundary. The important claim to check is that the intersection at the ideal boundary between the moving hyperplane and a lower dimensional plane, consisting entirely of  $d$  intersection points, corresponds to a corner-peeling move. We include two examples to illustrate the validity of this plane.

**Example 31** *In the case of a 1-maximum class coming from disjoint lines in  $\mathbb{H}^2$ , a cell can disappear when the sweeping hyperplane meets a line at an ideal point. This cell is indeed a vertex of the tree, i.e. a corner-vertex.*

**Example 32** *Assume that we have a family of planes in the unit ball which meet in pairs in single lines, but there are no triple points of intersection, corresponding to a 2-maximum class. A corner-peeling move occurs when a region bounded*

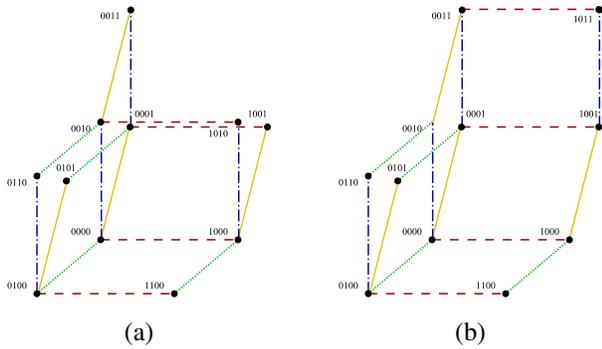


Figure 4: 2-maximum classes in  $\{0, 1\}^4$  that can be represented as hyperbolic arrangements but not as Euclidean arrangements.

by two half disks and an interval disappears, in the positive half space bounded by the sweeping hyperplane. Such a region can be visualized by taking a slice out of an orange. Note that the final point of contact between the hyperplane and the region is at the end of a line of intersection between two planes on the ideal boundary.

We next observe that sweeping by generic hyperbolic hyperplanes induces corner-peeling of the corresponding maximum class, extending Theorem 25. As the generic hyperplane sweeps across hyperbolic space, not only do swept cells correspond to corners of  $d$ -cubes but also to corners of lower dimensional cubes as well. Moreover, the order of the dimensions of the cubes which are corner-peeled can be arbitrary—lower dimensional cubes may be corner-peeled before all the higher dimensional cubes are corner-peeled. This is in contrast to Euclidean sweepouts (cf. Corollary 26). Similar to Euclidean sweepouts, hyperbolic sweepouts correspond to corner-peeling and not min-peeling.

**Theorem 33** Any  $d$ -maximum class  $C \subseteq \{0, 1\}^n$  corresponding to a simple hyperbolic  $d$ -arrangement  $A$  can be corner-peeled by sweeping  $A$  with a generic hyperbolic hyperplane.

**Proof:** We follow the same strategy of the proof of Theorem 25. For sweeping in hyperbolic space  $\mathbb{H}^k$ , the generic hyperplane  $h$  is initialized as tangent to  $\mathbb{H}^k$ . As  $h$  is swept across  $\mathbb{H}^k$ , new intersections appear with  $A$  just after  $h$  meets the non-empty intersection of a subset of hyperplanes of  $A$  with the ideal boundary. Each  $d$ -cube  $C'$  in  $C$  still corresponds to the cells adjacent to the intersection  $I_{C'}$  of  $d$  hyperplanes. But now  $I_{C'}$  is a  $(k - d)$ -dimensional hyperbolic hyperplane. A cell  $c$  adjacent to  $I_{C'}$  is corner-peeled precisely when  $h$  last intersects  $c$  at a point of  $I_{C'}$  at the ideal boundary. As for simple linear arrangements, the general position of  $A \cup \{h\}$  ensures that corner-peeling events never occur simultaneously. For the case  $k = d + 1$ , as for the simple linear arrangements just prior to the corner-peeling of  $c$ ,  $\mathcal{H}_+ \cap c$  is homeomorphic to a  $k$ -simplex with a missing face on the ideal boundary. And so as in the simple linear case, this  $d$ -intersection point corresponds to a corner  $d$ -cube. In the case  $k > d + 1$ ,  $\mathcal{H}_+ \cap c$  becomes a simplex as before

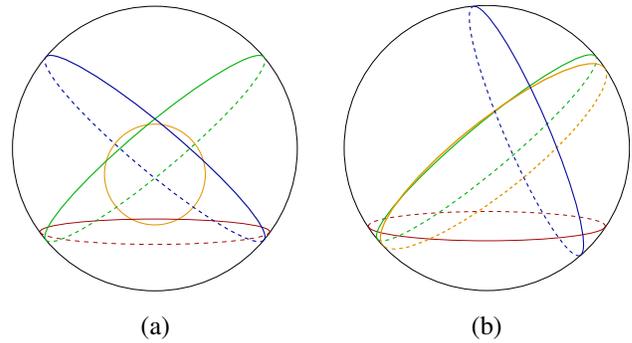


Figure 5: Hyperbolic hyperplane arrangements corresponding to the classes in Figure 4. In both cases the four hyperbolic planes meet in 6 straight line segments (not shown). The planes' colors correspond to the edges' colors in Figure 4.

multiplied by  $\mathbb{R}^{k-d-1}$ . If  $k = d$ , then the main difference is just before corner-peeling of  $c$ ,  $\mathcal{H}_+ \cap c$  is homeomorphic to a  $k$ -simplex which may be either closed or with a missing face on the ideal boundary. The rest of the argument remains the same, except for one important observation.

Although swept corners in hyperbolic arrangements can be of cubes of differing dimensions, these dimensions never exceed  $d$  and so the proof that sweeping simple linear arrangements induces  $d$ -compression schemes is still valid. ■

**Example 34** Constructed with lifting, Figure 4 completes the enumeration, up to symmetry, of the 2-maximum classes in  $\{0, 1\}^4$  begun with Example 24. These cases cannot be represented as simple Euclidean linear arrangements, since their boundaries do not satisfy the condition of Corollary 22 but can be represented as hyperbolic arrangements as in Figure 5. Figures 6 and 7 display the sweeping of a general hyperplane across the former arrangement and the corresponding corner-peeling. Notice that the corner-peeled cubes' dimensions decrease and then increase.

**Corollary 35** There is no constant  $c$  so that all maximal classes of VC dimension  $d$  can be embedded into maximum classes corresponding to simple hyperbolic hyperplane arrangements of VC dimension  $d + c$ .

This result follows from our counter-examples to Kuzmin & Warmuth's minimum degree conjecture [RBR08].

Corollary 30 gives a proper superset of simple linear hyperplane arrangement-induced maximum classes as hyperbolic arrangements. We will prove in the next section that all maximum classes can be represented as PL hyperplane arrangements in a ball. These are the topological analogue of hyperbolic hyperplane arrangements. If the boundary of the ball is removed, then we obtain an arrangement of PL hyperplanes in Euclidean space.

## 6 Infinite Euclidean and Hyperbolic Arrangements

We consider a simple example of an infinite maximum class which admits corner-peeling and a compression scheme analogous to those of previous sections.

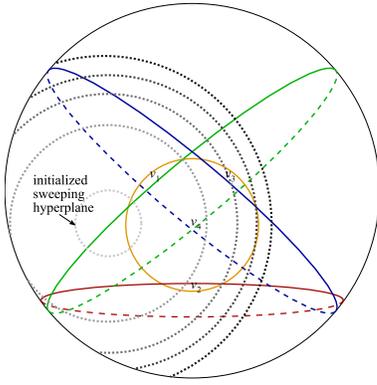


Figure 6: The simple hyperbolic arrangement of Figure 5.(a) with a generic sweeping hyperplane shown in several positions before and after it sweeps past four cells.

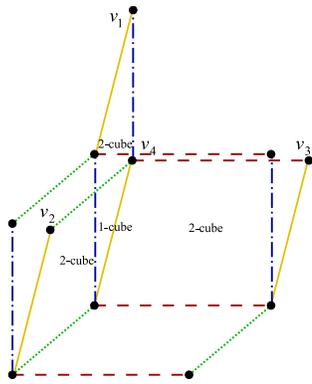


Figure 7: The 2-maximum class in  $\{0, 1\}^4$  of Figure 4.(a), with the first four corner-vertices peeled by the hyperbolic arrangement sweeping of Figure 6. Notice that three 2-cubes are peeled, then a 1-cube (all shown) followed by 2-cubes.

**Example 36** Let  $\mathcal{L}$  be the set of lines in the plane of the form  $L_{2m} = \{(x, y) \mid x = 2m\}$  and  $L_{2n+1} = \{(x, y) \mid y = 2n\}$  for  $m, n \in \mathbb{N}$ . Let  $v_{00}, v_{0n}, v_{m0},$  and  $v_{mn}$  be the cells bounded by the lines  $\{L_2, L_3\}, \{L_2, L_{2n+1}, L_{2n+3}\}, \{L_{2m}, L_{2m+2}, L_3\},$  and  $\{L_{2m}, L_{2m+2}, L_{2n+1}, L_{2n+3}\},$  respectively. Then the cubical complex  $C,$  with vertices  $v_{mn},$  can be corner-peeled and hence compressed, using a sweep-out by the lines  $\{(x, y) \mid x + (1 + \epsilon)y = t\}$  for  $t \geq 0$  and any small fixed irrational  $\epsilon > 0.$   $C$  is a 2-maximum class and the unlabeled compression scheme is also of size 2.

To verify the properties of this example, notice that sweeping as specified corresponds to corner-peeling the vertex  $v_{00},$  then the vertices  $v_{10}, v_{01},$  then the remaining vertices  $v_{mn}$  in order of increasing  $m + n.$  The lines  $x + (1 + \epsilon)y = t$  are generic as they pass through only one intersection point of  $\mathcal{L}$  at a time. Additionally, representing  $v_{00}$  by  $\emptyset, v_{0n}$  by  $\{L_{2n+1}\}, v_{m0}$  by  $\{L_{2m}\}$  and  $v_{mn}$  by  $\{L_{2m}, L_{2n+1}\}$  constitutes a valid unlabeled compression scheme. Note that the compression scheme is associated with sweeping across the arrangement in the direction of decreasing  $t.$  This is necessary to pick up the boundary vertices of  $C$  last in the sweep-

out process, so that they have either singleton representatives or the empty set. In this way, as in [KW07], we obtain a compression scheme so that every labeled sample of size 2 is associated with a unique concept in  $C,$  which is consistent with the sample. On the other hand to obtain corner-peeling, we need the sweepout to proceed with  $t$  increasing so that we can begin at the boundary vertices of  $C.$

In concluding this brief discussion, we note that many infinite collections of simple hyperbolic hyperplanes and Euclidean hyperplanes can also be corner-peeled and compressed, even if intersection points and cells accumulate. However a key requirement in the Euclidean case is that the concept class  $C$  has a non-empty boundary, when considered as a cubical complex. An easy approach is to assume that all the  $d$ -intersections of the arrangement lie in a half-space. Moreover, since the boundary must also admit corner-peeling, we require more conditions, similar to having all the intersection points lying in an octant.

**Example 37** In  $\mathbb{R}^3,$  choose the family of planes  $\mathcal{P}$  of the form  $P_{3n+i} = \{\mathbf{x} \in \mathbb{R}^3 \mid x_{i+1} = 1 - 1/n\}$  for  $n \geq 1$  and  $i \in \{0, 1, 2\}.$  A corner-peeling scheme is induced by sweeping a generic plane  $\{\mathbf{x} \in \mathbb{R}^3 \mid x_1 + \alpha x_2 + \beta x_3 = t\}$  across the arrangement, where  $t$  is a parameter and close to 1,  $\alpha, \beta$  are algebraically independent and  $\alpha, \beta$  are both close to 1. This example has similar properties to Example 36: the compression scheme is again given by decreasing  $t$  whereas corner-peeling corresponds to increasing  $t.$  Note that cells shrink to points, as  $\mathbf{x} \rightarrow \mathbf{1}$  and the volume of cells converge to zero as  $n \rightarrow \infty,$  or equivalently any  $x_i \rightarrow 1.$

**Example 38** In the hyperbolic plane  $\mathbb{H}^2,$  choose the family of lines  $\mathcal{L}$  given by  $L_{2n} = \{(x, y) \mid x = 1 - 1/n\}$  and  $L_{2n+1} = \{(x, y) \mid x + ny = 1\},$  for  $n \geq 1.$  This arrangement has corner-peeling and compression schemes given by sweeping across  $\mathcal{L}$  using the generic line  $\{y = t\}.$

## 7 Piecewise-Linear Arrangements

A PL hyperplane is the image of a proper piecewise-linear homeomorphism from the  $(k - 1)$ -ball  $B^{k-1}$  into  $B^k,$  i.e. the inverse image of  $S^k$  is  $S^{k-1}$  [RS82]. A simple PL  $d$ -arrangement is an arrangement of  $n$  PL hyperplanes such that every subcollection of  $j$  hyperplanes meet transversely in a  $(k - j)$ -dimensional PL plane for  $2 \leq j \leq d$  and every subcollection of  $d + 1$  hyperplanes are disjoint.

### 7.1 Maximum Classes are Represented by Simple PL Hyperplane Arrangements

Our aim is to prove the following theorem by a series of steps.

**Theorem 39** Every  $d$ -maximum class  $C \subseteq \{0, 1\}^n$  can be represented by a simple arrangement of PL hyperplanes in an  $n$ -ball. Moreover the corresponding simple arrangement of PL hyperspheres in the  $(n - 1)$ -sphere also represents  $C,$  so long as  $n > d + 1.$

#### 7.1.1 Embedding a $d$ -Maximum Cubical Complex in the $n$ -cube into an $n$ -ball.

We begin with a  $d$ -maximum cubical complex  $C \subseteq \{0, 1\}^n$  embedded into  $[0, 1]^n.$  This gives a natural embedding of  $C$

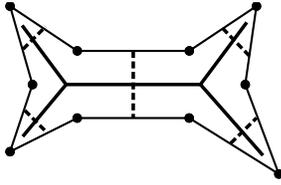


Figure 8: A 1-maximum class (thick solid lines) with its fattening (thin solid lines with points), bisecting sets (dashed lines) and induced complementary cells.

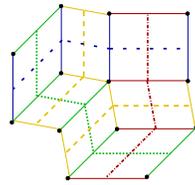


Figure 9: The top of Figure 4.(b) (i.e. the 2-cubes seen from above) gives part of the boundary of a regular neighborhood in  $\mathbb{R}^3$ .

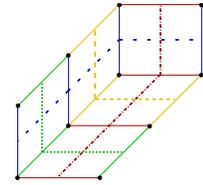


Figure 10: The bottom of Figure 4.(b) (i.e. the 2-cubes seen from below) gives the rest of the boundary of a regular neighborhood.

into  $\mathbb{R}^n$ . Take a small regular neighborhood  $\mathcal{N}$  of  $C$  so that the boundary  $\partial\mathcal{N}$  of  $\mathcal{N}$  will be a closed manifold of dimension  $n - 1$ . Note that  $\mathcal{N}$  is contractible because it collapses onto  $C$  and so  $\partial\mathcal{N}$  is a homology  $(n - 1)$ -sphere (by a standard, well-known argument from topology [Maz61]). Our aim is to prove that  $\partial\mathcal{N}$  is an  $(n - 1)$ -sphere and  $\mathcal{N}$  is an  $n$ -ball. There are two ways of proving this: show that  $\partial\mathcal{N}$  is simply connected and invoke the well-known solution to the generalised Poincaré conjecture [Sma61], or use the cubical structure of the  $n$ -cube and  $C$  to directly prove the result. We adopt the latter approach, although the former works fine. The advantage of the latter is that it produces the required hyperplane arrangement, not just the structures of  $\partial\mathcal{N}$  and  $\mathcal{N}$ .

### 7.1.2 Bisecting Sets

For each color  $i$ , there is a hyperplane  $P_i$  in  $\mathbb{R}^n$  consisting of all vectors with  $i^{\text{th}}$  coordinate equal to  $1/2$ . We can easily arrange the choice of regular neighborhood  $\mathcal{N}$  of  $C$  so that  $\mathcal{N}_i = P_i \cap \mathcal{N}$  is a regular neighborhood of  $C \cap P_i$  in  $P_i$ . (We call  $\mathcal{N}_i$  a *bisecting set* as it intersects  $C$  along the ‘center’ of the reduction in the  $i^{\text{th}}$  coordinate direction, see Figure 8.) But then since  $C \cap P_i$  is a cubical complex corresponding to the reduction  $C^i$ , by induction on  $n$ , we can assert that  $\mathcal{N}_i$  is an  $(n - 1)$ -ball. Similarly the intersections  $\mathcal{N}_i \cap \mathcal{N}_j$  can be arranged to be regular neighborhoods of  $(d - 2)$ -maximum classes and are also balls of dimension  $n - 2$ , etc. In this way, we see that if we can show that  $\mathcal{N}$  is an  $n$ -ball, then the induction step will be satisfied and we will have produced a PL hyperplane arrangement in a ball.

### 7.1.3 Shifting

To complete the induction step, we use the technique of shifting, [Alo83], [Fra83], [Hau95]. In our situation, this can be viewed as the converse of lifting. Namely if a color  $i$  is chosen, then the cubical complex  $C$  has a lifted reduction  $C'$  consisting of all  $d$ -cubes containing the  $i^{\text{th}}$  color. By shifting, we can move down any of the lifted components, obtained by splitting  $C$  open along  $C'$ , from the level  $x_i = 1$  to the level  $x_i = 0$ , to form a new cubical complex  $C^*$ . We claim that the regular neighborhood of  $C$  is a ball if and only if the same is true for  $C^*$ . But this is quite straightforward, since the operation of shifting can be thought of as sliding components of  $C$ , split open along  $C'$ , continuously from level  $x_i = 1$  to  $x_i = 0$ . So there is an isotopy of the attaching maps of the components onto the lifted reduction, using the product structure of the latter. It is easy then to check that

this does not affect the homeomorphism type of the regular neighborhood and so the claim of shift invariance is proved.

But repeated shifting finishes with the downwards closed maximum class consisting of all vertices in the  $n$ -cube with at most  $d$  coordinates being one and the remaining coordinates all being zero. It is easy to see that the corresponding cubical complex  $\tilde{C}$  is star-like, i.e. contains all the straight line segments from the origin to any point in  $\tilde{C}$ . If we choose a regular neighborhood  $\tilde{\mathcal{N}}$  to also be star-like, then it is obvious that  $\tilde{\mathcal{N}}$  is an  $n$ -ball. Hence our induction is complete and we have shown that any  $d$ -maximum class in  $\{0, 1\}^n$  can be represented by a family of PL hyperplanes in the  $n$ -ball.

### 7.1.4 Ideal Boundary

To complete the proof of Theorem 39, let  $\partial\mathcal{N} = S^{n-1}$  denote the boundary of the  $n$ -ball  $\mathcal{N}$  constructed above (cf. Figures 9 and 10). Each PL hyperplane intersects this sphere in a PL hypersphere of dimension  $n - 2$ . It remains to show this arrangement of hyperspheres gives the same cubical complex as  $C$ , unless  $n = d + 1$ .

Suppose that  $n > d + 1$ . Then it is easy to see that each cell  $c$  in the complement of the PL hyperplane arrangement in  $\mathcal{N}$  has part of its boundary on the ideal boundary  $\partial\mathcal{N}$ . Let  $\partial c = \partial c_+ \cup \partial c_-$ , where  $\partial c_+$  is the intersection of  $c$  with the ideal boundary and  $\partial c_-$  is the closure of  $\partial c \setminus \partial c_+$ .

It is now straightforward to verify that the face structure of  $\partial c_+$  is equivalent to the face structure of  $\partial c_-$ . Note that any family of at most  $d$  PL hyperplanes meet in a ball properly embedded in  $\mathcal{N}$ . Since  $n > d + 1$ , the smallest dimension of such a ball is two, and hence its boundary is connected. Then  $\partial c_-$  has faces which are balls obtained in this way of dimension varying between  $n - d$  and  $n - 1$ . Each of these faces has boundary a sphere which is a face of  $\partial c_+$ . So this establishes a bijection between the faces of  $\partial c_+$  and those of  $\partial c_-$ . It is easy to check that the cubical complexes corresponding to the PL hyperplanes and to the PL hyperspheres are the same.

Note that if  $n = d + 1$ , then any  $d$ -maximum class  $C \subseteq \{0, 1\}^{d+1}$  is obtained by taking all the  $d$ -faces of the  $(d + 1)$ -cube which contain a particular vertex. So  $C$  is a  $d$ -ball and the ideal boundary of  $\mathcal{N}$  is a  $d$ -sphere. The cubical complex associated with the ideal boundary is the double  $2C$  of  $C$ , i.e. two copies of  $C$  glued together along their boundaries. The proof of Theorem 39 is now complete.

**Example 40** Consider the bounded below 2-maximum class  $\tilde{C} \subseteq \{0, 1\}^5$ . We claim that  $\tilde{C}$  cannot be realized as an ar-

angement of PL hyperplanes in the 3-ball  $B^3$ . Note that our method gives  $\tilde{C}$  as an arrangement in  $B^5$  and this example shows that  $B^4$  is the best one might hope for in terms of dimension of the hyperplane arrangement.

For suppose that  $\tilde{C}$  could be realized by any PL hyperplane arrangement in  $B^3$ . Then clearly we can also embed  $\tilde{C}$  into  $B^3$ . The vertex  $v_0 = \{0\}^5$  has link given by the complete graph  $K$  on 5 vertices in  $\tilde{C}$ . (By link, we mean the intersection of the boundary of a small ball in  $B^3$  centered at  $v_0$  with  $\tilde{C}$ .) But as is well known,  $K$  is not planar, i.e. cannot be embedded into the plane or 2-sphere. This contradiction shows that no such arrangement is possible.

## 7.2 Maximum Classes with Manifold Cubical Complexes

We prove a partial converse to Corollary 22: if a  $d$ -maximum class has a ball as cubical complex, then it can always be realized by a simple PL hyperplane arrangement in  $\mathbb{R}^d$ .

**Theorem 41** *Suppose that  $C \subseteq \{0, 1\}^n$  is a  $d$ -maximum class. Then the following properties of  $C$ , considered as a cubical complex, are equivalent:*

- (i) *There is a simple arrangement  $A$  of  $n$  PL hyperplanes in  $\mathbb{R}^d$  which represents  $C$ .*
- (ii)  *$C$  is homeomorphic to the  $d$ -ball.*
- (iii)  *$C$  is a  $d$ -manifold with boundary.*

**Proof:** To prove (i) implies (ii), we can use exactly the same argument as Corollary 22. Next (ii) trivially implies (iii). So it remains to show that (iii) implies (i). The proof proceeds by double induction on  $n, d$ . The initial cases where either  $d = 1$  or  $n = 1$  are very easy.

Assume that  $C$  is a manifold. Let  $p$  denote the  $i^{\text{th}}$  coordinate projection. Then  $p(C)$  is obtained by collapsing  $C^i \times [0, 1]$  onto  $C^i$ , where  $C^i$  is the reduction. As before, let  $P_i$  be the linear hyperplane in  $\mathbb{R}^n$ , where the  $i^{\text{th}}$  coordinate takes value  $1/2$ . Viewing  $C$  as a manifold embedded in the  $n$ -cube, since  $P_i$  intersects  $C$  transversely, we see that  $C^i \times \{1/2\}$  is a proper submanifold of  $C$ . But it is easy to check that collapsing  $C^i \times [0, 1]$  to  $C^i$  in  $C$  produces a new manifold which is again homeomorphic to  $C$ . (The product region  $C^i \times [0, 1]$  in  $C$  can be expanded to a larger product region and so collapsing shrinks the larger region to one of the same homeomorphism type). So we conclude that the projection  $p(C)$  is also a manifold. By induction on  $n$ , it follows that there is a PL hyperplane arrangement  $A$ , consisting of  $n - 1$  PL hyperplanes in  $B^d$ , which represents  $p(C)$ .

Next, observe that the reduction  $C^i$  can be viewed as a properly embedded submanifold  $M$  in  $B^d$ , where  $M$  is a union of some of the  $(d - 1)$ -dimensional faces of the Voronoi cell decomposition corresponding to  $A$ , described in Corollary 22. By induction on  $d$ , we conclude that  $C^i$  is also represented by  $n$  PL hyperplanes in  $B^{d-1}$ . But then since condition (i) implies (ii), it follows that  $M$  is PL homeomorphic to  $B^{d-1}$ , since the underlying cubical complex for  $C^i$  is a  $(d - 1)$ -ball. So it follows that  $A \cup \{M\}$  is a PL hyperplane arrangement in  $B^d$  representing  $C$ . This completes the proof that condition (iii) implies (i). ■

## 8 Corner-Peeling 2-Maximum Classes

**Theorem 42** *Every 2-maximum class can be corner-peeled.*

**Proof:** By Theorem 39, we can represent any 2-maximum class  $C \subseteq \{0, 1\}^n$  by a simple family of hyperspheres  $\{S_i\}$  in  $S^{n-1}$ . Every pair of hyperspheres  $S_i, S_j$  intersects in an  $(n - 3)$ -sphere  $S_{ij}$  and there are no intersection points between any three of these hyperspheres. Consider the family of spheres  $S_{ij}$ , for  $i$  fixed. These are disjoint hyperspheres in  $S_i$  so we can choose an innermost one  $S_{ik}$  which bounds an  $(n - 2)$ -ball  $B_1$  in  $S_i$  not containing any other of these spheres. Moreover there are two balls  $B_2, B_3$  bounded by  $S_{ik}$  on  $S_k$ . We call the two  $(n - 1)$ -balls  $Q_2, Q_3$  bounded by  $B_1 \cup B_2, B_1 \cup B_3$  respectively in  $S^{n-1}$ , which intersect only along  $B_1$ , *quadrants*.

Assume  $B_2$  is innermost on  $S_k$ . Then the quadrant  $Q_2$  has both faces  $B_1, B_2$  innermost. It is easy to see that such a quadrant corresponds to a corner vertex in  $C$  which can be peeled. Moreover, after peeling, we still have a family of PL hyperspheres which give an arrangement corresponding to the new peeled class. The only difference is that cell  $Q_2$  disappears, by interchanging  $B_1, B_2$  on the corresponding spheres  $S_i, S_k$  and then slightly pulling the faces apart. (If  $n = 3$ , we can visualize a pair of disks on two intersecting spheres with a common boundary circle. Then peeling can be viewed as moving these two disks until they coincide and then pulling first past the second). So it is clear that if we can repeatedly show that a quadrant can be found with two innermost faces, until all the intersections between the hyperspheres have been removed, then we will have corner-peeled  $C$  to a 1-maximum class, i.e. a tree. So peeling will be established.

Suppose neither of the two quadrants  $Q_2, Q_3$  has both faces innermost. Consider  $Q_2$  say and let  $\{S_\alpha\}$  be the family of spheres intersecting the interior of the face  $B_2$ . Amongst these spheres, there is clearly at least one  $S_\beta$  so that the intersection  $S_{k\beta}$  is innermost on  $S_k$ . But then  $S_{k\beta}$  bounds an innermost ball  $B_4$  in  $S_k$  whose interior is disjoint from all the spheres  $\{S_\alpha\}$ . Similarly, we see that  $S_{k\beta}$  bounds a ball  $B_5$  which is the intersection of the sphere  $S_\beta$  with the quadrant  $Q_2$ . We get a new quadrant bounded by  $B_4 \cup B_5$  which is strictly smaller than  $Q_2$  and has at least one innermost face. But clearly this process must terminate—we cannot keep finding smaller and smaller quadrants and so a smallest one must have both faces innermost. ■

## 9 Conclusions and Open Problems

We saw in Corollary 22 that  $d$ -maximum classes represented by simple linear hyperplane arrangements in  $\mathbb{R}^d$  have underlying cubical complexes that are homeomorphic to a  $d$ -ball. Hence the VC dimension and the dimension of the cubical complex are the same. Moreover in Theorem 41, we proved that  $d$ -maximum classes represented by PL hyperplane arrangements in  $\mathbb{R}^d$  are those whose underlying cubical complexes are manifolds or equivalently  $d$ -balls.

**Question 43** *Does every simple PL hyperplane arrangement in  $B^d$ , where every subcollection of  $d$  planes transversely meet in a point, represent the same concept class as some simple linear hyperplane arrangement?*

**Question 44** What is the connection between the VC dimension of a maximum class induced by a simple hyperbolic hyperplane arrangement and the smallest dimension of hyperbolic space containing such an arrangement? In particular, can the hyperbolic space dimension be chosen to only depend on the VC dimension and not the dimension of the binary cube containing the class?

We gave an example of a 2-maximum class in the 5-cube that cannot be realized as a hyperbolic hyperplane arrangement in  $\mathcal{H}^3$ . Note that the Whitney embedding theorem [RS82] proves that any cubical complex of dimension  $d$  embeds in  $\mathbb{R}^{2d}$ . Can such an embedding be used to construct a hyperbolic arrangement in  $\mathcal{H}^{2d}$  or a PL arrangement in  $\mathbb{R}^{2d}$ ?

The structure of the boundary of a maximum class is strongly related to corner-peeling. For Euclidean hyperplane arrangements, the boundary of the corresponding maximum class is homeomorphic to a sphere by Corollaries 21 and 22.

**Question 45** Is there a characterization of the cubical complexes that can occur as the boundary of a maximum class? Characterize maximum classes with isomorphic boundaries.

**Question 46** Does a corner-peeling scheme exist with corner vertex sequence having minimum degree?

Theorem 39 suggests the following.

**Question 47** Can any  $d$ -maximum class in  $\{0, 1\}^n$  be represented by a simple arrangement of hyperplanes in  $\mathbb{H}^n$ ?

**Question 48** Which compression schemes arise from sweeping across simple hyperbolic hyperplane arrangements?

Kuzmin & Warmuth note that there are unlabeled compression schemes that are cyclic [KW07]. In Proposition 15 we show that corner-peeling compression schemes (like min-peeling) are acyclic. So compression schemes arising from sweeping across simple arrangements of hyperplanes in Euclidean or Hyperbolic space are also acyclic. Does acyclicity characterize such compression schemes?

**Acknowledgments:** We thank Peter Bartlett and the first anonymous referee for their very helpful feedback.

## References

- [Alo83] N. Alon. On the density of sets of vectors. *Discrete Math.*, 46(2):199–202, 1983.
- [BDL98] S. Ben-David and A. Litman. Combinatorial variability of Vapnik-Chervonenkis classes with applications to sample compression schemes. *Discrete Applied Math.*, 86(1):3–25, 1998.
- [BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M.K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.
- [Dud85] R.M. Dudley. The structure of some Vapnik-Chervonenkis classes. In L.M. Le Cam and R.A. Olshen, editors, *Proceedings of the Berkeley Conference in Honor of Jerzy Neyman*, volume II, pages 495–507. Wadsworth, 1985.

- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1987.
- [Flo89] S. Floyd. Space-bounded learning and the Vapnik-Chervonenkis dimension. Technical Report TR-89-061, ICSI, UC Berkeley, 1989.
- [Fra83] P. Frankl. On the trace of finite sets. *Journal of Comb. Theory (A)*, 34(1):41–45, 1983.
- [GW94] B. Gärtner and E. Welzl. Vapnik-Chervonenkis dimension and (pseudo-) hyperplane arrangements. *Discrete and Comp. Geometry*, 12:399–432, 1994.
- [Hau95] D. Haussler. Sphere packing numbers for subsets of the boolean  $n$ -cube with bounded Vapnik-Chervonenkis dimension. *Journal of Comb. Theory (A)*, 69:217–232, 1995.
- [HLW94] D. Haussler, N. Littlestone, and M.K. Warmuth. Predicting  $\{0, 1\}$  functions on randomly drawn points. *Information and Computation*, 115(2):284–293, 1994.
- [KW07] D. Kuzmin and M.K. Warmuth. Unlabeled compression schemes for maximum classes. *Journal of Machine Learning Research*, 8(Sep):2047–2081, 2007.
- [LW86] N. Littlestone and M.K. Warmuth. Relating data compression and learnability. Unpublished manuscript, 1986.
- [Maz61] B. Mazur. A note on some contractible 4-manifolds. *Annals of Math.*, 73:221–228, 1961.
- [Ney06] T. Neylon. *Sparse Solutions for Linear Prediction Problems*. PhD thesis, NYU, 2006.
- [Rat94] J. G. Ratcliffe. *Foundations of Hyperbolic Manifolds*. Springer-Verlag, 1994.
- [RBR08] B. I. P. Rubinstein, P. L. Bartlett, and J. H. Rubinstein. Shifting: one-inclusion mistake bounds and sample compression. *Journal of Computer and System Sciences: Special Issue on Learning Theory 2006*, 2008. in press.
- [RS82] C. Rourke and B. Sanderson. *Introduction to Piecewise-Linear Topology*. Springer-Verlag, 1982.
- [Sau72] N. Sauer. On the density of families of sets. *Journal of Comb. Th. (A)*, 13:145–147, 1972.
- [She72] S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Math.*, 41(1):247–261, 1972.
- [Sma61] S. Smale. Generalized Poincaré conjecture in dimensions greater than four. *Annals of Math.*, 74:391–406, 1961.
- [VC71] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Prob. and its Applic.*, 16(2):264–280, 1971.
- [War03] M.K. Warmuth. Compressing to VC dimension many points. In *Proceedings of the 16th Annual Conference on Learning Theory*, 2003.
- [Wel87] E. Welzl. Complete range spaces. Unpublished notes, 1987.

---

# On the Equivalence of Weak Learnability and Linear Separability: New Relaxations and Efficient Boosting Algorithms

---

**Shai Shalev-Shwartz**

Toyota Technological Institute, Chicago, USA  
SHAI@TTI-C.ORG

**Yoram Singer**

Google, Mountain View, USA  
SINGER@GOOGLE.COM

## Abstract

Boosting algorithms build highly accurate prediction mechanisms from a collection of low-accuracy predictors. To do so, they employ the notion of weak-learnability. The starting point of this paper is a proof which shows that weak learnability is equivalent to linear separability with  $\ell_1$  margin. While this equivalence is a direct consequence of von Neumann's minimax theorem, we derive the equivalence directly using Fenchel duality. We then use our derivation to describe a family of relaxations to the weak-learnability assumption that readily translates to a family of relaxations of linear separability with margin. This alternative perspective sheds new light on known soft-margin boosting algorithms and also enables us to derive several new relaxations of the notion of linear separability. Last, we describe and analyze an efficient boosting framework that can be used for minimizing the loss functions derived from our family of relaxations. In particular, we obtain efficient boosting algorithms for maximizing hard and soft versions of the  $\ell_1$  margin.

## 1 Introduction

Boosting is a popular and successful method for building highly accurate predictors from a set of low-accuracy base predictors. For an overview see for example [FS99, Sch03, MR03]. The first boosting algorithm was used for showing the equivalence between weak learnability and strong learnability [Sch90]. Weak learnability means that for any distribution over a set of examples there exists a single feature, also referred to as weak hypothesis, that performs slightly better than random guessing. Schapire [Sch90] was the first to show that if the weak learnability assumption holds then it is possible to construct a highly accurate classifier, to the point that it perfectly classifies all the examples in the training set. This highly accurate classifier is obtained by taking the sign of a weighted combination of weak hypotheses. Put another way, [Sch90] showed that if the weak learnability assumption holds then the set of examples is linearly separable.

Studying the generalization properties of the AdaBoost algorithm, Schapire et al. [SFBL97] showed that AdaBoost

in fact finds a linear separator with a large margin. However, AdaBoost does not converge to the max margin solution [RW05, RSD07]. Interestingly, the equivalence between weak learnability and linear separability is not only qualitative but also quantitative: weak learnability with edge  $\gamma$  is equivalent to linear separability with an  $\ell_1$  margin of  $\gamma$ . We give a precise statement and a simple proof of the equivalence in Thm. 4. We note that the equivalence can be also derived from von Neumann's minimax theorem [vN28]. Nevertheless, our proof is instructive and serves as a building block for the derivation of our main results.

Since the weak learnability assumption is equivalent to linear separability, it implies that the weak-learnability assumption is non-realistic due to its high sensitivity to even small amounts of label noise. For example, assume that we have a dataset that is perfectly separable with a large margin with the exception of two examples. These two examples share the same instance but attain opposite labels. Since such a dataset is non-separable, the weak learnability assumption fails to hold as well. To cope with this problem, we must somehow relax the weak learnability, which is equivalent to relaxing the linear separability assumption. In this paper we propose a family of relaxations of the linear separability assumption, which stems from the equivalence of weak-learnability and linear-separability. The guiding tool is to first define a natural family of relaxations of the weak learnability assumption, and then analyze its implication on the separability assumption.

In addition to our analysis and relaxations outline above, we also propose and analyze an algorithmic framework for boosting that efficiently solve the problems derived from our family of relaxations. The algorithm finds an  $\epsilon$  accurate solution after performing at most  $O(\log(m)/\epsilon^2)$  iterations, where  $m$  is the number of training examples. The number of iterations upper bounds the number of different weak-hypotheses constituting the solution. Therefore, we cast a natural trade-off between the desired accuracy level,  $\epsilon$ , of the (possibly relaxed) margin attained by the weight vector learned by the boosting algorithm, and the sparseness of the resulting predictor. In particular, we obtain new algorithms for maximizing the hard and soft  $\ell_1$  margin. We also provide an  $O(m \log(m))$  procedure for entropic projections onto  $\ell_\infty$  balls. Combined with this procedure, the total complexity of each iteration of our algorithm for minimizing the soft  $\ell_1$  margin is almost the same as the complexity of each iteration

of AdaBoost, assuming that the complexity of each activation of the weak learning algorithm requires  $\Omega(m)$  time.

**Related Work** As mentioned above, the equivalence between weak learnability and linear separability with  $\ell_1$  margin is a direct consequence of von Neumann’s minimax theorem in game theory [vN28]. Freund and Schapire [FS96] were the first to use von Neumann’s result to draw a connection between weak learnability and separability. They showed that if the weak learnability assumption holds then the data is linearly separable. The exact quantification of the weak learnability parameter and the  $\ell_1$  margin parameter was spelled out later in [RW05].

Schapire et al. [SFBL97] showed that the AdaBoost algorithm finds a large margin solution. However, as pointed out by [RW05, RSD07], AdaBoost does not converge to the max margin solution. Ratsch and Warmuth [RW05] suggested an algorithm called AdaBoost<sub>\*</sub> which converges to the maximal margin solution in  $O(\log(m)/\epsilon^2)$  iterations. The family of algorithms we propose in this paper entertains the same convergence properties. Rudin et al. [RSD07] provided a more accurate analysis of the margin attained by AdaBoost and also presented algorithms for achieving the max-margin solution. However, their algorithm may take  $O(1/\epsilon^3)$  iterations to find an  $\epsilon$  accurate predictor.

The above algorithms are effective when the data is linearly separable. Over the years, many boosting algorithms were suggested for non-separable datasets. We list here few examples. The LogLoss Boost algorithm [CSS02] tries to minimize the cumulative logistic loss, which is less sensitive to noise. MadaBoost [DW00] is another example of an algorithm that copes with non-separability. It does so by capping from the above the importance weights produced by the boosting algorithm. MadaBoost shares similarities with some of the relaxations presented in this paper. However, MadaBoost does not exploit the aforementioned equivalence and has a convergence rate that seems to be inferior to the rate obtained by the relaxations we consider in this paper. Another notable example for a boosting algorithm that works well in the non-separable case and is noise tolerant is the BrownBoost algorithm [Fre01]. BrownBoost uses the error-function (erf) as a margin-based loss function. The error-function reaches an asymptote when its input (margin in the context of BrownBoost) tends to  $-\infty$ . It thus constitutes a robust alternative to a convex loss function, including the LogLoss function. Since the error function is non-convex, all the results presented in this paper are not applicable to BrownBoost. In the support vector machine literature, the common relaxation of the separability assumption is obtained by using the hinge-loss (see for example [CST00]). Warmuth, Glocer and Ratsch [WGR07] recently proposed the SoftBoost algorithm that directly minimizes the hinge-loss function. The relaxation described in [WGR07] is a special case of the family of relaxations we present in this paper. The SoftBoost algorithm also builds on the idea of relaxing the weak learnability assumption by capping the maximal weight of a single example. A similar idea was also used by the SmoothBoost algorithm [Ser03]. Our presentation leads to an interesting perspective on this relaxation, showing that maximizing the margin while minimizing the hinge-loss is equivalent to maximizing the average margin

of the  $k$  examples with the worst margin. This equivalence is also implied from the work presented in [WGR07]. More importantly, in this paper we present a much simple algorithm which does not employ a convex optimization procedure on each round of boosting. Our approach stands in contrast to the algorithm of [WGR07], which requires “totally corrective” updates (see also [WLR06]) and needs to solve a rather complex optimization problem on each iteration.

The family of boosting algorithms we derive is reminiscent of the boosting algorithm proposed by Zhang [Zha03]. However, our analysis is different and allows us to: (i) provide an analytic solution for the step size; (ii) tackle complicated loss functions, including cases when the loss function does not take an explicit form. Our analysis stems from the primal-dual view of online convex programming [SSS06a, SSS07, SS07] and also borrows ideas from the analysis given in [SVL07]. The main difference between our analysis and that of [SVL07, Zha03] is that we do not impose any assumption on the second order derivatives of the objective function. Instead, we rely on a duality argument and require a strongly convex assumption on the Fenchel conjugate of the loss function. As we show, in many interesting cases, it is simple to verify that our assumption holds, while it is very complex to analyze the second order derivatives of the loss function in hand.

Throughout this paper, we focus on the analysis of the empirical loss over the training set. There has been extensive work on obtaining generalization bounds for boosting algorithms and for margin-based hypotheses. We refer the reader for example to [SFBL97, MBB98, KPL01]. A complementary question, left out of the scope of this paper, is whether the equivalence between weak learnability and linear separability with margin can be exploited for obtaining improved generalization bounds.

## 2 Notation and basic definitions

Let  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$  be a sequence of  $m$  examples, where for all  $i$ ,  $\mathbf{x}_i \in \mathcal{X}$  and  $y_i \in \{+1, -1\}$ . Let  $\mathcal{H}$  be a set of base hypotheses, namely, each  $h \in \mathcal{H}$  is a function from  $\mathcal{X}$  into  $\{+1, -1\}$ . For simplicity, we assume that  $\mathcal{H}$  is finite and thus  $\mathcal{H} = \{h_1, \dots, h_n\}$ . Let  $A$  be a matrix of size  $m \times n$  over  $\{+1, -1\}$  where the  $(i, j)$  entry of  $A$  is  $A_{i,j} = y_i h_j(\mathbf{x}_i)$ . We note that boosting algorithms solely use the matrix  $A$  and do not directly work with the set of examples. Therefore, throughout the rest of the paper we focus on the properties of the matrix  $A$ .

We denote column vectors with bold face letters, e.g.  $\mathbf{d}$  and  $\mathbf{w}$ , and use the notation  $\mathbf{d}^\dagger, \mathbf{w}^\dagger$  for denoting their corresponding row vectors. The inner product between vectors is denoted by  $\langle \mathbf{d}, \mathbf{w} \rangle = \mathbf{d}^\dagger \mathbf{w}$ . We denote by  $A^\dagger$  the transpose of the matrix  $A$ . The vector obtained by multiplying a matrix  $A$  with a vector  $\mathbf{d}$  is designated as  $A\mathbf{d}$  and its  $i$ th element as  $(A\mathbf{d})_i$ .

The set of non-negative real numbers is denoted as  $\mathbb{R}_+$  and the set of integers  $\{1, \dots, n\}$  as  $[n]$ . The  $m$  dimensional probability simplex is denoted by  $\mathbb{S}^m = \{\mathbf{d} \in \mathbb{R}_+^m : \|\mathbf{d}\|_1 = 1\}$ . We denote the  $m$  dimensional  $\ell_1$  ball of radius  $r$  by  $\mathbb{B}_1^m(r) = \{\mathbf{w} \in \mathbb{R}^m : \|\mathbf{w}\|_1 \leq r\}$ . For the unit  $\ell_1$  ball, we often omit  $r$  and use the shorthand  $\mathbb{B}_1^m$ . Similarly, we denote the  $m$  dimensional  $\ell_p$  ball by  $\mathbb{B}_p^m(r) = \{\mathbf{w} \in \mathbb{R}^m :$

$\|\mathbf{w}\|_p \leq r\}$  and again omit  $r$  whenever it is equals to 1.

**Definition 1 (separability with  $\ell_1$  margin  $\gamma$ )** A matrix  $A$  is linearly separable with  $\ell_1$  margin  $\gamma$  if there exists  $\mathbf{w} \in \mathbb{B}_1^n$  such that  $\min_{i \in [m]} (A\mathbf{w})_i \geq \gamma$ , and  $\gamma$  is the largest scalar that satisfies the above inequality, namely,

$$\gamma = \max_{\mathbf{w} \in \mathbb{B}_1^n} \min_{i \in [m]} (A\mathbf{w})_i .$$

**Definition 2 ( $\gamma$ -weak-learnability)** A matrix  $A$  is  $\gamma$ -weak-learnable if for all  $\mathbf{d} \in \mathbb{S}^m$  there exists  $j \in [n]$  such that  $|(d^\dagger A)_j| \geq \gamma$ , and  $\gamma$  is the largest scalar that satisfies the above. Namely,

$$\gamma = \min_{\mathbf{d} \in \mathbb{S}^m} \max_{j \in [n]} |(d^\dagger A)_j| .$$

We next give a few basic definitions from convex analysis. A set  $S \subset \mathbb{R}^n$  is convex if for any two vectors  $\mathbf{d}_1, \mathbf{d}_2$  in  $S$ , all the line between  $\mathbf{d}_1$  and  $\mathbf{d}_2$  is also in  $S$ , that is,  $\{\alpha \mathbf{d}_1 + (1 - \alpha) \mathbf{d}_2 : \alpha \in [0, 1]\} \subseteq S$ . A function  $f : S \rightarrow \mathbb{R}$  is closed and convex if for any scalar  $r$ , the level set  $\{\mathbf{d} : f(\mathbf{d}) \leq r\}$  is closed and convex. We allow functions to output  $+\infty$  and denote by  $\text{dom}(f)$  the set  $\{\mathbf{d} : f(\mathbf{d}) < +\infty\}$ . The core of a set  $C \in \mathbb{R}^n$ , denoted  $\text{core}(C)$ , is the set of all points in  $\mathbf{x} \in C$  such that for all  $\mathbf{d} \in \mathbb{R}^n$  there exists  $\tau' > 0$  for which for all  $\tau \in [0, \tau']$  we have  $\mathbf{x} + \tau \mathbf{d} \in C$ . The Fenchel conjugate of a function  $f : S \rightarrow \mathbb{R}$  is defined as

$$f^*(\boldsymbol{\theta}) = \max_{\mathbf{d} \in S} \langle \mathbf{d}, \boldsymbol{\theta} \rangle - f(\mathbf{d}) . \quad (1)$$

If  $f$  is closed and convex then  $f^{**} = f$ .

Our derivation makes an extensive use of the following theorem.

**Theorem 3 (Fenchel Duality: Theorem 3.3.5 in [BL06])**

Let  $f : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{\infty\}$  and  $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$  be two closed and convex functions and let  $A$  be a matrix of dimension  $m \times n$ . Then,

$$\max_{\mathbf{w}} -f^*(-A\mathbf{w}) - g^*(\mathbf{w}) \leq \min_{\mathbf{d}} f(\mathbf{d}) + g(d^\dagger A) .$$

The above holds with equality if in addition we have

$$\mathbf{0} \in \text{core}(\text{dom}(g) - A^\dagger \text{dom}(f)) .$$

We denote an arbitrary norm by  $\|\cdot\|$  and its dual norm by  $\|\cdot\|_*$ . That is,

$$\|\mathbf{w}\|_* = \max_{\mathbf{d} : \|\mathbf{d}\| \leq 1} \langle \mathbf{w}, \mathbf{d} \rangle .$$

Two dual norms that we extensively use are  $\|\mathbf{w}\|_1 = \sum_i |w_i|$  and  $\|\mathbf{w}\|_\infty = \max_i |w_i|$ .

For a set  $C$ , we denote by  $I_C(\mathbf{d})$  the indicator function of  $C$ , that is,  $I_C(\mathbf{d}) = 0$  if  $\mathbf{d} \in C$  and otherwise  $I_C(\mathbf{d}) = \infty$ . The definition of  $\|\mathbf{w}\|_*$  implies that the Fenchel conjugate of  $I_C(\mathbf{d})$  where  $C = \{\mathbf{d} : \|\mathbf{d}\| \leq 1\}$ , is the function  $\|\cdot\|_*$ . To conclude this section, we would like to point the reader to Table 1 which summarizes our notations.

Table 1: Summary of notations.

$\mathbf{x}, \mathbf{x}^\dagger$	column vector and its transpose
$\langle \mathbf{x}, \mathbf{v} \rangle$	inner product ( $= \mathbf{x}^\dagger \mathbf{v}$ )
$A$	matrix of size $m \times n$
$\mathbb{S}^m$	$m$ dimensional probability simplex
$\mathbb{B}_p^m(\nu)$	$\ell_p$ ball $\{\mathbf{w} \in \mathbb{R}^m : \ \mathbf{w}\ _p \leq \nu\}$
$I_C(\mathbf{d})$	indicator function ( $= 0$ if $\mathbf{d} \in C$ and $= \infty$ else)
$[\mathbf{x}]_+$	vector whose $i$ th element equals $\max\{0, x_i\}$
$\ \cdot\ , \ \cdot\ _*$	norm and its dual norm
$f, f^*$	function and its Fenchel conjugate
$\mathbf{e}^i$	all zeros vector except 1 in the $i$ th position
$[m]$	the set $\{1, \dots, m\}$

### 3 Weak-learnability and linear-separability

In this section we establish the equivalence between weak learnability and linear separability with  $\ell_1$  margin. As mentioned before, this result can be derived from von Neumann's minimax theorem. The purpose of the proof below is to underscore the duality between weak learnability and separability, which becomes useful in the next sections.

**Theorem 4** A matrix  $A$  is  $\gamma$ -weak-learnable if and only if it is linearly separable with  $\ell_1$  margin of  $\gamma$ .

**Proof:** We prove the theorem using Fenchel duality (Thm. 3). For convenience, we refer to the optimization problem on the right (left) hand side of Thm. 3 as the primal (dual) optimization problem. Let  $f$  be the indicator function of the  $m$ -dimensional simplex, i.e.  $f(\mathbf{d}) = 0$  if  $\mathbf{d} \in \mathbb{S}^m$  and otherwise  $f(\mathbf{d}) = \infty$ , and let  $g(\mathbf{w}) = \|\mathbf{w}\|_\infty$ . Then, the primal problem is

$$P^* = \min_{\mathbf{d}} f(\mathbf{d}) + g(d^\dagger A) = \min_{\mathbf{d} \in \mathbb{S}^m} \|\mathbf{d}^\dagger A\|_\infty .$$

The definition of  $\gamma$ -weak-learnability conveys that  $A$  is  $P^*$ -weak-learnable. Next, we turn to the dual problem. The Fenchel conjugate of  $g$  is the indicator function of the set  $\mathbb{B}_1^n$  (see Sec. 2) and the Fenchel conjugate of  $f$  is

$$f^*(\boldsymbol{\theta}) = \max_{\mathbf{d} \in \mathbb{S}^m} \langle \boldsymbol{\theta}, \mathbf{d} \rangle - f(\mathbf{d}) = \max_{\mathbf{d} \in \mathbb{S}^m} \langle \boldsymbol{\theta}, \mathbf{d} \rangle = \max_{i \in [m]} \theta_i .$$

Therefore,

$$D^* = \max_{\mathbf{w} \in \mathbb{R}^n} -f^*(-A\mathbf{w}) - g^*(\mathbf{w}) = \max_{\mathbf{w} \in \mathbb{B}_1^n} \min_{i \in [m]} (A\mathbf{w})_i .$$

Definition 1 implies that  $A$  is separable with  $\ell_1$  margin of  $D^*$ . To conclude our proof, it is left to show that  $P^* = D^*$ . First, we note that for  $\mathbf{w} = \mathbf{0}$  the value of  $D$  is zero, and thus  $D^* \geq 0$ . Therefore, if  $P^* = 0$  then  $0 = P^* \geq D^* \geq 0$  so in this case we clearly have  $P^* = D^*$ . Assume now that  $P^* = \gamma > 0$ . Based on Thm. 3 and the definition of the core operator, it suffices to show that for any vector  $\mathbf{v}$  there exists  $\tau' > 0$  such that for all  $\tau \in [0, \tau']$  we have  $\tau \mathbf{v} \notin \{A^\dagger \mathbf{d} : \mathbf{d} \in \mathbb{S}^m\}$ . This property holds true since for any  $\mathbf{d} \in \mathbb{S}^m$  we have  $\|A^\dagger \mathbf{d}\|_\infty \geq P^*$  while for sufficiently small  $\tau'$  we must have  $\|\tau \mathbf{v}\|_\infty < P^*$  for all  $\tau \in [0, \tau']$ . ■

## 4 A family of relaxations

In the previous section we showed that weak learnability is equivalent to separability. The separability assumption is problematic since even a perturbation of a single example can break it. In this section we propose a family of relaxations of the separability assumption. The motivation for these relaxations stems from the equivalence between weak-learnability and separability. The main idea is to first define a natural family of relaxations of the weak learnability assumption, and then analyze the implication to the separability assumption. To simplify the presentation, we start with a particular relaxation that was studied in [Ser03, WLR06]. We then generalize the example and describe the full family of relaxations.

### 4.1 A first relaxation: capped probabilities and soft margin

To motivate the first simple relaxation, consider a matrix  $A$  whose  $i$ th row equals to the negation of its  $j$ th row. That is, our training set contains an instance which appears twice, each time with a different label. Clearly, this training set is not separable even though the rest of the training set can be perfectly separable with a large margin. The equivalence between weak learnability and linear separability implies that  $A$  is also not weak learnable. To derive this property directly, construct the distribution  $\mathbf{d}$  with  $d_i = d_j = \frac{1}{2}$  (and  $d_r = 0$  for  $r \neq i$  and  $r \neq j$ ) and note that  $\mathbf{d}^\dagger A = \mathbf{0}$ .

In the above example, the weak learnability assumption fails because we place excessive weight on the problematic examples  $i, j$ . Indeed, it was observed that AdaBoost overweights examples, which partially explains its poor performance on noisy data. To overcome this problem, it was suggested (see for instance [Ser03, WLR06]) to restrict the set of admissible distributions by capping the maximum importance weight of each example. That is, the weak learner should return a weak hypothesis only when its input distribution satisfies  $\|\mathbf{d}\|_\infty \leq \frac{1}{k}$ , for a predefined integer  $k \in [m]$ .

Plugging the above restriction on  $\mathbf{d}$  into Definition 2 we obtain the following relaxed weak learnability value,

$$\rho = \min_{\mathbf{d} \in \mathbb{S}^m: \|\mathbf{d}\|_\infty \leq \frac{1}{k}} \max_{j \in [n]} |(\mathbf{d}^\dagger A)_j|. \quad (2)$$

Assume that a matrix  $A$  satisfies the above with  $\rho > 0$ . The immediate question that surfaces is what is the implication on the separability properties of  $A$ ? To answer this question, we need to refine the duality argument given in the proof of Thm. 4.

Let  $f(\mathbf{d})$  be the indicator function of  $\mathbb{S}^m \cap \mathbb{B}_\infty^m(\frac{1}{k})$  and let  $g(\mathbf{w}) = \|\mathbf{w}\|_\infty$ . The optimization problem given in Eq. (2) can be rewritten as  $\min_{\mathbf{d}} f(\mathbf{d}) + g(\mathbf{d}^\dagger A)$ . To derive the dual optimization problem, we find the Fenchel conjugate of  $f$ ,

$$f^*(\boldsymbol{\theta}) = \max_{\mathbf{d} \in \mathbb{S}^m: \|\mathbf{d}\|_\infty \leq \frac{1}{k}} \langle \mathbf{d}, \boldsymbol{\theta} \rangle.$$

To maximize the inner product  $\langle \mathbf{d}, \boldsymbol{\theta} \rangle$  we should allocate the largest admissible weight to the largest element of  $\boldsymbol{\theta}$ , allocate the largest of the remaining weights to the second largest element of  $\boldsymbol{\theta}$ , and so on and so forth. For each  $i \in [m]$ ,

let  $s_i(\boldsymbol{\theta})$  be the  $i$ th largest element of  $\boldsymbol{\theta}$ , that is,  $s_1(\boldsymbol{\theta}) \geq s_2(\boldsymbol{\theta}) \geq \dots$ . Then, the above argument yields

$$f^*(\boldsymbol{\theta}) = \frac{1}{k} \sum_{j=1}^k s_j(\boldsymbol{\theta}).$$

Combining the form of  $f^*$  with Thm. 3 we obtain that the dual problem of Eq. (2) is

$$\max_{\mathbf{w} \in \mathbb{B}_1^n} \frac{1}{k} \sum_{j=0}^{k-1} s_{m-j}(A\mathbf{w}). \quad (3)$$

Using the same technique as in the proof of Thm. 4 it is easy to verify that strong duality holds as well. We therefore obtain the following corollary.

**Corollary 5** *Let  $A$  be a matrix and let  $k \in [m]$ . For a vector  $\boldsymbol{\theta}$ , let  $\text{AvgMin}_k(\boldsymbol{\theta})$  be the average of the  $k$  smallest elements of  $\boldsymbol{\theta}$ . Let  $\rho$  be as defined in Eq. (2). Then,*

$$\max_{\mathbf{w} \in \mathbb{B}_1^n} \text{AvgMin}_k(A\mathbf{w}) = \rho.$$

Let us now discuss the role of the parameter  $k$ . First, if  $k = 1$  then the function  $\text{AvgMin}_k$  reduces to the minimum over the vector provided as its argument, and therefore we revert back to the traditional definition of margin. When  $k = m$ , the only admissible distribution is the uniform distribution. In this case, it is easy to verify that the optimal weight vector associates  $w_j = 1$  with the feature that maximizes  $|(\mathbf{d}^\dagger A)_j|$  (while  $\mathbf{d}$  being the uniform distribution) and  $w_j = 0$  with the rest of the features. That is, the performance of the optimal strong hypothesis is equal to the performance of the best single weak hypothesis, and no boosting process takes place. The interesting regime is when  $k$  is proportional to  $m$ , for example  $k = 0.1m$ . In this case, if  $\rho > 0$ , then we are guaranteed that 90% of the examples can be separated by margin of at least  $\rho$ .

It is also possible to set  $k$  based on knowledge of the number of noisy examples in the training set and the separability level of the rest of the examples. For example, assume that all but  $\nu$  of the examples are separable with margin  $\gamma$ . Then, the worst objective value that  $\mathbf{w}$  can attain is,  $\text{AvgMin}_k(A\mathbf{w}) = \frac{-\nu + (k-\nu)\gamma}{k}$ . Constraining the right hand side of this equality above to be at least  $\frac{\gamma}{2}$  and solving for  $k$  yields that for  $k \geq 2\nu(\gamma + 1)/\gamma$  at least  $m - k$  examples attain a margin value of at least  $\gamma/2$ .

### 4.2 A general relaxation scheme

We now generalize the above relaxation and present our general relaxation scheme. To do so, we first rewrite Eq. (2) as follows. Denote  $C = \mathbb{B}_\infty^m(1/k)$  and recall that  $I_C(\mathbf{d})$  is the indicator function of the set  $C$ . We can now rewrite Eq. (2) as

$$\rho = \min_{\mathbf{d} \in \mathbb{S}^m} \left( \max_{j \in [n]} |(\mathbf{d}^\dagger A)_j| + I_C(\mathbf{d}) \right). \quad (4)$$

The general relaxation scheme is obtained by replacing  $I_C$  with a large family of functions. Before specifying the properties of allowed functions, let us first define the following generalized notion of weak learnability.

**Definition 6** ( $(\rho, f)$ -weak-learnability) Let  $f$  be an arbitrary function. A matrix  $A$  is  $(\rho, f)$ -weak-learnable if

$$\rho = \min_{\mathbf{d} \in \mathbb{S}^m} \left( \max_{j \in [n]} |(\mathbf{d}^\dagger A)_j| + f(\mathbf{d}) \right).$$

Intuitively, we can think on  $\rho$  as the minimum of the maximal edge plus a regularization term  $f(\mathbf{d})$ . In the case of capped importance weights, the regularization function is a barrier function that does not penalize distributions inside  $\mathbb{B}_\infty^m(1/k)$  and places an infinite penalty for the rest of the distributions.

The following theorem shows how the fact that a matrix  $A$  is  $(\rho, f)$ -weak-learnable affects its separability properties. To remind the reader, we denote by  $\mathbf{e}^i$  the vector whose  $i$ th element is 1 and the rest of its elements are zero. The notation  $[\mathbf{x}]_+$  represents the vector whose  $i$ th element is  $\max\{0, x_i\}$ .

**Theorem 7** Let  $f$  be a convex function,  $\rho$  be a scalar, and  $A$  be a  $(\rho, f)$ -weak-learnable matrix. If the following assumptions hold,

- (i)  $\min_{\mathbf{d}} f(\mathbf{d}) = 0$ ,
- (ii)  $\mathbf{0} \in \text{core}(\text{dom}(f))$ ,
- (iii)  $\forall \theta \in \mathbb{R}^m, \forall i \in [m], \forall \alpha \in [0, 1]$ , the Fenchel conjugate of  $f$  satisfies

$$f^*(\theta) \geq f^*(\theta - \alpha \theta_i \mathbf{e}^i)$$

then,

$$\max_{\mathbf{w} \in \mathbb{B}_1^n, \gamma \in \mathbb{R}} \left( \gamma - f^*([\gamma - A\mathbf{w}]_+) \right) = \rho.$$

The proof of the theorem is again based on the Fenchel duality theorem. The vector  $[\gamma - A\mathbf{w}]_+$  appearing in the dual problem is the vector of hinge-losses. Before diving into the details of the proof, let us give two concrete family of functions that satisfy the requirement given in the theorem.

**Example 1** Let  $f$  be the indicator function of a ball of radius  $\nu$ ,  $\{\mathbf{d} : \|\mathbf{d}\| \leq \nu\}$ , where  $\|\cdot\|$  is an arbitrary norm and  $\nu$  is a scalar such that the intersection of this ball with the simplex is non-empty. Then,  $f^*(\mathbf{w}) = \nu \|\mathbf{w}\|_*$  and the condition given in the theorem clearly holds. In this case, we obtain that

$$\max_{\mathbf{w} \in \mathbb{B}_1^n, \gamma \in \mathbb{R}} \left( \gamma - \nu \|[\gamma - A\mathbf{w}]_+\|_* \right) = \min_{\mathbf{d} \in \mathbb{S}^m : \|\mathbf{d}\| \leq \nu} \|\mathbf{d}^\dagger A\|_\infty.$$

In particular, if  $\|\cdot\|$  is the  $\ell_\infty$  norm we obtain again the example of capped sample weights. Since the 1-norm and  $\infty$ -norm are dual we get that in the dual problem we are maximizing the margin parameter  $\gamma$  while minimizing the cumulative hinge-loss. Combining this fact with Corollary 5 we get that

$$\text{AvgMin}_k(A\mathbf{w}) = \max_{\gamma \in \mathbb{R}} \left( \gamma - \frac{1}{k} \sum_{i=1}^m [\gamma - (A\mathbf{w})_i]_+ \right).$$

The right hand side of the above is usually called the ‘‘soft-margin’’. The above equality tells us that the soft margin is equivalent to the average margin of the  $k$  worst examples (see also [WLR06, SSWB98]).

**Example 2** Let  $f(\mathbf{d}) = \nu \|\mathbf{d}\|$  where  $\|\cdot\|$  is an arbitrary norm and  $\nu$  is a scalar. Then,  $f^*(\mathbf{w})$  is the indicator function of the ball of radius  $\nu$  with respect to the dual norm  $\{\mathbf{w} : \|\mathbf{w}\|_* \leq \nu\}$ . The condition given in the theorem clearly holds here as well and we obtain the dual problem

$$\max_{\mathbf{w} \in \mathbb{B}_1^n, \gamma \in \mathbb{R}} \gamma \quad \text{s.t.} \quad \|[\gamma - A\mathbf{w}]_+\|_* \leq \nu.$$

That is, we are now maximizing the margin subject to a constraint on the vector of hinge-losses.

We now turn to proving Thm. 7. First, we need the following lemma which characterizes the Fenchel conjugate of  $f + I_{\mathbb{S}^m}$ .

**Lemma 8** Assume that  $f$  satisfies the conditions given in Thm. 7 and denote  $\tilde{f}(\mathbf{d}) = f(\mathbf{d}) + I_{\mathbb{S}^m}(\mathbf{d})$ . Then,

$$\tilde{f}^*(\theta) = - \max_{\gamma \in \mathbb{R}} (\gamma - f^*([\gamma + \theta]_+)).$$

**Proof:** We first rewrite  $\tilde{f}^*$  as

$$\begin{aligned} \tilde{f}^*(\theta) &= \max_{\mathbf{d}} -f(\mathbf{d}) - (I_{\mathbb{S}^m}(\mathbf{d}) - \langle \theta, \mathbf{d} \rangle) \\ &= - \left( \min_{\mathbf{d}} f(\mathbf{d}) + (I_{\mathbb{S}^m}(\mathbf{d}) - \langle \theta, \mathbf{d} \rangle) \right) \end{aligned}$$

Denote  $g(\mathbf{d}) = I_{\mathbb{S}^m}(\mathbf{d}) - \langle \theta, \mathbf{d} \rangle$ . It is easy to verify that  $g^*(\mathbf{x}) = \max_i(\theta_i + x_i)$ . Next, note that  $\mathbf{0} \in \text{core}(\text{dom}(f))$  by assumption and that  $\text{dom}(g) = \mathbb{S}^m$ . Therefore, strong duality holds and we can use Thm. 3 which yields,

$$\begin{aligned} -\tilde{f}^*(\theta) &= \max_{\mathbf{x}} (-f^*(\mathbf{x}) - g^*(-\mathbf{x})) \\ &= \max_{\mathbf{x}} \left( -f^*(\mathbf{x}) - \max_i(\theta_i - x_i) \right). \end{aligned}$$

Let  $C_\gamma = \{\mathbf{x} : \forall i, x_i \geq \theta_i + \gamma\}$ . We show in the sequel that for any  $\gamma$ , the vector  $[\theta + \gamma]_+$  is a minimizer of  $f^*(\mathbf{x})$  over  $\mathbf{x} \in C_\gamma$ . Combining this with the above expression for  $-\tilde{f}^*(\theta)$  we get that

$$-\tilde{f}^*(\theta) = \max_{\gamma} (\gamma - f^*([\theta + \gamma]_+)),$$

as required. Therefore, it is left to show that the vector  $[\theta + \gamma]_+$  is indeed a minimizer of  $f^*(\mathbf{x})$  over  $C_\gamma$ . Clearly,  $[\theta + \gamma]_+ \in C$ . In addition, for any  $\mathbf{x} \in C_\gamma$  we can make a sequence of modifications to  $\mathbf{x}$  until  $\mathbf{x} = [\theta + \gamma]_+$  as follows. Take some element  $i$ . If  $x_i > [\theta_i + \gamma]_+$  then based on assumption (iii) of Thm. 7 we know that

$$f^* \left( \mathbf{x} - \frac{x_i - [\theta_i + \gamma]_+}{x_i} x_i \mathbf{e}^i \right) \leq f^*(\mathbf{x}).$$

If  $x_i < [\theta_i + \gamma]_+$  we must have that  $[\theta_i + \gamma]_+ = 0$  since we assume that  $\mathbf{x} \in C_\gamma$  and thus  $x_i \geq \theta_i + \gamma$ . Thus,  $x_i < 0$  but now using assumption (iii) of Thm. 7 again we obtain that  $f^*(\mathbf{x} - x_i \mathbf{e}^i) \leq f^*(\mathbf{x})$ . Repeating this for every  $i \in [m]$  makes  $\mathbf{x}$  equals to  $[\theta + \gamma]_+$  while the value of  $f^*(\mathbf{x})$  is non-increasing along this process. We therefore conclude that  $[\theta + \gamma]_+$  is a minimizer of  $f^*(\mathbf{x})$  over  $\mathbf{x} \in C_\gamma$  and our proof is concluded. ■

Based on the above lemma the proof of Thm. 7 is easily derived.

**Proof:**[of Thm. 7] The proof uses once more the Fenchel duality theorem. Define the function  $\tilde{f}(\mathbf{d}) = f(\mathbf{d}) + I_{\mathbb{S}^m}(\mathbf{d})$ . Therefore, Thm. 3 tells us that the dual of the problem  $\min_{\mathbf{d}} \tilde{f}(\mathbf{d}) + \|\mathbf{d}^\dagger A\|_\infty$  is the problem  $\max_{\mathbf{w} \in \mathbb{B}_1^n} \left( -\tilde{f}^*(-A\mathbf{w}) \right)$ . Using Lemma 8 we obtain that the dual of the problem given in Definition 6 is the same maximization problem as stated in the theorem. To conclude the proof it is left to show that strong duality also holds here. First, using the assumption  $\min_{\mathbf{d}} f(\mathbf{d}) = 0$  we get that  $f^*(\mathbf{0}) = 0$ . By setting  $\mathbf{w} = \mathbf{0}$  and  $\gamma = 0$  we get that the dual problem is bounded below by zero. Thus, if  $\rho = 0$  then strong duality holds. If  $\rho > 0$  then we can use the fact that  $\text{dom}(\tilde{f}) \subseteq \text{dom}(f)$  and therefore the same arguments as in the end of the proof of Thm. 4 holds here as well. ■

## 5 Boosting algorithms

In this section we derive a boosting algorithm for solving the max-relaxed-margin problem described in the previous section, namely,

$$\max_{\mathbf{w} \in \mathbb{B}_1^n} \max_{\gamma \in \mathbb{R}} (\gamma - f^*([\gamma - A\mathbf{w}]_+)) . \quad (5)$$

The function  $f^*$  should satisfy the conditions stated in Thm. 7. In particular, if  $f^*(\mathbf{x}) = \nu \|\mathbf{x}\|_1$  we obtain the soft margin problem

$$\max_{\mathbf{w} \in \mathbb{B}_1^n} \max_{\gamma \in \mathbb{R}} \left( \gamma - \nu \sum_{i=1}^m [\gamma - (A\mathbf{w})_i]_+ \right) , \quad (6)$$

while if  $f^*(\mathbf{x}) = \max_i x_i$  then we obtain the non-relaxed max margin problem

$$\max_{\mathbf{w} \in \mathbb{B}_1^n} \min_{i \in [m]} (A\mathbf{w})_i .$$

The boosting algorithm for solving Eq. (5) is described in Fig. 1. To simplify the presentation, let us first describe the algorithm for the non-relaxed max-margin problem, that is,  $f^*(\mathbf{x}) = \max_i x_i$ . As we have shown in the proof of Thm. 4, the corresponding Fenchel conjugate  $f(\mathbf{d})$  is the indicator function of  $\mathbb{S}^m$ . The algorithm initializes the weight vector to be the zero vector,  $\mathbf{w}_1 = \mathbf{0}$ . On round  $t$ , we define a distribution over the examples

$$\begin{aligned} \mathbf{d}_t &= \operatorname{argmax}_{\mathbf{d} \in \mathbb{S}^m} \left( \langle -A\mathbf{w}_t, \mathbf{d} \rangle - (f(\mathbf{d}) + \beta h(\mathbf{d})) \right) \\ &= \operatorname{argmin}_{\mathbf{d} \in \mathbb{S}^m} \left( \langle A\mathbf{w}_t, \mathbf{d} \rangle + (f(\mathbf{d}) + \beta h(\mathbf{d})) \right) , \end{aligned}$$

where  $h(\mathbf{d})$  is the relative entropy function. Since we are now dealing with the case  $f(\mathbf{d}) = I_{\mathbb{S}^m}$ , we can use Lemma 18 in the appendix and get that  $\mathbf{d}_t$  is the gradient of the Fenchel conjugate of the function  $\beta h(\mathbf{d})$ . In the appendix we list several Fenchel conjugate pairs. In particular, the Fenchel conjugate of the relative entropy is the soft-max function

$$h^*(\boldsymbol{\theta}) = \log \left( \frac{1}{m} \sum_{i=1}^m e^{\theta_i} \right) .$$

INPUT: matrix  $A \in [+1, -1]^{m,n}$   
 Relaxation function  $f^*$   
 Desired accuracy  $\epsilon$   
 DEFINE:  $h(\mathbf{d}) = \sum_{i=1}^m d_i \log(d_i) + \log(m)$   
 $f(d) = \text{Fenchel conjugate of } f^*$   
 INITIALIZE:  $\mathbf{w}_1 = \mathbf{0}$ ,  $\beta = \frac{\epsilon}{2 \log(m)}$   
 FOR  $t = 1, 2, \dots, T$   
 $\mathbf{d}_t = \operatorname{argmin}_{\mathbf{d} \in \mathbb{S}^m} \left( \langle A\mathbf{w}_t, \mathbf{d} \rangle + (f(\mathbf{d}) + \beta h(\mathbf{d})) \right)$   
 $j_t \in \operatorname{argmax}_j |(\mathbf{d}_t^\dagger A)_j|$   
 (w.l.o.g. assume  $\text{sign}(\mathbf{d}_t^\dagger A)_{j_t} = 1$ )  
 $\eta_t = \max \left\{ 0, \min \left\{ 1, \frac{\beta \mathbf{d}_t^\dagger A(\mathbf{e}^{j_t} - \mathbf{w}_t)}{\|A(\mathbf{e}^{j_t} - \mathbf{w}_t)\|_\infty^2} \right\} \right\}$   
 $\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t \mathbf{e}^{j_t}$   
 OUTPUT:  $\mathbf{w}_{T+1}$

Figure 1: A Boosting Algorithm for maximizing the relaxed margin given in Eq. (5).

Using the property  $(\beta h)^*(\boldsymbol{\theta}) = \beta h^*(\boldsymbol{\theta}/\beta)$  we obtain that

$$d_{t,i} \propto e^{-\frac{1}{\beta}(A\mathbf{w}_t)_i} .$$

That is, the log of the probability assigned to the  $i$ th example is negatively proportional to the margin of the example according to the current weight vector  $\mathbf{w}_t$ . Therefore, the algorithm allocates larger importance weights to the erroneous examples, in a similar fashion to the weighting scheme of many other boosting algorithms, such as Adaboost.

Next, we perform a step analogous to calling a weak-learner by finding a single column of  $A$  with the best edge. We would like to note that it is possible to extend the algorithm so that the weak learner may find a column whose edge is only approximately optimal. For simplicity we confine the description to weak learners that return the column with the largest edge. Finally, we set  $\mathbf{w}_{t+1}$  to be the convex combination of  $\mathbf{w}_t$  and the new hypothesis. The coefficient of the convex combination, denoted  $\eta_t$ , is calculated analytically based on our analysis. Note that the update form guarantees that  $\|\mathbf{w}_t\|_1 \leq 1$  for all  $t$ .

The sole modification of the algorithm when running with other relaxation functions is concerned with the definition of  $\mathbf{d}_t$ . In Sec. 5.2 we further elaborate on how to solve the optimization problem which appears in the definition of  $\mathbf{d}_t$ . We provide a few general tools and also present an efficient procedure for the case where  $f$  is the indicator function of  $\mathbb{B}_\infty^m(\nu)$ .

The following theorem provides analysis of the rate of convergence of the algorithm.

**Theorem 9** Assume that the algorithm given in Fig. 1 is run for  $T = \Omega(\log(m)/\epsilon^2)$  iterations. Then, the algorithm outputs an  $\epsilon$ -accurate solution,

$$\max_{\gamma} (\gamma - f^*([\gamma - A\mathbf{w}_{T+1}]_+)) \geq \rho - \epsilon ,$$

where  $\rho$  is the optimal value of the solution as defined in Thm. 7.

Before turning into the proof of Thm. 9 let us first discuss its implications. First we note that the number of iterations of the algorithm upper bounds the number of non-zero elements of the solution. Therefore, we have a trade-off between the desired accuracy level,  $\epsilon$ , and the level of sparsity of the solution,  $\mathbf{w}_{T+1}$ .

The algorithm can be used for maximizing the hard margin using  $O(\log(m)/\epsilon^2)$  iterations. In this case, the algorithm shares the simplicity of the popular AdaBoost approach. The rate of convergence we obtain matches the rate of the AdaBoost<sub>\*</sub> described by Ratsch and Warmuth [RW05] and is better than the rate obtained in Rudin et al. [RSD07]. We note also that if  $A$  is  $\gamma$ -separable and we set  $\epsilon = \gamma/2$  then we would find a solution with half the optimal margin in  $O(\log(m)/\gamma^2)$  iterations. AdaBoost seemingly attains an exponentially fast decay of the empirical error of  $e^{-\gamma^2 T}$ . Thus,  $T$  should be at least  $1/\gamma^2$ . Further careful examination also reveals a factor of  $\log(m)$  in the convergence rate of AdaBoost. Therefore, our algorithm attains the same rate of convergence of AdaBoost while both algorithms obtain a margin which is half of the optimal margin. (See also the margin analysis of AdaBoost described in Rudin et al. [RSD07].)

We can also use the algorithm for maximizing the soft margin given in Eq. (6). In Sec. 5.2 we show how to calculate  $\mathbf{d}_t$  in  $\tilde{O}(m)$  time. Therefore, the complexity of the resulting algorithm is roughly the same as the complexity of AdaBoost. The bound on the number of iterations that we obtain matches the bound of the SoftBoost algorithm, recently proposed by Warmuth et al. [WLR06]. However, our algorithm is simpler to implement and the time complexity of each iteration of our algorithm is substantially lower than the one described in [WLR06].

### 5.1 Proof of convergence rate

To motivate our proof technique, let us focus first on the max-margin case without any relaxation. As we showed before, the AdaBoost algorithm approximates the max operator,  $\max_i \theta_i$ , with a soft-max operator,  $\log(\frac{1}{m} \sum_i e^{\theta_i})$ , also known as the exp-loss. We can think of this approximation as another form of relaxation of the max margin. To distinguish this type of relaxation from the family of relaxations described in the previous section, we refer to it as an ‘‘algorithmic’’ relaxation, since this relaxation is driven by algorithmic factors and not directly by the concept of relaxing the margin. The algorithmic relaxation of AdaBoost encapsulates the following relaxation of weak learnability: replace the indicator function of the simplex with the relative entropy function over the simplex, which we denote by  $h(\mathbf{d})$  (see also the definition in Fig. 1). The advantage of endowing the simplex with the relative entropy stems from the fact that the relative entropy is *strongly* convex with respect to the  $\ell_1$  norm, as we formally define now.

**Definition 10** A continuous function  $f$  is  $\sigma$ -strongly convex over a convex set  $S$  with respect to a norm  $\|\cdot\|$  if  $S$  is contained in the domain of  $f$  and for all  $\mathbf{v}, \mathbf{u} \in S$  and  $\alpha \in [0, 1]$

we have

$$f(\alpha \mathbf{v} + (1 - \alpha) \mathbf{u}) \leq \alpha f(\mathbf{v}) + (1 - \alpha) f(\mathbf{u}) - \frac{\sigma}{2} \alpha (1 - \alpha) \|\mathbf{v} - \mathbf{u}\|^2.$$

In the above definition, if  $\sigma = 0$  we revert back to the standard definition of convexity. Strong convexity quantifies the difference between the value of the function at the convex combination and the convex combination of the values of the function. The relative entropy is 1-strongly convex with respect to the  $\ell_1$  norm over the probabilistic simplex (see Lemma 16 in [SS07]). Few important properties of *strongly* convex functions are summarized in Lemma 18 (in the appendix). We use these properties in our proofs below.

Continuing with our motivating discussion, we view the algorithmic relaxation of AdaBoost as a replacement of the convex function  $I_{\mathbb{S}^m}(\mathbf{d})$  by the strongly convex function  $h(\mathbf{d})$ . More generally, recall the definition  $\tilde{f}(\mathbf{d}) = f(\mathbf{d}) + I_{\mathbb{S}^m}(\mathbf{d})$  from Sec. 4 and that solving Eq. (5) is equivalent to maximizing  $-\tilde{f}^*(-A\mathbf{w})$  over  $\mathbf{w} \in \mathbb{B}_1^n$ . As in the algorithmic relaxation of AdaBoost, we replace  $\tilde{f}(\mathbf{d})$  by the function

$$\hat{f}(\mathbf{d}) = \tilde{f}(\mathbf{d}) + \beta h(\mathbf{d}),$$

where  $\beta \in (0, 1)$ . Since for all  $\mathbf{d} \in \mathbb{S}^m$  we have  $0 \leq h(\mathbf{d}) \leq \log(m)$ , by setting  $\beta = \epsilon/(2 \log(m))$  we obtain that

$$\forall \mathbf{d} \in \mathbb{S}^m, \hat{f}(\mathbf{d}) - \epsilon/2 \leq \tilde{f}(\mathbf{d}) \leq \hat{f}(\mathbf{d}).$$

Using Lemma 19 in the appendix we obtain that

$$\forall \theta, \hat{f}^*(\theta) \leq \tilde{f}^*(\theta) \leq \hat{f}^*(\theta) + \epsilon/2. \quad (7)$$

The above implies that maximizing  $-\hat{f}^*(-A\mathbf{w})$  gives an  $\epsilon/2$  accurate solution to the problem of maximizing  $-\tilde{f}^*(-A\mathbf{w})$ . This argument holds for the entire family of functions discussed in Sec. 4. An appealing property of strong convexity that we exploit is that by adding a convex function to a strongly convex function we retain at least the same strong convexity level. Therefore, for all the functions  $\tilde{f}(\mathbf{d})$  discussed in Sec. 4 the corresponding  $\hat{f}(\mathbf{d})$  retains the strongly convex property of the relative entropy.

The algorithm in Fig. 1 is designed for maximizing  $-\hat{f}^*(-A\mathbf{w})$  over  $\mathbb{B}_1^n$ . Based on the above discussion, this maximization translates to an approximate maximization of  $-\tilde{f}^*(-A\mathbf{w})$ . Using again Thm. 3 we obtain that

$$\max_{\mathbf{w} \in \mathbb{B}_1^n} -\hat{f}^*(-A\mathbf{w}) \leq \min_{\mathbf{d}} \hat{f}(\mathbf{d}) + \|\mathbf{d}^\dagger A\|_\infty.$$

Denote by  $\mathcal{D}(\mathbf{w})$  and  $\mathcal{P}(\mathbf{d})$  the dual and primal objective values of the above equation. We also denote by  $\epsilon_t$  the sub-optimality value attained at iteration  $t$  of the algorithm, namely,

$$\epsilon_t = \max_{\mathbf{w} \in \mathbb{B}_1^n} \mathcal{D}(\mathbf{w}) - \mathcal{D}(\mathbf{w}_t).$$

The following key lemma lower bounds the improvement of the algorithm in terms of its current sub-optimality.

**Lemma 11** Let  $\epsilon_t$  be the sub-optimality value of the algorithm in Fig. 1 at iteration  $t$  and assume that  $\epsilon_t \leq 1$ . Then,  $\epsilon_t - \epsilon_{t+1} \geq \beta \epsilon_t^2/8$ .

**Proof:** Denote  $\Delta_t = \epsilon_t - \epsilon_{t+1}$  and based on the definition of  $\epsilon_t$  we clearly have that  $\Delta_t = \mathcal{D}(\mathbf{w}_{t+1}) - \mathcal{D}(\mathbf{w}_t)$ . To simplify our notation, we use the shorthand  $j$  for  $j_t$  and  $\eta$  for  $\eta_t$ . Since

$$\mathbf{w}_{t+1} = (1 - \eta)\mathbf{w}_t + \eta\mathbf{e}^j$$

we get that

$$\Delta_t = \mathcal{D}(\mathbf{w}_t + \eta(\mathbf{e}^j - \mathbf{w}_t)) - \mathcal{D}(\mathbf{w}_t) .$$

Using the definition of  $\mathcal{D}$  we further rewrite  $\Delta_t$  as

$$\Delta_t = \hat{f}^*(-A\mathbf{w}_t) - \hat{f}^*(-A\mathbf{w}_t - \eta A(\mathbf{e}^j - \mathbf{w}_t)) . \quad (8)$$

The key property that we use is that  $\hat{f}^*$  is the Fenchel conjugate of a  $\beta$ -strongly convex function over the simplex with respect to the  $\ell_1$  norm. Therefore, using Lemma 18 in the appendix, we know that for any  $\theta_1$  and  $\theta_2$ :

$$\hat{f}^*(\theta_1 + \theta_2) - \hat{f}^*(\theta_1) \leq \langle \nabla, \theta_2 \rangle + \frac{\|\theta_2\|_\infty^2}{2\beta} ,$$

where  $\nabla = \arg \max_{\mathbf{d}} \langle \theta_1, \mathbf{d} \rangle - \hat{f}(\mathbf{d})$ . Combining this property with Eq. (8) and using the definition of  $\mathbf{d}_t$  we obtain that

$$\Delta_t \geq \eta \langle \mathbf{d}_t, A(\mathbf{e}^j - \mathbf{w}_t) \rangle - \frac{\eta^2 \|A(\mathbf{e}^j - \mathbf{w}_t)\|_\infty^2}{2\beta} . \quad (9)$$

Using the assumption  $A \in [+1, -1]^{m \times n}$ , the fact that  $\mathbf{w}_t \in \mathbb{B}_1^n$ , and the triangle inequality we get that

$$\|A(\mathbf{e}^j - \mathbf{w}_t)\|_\infty \leq 2$$

and thus

$$\Delta_t \geq \eta \langle \mathbf{d}_t, A(\mathbf{e}^j - \mathbf{w}_t) \rangle - 2\eta^2/\beta . \quad (10)$$

Next, we show that  $\langle \mathbf{d}_t, A(\mathbf{e}^j - \mathbf{w}_t) \rangle = \mathcal{P}(\mathbf{d}_t) - \mathcal{D}(\mathbf{w}_t)$ . To do so, we first use Lemma 17 to get that  $\langle \mathbf{d}_t, -A\mathbf{w}_t \rangle = \hat{f}(\mathbf{d}_t) + \hat{f}^*(-A\mathbf{w}_t)$  and second we use the definition of  $j$  to get that  $\langle \mathbf{d}_t, A\mathbf{e}^j \rangle = \|\mathbf{d}_t^\dagger A\|_\infty$ . Combining this with Eq. (10) yields

$$\Delta_t \geq \eta(\mathcal{P}(\mathbf{d}_t) - \mathcal{D}(\mathbf{w}_t)) - 2\eta^2/\beta . \quad (11)$$

The weak duality property tells us that  $\mathcal{P}(\mathbf{d}_t) \geq \max_{\mathbf{w} \in \mathbb{B}_1^n} \mathcal{D}(\mathbf{w})$  and therefore  $\Delta_t \geq \eta\epsilon_t - 2\eta^2/\beta$ . Denote  $\eta' = \epsilon_t\beta/4$  and note that  $\eta' \in [0, 1]$ . Had we set  $\eta_t = \eta'$  we could have obtained that  $\Delta_t \geq \beta\epsilon_t^2/8$  as required. Since we set  $\eta_t$  to be the maximizer of the expression in Eq. (9) over  $[0, 1]$ , we get an even larger value for  $\Delta_t$ . This concludes our proof.  $\blacksquare$

Based on Lemma 11 the proof of Thm. 9 easily follows.

**Proof:**[of Thm. 9] We first show that  $\epsilon_1 \leq 1$ . To see this, we use the weak duality to get that  $\epsilon_1 \leq \mathcal{P}(\mathbf{d}_1) - \mathcal{D}(\mathbf{w}_1)$ . Next, we recall that in the proof of Lemma 11 we have shown that for all  $t$ ,  $\mathcal{P}(\mathbf{d}_t) - \mathcal{D}(\mathbf{w}_t) = \langle \mathbf{d}_t, A(\mathbf{e}^{j_t} - \mathbf{w}_t) \rangle$ . Since  $\mathbf{w}_1 = \mathbf{0}$  we get that  $\epsilon_1 \leq \langle \mathbf{d}_1, A\mathbf{e}^{j_1} \rangle = \|\mathbf{d}_1^\dagger A\|_\infty \leq 1$ .

We can now apply Lemma 11 for  $t = 1$  and get that  $\epsilon_2 \leq \epsilon_1$ . By induction, we obtain that Lemma 11 holds for all  $t$ . Applying Lemma 20 (given in the appendix) we get that  $\epsilon_t \leq \frac{8}{\beta(t+1)}$ .

Plugging the definition of  $\beta = \epsilon/(2\log(m))$  into the upper bound on  $\epsilon_{T+1}$  we get  $\epsilon_{T+1} \leq \frac{16\log(m)}{(T+2)\epsilon}$ . Therefore, if  $T + 2 \geq 32\log(m)/\epsilon^2$  we get that  $\epsilon_{T+1} \leq \epsilon/2$ . Finally, Let  $\epsilon'$  be the error of  $\mathbf{w}_{T+1}$  on the original  $f$  then using Eq. (7) we obtain that  $\epsilon' \leq \epsilon_{T+1} + \epsilon/2 = \epsilon$ .  $\blacksquare$

## 5.2 Efficient implementation for soft margin

In this section we provide an efficient procedure for calculating the distribution  $\mathbf{d}_t$  as described in Fig. 1 when  $f(\mathbf{d})$  is the indicator function of  $\{\mathbf{d} : \|\mathbf{d}\|_\infty \leq \nu\}$ . As we showed above, this case corresponds to the maximization of the soft margin.

We first present a lemma that provides us with an alternative method for finding  $\mathbf{d}$ , which is based on Bregman divergences. The Bregman divergence with respect to a convex function  $h$  between two vectors  $\mathbf{d}$  and  $\mathbf{d}_0$  is defined as,

$$B_h(\mathbf{d}||\mathbf{d}_0) = h(\mathbf{d}) - h(\mathbf{d}_0) - \langle \nabla h(\mathbf{d}_0), \mathbf{d} - \mathbf{d}_0 \rangle .$$

See [CZ97] for a rigorous definition of the Bregman divergence.

**Lemma 12** *Let  $h : S \rightarrow \mathbb{R}$  be a strongly convex and differentiable function, let  $f$  be a convex function, and denote  $\hat{f} = h + f$ . Let  $\theta$  be a vector and denote  $\mathbf{d}_0 = \nabla h^*(\theta)$ , where  $h^*$  is the Fenchel conjugate of  $h$ . Then,*

$$\nabla \hat{f}^*(\theta) = \underset{\mathbf{d}}{\operatorname{argmin}} (B_h(\mathbf{d}||\mathbf{d}_0) + f(\mathbf{d})) .$$

**Proof:** Since  $h$  is strongly convex and differentiable we have that  $\nabla h(\mathbf{d}_0) = \theta$ . Therefore,

$$\begin{aligned} \nabla \hat{f}^*(\theta) &= \underset{\mathbf{d}}{\operatorname{argmax}} \langle \mathbf{d}, \theta \rangle - \hat{f}(\mathbf{d}) \\ &= \underset{\mathbf{d}}{\operatorname{argmin}} h(\mathbf{d}) - \langle \mathbf{d}, \theta \rangle + f(\mathbf{d}) \\ &= \underset{\mathbf{d}}{\operatorname{argmin}} h(\mathbf{d}) - \langle \mathbf{d}, \nabla h(\mathbf{d}_0) \rangle + f(\mathbf{d}) \\ &= \underset{\mathbf{d}}{\operatorname{argmin}} B_h(\mathbf{d}||\mathbf{d}_0) + f(\mathbf{d}) . \end{aligned}$$

$\blacksquare$

Applying the above lemma with  $f = I_C$  for some convex set  $C$  we obtain the following corollary.

**Corollary 13** *Assume that the conditions stated in Lemma 12 hold and that  $f(\mathbf{d}) = I_C(\mathbf{d})$  for some convex set  $C$ . Then,*

$$\nabla (h + f)^*(\theta) = \underset{\mathbf{d} \in C}{\operatorname{argmin}} B_h(\mathbf{d}||\nabla h^*(\theta)) .$$

We now get back to the problem of finding  $\mathbf{d}_t$  when  $f(\mathbf{d})$  is  $I_C(\mathbf{d})$  for  $C = \{\mathbf{d} : \|\mathbf{d}\|_\infty \leq \nu\}$ . Based on Corollary 13 we can first define the distribution vector  $\mathbf{d}_0$  such that  $\mathbf{d}_{0,i} \propto \exp(-\frac{1}{\beta}(A\mathbf{w}_t)_i)$  and then set

$$\mathbf{d}_t = \underset{\mathbf{d} \in \mathbb{S}^m : \|\mathbf{d}\|_\infty \leq \nu}{\operatorname{argmin}} B_h(\mathbf{d}||\mathbf{d}_0) . \quad (12)$$

We are therefore left with the problem of solving the entropic projection problem given in Eq. (12). A similar problem was tackled by Herbster and Warmuth [HW01], who provided  $O(m \log(m))$  and  $O(m)$  algorithms for performing entropic projections. For completeness, in the rest of this section we outline the simpler  $O(m \log(m))$  algorithm. To do so, we first show that the entropic projection preserves the relative order of components of the projected vector.

**Lemma 14** *Let  $\mathbf{d}_t$  be the solution of Eq. (12) and let  $i, j$  be two indices such that  $d_{0,i} > d_{0,j}$ . Then,  $d_{t,i} \geq d_{t,j}$ .*

```

INPUT: A vector  $\mathbf{d}_0 \in \mathbb{S}^m$  and a scalar  $\nu \in (0, 1)$ 
Sort  $\mathbf{d}_0$  in non-increasing order  $\Rightarrow \mathbf{u}$ 
INITIALIZE:  $Z = \sum_{r=1}^m u_r$ 
FOR  $i = 1, \dots, m$ 
   $\theta = \frac{1 - \nu(i-1)}{Z}$ 
  IF  $\theta u_i \leq \nu$ 
    BREAK
  ENDFOR
   $Z \leftarrow Z - u_i$ 
ENDFOR
OUTPUT:  $\mathbf{d}_t$  s.t.  $d_{t,r} = \min\{\nu, \theta d_{0,r}\}$ 

```

Figure 2: An  $O(m \log(m))$  Procedure for solving the Entropic Projection problem defined by Eq. (12).

**Proof:** Assume that the claim of the proof is not true. Let  $i$  and  $j$  be two indices which violate the claim, therefore  $d_{t,i} < d_{t,j}$ . We now construct a vector  $\tilde{\mathbf{d}}$  which resides in  $\mathbb{S}^m$  and whose components do not exceed  $\nu$ . We set all the components of  $\tilde{\mathbf{d}}$ , except for the  $i$ th and  $j$ th components, to be equal to the corresponding components of  $\mathbf{d}_t$ . Next, we set  $\tilde{d}_{t,i} = d_{t,j}$  and  $\tilde{d}_{t,j} = d_{t,i}$ . Clearly,  $\tilde{\mathbf{d}}$  constitutes a feasible solution. Taking the difference between the Bregman divergence of the two vectors each to  $\mathbf{d}_0$  we get,

$B_h(\mathbf{d}_t \| \mathbf{d}_0) - B_h(\tilde{\mathbf{d}} \| \mathbf{d}_0) = (d_j - d_i) \log(d_{0,i}/d_{0,j}) > 0$ , which contradicts the fact that  $\mathbf{d}_t$  is the vector attaining the smallest Bregman divergence to  $\mathbf{d}_0$ . ■

Without loss of generality, assume that  $\mathbf{d}_0$  is sorted in a non-increasing order. Therefore, using Lemma 14 we know that  $\mathbf{d}_t$  has the form  $(\nu, \dots, \nu, d_{t,i}, \dots, d_{t,j}, 0, \dots, 0)$  where for each  $r \in \{i, \dots, j\}$  we have  $d_{t,r} \in (0, \nu)$ . Moreover, the following lemma provides us with a simple way to find all the rest of the elements of  $\mathbf{d}_t$ .

**Lemma 15** Assume that  $\mathbf{d}_0$  is sorted in a non-increasing order and that  $\mathbf{d}_t = (\nu, \dots, \nu, d_{t,i}, \dots, d_{t,j}, 0, \dots, 0)$ . Then, for all  $r \in \{i, \dots, j\}$  we have

$$d_{t,r} = \theta d_{0,r} \text{ where } \theta = \frac{1 - \nu(i-1)}{\sum_{r=i}^j d_{0,r}}.$$

**Proof:** Let  $\mathbf{v}$  denotes the gradient of  $B_h(\mathbf{d} \| \mathbf{d}_0)$  with respect to  $\mathbf{d}$  at  $\mathbf{d}_t$ , namely,

$$v_i = \log(d_{t,i}) + 1 - \log(d_{0,i}).$$

Let  $I = \{i, \dots, j\}$ . Note that for the elements in  $I$  the optimization problem has a single linear equality constraint and the solution is in the interior of the set  $(0, \nu)^{|I|}$ . Therefore, using Corollary 2.1.3 in [BL06] we obtain that there exists a constant  $\theta'$  such that for all  $i \in I$ ,  $v_i = \theta' - 1$  or equivalently

$$\forall i \in I, d_{t,i} = d_{t,0} e^{\theta' - 1}.$$

Let us denote  $\theta = e^{\theta' - 1}$ . Using this form in the equation  $\sum_i d_{t,i} = 1$  we get that,

$$1 = \sum_{r=1}^m d_{t,r} = \nu(i-1) + \theta \sum_{r=i}^j d_{0,r},$$

which immediately yields that  $\theta$  attains the value stated in the lemma. ■

We are left with the problem of finding the indices  $i$  and  $j$ . The next lemma tells us that not a single element of the optimal vector attains a value of zero.

**Lemma 16** Assume that the vector  $\mathbf{d}_0$  is provided in a non-increasing order of elements and that all of its elements are positive. Then, the optimal solution of Eq. (12) is of the form,  $(\nu, \dots, \nu, d_{t,i}, \dots, d_{t,m})$  where  $d_{t,m} > 0$ .

**Proof:** Plugging the value of  $\theta$  from the previous lemma into the objective function and performing simple algebraic manipulations we obtain the following objective value,

$$B_h(\mathbf{d}_t \| \mathbf{d}_0) = \sum_{r=1}^{i-1} \nu \log\left(\frac{\nu}{d_{0,r}}\right) + (1 - \nu(i-1)) \log(\theta).$$

Therefore, the objective is monotonically increasing in  $\theta$ . This in turn implies that we should set  $\theta$  to be as small as possible in order to find the minimal Bregman divergence. Next, note that the value of  $\theta$  as defined in Lemma 15 is decreasing as a function of  $j$ . The optimal solution is obtained for  $j = m$ . ■

Finally, we are left with the task of finding the index  $i$ . Once it is found we readily obtain  $\theta$ , which immediately translates into a closed form solution for  $\mathbf{d}_t$ . Lemma 14 in conjunction with a property presented in the sequel, implies that the *first* index for which  $\mathbf{d}_t$ , as defined by Lemma 15 with  $j = m$ , constitutes the optimal index for  $i$ . The pseudo-code describing the resulting efficient procedure for solving the problem in Eq. (12) is given in Fig. 2. The algorithm starts by sorting the vector  $\mathbf{d}_0$ . Then, it checks each possible index  $i$  of the sorted vector as the position to stop capping the weights. More formally, given an index  $i$  the algorithm checks whether  $\mathbf{d}_t$  can take the form  $(\nu, \dots, \nu, d_{t,i}, \dots, d_{t,m})$  where  $d_{t,i} < \nu$ . To check each index  $i$  the algorithm calculates  $\theta$  as given by Lemma 15. The same lemma also implies that  $d_{t,i} = \theta d_{0,i}$ . Thus, if the assumption on the index  $i$  is correct, the following inequality must hold,  $\nu > d_{t,i} = \theta d_{0,i}$ . In case the index  $i$  under examination indeed satisfies the inequality the algorithm breaks out of the loop. Therefore, the algorithm outputs the feasible solution with the smallest number of weights at the bound  $\nu$ . It thus remains to verify that the feasible solution with the smallest number of capped weights is indeed optimal. This property follows from a fairly straightforward yet tedious lemma which generalizes Lemma 3 from [SSS06b] and is thus omitted. Note also that the time complexity of the resulting algorithm is  $O(m \log(m))$  which renders it applicable to boosting-based applications with large datasets.

## 6 Discussion

The starting point of this paper was an alternative view of the equivalence of weak-learnability and linear-separability. This view lead us to derive new relaxations of the notion of margin, which are useful in the noisy non-separable case. In turn, the new relaxations of the margin motivated us to derive new boosting algorithms which maintain distributions over

the examples that are restricted to a subset of the simplex. There are a few future direction research we plan to pursue. First, we would like to further explore additional constraints of the distribution  $\mathbf{d}_t$ , such as adding  $\ell_2$  constraints. We also would like to replace the relative entropy penalty for the distribution  $\mathbf{d}_t$  with binary entropies of each of the components of  $\mathbf{d}_t$  with respect to the two dimensional vector  $(\frac{1}{2}, \frac{1}{2})$ . The result is a boosting-based apparatus for the log-loss. Last, we would like to explore alternative formalisms for the primal problem that also modify the definition of the function  $g(\mathbf{d}) = \|\mathbf{d}^\dagger A\|_\infty$ , which may lead to a regularization term of the vector  $\mathbf{w}$  rather than the domain constraint we currently have.

## A Technical lemmas

The first lemma states a sufficient condition under which the Fenchel-Young inequality holds with equality. Its proof can be found in ([BL06], Proposition 3.3.4).

**Lemma 17** *Let  $f$  be a closed and convex function and let  $\partial f(\mathbf{w})$  be its differential set at  $\mathbf{w}$ . Then, for all  $\boldsymbol{\theta} \in \partial f(\mathbf{w})$  we have,  $f(\mathbf{w}) + f^*(\boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \mathbf{w} \rangle$ .*

The next lemma underscores the importance of strongly convex functions. The proof of this lemma follows from Lemma 18 in [SS07].

**Lemma 18** *Let  $f$  be a closed and  $\sigma$ -strongly convex function over  $S$  with respect to a norm  $\|\cdot\|$ . Let  $f^*$  be the Fenchel conjugate of  $f$ . Then,  $f^*$  is differentiable and its gradient satisfies  $\nabla f^*(\boldsymbol{\theta}) = \arg \max_{\mathbf{w} \in S} \langle \mathbf{w}, \boldsymbol{\theta} \rangle - f(\mathbf{w})$ . Furthermore, for all  $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^n$ , we have*

$$f^*(\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2) - f^*(\boldsymbol{\theta}_1) \leq \langle \nabla f^*(\boldsymbol{\theta}_1), \boldsymbol{\theta}_2 \rangle + \frac{1}{2\sigma} \|\boldsymbol{\theta}_2\|_*^2$$

**Lemma 19** *Let  $f, g$  be two functions and assume that for all  $\mathbf{w} \in S$  we have  $g(\mathbf{w}) \geq f(\mathbf{w}) \geq g(\mathbf{w}) - c$  for some constant  $c$ . Then,  $g^*(\boldsymbol{\theta}) \leq f^*(\boldsymbol{\theta}) \leq g^*(\boldsymbol{\theta}) + c$ .*

**Proof:** There exists some  $\mathbf{w}'$  s.t.

$$\begin{aligned} g^*(\boldsymbol{\theta}) &= \langle \mathbf{w}', \boldsymbol{\theta} \rangle - g(\mathbf{w}') \\ &\leq \langle \mathbf{w}', \boldsymbol{\theta} \rangle - f(\mathbf{w}') \\ &\leq \max_{\mathbf{w}} \langle \mathbf{w}, \boldsymbol{\theta} \rangle - f(\mathbf{w}) = f^*(\boldsymbol{\theta}). \end{aligned}$$

This proves the first inequality. The second inequality follows from the fact that the conjugate of  $g(\mathbf{w}) - c$  is  $g^*(\boldsymbol{\theta}) + c$ . ■

**Lemma 20** *Let  $1 \geq \epsilon_1 \geq \epsilon_2 \geq \dots$  be a sequence such that for all  $t \geq 1$  we have  $\epsilon_t - \epsilon_{t+1} \geq r \epsilon_t^2$  for some constant  $r \in (0, 1/2)$ . Then, for all  $t$  we have  $\epsilon_t \leq \frac{1}{r(t+1)}$ .*

**Proof:** We prove the lemma by induction. First, for  $t = 1$  we have  $\frac{1}{r(t+1)} = \frac{1}{2r} \geq 1$  and the claim clearly holds. Assume that the claim holds for some  $t$ . Then,

$$\epsilon_{t+1} \leq \epsilon_t - r\epsilon_t^2 \leq \frac{1}{r(t+1)} - \frac{1}{r(t+1)^2}, \quad (13)$$

where we used the fact that the function  $x - rx^2$  is monotonically increasing in  $[0, 1/(2r)]$  along with the inductive assumption. We can rewrite the right-hand side of Eq. (13) as

$$\frac{1}{r(t+2)} \left( \frac{(t+1)+1}{t+1} \cdot \frac{(t+1)-1}{t+1} \right) = \frac{1}{r(t+2)} \left( \frac{(t+1)^2-1}{(t+1)^2} \right).$$

The term  $\frac{(t+1)^2-1}{(t+1)^2}$  is smaller than 1 and thus  $\epsilon_{t+1} \leq \frac{1}{r(t+2)}$ , which concludes our proof. ■

## B Fenchel conjugate pairs

We now list a few useful Fenchel-conjugate pairs. Proofs can be found in ([BV04] Section 3.3, [BL06] Section 3.3., [SS07] Section A.3).

$f(\mathbf{d})$	$f^*(\boldsymbol{\theta})$
$I_C(\mathbf{d})$ for $C = \{\mathbf{d} : \ \mathbf{d}\  \leq \nu\}$	$\nu \ \boldsymbol{\theta}\ _*$
$I_{\mathbb{S}^m}(\mathbf{d})$	$\max_i \theta_i$
$I_{\mathbb{S}^m}(\mathbf{d}) + \sum_{i=1}^m d_i \log(\frac{d_i}{1/m})$	$\log(\frac{1}{m} \sum_{i=1}^m e^{\theta_i})$
$\frac{1}{2} \ \mathbf{d}\ ^2$	$\frac{1}{2} \ \boldsymbol{\theta}\ _*^2$
$c f(\mathbf{d})$ for $c > 0$	$c f^*(\boldsymbol{\theta}/c)$
$f(\mathbf{d} + \mathbf{d}_0)$	$f^*(\boldsymbol{\theta}) - \langle \boldsymbol{\theta}, \mathbf{d}_0 \rangle$
$f(c \mathbf{d})$ for $c \neq 0$	$f^*(\boldsymbol{\theta}/c)$

## References

- [BL06] J. Borwein and A. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer, 2006.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [CSS02] M. Collins, R.E. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 47(2/3):253–285, 2002.
- [CST00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
- [CZ97] Y. Censor and S.A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, New York, NY, USA, 1997.
- [DW00] C. Domingo and O. Watanabe. Madaboost: A modification of adaboost. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, 2000.
- [Fre01] Y. Freund. An adaptive version of the boost by majority algorithm. *Machine Learning*, 43(3):293–318, 2001.

- [FS96] Y. Freund and R.E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 325–332, 1996.
- [FS99] Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.
- [HW01] M. Herbster and M. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.
- [KPL01] V. Koltchinskii, D. Panchenko, and F. Lozano. Some new bounds on the generalization error of combined classifiers. In *Advances in Neural Information Processing Systems 14*, 2001.
- [MBB98] Llew Mason, Peter Bartlett, and Jonathan Baxter. Direct optimization of margins improves generalization in combined classifiers. Technical report, Department of Systems Engineering, Australian National University, 1998.
- [MR03] R. Meir and G. Rätsch. An introduction to boosting and leveraging. In S. Mendelson and A. Smola, editors, *Advanced Lectures on Machine Learning*, pages 119–184. Springer, 2003.
- [RSD07] C. Rudin, R.E. Schapire, and I. Daubechies. Analysis of boosting algorithms using the smooth margin function. *Annals of Statistics*, 2007.
- [RW05] G. Ratsch and M. Warmuth. Efficient margin maximizing with boosting. *Journal of Machine Learning Research*, pages 2153–2175, 2005.
- [Sch90] R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
- [Sch03] R.E. Schapire. The boosting approach to machine learning: An overview. In D.D. Denison, M.H. Hansen, C. Holmes, B. Mallick, and B. Yu, editors, *Nonlinear Estimation and Classification*. Springer, 2003.
- [Ser03] R.A. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003.
- [SFBL97] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Machine Learning: Proceedings of the Fourteenth International Conference*, pages 322–330, 1997. To appear, *The Annals of Statistics*.
- [SS07] S. Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University, 2007.
- [SSS06a] S. Shalev-Shwartz and Y. Singer. Convex repeated games and fenchel duality. In *Advances in Neural Information Processing Systems 20*, 2006.
- [SSS06b] S. Shalev-Shwartz and Y. Singer. Efficient learning of label ranking by soft projections onto polyhedra. *Journal of Machine Learning Research*, 7 (July):1567–1599, 2006.
- [SSS07] S. Shalev-Shwartz and Y. Singer. A primal-dual perspective of online learning algorithms. *Machine Learning Journal*, 2007.
- [SSWB98] B. Schölkopf, A. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. Technical Report NC2-TR-1998-053, NeuroColt2, 1998.
- [SVL07] A. Smola, S.V.N. Vishwanathan, and Q. Le. Bundle methods for machine learning. In *Advances in Neural Information Processing Systems 21*, 2007.
- [vN28] J. von Neumann. Zur theorie der gesellschaftsspiele (on the theory of parlor games). *Math. Ann.*, 100:295–320, 1928.
- [WGR07] M. Warmuth, K. Glocer, and G. Ratsch. Boosting algorithms for maximizing the soft margin. In *Advances in Neural Information Processing Systems 21*, 2007.
- [WLR06] M. Warmuth, J. Liao, and G. Ratsch. Totally corrective boosting algorithms that maximize the margin. In *Proceedings of the 23rd international conference on Machine learning*, 2006.
- [Zha03] T. Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Transaction on Information Theory*, 49:682–691, 2003.



---

# Adaptive Aggregation for Reinforcement Learning with Efficient Exploration: Deterministic Domains

---

Andrey Bernstein\*

Department of Electrical Engineering  
Technion – Israel Institute of Technology  
Haifa, Israel  
andreyb@tx.technion.ac.il

Nahum Shimkin

Department of Electrical Engineering  
Technion – Israel Institute of Technology  
Haifa, Israel  
shimkin@ee.technion.ac.il

## Abstract

We propose a model-based learning algorithm, the Adaptive Aggregation Algorithm (AAA), that aims to solve the online, continuous state space reinforcement learning problem in a deterministic domain. The proposed algorithm uses an adaptive state aggregation approach, going from coarse to fine grids over the state space, which enables to use finer resolution in the “important” areas of the state space, and coarser resolution elsewhere. We consider an on-line learning approach, in which we discover these important areas on-line, using an uncertainty intervals exploration technique. Polynomial learning rates in terms of mistake bound (in a PAC framework) are established for this algorithm, under appropriate continuity assumptions.

## 1 Introduction

Markov Decision Processes (MDPs) provide a standard framework for handling control and sequential decision making tasks under uncertainty ([4, 17]). Solid theory and a variety of algorithms enable the efficient computation of optimal control policies in MDPs when the state and action spaces are finite. However, an exact solution becomes intractable when the number of states is large or infinite. In this case, some approximation schemes are required. See [4] and [6] for a thorough discussion. Also, see such recent works as [1, 11, 14].

One natural approximation approach is *state aggregation*, in which the state space is discretized into a (relatively small) finite collection of *cells*. Each cell is said to *aggregate* the states that fall in this cell. Once the aggregation is performed, the new problem is a planning problem in a reduced state space, which can be solved by regular techniques. The main question that arises here is how to perform the aggregation, so that, on the one hand, obtain a “good” approximation of an optimal policy, and on the other hand minimize the problem complexity. This question was addressed by many works, such as [21, 9], which provide formal answers under some continuity assumptions on the model parameters.

---

\*We would like to thank the reviewers for their useful comments on this paper.

An extra difficulty is added when dealing with *learning problems*, namely situations where the model of the MDP is initially unknown. Reinforcement Learning (RL) encompasses a wide range of techniques for solving this problem by interacting with the environment. An important part of an RL algorithm is the exploration scheme. The role of exploration is to gain new information by appropriate action selection which directs the agent towards unknown states of the MDP.

Recently, efficient learning algorithms were presented and proved to learn nearly optimal behavior (with high probability) within a time or error bound that is polynomial in the problem size. These include the  $E^3$  [13], R-MAX [7], MBIE [18], GCB [8], UCRL [2] and OLP [20] algorithms. These algorithms use efficient exploration techniques, which are often based on the so-called “optimism in face of uncertainty” principle. However, these algorithms are infeasible in cases where the state and/or action spaces are very large or infinite, since their time and space complexity is typically polynomial in the size of the space.

On the other hand, most of the existing algorithms for solving “large” problems that rely on state aggregation, are heuristic in nature, without formal guarantees. These include such works on adaptive aggregation as [15], [16] and [5]. An exception is the algorithm proposed by Diuk et al. [10], which uses R-MAX as the basis. However, this algorithm requires a specific structure of the problem; otherwise, its total mistake bound is polynomial in the size of the state space, which can be very large or infinite.

In this paper we focus on the online reinforcement learning problem in MDPs with very large or infinite state space, finite action space, discounted return criterion, and with *deterministic* dynamics and rewards. For concreteness we will focus on the continuous state case; however our schemes and results also apply to the discrete case, where the number of states is very large or countably infinite. The proposed algorithms use an *adaptive state aggregation* approach, going from coarse to fine grids over the state space, which enables to use finer resolution in the “important” areas of the state space, and coarser resolution elsewhere. We consider an on-line learning approach, in which we discover these important areas on-line, using an *uncertainty intervals* exploration technique<sup>1</sup>. Certain continuity assumptions on the ba-

---

<sup>1</sup>Our uncertainty intervals are the analogue of the *confidence intervals* used in the stochastic case [18]. However, the origin of

basic model parameters will be imposed. Such assumptions are essential for generalization in continuous state space, especially when using the state aggregation approach.

The principle that governs our basic scheme is simply *to split frequently visited cells*. The idea behind this principle is as follows. As time progresses, we will visit cells that are “close” to the optimal trajectory; on the optimal trajectory, we need high resolution. Perhaps surprisingly, this principle is not sufficient to obtain theoretical results. Consequently, we will propose an improved variant of the basic algorithm, for which learning rates in terms of total mistake bound (see below) will be established. In this variant, in addition to splitting the visited cells, we also split cells that the algorithm “could have” visited (according to the uncertainty in the model of the MDP that was learned so far).

We will use the *total mistake bound* as a performance metric for our algorithms. This metric counts the total number of time-steps in which the algorithm’s implemented policy is strictly suboptimal from the current state. This metric has been used in a number of recent works on on-line learning in discounted MDP problems<sup>2</sup> [12, 18, 19]. In our case we will establish two types of mistake bounds, which we call the *prior bound* and the *posterior bound*. The first type ensures that our algorithm is not worse than a non-adaptive algorithm, which uses a single *uniformly dense* grid. In this case, our mistake bound is thus polynomial in the number of cells in this grid. The second type ensures that the mistake bound is polynomial in the number of cells in the *actually used* grid.

In our analysis, we need to distinguish between two cases: The “contractive” case, characterized by  $\gamma\beta < 1$ , where  $\gamma$  is the MDP discount factor and  $\beta$  is a (Lipschitz) continuity parameter of the transition function (cf. equation (6b)); and the “expansive” case, where  $\gamma\beta > 1$ . Due to space constraints, we treat here in detail the former, while the results for the latter are presented without proofs, which can be found in [3].

The paper is structured as follows. In Section 2 we present the model and the notation. In Section 3 we introduce some further definitions and assumptions. In Section 4 we propose a basic version of our AAA algorithm, while Section 5 presents an improved variant of the algorithm that is required for its convergence. In Section 6 polynomial bounds on the total mistake count of this improved algorithm are presented for the “contractive” case, while Section 7 proves these bounds. In Section 8 we provide results for the “expansive” case, without proof. Finally, conclusions and future work are presented in Section 9.

## 2 Model and Performance Metrics

We denote a deterministic MDP by the 4-tuple  $M = (\mathbb{X}, \mathbb{A}, f, r)$ , where  $\mathbb{X}$  is a state space,  $\mathbb{A}$  is an action space,  $f(x, a)$  is the transition function which specifies the next state  $x' \in \mathbb{X}$  given the previous state  $x \in \mathbb{X}$  and action  $a \in \mathbb{A}$ , and  $r(x, a) \in [r_{min}, r_{max}]$  is the immediate reward function

uncertainty in our model is the (deterministic) aggregation error, rather than stochastic sampling error.

<sup>2</sup>These works refer to this metric as the *sample complexity of exploration*.

which specifies the reward of performing action  $a \in \mathbb{A}$  in state  $x \in \mathbb{X}$ .

Let  $d : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  be a fixed metric on  $\mathbb{X}$ . We assume the following regarding the state and action spaces.

### Assumption 1

1. The action space  $\mathbb{A}$  is a finite set.
2. The state space  $\mathbb{X}$  is a bounded subset of  $\mathbb{R}^n$ . That is, there exists a constant  $\Delta_{max} < \infty$  such that for all  $x, x' \in \mathbb{X}$ ,  $d(x, x') \leq \Delta_{max}$ .

The MDP  $M$  is used to model an environment, or a dynamic system, with which a learning agent interacts. The interaction proceeds as follows: At time  $t$  the agent observes the state  $x_t \in \mathbb{X}$ , chooses an action  $a_t \in \mathbb{A}$ , receives a reward  $r_t = r(x_t, a_t)$ , and the process moves to state  $x_{t+1} = f(x_t, a_t)$ .

Let  $h_t \triangleq \{x_0, a_0, x_1, a_1, \dots, x_{t-1}, a_{t-1}, x_t\}$  denote the *history* of observed states and actions, that is available to the agent at time  $t$  to make its choice of  $a_t$ . Also, let  $\mathbb{H}_t \triangleq (\mathbb{X} \times \mathbb{A})^t \times \mathbb{X}$  denote the space of all possible histories up to time  $t$ . Then, at each time  $t$ , the agent makes its decision according to some *decision rule*  $\pi_t : \mathbb{H}_t \rightarrow \mathbb{A}$ , so that  $a_t = \pi_t(h_t)$ ,  $t \geq 0$ . The collection  $\pi = \{\pi_t\}_{t=0}^{\infty}$  is the *control policy*. A policy is *stationary* if the decision rule does not change over time, and depends only on the last state observed. We shall slightly abuse notation and identify the stationary policy  $\pi$  with the map  $\pi : \mathbb{X} \rightarrow \mathbb{A}$ , so that at each time  $t$ ,  $a_t = \pi(x_t)$ .

In this paper we focus on the *discounted return criterion*. For a given initial state  $x_0 = x$ , we denote the infinite horizon discounted return of state  $x$ , for a given policy  $\pi$ , in MDP  $M$ , by

$$J_M^\pi(x) \triangleq \sum_{t=0}^{\infty} \gamma^t r(x_t, \pi_t(h_t)),$$

where  $0 < \gamma < 1$  is the *discount factor*. The optimal return is denoted by  $V_M(x) \triangleq \sup_{\pi} J_M^\pi(x)$ , which is also called *the optimal value function*. We often drop  $M$  from the notation above, if it does not cause confusion. A policy  $\pi$  is *optimal* if  $J^\pi(x) = V(x)$  holds for all  $x \in \mathbb{X}$ . For any  $\epsilon > 0$ , a policy  $\pi$  is  $\epsilon$ -*optimal* if  $J^\pi(x) \geq V(x) - \epsilon$  holds for all  $x \in \mathbb{X}$ .

It is well known ([17]) that the optimal value function satisfies Bellman’s equation

$$V(x) = \max_{a \in \mathbb{A}} \{r(x, a) + \gamma V(f(x, a))\}, \quad x \in \mathbb{X}, \quad (1)$$

and any stationary deterministic policy  $\pi^*$  which satisfies  $\pi^*(x) \in \operatorname{argmax}_{a \in \mathbb{A}} \{r(x, a) + \gamma V(f(x, a))\}$ ,  $x \in \mathbb{X}$ , is an optimal policy. Let  $Q(x, a) \triangleq r(x, a) + \gamma V(f(x, a))$  denote the action-value function, or  $Q$ -function, which provides the return of choosing an action  $a$  in state  $x$ , and then following an optimal policy. Also, we let  $V_{max} \triangleq \frac{r_{max}}{1-\gamma}$ . Note that  $V_{max}$  is the maximal possible discounted return of any policy.

Our main performance metric will be *mistake bound* (or *policy-mistake bound*), introduced for RL in [12]. It counts the number of time steps  $t$  in which the algorithm executes a not  $\epsilon$ -optimal policy from the current state,  $x_t$ . Specifically, let  $\pi_t$  be the decision rule that the algorithm uses at time

$t$  to choose its action. Then, given  $h_t, \mathcal{A}_t = \{\pi_k\}_{k=t}^\infty$  is a (non-stationary) policy that the algorithm implements at time  $t$ , and  $\sum_{k=t}^\infty \gamma^{k-t} r_k \triangleq J^{\mathcal{A}_t}(x_t)$  can be interpreted as the return of this policy from time  $t$  onward, where  $r_k = r(x_k, \pi_k(h_k))$  and  $x_{k+1} = f(x_k, \pi_k(h_k))$ . Now, the policy-mistake count is defined as

$$\text{PM}(\epsilon) \triangleq \sum_{t=0}^{\infty} \mathbb{I} \{ J^{\mathcal{A}_t}(x_t) < V(x_t) - \epsilon \}. \quad (2)$$

For deterministic domains with finite state-space, we have the following near-optimality criterion.

### The Policy-Mistake Bound Criterion

A learning algorithm is PAC (Probably Approximately Correct) if there exists a polynomial

$$B = B \left( |\mathbb{X}|, |\mathbb{A}|, \frac{1}{1-\gamma}, \frac{1}{\epsilon} \right)$$

such that for all  $\epsilon > 0$ ,  $\text{PM}(\epsilon) < B$ .

Note, that while the ‘‘probably’’ aspect is absent in our deterministic case, we will find it convenient to keep the PAC terminology here.

A possible alternative to the policy-mistake count is the action-mistake count, defined as follows:

$$\text{AM}(\epsilon) \triangleq \sum_{t=0}^{\infty} \mathbb{I} \{ Q(x_t, a_t) < V(x_t) - \epsilon \}. \quad (3)$$

This criterion counts the number of sub-optimal actions, that is, the number of times that an algorithm executed an action whose action-value is  $\epsilon$ -inferior to the optimal value. It is easily verified (see Corollary 1 in [3]) that policy-mistake count is a stronger criterion, in the sense that  $\text{AM}(\epsilon) \leq \text{PM}(\epsilon)$ . Hence we focus here on the former.

In the above definition, the bound  $B$  depends on the number of states  $|\mathbb{X}|$ . In case  $|\mathbb{X}|$  is infinite, some other measures of  $\mathbb{X}$  must be considered. As already mentioned, in our case we will replace  $|\mathbb{X}|$  by the number of cells in sufficiently dense grid over the state space.

## 3 Preliminaries

### a. Grid-Cell Notation

A grid  $\mathbb{S}$  over the state space  $\mathbb{X}$  is a partition of  $\mathbb{X}$  into disjoint elements that covers the whole of  $\mathbb{X}$ . We call any  $s \in \mathbb{S}$  a cell. We say that a grid  $\mathbb{S}_2$  is a refinement of a grid  $\mathbb{S}_1$ , if for every cell  $s \in \mathbb{S}_2$  there exists  $s' \in \mathbb{S}_1$ , such that  $s \subseteq s'$ . We denote this relation by  $\mathbb{S}_2 \preceq \mathbb{S}_1$ . For any sets  $A$  and  $B$  of cells, we define the *intersection operator* between these two sets as

$$A \wedge B \triangleq \{s_A \cap s_B : s_A \in A, s_B \in B\} \setminus \{\emptyset\}. \quad (4)$$

For a given cell  $s \in \mathbb{S}$ , let  $\Delta(s) \triangleq \sup_{x, x' \in s} d(x, x')$  denote the cell size (or *diameter*) in the given metric  $d$ . For two given cells  $s, s' \in \mathbb{S}$ , we define the *biased* distance between these cells as

$$d_b(s, s') \triangleq \begin{cases} \inf_{x \in s, x' \in s'} d(x, x'), & s \neq s', \\ -\Delta(s), & s = s'. \end{cases} \quad (5)$$

This definition will be justified in Section 5 (see Definition 3 and the remark after this Definition). Of course  $d_b(s, s')$  is not a distance in a regular sense, since it can be negative. Also, for given state  $x \in \mathbb{X}$ , let  $s_x \in \mathbb{S}$  be the cell that includes  $x$  ( $x \in s_x$ ). Then, given a cell  $s \in \mathbb{S}$ , we define the biased state-to-cell distance  $d_b(s, x) \triangleq d_b(s, s_x)$ .

### b. Feasible Splitting Schemes and Grids

Given a source grid  $\mathbb{S}_1$  and a candidate  $s \in \mathbb{S}_1$  to split, a splitting scheme tells us how to split  $s$  into  $c_{split}$  cells  $s_1, \dots, s_{c_{split}}$ , with  $s_i \cap s_j = \emptyset$ ,  $\cup_i s_i = s$ , to form a refined (target) grid  $\mathbb{S}_2 \preceq \mathbb{S}_1$ . We are interested in splitting schemes that decrease the size  $\Delta(s)$  of a cell  $s$ . More formally, we require the following condition on the splitting scheme.

**Definition 1 (Feasible Splitting Scheme)** A splitting scheme with splitting coefficient  $c_{split}$  is feasible if there exists  $0 < \lambda < 1$  (independent of  $s$ ) such that  $\Delta(s_i) \leq \lambda \Delta(s)$  for  $i = 1, \dots, c_{split}$ .

Now, given a fixed feasible splitting scheme and an initial grid  $\mathbb{S}_0$  over the state space  $\mathbb{X}$ , we define the set of *feasible grids* as the set of all grids that can be obtained by using this given scheme starting from  $\mathbb{S}_0$ .

### c. Continuity Assumption

The following continuity assumption will be imposed on the basic model parameters.

**Assumption 2** There exist constants  $\alpha > 0$  and  $\beta > 0$  such that, for all  $x_1, x_2 \in \mathbb{X}$  and  $a \in \mathbb{A}$  it holds that

$$|r(x_1, a) - r(x_2, a)| \leq \alpha \cdot d(x_1, x_2), \quad (6a)$$

$$d(f(x_1, a), f(x_2, a)) \leq \beta \cdot d(x_1, x_2). \quad (6b)$$

A continuity assumption of some kind is obviously essential for generalization in continuous state spaces. Assumptions of similar nature to the one above were used in various works on state aggregation, such as [21, 9]. However, we note that the specific assumptions used in these papers refer to continuity of probability densities. Consequently they are too strong for the continuous deterministic case as they imply that all states are mapped to the same target state.

In this paper we will treat in detail the case  $\gamma\beta < 1$  (where  $\gamma$  is the discount factor), in which there is some ‘‘contraction’’ effect in the system dynamics. Results for the complementary case of  $\gamma\beta > 1$  are presented without proof in Section 8.

We assume that both  $\alpha$  and  $\beta$  are known for the purpose of learning.

## 4 The Basic AAA Algorithm

In this section we present the basic variant of the AAA algorithm, which is directly based on the principle of splitting frequently visited cells. As it turns out, this algorithm may fail in some cases, and therefore no theoretical guarantees will be presented. Instead, we will provide an example, showing the source of the problem. This will provide the motivation for the improved scheme in the next section.

In the following subsections we present the different parts of this algorithm in detail. An outline of the complete algorithm is presented as Algorithm 1.

---

**Algorithm 1** Basic Adaptive Aggregation Algorithm (outline)

---

**Input parameters:**

- Maximal reward  $r_{max}$ ,
- Lipschitz continuity parameters  $\alpha$  and  $\beta$ ,
- Count threshold  $\mathcal{N}$ ,
- Cell size threshold  $\Delta_\epsilon$ .

**Initialization:**

1. Initialize the grid to some initial grid  $\mathbb{S}_0(a) = \mathbb{S}_0$  for all  $a \in \mathbb{A}$ , and the cell count  $N(s, a) = 0$ , for all  $a \in \mathbb{A}, s \in \mathbb{S}_0$ ;
2. For all  $s \in \mathbb{S}_0(a)$  and  $a \in \mathbb{A}$ , initialize the reward upper bound and the transition uncertainty set:

$$\tilde{r}(s, a) = r_{max}, \quad \text{CI}_f(s, a) = \mathbb{S}_0(a).$$

**For times  $t = 0, 1, 2, \dots$  do:**

1. **Policy Computation:** Algorithm 2
  2. **Policy Execution:** Algorithm 3
  3. **Cell Splitting:** Algorithm 4
- 

**a. Action-Grids and Common Grid**

In our algorithm, we will use a separate grid for every action. This will allow to use a different resolution for each action. We denote by  $\mathbb{S}_t(a)$  the grid that is used by the algorithm at time  $t$  for action  $a$ . We denote by  $\mathbb{S}_t$  the coarsest grid which is a refinement of all  $\mathbb{S}_t(a)$  at time  $t$ . That is

$$\mathbb{S}_t \triangleq \bigwedge_{a \in \mathbb{A}} \mathbb{S}_t(a),$$

where the intersection operator is defined in (4). We call this grid a *common grid* (at time  $t$ ). This grid will be used to compute the value function, while the action-grids are used for empirical model estimation.

**b. Empirical Model**

We use a *single sample* to estimate empirically the reward and transition. Specifically, suppose that we choose action  $a$  in cell  $s$ . We thus obtain the sample  $(x, a, r = r(x, a), x' = f(x, a))$ , with  $x \in s$  and  $x' \in s'$ . We define the empirical model based on this single sample:

$$\hat{r}(s, a) = r, \tag{7}$$

$$\hat{f}(s, a) = s'. \tag{8}$$

Once the sample from  $(s, a)$  is obtained, the model remains unchanged for this pair (until the cell is split).

**c. Uncertainty Intervals and Upper Value Function**

In the AAA algorithm we will use an uncertainty intervals exploration technique as it applies to deterministic systems due to aggregation. Below we present the definition of the uncertainty intervals in case of continuous state space, and how we use them in the algorithm.

At any time  $t$ , and for every  $a \in \mathbb{A}$  and  $s \in \mathbb{S}_t(a)$ , we define the reward uncertainty interval around the empirical reward (7) as<sup>3</sup>:

$$\begin{aligned} \text{CI}_r(s, a) &\triangleq [\underline{r}(s, a), \tilde{r}(s, a)] \\ &= [\hat{r}(s, a) - \alpha\Delta(s), \hat{r}(s, a) + \alpha\Delta(s)] \end{aligned}$$

if the pair  $(s, a)$  was sampled till time  $t$ ; otherwise, the reward uncertainty interval for this pair is inherited from the parent cell. By the continuity Assumption 2, this uncertainty interval satisfies that  $r(x, a) \in \text{CI}_r(s, a), \forall x \in s$ . Also, the transition uncertainty set is defined as:

$$\text{CI}_f(s, a) \triangleq \left\{ s' \in \mathbb{S}_t : d_b(s', \hat{f}(s, a)) \leq \beta\Delta(s) \right\},$$

where  $d_b$  is the biased distance defined in (5). (If the pair  $(s, a)$  was not sampled till time  $t$ , the uncertainty set is inherited from the parent cell as in the reward case). Again, by the continuity assumption, this uncertainty set satisfies:  $f(x, a) \in \text{CI}_f(s, a), \forall x \in s$ . Using this notation, we define the following dynamic programming operator.

**Definition 2** The upper DP operator at time  $t$  for any given function  $g : \mathbb{S}_t \rightarrow \mathbb{R}$  is

$$\mathcal{T}_1 g(s) = \max_{a \in \mathbb{A}} \left\{ \tilde{r}(s, a) + \gamma \max_{s' \in \text{CI}_f(s, a)} g(s') \right\}.$$

Now, using this operator, we define the *upper value function* (UVF) as the solution of the following fixed point equation:

$$\tilde{V}_t(s) = \mathcal{T}_1 \tilde{V}_t(s), s \in \mathbb{S}_t.$$

It can be shown that this equation has a unique solution, which can be found using Value Iteration or linear programming. Moreover, we can show that this solution is indeed an upper bound on the optimal value function  $V$ . In addition, on dense enough grid this solution is also very close to  $V$ . We do not provide here proofs for these claims. However, these claims easily follow from our analysis of the improved algorithm in Section 7.

The policy that is used in the algorithm is now the optimal (or *greedy*) policy with respect to  $\tilde{V}_t(s)$ :

$$\pi_t(s) = \operatorname{argmax}_{a \in \mathbb{A}} \left\{ \tilde{r}(s, a) + \gamma \max_{s' \in \text{CI}_f(s, a)} \tilde{V}_t(s') \right\}, s \in \mathbb{S}_t.$$

This policy is recalculated only when a cell-action pair is visited for the first time, or some cell is split.

We summarize the UVF and policy computation algorithm in Algorithm 2.

**d. Policy Execution**

As we have seen, the decision rule  $\pi_t$  that is used at each time  $t$ , is determined by the UVF, as presented in Algorithm 2 (equation (10)). In addition, in each execution of the decision rule, a new sample is obtained, and the empirical model and the uncertainty intervals of the corresponding cell-action pairs are updated. This process is summarized in Algorithm 3.

---

<sup>3</sup>We drop the time index from most of our notation for ease of exposition.

---

**Algorithm 2** Policy Computation

---

**If the model has been changed** (that is, some cell-action pair has been visited for the first time, or some cell has been split):

1. Compute the UVF over  $\mathbb{S}_t = \bigwedge_{a \in \mathbb{A}} \mathbb{S}_t(a)$  by solving

$$\tilde{V}_t(s) = \mathcal{T}_1 \tilde{V}_t(s), s \in \mathbb{S}_t, \quad (9)$$

where  $\mathcal{T}_1$  is defined in Definition 2.

2. Compute the corresponding optimal policy

$$\pi_t(s) = \operatorname{argmax}_{a \in \mathbb{A}} \left\{ \tilde{r}(s, a) + \gamma \max_{s' \in \text{CI}_f(s, a)} \tilde{V}_t(s') \right\}. \quad (10)$$

If more than one action achieves the maximum, choose the first one in lexicographic order.

**Otherwise**, use the previously computed value and policy:

$$\tilde{V}_t = \tilde{V}_{t-1} \text{ and } \pi_t = \pi_{t-1}.$$

---

### e. Splitting Method

Assume that a fixed feasible splitting scheme is used throughout (cf. Definition 1). Define a count threshold  $\mathcal{N}$ . We will split a cell if the number of visits to it exceeds  $\mathcal{N}$ . In addition to this splitting criterion, we also employ a “stop-splitting” rule, based on the size of the cell. Let  $\Delta_\epsilon$  be a (small) *cell size threshold* parameter. Then, if a cell  $s$  satisfies  $\Delta(s) \leq \Delta_\epsilon$ , it will not be split anymore. Since the number of times that the algorithm encounters a pair with  $\Delta(s) > \Delta_\epsilon$  can be bounded, it follows that the number of different (stationary) policies that the algorithm uses can also be bounded. This will eventually enable us to prove a bound on the policy-mistake count, in Section 7.

Now, under a fixed feasible splitting scheme, we denote by  $\mathbb{S}_\epsilon$  the coarsest feasible grid with  $\Delta(s) \leq \Delta_\epsilon$  for all  $s \in \mathbb{S}_\epsilon$ . We call this grid  *$\epsilon$ -optimal grid*. The number of cells in  $\mathbb{S}_\epsilon$  can be bounded as follows (see Lemma 6 in Section 7):

$$N_\epsilon \triangleq |\mathbb{S}_\epsilon| \leq |\mathbb{S}_0| c_{split} \left( \frac{\Delta_{max}}{\Delta_\epsilon} \right)^{\log_{1/\lambda}(c_{split})}, \quad (13)$$

where  $\mathbb{S}_0$  is the initial grid,  $\lambda$  and  $c_{split}$  are the parameters of the splitting scheme (Definition 1), and  $\Delta_{max}$  is the diameter of the state space (see Assumption 1).

We summarize the splitting process in Algorithm 4. Recall that the complete AAA algorithm is outlined in Algorithm 1.

### f. Why the Basic AAA Scheme Might Fail

To realize the problem, consider some cell  $s$ . The value of the function  $\tilde{V}$  at that cell is computed based on the optimistic next-step cell  $s_1$ :

$$s_1 \triangleq \operatorname{argmax}_{s' \in \text{CI}_f(s, a)} \tilde{V}_t(s') \quad (14)$$

(cf. equation (9) and Definition 2). However, it may happen that the actual process never visits  $s_1$ , but rather some other cell in  $\text{CI}_f(s, a)$ . This may happen irrespectively of

---

**Algorithm 3** Policy Execution

---

- (i) Execute the action  $a_t = \pi_t(s_t)$ , with  $s_t \in \mathbb{S}_t$  being the current common grid cell. For the visited cell-action pair  $(s_t, a_t) = (s_t, a)$ , let  $s \in \mathbb{S}_t(a)$  be the cell in the action-grid that contains  $s_t$ .
  - (ii) Update the counter:  $N(s, a) := N(s, a) + 1$ .
  - (iii) If  $(s, a)$  is visited *for the first time*, compute the model of this pair. Namely,
    - (a) Compute the empirical reward and transition according to equations (7) and (8).
    - (b) Compute the upper reward value
$$\tilde{r}(s, a) := \hat{r}(s, a) + \alpha \Delta(s), \quad (11)$$
    - (c) Compute the transition uncertainty set
$$\text{CI}_f(s, a) := \left\{ s' \in \mathbb{S}_t : d_b(s', \hat{f}(s, a)) \leq \beta \Delta(s) \right\}. \quad (12)$$
    - (d) Save the basic sample  $(x, a, x')$  obtained for this  $(s, a)$ , with  $x = x_t$  and  $x' = x_{t+1}$ .
- 

---

**Algorithm 4** Splitting Algorithm

---

1. Initialize  $\mathbb{S}_{t+1}(a) = \mathbb{S}_t(a)$ , for all  $a \in \mathbb{A}$ .
  2. For each cell-action pair  $(s, a)$ , with  $s \in \mathbb{S}_t(a)$ , which satisfy  $N(s, a) \geq \mathcal{N}$  and  $\Delta(s) > \Delta_\epsilon$ , perform the following:
    - (a) Split this cell-action pair according to the given (feasible) splitting scheme. Let  $s_1, \dots, s_{c_{split}} \in \mathbb{S}_{t+1}(a)$  be the resulting sub-cells after this split. Let  $s_k$  be the cell that contains the sample of the parent cell  $s$ .
    - (b) Initialize the reward upper bounds of the new cells:
$$\tilde{r}(s_j, a) = \tilde{r}(s, a), \quad \forall j \neq k,$$
$$\tilde{r}(s_k, a) = \hat{r}(s, a) + \alpha \Delta(s_k),$$
    - (c) Initialize the transition uncertainty sets of the new cells:
$$\text{CI}_f(s_j, a) = \text{CI}_f(s, a), \quad \forall j \neq k,$$
$$\text{CI}_f(s_k, a) = \left\{ s' \in \mathbb{S}_t : d_b(s', \hat{f}(s, a)) \leq \beta \Delta(s_k) \right\},$$
    - (d) Update the counts of the new cells as follows:
$$N(s_j, a) = 0, \quad \forall j \neq k,$$
$$N(s_k, a) = 1.$$
- 

how small  $s$  is, or how many times it is visited. This might be the case, for example, if some points (states) in  $s$  map under  $f(x, a)$  to the border between  $s_1$  and some adjacent cell  $s_2$ , and all visits to  $s$  are to that part that maps to  $s_2$ . In that case, cell  $s_1$  which is not visited will remain large, hence

with potentially large error in its empirical estimates. This may lead to a large error in the estimated value function at  $s$ , and consequently to an error in the computed policy. We propose a solution to this problem in the next section.

## 5 The AAA Algorithm

To overcome the potential pitfalls of the basic AAA algorithm above, we need to modify the definition of the upper value function so that it will more closely approximate the optimal one. Two modifications will be introduced. First, we propose splitting of the optimistic next-step cells (recall the cell  $s_1$  defined in (14)), in addition to actually visited ones. Those cells, which we call “virtually visited” cells, will be defined formally in Definition 5 below. However, splitting those cells is not enough to fix the problem, since it may happen that no actual samples are obtained for the created cells. Thus, we introduce a *smoothing* operator in the DP equations. This operator, which is specified in Definition 3 below, allows to improve the accuracy of the upper value function in (small) cells even if they are not actually visited (hence not actually sampled), based on the values of their geometric neighbors.

In what follows we will focus on the case  $\gamma\beta < 1$ . As can be seen in the proof of Lemma 1 (Section 7), in this case we have some sort of a contraction effect. Thus, the results are technically much simpler than for  $\gamma\beta > 1$ . The latter case will be briefly discussed in Section 8, and is treated in detail in [3].

**Definition 3** *Let the continuity function of the optimal value be defined as*

$$\omega(\theta) \triangleq \frac{\alpha}{1 - \gamma\beta} \theta, \quad \theta \geq 0.$$

*The smoothing operator at time  $t$  for any given function  $g : \mathbb{S}_t \rightarrow \mathbb{R}$  is*

$$\mathcal{T}_2 g(s) \triangleq \min_{s' \in \mathbb{S}_t} \{g(s') + \omega(\Delta(s) + d_b(s, s'))\}.$$

**Remark.** Note that by definition of the biased distance  $d_b$  in (5), the above minimized set also includes  $g(s)$ , since for  $s' = s$ ,

$$g(s) + \omega(\Delta(s) + d_b(s, s)) = g(s) + \omega(0) = g(s).$$

Therefore,  $\mathcal{T}_2 g(s) \leq g(s)$  for all  $s \in \mathbb{S}_t$ .

It is shown in Lemma 1 in Section 7, that the continuity function  $\omega$  is in fact a bound on the modulus of continuity of the optimal value function  $V$ . The definition of the smoothing operator is then formally justified in Lemma 3, which states that if  $g$  is an upper value function, that is  $g(s) \geq V(x)$  for all  $s$  and  $x \in s$ , then so is  $\mathcal{T}_2 g$ . Thus, the smoothing operator  $\mathcal{T}_2$  tightens the upper value function  $g$  based on the values in adjacent cells.

Now, using the smoothing operator, we modify the definition of the upper DP operator (Definition 2).

**Definition 4** *The smoothed upper DP operator at time  $t$  is defined by  $\tilde{\mathcal{T}} \triangleq \mathcal{T}_1 \mathcal{T}_2$ . That is, for given function  $g : \mathbb{S}_t \rightarrow \mathbb{R}$ ,*

$$\tilde{\mathcal{T}} g(s) = \max_{a \in \mathbb{A}} \left\{ \tilde{r}(s, a) + \gamma \max_{s' \in \text{Cl}_f(s, a)} \mathcal{T}_2 g(s') \right\}.$$

This new operator smooths  $g(s)$  before applying to it the DP operation. As before, we define the UVF as the solution of the fixed point equation:

$$\tilde{V}_t(s) = \tilde{\mathcal{T}} \tilde{V}_t(s), \quad s \in \mathbb{S}_t.$$

Our second modification involves splitting of “virtually visited cells”. We next define the required notation.

**Definition 5 (Virtually Visited Cells)** *At any period  $[t_0, t_1]$  of the algorithm’s execution:*

1. *Let  $\{s_t\}_{t=t_0}^{t_1}$  be the **actually visited cells**  $s_t \in \mathbb{S}_t$  that are visited during this period, and  $\{a_t\}_{t=t_0}^{t_1}$  be the corresponding actions, with  $a_t = \pi_t(s_t)$ .*
2. *Let  $\{(s'_t, a_t)\}_{t=t_0}^{t_1}$  be the **actually visited cell-action pairs** during this period, with  $s'_t \in \mathbb{S}_t(a_t)$ , such that  $s_t \subseteq s'_t$ .*
3. *Denote the **virtually visited cells** during this period by  $\{s_t^*\}_{t=t_0+1}^{t_1+1}$ , where  $s_t^* \in \mathbb{S}_t$  and is the argument of the maximization*

$$s_{t+1}^* \triangleq \operatorname{argmax}_{s' \in \text{Cl}_f(s'_t, a_t)} \mathcal{T}_2 \tilde{V}_t(s').$$

4. *For each virtually visited common grid cell  $s_t^* \in \mathbb{S}_t$ , let  $\{s_a\}_{a \in \mathbb{A}}$  be the action-grid cells ( $s_a \in \mathbb{S}_t(a)$ ) which contain  $s_t^*$ . We define the **virtually visited cell-action pair** as the pair  $(\tilde{s}_t, \tilde{a}_t) = (s_a, a)$  with the cell  $s_a$  having the smallest diameter among those action-grid cells.*

In the course of the algorithm, both actually and virtually visited cell-action pairs will be split (using a count threshold as before). We note that the splitting of virtually visited cells is needed in the common grid  $\mathbb{S}_t$ , to avoid the problem presented in Section 4. This splitting can be done directly in  $\mathbb{S}_t$ . However, we have chosen to keep the relation  $\mathbb{S}_t = \bigwedge_{a \in \mathbb{A}} \mathbb{S}_t(a)$ . Thus, we will split the *smallest* cell  $\tilde{s} \in \mathbb{S}_t(\tilde{a})$  which contains the virtually visited cell  $s^* \in \mathbb{S}_t$  that is candidate for splitting. In this way, the split is inherited by  $\mathbb{S}_t$ .

To summarize, the following modifications are introduced in the AAA algorithm: In Algorithm 2, equation (9) is replaced by

$$\tilde{V}_t(s) = \tilde{\mathcal{T}} \tilde{V}_t(s), \quad s \in \mathbb{S}_t, \quad (15)$$

where  $\tilde{\mathcal{T}}$  is defined in Definition 4. Also, equation (10) is replaced by

$$\pi_t(s) = \operatorname{argmax}_{a \in \mathbb{A}} \left\{ \tilde{r}(s, a) + \gamma \max_{s' \in \text{Cl}_f(s, a)} \mathcal{T}_2 \tilde{V}_t(s') \right\}. \quad (16)$$

Finally, in Algorithm 3, step (ii), we update the counters of both the actually and virtually visited cell-action pairs:

$$N(s'_t, a_t) := N(s'_t, a_t) + 1, \quad N(\tilde{s}_t, \tilde{a}_t) := N(\tilde{s}_t, \tilde{a}_t) + 1.$$

## 6 Main Results ( $\gamma\beta < 1$ )

In this section we summarize the main results regarding the AAA algorithm. Proofs are deferred to the next section.

Recall the definition of the mistake count (2) and the corresponding near-optimality criterion. Also, recall that  $\mathbb{S}_\epsilon$  is

the coarsest (feasible) grid with  $\Delta(s) \leq \Delta_\epsilon$ , which satisfies (13). First we present the main theorem, which provides a mistake bound of modified AAA scheme in terms of the number of cells in  $\mathbb{S}_\epsilon$ .

**Theorem 1** *Let  $\epsilon > 0$  and assume that the AAA algorithm receives an input*

$$\Delta_\epsilon = \frac{(1-\gamma)(1-\gamma\beta)}{2\alpha(\gamma+2)}\epsilon.$$

*Then, the policy-mistake count of the algorithm is bounded by*

$$\text{PM}(\epsilon) \leq \frac{|\mathbb{S}_\epsilon| |\mathbb{A}| (2\mathcal{N} + 1)}{1-\gamma} \ln \frac{2(r_{max} - r_{min})}{\epsilon(1-\gamma)}.$$

In addition to the above theorem, we can obtain a possibly tighter mistake bound in terms of the *posterior number of cells actually used in the course of the algorithm*. In fact, the purpose of adaptive aggregation is that as time progresses, the algorithm will split cells only in the vicinity of the optimal trajectory. Therefore, the actual number of grid cells “at infinity” will be much less than  $|\mathbb{S}_\epsilon|$ . We make this more formal below.

**Definition 6** *Let  $x_0$  be the initial state and let  $N_\infty(x_0, a)$  be the number of cells in the grid  $\lim_{t \rightarrow \infty} \mathbb{S}_t(a)$ , that is*

$$N_\infty(x_0, a) \triangleq \lim_{t \rightarrow \infty} |\mathbb{S}_t(a)|. \quad (17)$$

*Also, let  $N_\infty(x_0) \triangleq \sum_{a \in \mathbb{A}} N_\infty(x_0, a)$ .*

We note that the limit in (17) exists and is finite, since  $|\mathbb{S}_t(a)|$  increases in  $t$ , while  $|\mathbb{S}_t(a)| \leq |\mathbb{S}_\epsilon|$  due to the enforced “stop-splitting” rule. For the same reason, we have the trivial bound  $N_\infty(x_0) \leq |\mathbb{S}_\epsilon| |\mathbb{A}|$ .

**Theorem 2** *Let  $\epsilon > 0$  and assume that the AAA algorithm receives an input  $\Delta_\epsilon$  as in Theorem 1. Then, it holds that*

$$\text{PM}(\epsilon) \leq \frac{4N_\infty(x_0)\mathcal{N}}{1-\gamma} \ln \frac{2(r_{max} - r_{min})}{\epsilon(1-\gamma)}.$$

Note that the bound of Theorem 2 becomes better than the bound of Theorem 1 if  $N_\infty(x_0) \leq \frac{1}{2} |\mathbb{S}_\epsilon| |\mathbb{A}|$ .

**Remark.** Since the action-mistake count satisfies  $\text{AM}(\epsilon) \leq \text{PM}(\epsilon)$ , the policy-mistake bounds of Theorems 1 and 2 apply also to the action-mistake.

**Discussion.** Theorem 1 implies that the mistake bound is linear in  $|\mathbb{S}_\epsilon|$ . Therefore, using equation (13), we obtain the following explicit dependence on  $\epsilon$  (ignoring the log factor):

$$\text{PM}(\epsilon) \leq C (1/\epsilon)^n |\mathbb{A}| (2\mathcal{N} + 1), \quad (18)$$

where the constant  $C$  is polynomial in  $\alpha$ ,  $1/(1-\gamma)$  and in  $1/(1-\gamma\beta)$ . Note however the exponential dependence on the dimension  $n$  of the state-space, which is an obvious artifact of the dense aggregation approach.

In the context of the posterior bound (Theorem 2), it should be noted that there is a trade-off between the choice of the count threshold  $\mathcal{N}$  and the number of cells at infinity

$N_\infty(x_0)$ . If we choose  $\mathcal{N}$  too small, the algorithm will perform many splits, and consequently  $N_\infty(x_0)$  will be large. In this case it may happen that the algorithm will produce redundant cells, which are not actually needed for near-optimal performance. On the other hand, if we choose large  $\mathcal{N}$ , the algorithm will perform less splits, resulting in a smaller  $N_\infty(x_0)$ . This however may lead to a slower convergence to the optimal trajectory.

Two comparisons that may be of interest follow. First, consider our algorithm for the “flat” model, which uses a sufficiently fine grid (namely,  $\mathbb{S}_\epsilon$ ) over the state space. It can be shown that the mistake bound in such case will be as in Theorem 1, with  $2\mathcal{N} + 1$  replaced by 1. Clearly, however, such an algorithm is not feasible if  $|\mathbb{S}_\epsilon|$  is large.

Moreover, consider a naïve approach, where the “flat” model is treated as a finite-state MDP, and an efficient exploration technique is used on this MDP (such as the R-MAX algorithm [7]). In the ideal case when the MDP assumption happens to hold true, such an algorithm will again have the mistake bound as in Theorem 1, with  $2\mathcal{N} + 1$  replaced by 1 (see for instance [12], Theorem 8.3.5). However, as this assumption generally is not satisfied, the computed value function might *underestimate* the optimal one, resulting in algorithm’s failure (and, in fact, in infinite mistake count).

## 7 Analysis of the AAA Scheme

Below is the outline of the analysis. First we show that the optimal value function  $V$  possesses some continuity property, which will justify the use of the smoothing operator  $\mathcal{T}_2$ . Then, we show that there exists a unique solution to equation (15), and that this solution upper bounds the optimal value  $V$ . Finally, we prove that under certain conditions on the grid, the optimal policy with respect to the UVF (equation (16)) is an  $\epsilon$ -optimal policy, which will enable us to prove a polynomial bound on the policy-mistake count of the algorithm.

For ease of exposition, throughout the analysis we write CI for the transition uncertainty set (instead of  $\text{CI}_f$ ), and denote by  $V_b \triangleq \frac{1}{1-\gamma}(r_{max} - r_{min})$  the maximal difference between two returns of any two policies. Also, recall that the proofs presented below are limited to the  $\gamma\beta < 1$  case.

### a. Continuity of the Optimal Value Function

In this subsection we show that under the continuity Assumption 2, the optimal value function is also Lipschitz continuous<sup>4</sup>.

**Lemma 1** *For any given  $x_1, x_2 \in \mathbb{X}$ , we have that*

$$|V(x_1) - V(x_2)| \leq \frac{\alpha}{1-\gamma\beta} d(x_1, x_2) \triangleq \omega(d(x_1, x_2)).$$

*Proof.* Fix  $x_1, x_2 \in \mathbb{X}$ . From the optimality equation (1), we have that

$$\begin{aligned} & |V(x_1) - V(x_2)| \\ & \leq \max_a |r(x_1, a) - r(x_2, a)| \\ & \quad + \gamma \max_a |V(f(x_1, a)) - V(f(x_2, a))| \\ & \leq \alpha d(x_1, x_2) + \gamma \max_a |V(f(x_1, a)) - V(f(x_2, a))|, \end{aligned}$$

<sup>4</sup>In case  $\gamma\beta > 1$  it is Hölder continuous, see [3] for details.

where the second inequality follows by Assumption 2. Also by this assumption, we have that

$$d(f(x_1, a), f(x_2, a)) \leq \beta d(x_1, x_2),$$

for any  $a$ . Applying the above inequalities iteratively, for any integer  $H > 0$ , we obtain the following bound:

$$|V(x_1) - V(x_2)| \leq \alpha d(x_1, x_2) \sum_{k=0}^{H-1} (\gamma\beta)^k + \gamma^H V_b.$$

Now, since  $\gamma\beta < 1$ , we can take  $H = \infty$  in the above bound, and obtain the desired result.  $\square$

## b. The Upper Value Function

First we prove the contraction property of the upper DP operator used in the fixed point equation (15).

**Lemma 2** *The operator  $\tilde{T}$  is a contraction mapping in the  $\ell_\infty$  norm, with the contraction factor  $\gamma$ . Thus, there exists a unique solution to equation (15).*

*Proof.* Given two functions  $g_1$  and  $g_2$ , we have the following sequence of inequalities:

$$\begin{aligned} & \left| (\tilde{T}g_1)(s) - (\tilde{T}g_2)(s) \right| \\ & \leq \gamma \max_{a \in \mathbb{A}} \left| \max_{s' \in \text{CI}(s, a)} \mathcal{T}_2 g_1(s') - \max_{s' \in \text{CI}(s, a)} \mathcal{T}_2 g_2(s') \right| \\ & \leq \gamma \max_{a \in \mathbb{A}} \max_{s' \in \text{CI}(s, a)} |\mathcal{T}_2 g_1(s') - \mathcal{T}_2 g_2(s')| \\ & \leq \gamma \max_{s' \in \mathbb{S}_t} |\mathcal{T}_2 g_1(s') - \mathcal{T}_2 g_2(s')|. \end{aligned}$$

Now, since

$$\begin{aligned} & |\mathcal{T}_2 g_1(s') - \mathcal{T}_2 g_2(s')| \\ & = \left| \min_{s'' \in \mathbb{S}_t} \{g_1(s'') + \omega(\Delta(s') + d_b(s', s''))\} - \min_{s'' \in \mathbb{S}_t} \{g_2(s'') + \omega(\Delta(s') + d_b(s', s''))\} \right| \\ & \leq \max_{s'' \in \mathbb{S}_t} |g_1(s'') - g_2(s'')| = \|g_1 - g_2\|_\infty, \end{aligned}$$

it follows that  $\left| (\tilde{T}g_1)(s) - (\tilde{T}g_2)(s) \right| \leq \gamma \|g_1 - g_2\|_\infty$  for all  $s \in \mathbb{S}_t$ . Hence,  $\left\| \tilde{T}g_1 - \tilde{T}g_2 \right\|_\infty \leq \gamma \|g_1 - g_2\|_\infty$ , which proves the result.  $\square$

We will need the following property of the smoothing operator  $\mathcal{T}_2$ .

**Lemma 3** *If  $g_1 : \mathbb{S}_t \rightarrow \mathbb{R}$  is an upper bound on the value function (that is,  $g_1(s) \geq V(x)$  for all  $s \in \mathbb{S}_t$  and  $x \in s$ ), then so is  $g_2 \triangleq \mathcal{T}_2 g_1$ .*

*Proof.* For given  $s \in \mathbb{S}_t$ , let  $s^*$  be the cell that achieves the minimum in the smoothing operator  $\mathcal{T}_2$ :

$$s^* = \operatorname{argmin}_{s' \in \mathbb{S}_t} \{g_1(s') + \omega(\Delta(s) + d_b(s, s'))\}.$$

If  $s = s^*$ , then by definition of the biased distance (5) we have that  $d_b(s, s^*) = -\Delta(s)$ , implying that

$$\omega(\Delta(s) + d_b(s, s^*)) = \omega(0) = 0.$$

Thus,  $g_2(s) = g_1(s) \geq V(x)$  for all  $x \in s$ . Otherwise, let<sup>5</sup>  $x_{\min} \in \bar{s}$  and  $x_{\min}^* \in \bar{s}^*$  be such that  $d_b(s, s^*) = d(x_{\min}, x_{\min}^*)$ . We have that

$$\begin{aligned} g_2(s) & \triangleq g_1(s^*) + \omega(\Delta(s) + d_b(s, s^*)) \\ & \geq V(x_{\min}^*) + \omega(\Delta(s) + d_b(s, s^*)) \\ & \geq V(x), \end{aligned}$$

where the first inequality follows by hypothesis for the state  $x_{\min}^* \in \bar{s}^*$ , and the second inequality holds for every  $x \in s$  by Lemma 1, since

$$d(x, x_{\min}^*) \leq d(x_{\min}, x_{\min}^*) + d(x, x_{\min}) \leq d_b(s, s^*) + \Delta(s). \quad \square$$

**Lemma 4** *The UVF  $\tilde{V}_t$  is indeed an upper bound on the optimal value function. That is, at every time  $t$ , we have that  $\tilde{V}_t(s) \geq V(x)$ ,  $\forall s \in \mathbb{S}_t, \forall x \in s$ .*

*Proof.* Since, by Lemma 2,  $\tilde{T}$  is a contraction operator, we can prove the claim by induction on the steps of value iteration. For the base case, let  $\tilde{V}^0(s) \equiv V_{\max} \geq V(x), \forall x \in \mathbb{X}$ . Now assume that the claim holds for  $n$ -th iteration. For  $n+1$ -th iteration we have by the Lipschitz continuity of the reward (Assumption 2) and by the definition of  $\tilde{r}(s, a)$ , that for all  $s \in \mathbb{S}_t$  and  $x \in s$ ,

$$\tilde{r}(s, a) = r(x_s, a) + \alpha \Delta(s) \geq r(x, a),$$

where  $x_s$  is a sample point in  $s$ . Also, by Assumption 2 and by the definition of  $\text{CI}(s, a)$ , it follows for any  $x \in s$ , that  $f(x, a) \in s'$ , with  $s' \in \text{CI}(s, a)$ . Thus,

$$\max_{s' \in \text{CI}(s, a)} \mathcal{T}_2 \tilde{V}^n(s') \geq \mathcal{T}_2 \tilde{V}^n(s' : f(x, a) \in s') \geq V(f(x, a)),$$

where the last inequality follows by the induction assumption and Lemma 3. Therefore, we have

$$\begin{aligned} \tilde{V}^{n+1}(s) & = \max_{a \in \mathbb{A}} \left\{ \tilde{r}(s, a) + \gamma \max_{s' \in \text{CI}(s, a)} \mathcal{T}_2 \tilde{V}^n(s') \right\} \\ & \geq \max_{a \in \mathbb{A}} \{r(x, a) + \gamma V(f(x, a))\} \\ & = V(x). \end{aligned}$$

which completes the induction proof. Since  $\tilde{V}^n \rightarrow \tilde{V}$ , the result follows.  $\square$

## c. Near-Optimality of the UVF Optimal Policy

In this section we provide a sufficient condition on the grid, which ensures that the return obtained by the policy  $\mathcal{A}_t = \{\pi_\tau\}_{\tau=t}^\infty$  which the algorithm implements at time  $t$ , is  $\epsilon$ -close to the UVF:  $\tilde{V}_t(s) - J_M^{\mathcal{A}_t}(x) \leq \epsilon$ , for a given  $s \in \mathbb{S}_t$  and all  $x \in s$ . This will imply that  $V(x) - J_M^{\mathcal{A}_t}(x) \leq \epsilon$ , since  $\tilde{V}_t$  is an upper bound on the optimal value; namely, this will imply that  $\mathcal{A}_t$  is an  $\epsilon$ -optimal policy.

To proceed, we introduce the definitions of *known cell-action pairs* and the *escape event*.

<sup>5</sup> $\bar{A}$  denotes the closure of a set  $A$ .



By the definition of  $T_{\epsilon/2}$  (Definition 8), we have  $\gamma^{T_{\epsilon/2}} V_b \leq \frac{\epsilon}{2}$ . Now, we have to check that the condition (19) of the lemma regarding  $\Delta_\epsilon$ , implies that

$$\sum_{t=0}^{T_{\epsilon/2}-1} \gamma^t [2\alpha\Delta(s'_t) + \gamma\omega(\Delta(s_{t+1}^*) + 2\beta\Delta(s'_t))] \leq \frac{\epsilon}{2}.$$

Indeed,

$$\begin{aligned} & \sum_{t=0}^{T_{\epsilon/2}-1} \gamma^t [2\alpha\Delta(s'_t) + \gamma\omega(\Delta(s_{t+1}^*) + 2\beta\Delta(s'_t))] \\ & \leq \sum_{t=0}^{T_{\epsilon/2}-1} \gamma^t [2\alpha\Delta_\epsilon + \gamma\omega(\Delta_\epsilon + 2\beta\Delta_\epsilon)] \\ & \leq \sum_{t=0}^{\infty} \gamma^t [2\alpha\Delta_\epsilon + \gamma\omega(\Delta_\epsilon + 2\beta\Delta_\epsilon)] \\ & = \frac{1}{1-\gamma} \left[ 2\alpha\Delta_\epsilon + \gamma \frac{\alpha}{1-\gamma\beta} (1+2\beta)\Delta_\epsilon \right] \\ & = \frac{2\alpha + \gamma\alpha}{(1-\gamma)(1-\gamma\beta)} \Delta_\epsilon = \frac{\epsilon}{2}, \end{aligned}$$

where the first inequality follows since we are addressing case (a), in which all actually and virtually visited cells are smaller than  $\Delta_\epsilon$ , the second inequality holds by taking the infinite sum instead of the finite one, the first equality follows by the definition of  $\omega$  (see Definition 3), and the last equality follows by the hypothesis (19) of the Lemma. This completes the proof of the Lemma.  $\square$

#### d. Number of Cells in $\epsilon$ -Optimal Grid

Before proving the mistake bounds, we provide an upper bound on the number of cells  $N_\epsilon = |\mathbb{S}_\epsilon|$ .

**Lemma 6** *For a fixed feasible splitting scheme, with parameters  $c_{split}$  and  $\lambda$ , and a single initial grid  $\mathbb{S}_0$ , we have that*

$$N_\epsilon \leq |\mathbb{S}_0| c_{split} \left( \frac{\Delta_{max}}{\Delta_\epsilon} \right)^{\log_{1/\lambda}(c_{split})}.$$

*Proof.* For every  $s \in \mathbb{S}_0$ , consider performing  $k(s)$  splits iteratively, such that at each iteration we obtain new  $c_{split}$  cells instead of the original one. It follows that after  $k(s)$  such splits, the size of a split cell  $s' \subseteq s$  satisfies  $\Delta(s') \leq \lambda^{k(s)} \Delta(s)$ . In addition, the number of cells in the grid that contains all such cells  $s'$  is

$$N = \sum_{s \in \mathbb{S}_0} c_{split}^{k(s)}. \quad (20)$$

Thus, for each  $s \in \mathbb{S}_0$ , we need to find the minimal  $k(s)$ , such that

$$\lambda^{k(s)} \Delta(s) \leq \Delta_\epsilon. \quad (21)$$

From (21), it follows that this minimal  $k(s) = k^*(s)$  satisfies

$$\log_{1/\lambda} \left( \frac{\Delta(s)}{\Delta_\epsilon} \right) \leq k^*(s) < \log_{1/\lambda} \left( \frac{\Delta(s)}{\Delta_\epsilon} \right) + 1.$$

Substituting the last inequality in (20) yields

$$\begin{aligned} N_\epsilon & = \sum_{s \in \mathbb{S}_0} c_{split}^{k^*(s)} \\ & \leq c_{split} \sum_{s \in \mathbb{S}_0} (c_{split})^{\log_{1/\lambda} \left( \frac{\Delta(s)}{\Delta_\epsilon} \right)} \\ & = c_{split} \sum_{s \in \mathbb{S}_0} \left( \frac{\Delta(s)}{\Delta_\epsilon} \right)^{\log_{1/\lambda}(c_{split})} \\ & \leq |\mathbb{S}_0| c_{split} \left( \frac{\Delta_{max}}{\Delta_\epsilon} \right)^{\log_{1/\lambda}(c_{split})}, \end{aligned}$$

which completes the proof.  $\square$

**Remark.** We note that Lemma 6 shows an *exponential* dependence of  $N_\epsilon$  on the state space dimension  $n$  since in most cases  $\log_{1/\lambda}(c_{split})$  is of order of  $n$ .

#### e. Proof of Theorem 1

First, we note that the escape event  $E_t(s)$  (Definition 9) can be viewed as an *exploration* event. If it occurs at some time  $t \geq 0$ , the algorithm will encounter (in an execution of length  $T_{\epsilon/2}$ ) a cell-action pair  $(s, a)$  (either actually, or virtually), with  $s$  that is not in  $\mathbb{S}_\epsilon$ . In addition, in case of actual escape event (see Definition 9), this pair was not sampled. This fact can be interpreted as a “discovery” of new information, since every such occurrence of an “unknown” pair will lead to an increase of the count of such pair, and, eventually, to split of such a pair.

Next two lemmas show that the number of times that “actual” and “virtual” escape events can occur is bounded.

**Lemma 7** *The number of times that  $AE_t(s)$  can occur is bounded by  $N_\epsilon |\mathbb{A}| (\mathcal{N} + 1) T_{\epsilon/2}$ .*

*Proof.* Note that any cell  $s \in \mathbb{S}_t(a)$  for any  $a$  and  $t$ , can be visited no more than  $\mathcal{N}$  times – after this number of times, the cell is split. Now think of the grid representation of the state space as a tree, with cells as leaves. The internal nodes in such tree represent the larger aggregations, that were used in previous episodes. Now, the number of such internal nodes is less or equal to the number of leaves, since the splitting coefficient is greater or equal to 2. Using this tree representation, the visit to the “unknown” pair can be interpreted as a visit to *an internal node of  $\mathbb{S}_\epsilon$* . Since the counter of this pair is incremented in this visit, by a simple counting argument (a.k.a. the Pigeonhole Principle), the number of times that the algorithm can encounter an internal node of  $\mathbb{S}_\epsilon$  is bounded by

$$(\text{number of internal nodes of } \mathbb{S}_\epsilon) \cdot \mathcal{N} \cdot |\mathbb{A}| \leq N_\epsilon \mathcal{N} |\mathbb{A}|.$$

Finally, when the algorithm encounters a leaf of  $\mathbb{S}_\epsilon$ , then only one such occurrence is sufficient in order to the desired cell to become sampled. Again, by a simple counting argument, the number of times this can occur is bounded by

$$(\text{number of leaves of } \mathbb{S}_\epsilon) \cdot |\mathbb{A}| = N_\epsilon |\mathbb{A}|.$$

To conclude, the number of times that an “unknown” cell-action pair can be encountered is bounded by

$$N_\epsilon \mathcal{N} |\mathbb{A}| + N_\epsilon |\mathbb{A}| = N_\epsilon (\mathcal{N} + 1) |\mathbb{A}|.$$

At each time  $t$ , consider the execution of a (non-stationary) policy  $\mathcal{A}_t$  for  $T_{\epsilon/2}$  time steps in  $M$ . We have two mutually exclusive cases: (a) If starting at time  $t$ , we execute the policy  $\mathcal{A}_t$  for  $T_{\epsilon/2}$  time steps, without encountering an “unknown” pair (that is, a pair not in  $AK_t$ ), there is no occurrence of the escape event  $AE_t(s)$ . (b) If starting at time  $t$ , we execute the policy  $\mathcal{A}_t$  for  $T_{\epsilon/2}$  time steps, and encounter at least one unknown pair at time  $t \leq t' \leq t + T_{\epsilon/2}$ , the escape event  $AE_t(s)$  occurs.

We then wish to bound the number of time steps that (b) is the case. In the worst case we will encounter an unknown pair at the end of the execution period of length  $T_{\epsilon/2}$ . In this case, we have that all the succeeding executions for  $t < t' \leq t + T_{\epsilon/2}$  will also encounter this unknown pair. That is, if  $AE_t(x_t)$  occurs at some time  $t$ , also  $AE_{t'}(x_{t'})$  for  $t < t' \leq t + T_{\epsilon/2}$  will occur, in the worst case. Since after  $N_\epsilon(\mathcal{N} + 1)|\mathbb{A}|$  visits to unknown pairs, all the pairs will become known,  $AE_t(s)$  can occur at most  $N_\epsilon(\mathcal{N} + 1)|\mathbb{A}|T_{\epsilon/2}$  times.  $\square$

**Lemma 8** *The number of times that  $VE_t(s)$  can occur is bounded by  $N_\epsilon|\mathbb{A}|\mathcal{N}T_{\epsilon/2}$ .*

*Proof.* The proof is similar to the proof of Lemma 7, with the difference that the virtual escape event cannot occur on the leaves of  $\mathbb{S}_\epsilon$ .  $\square$

**Lemma 9** *The number of times that the escape event  $E_t(s)$  can occur is bounded by  $N_\epsilon|\mathbb{A}|(2\mathcal{N} + 1)T_{\epsilon/2}$ .*

*Proof.* Follows by Lemmas 7 and 8 and Definition 9.  $\square$

Finally, we prove the main theorem regarding the mistake bound of the AAA algorithm.

*Proof of Theorem 1.* For each time  $t$ , we consider the execution of policy  $\mathcal{A}_t$  for  $T_{\epsilon/2}$  time-steps in  $M$ , with the initial state in each such execution  $x_t (x_t \in s_t)$ . We then have that

$$\begin{aligned} J_M^{A_t}(x_t) &\geq \tilde{V}_t(s_t) - \epsilon - \mathbb{I}\{E_t(s_t)\}V_b \\ &\geq V(x_t) - \epsilon - \mathbb{I}\{E_t(s_t)\}V_b, \end{aligned}$$

where the first inequality holds by Lemma 5, and the second inequality holds by Lemma 4. However, by Lemma 9, the number of times the event  $E_t(s_t)$  can occur is bounded by  $N_\epsilon(2\mathcal{N} + 1)|\mathbb{A}|T_{\epsilon/2}$ , implying that

$$\sum_{t=0}^{\infty} \mathbb{I}\left\{J_M^{A_t}(x_t) < V(x_t) - \epsilon\right\} \leq N_\epsilon|\mathbb{A}|(2\mathcal{N} + 1)T_{\epsilon/2},$$

which completes the proof of the theorem, using the definition of the  $\epsilon/2$ -horizon time, and the fact that  $\log_{1/\gamma} C \leq \frac{1}{1-\gamma} \ln C$ , for any  $C$ .  $\square$

## f. Proof of Theorem 2

Recall the definition of the posterior number  $N_\infty(x_0)$  of actually used cells in the course of the algorithm (Definition 6). We only need to prove the analogue of Lemma 9 in this case. The rest of the proof is exactly the same as that of Theorem 1.

Thus, we need to bound the number of times that an escape event occurs. Here we consider the trees that represent the grids “at infinity”, namely  $\mathbb{S}_\infty(a) = \lim_{n \rightarrow \infty} \mathbb{S}_t(a)$ ,  $a \in \mathbb{A}$ , instead of the  $\epsilon$ -optimal grid  $\mathbb{S}_\epsilon$ . First, consider the actual escape event. As previously, this event can occur on *internal nodes* of  $\mathbb{S}_\infty(a)$  no more than

$$(\text{number of internal nodes of } \mathbb{S}_\infty(a)) \cdot \mathcal{N} \leq N_\infty(x_0, a)\mathcal{N}$$

times. The *leaves* of the tree (which are the cells of  $\mathbb{S}_\infty(a)$ ) can be classified into two groups: (a) “Small” leaves, with  $\Delta(s) \leq \Delta_\epsilon$ , and (b) “Large” leaves, with  $\Delta(s) > \Delta_\epsilon$ . On “small” leaves, only one occurrence of the escape event is possible, since such a cell becomes known (Definition 7) after one sample. On “large” leaves, there cannot be more than  $\mathcal{N}$  occurrences of the escape event – otherwise these cells would have been split. Thus, the number of times the escape event can occur on leaves is bounded by

$$(\text{number of leaves of } \mathbb{S}_\infty(a))\mathcal{N} = N_\infty(x_0, a)\mathcal{N}.$$

To summarize, the number of times that the (actual) escape event can occur on all nodes, for all actions  $a \in \mathbb{A}$ , is bounded by  $\sum_{a \in \mathbb{A}} 2N_\infty(x_0, a)\mathcal{N}$ .

Similarly, the virtual escape event cannot occur more than  $\sum_{a \in \mathbb{A}} 2N_\infty(x_0, a)\mathcal{N}$  times. Note that there is no difference in bounds on the number of occurrences of actual and virtual escape events, since the cells of  $\mathbb{S}_\infty(a)$  can be “large” ones, that is having  $\Delta(s) > \Delta_\epsilon$ . Thus, the virtual escape event can also happen on leaves. By the same arguments as in proof of Theorem 1, the sum of the above two bounds times the  $\epsilon/2$ -horizon time  $T_{\epsilon/2}$  is the mistake bound of the algorithm.

## 8 The Expansive Case ( $\gamma\beta > 1$ )

Our analysis above focused on case  $\gamma\beta < 1$ . When  $\gamma\beta > 1$ , the analysis becomes more involved. This can be observed for example, from the bound on the distance between optimal values of two states, presented in proof of Lemma 1:

$$|V(x_1) - V(x_2)| \leq \alpha d(x_1, x_2) \sum_{k=0}^{H-1} (\gamma\beta)^k + \gamma^H V_b. \quad (22)$$

If  $\gamma\beta > 1$ , instead of bounding the infinite sum of distances between future rewards, we have to employ a “cut-off tactics”. Specifically, we have to make a balance between the first term in (22), which grows exponentially in  $H$ , and the second term, which decays exponentially in  $H$ . A detailed treatment of this case can be found in [3] and is omitted here due to space limitations. Using the approach outlined above, it is shown that to obtain the mistake bounds of Theorems 1 and 2, the cell size threshold should be taken as  $\Delta_\epsilon = K\epsilon^\xi$ , where  $\xi \triangleq \log_{1/\gamma} \beta$ , and  $K$  is polynomial in  $\alpha, \beta, 1/(1-\gamma)$  and *exponential* in  $\xi$ ; note that  $\xi > 1$  and compare this condition to (19) in case  $\gamma\beta < 1$ . As a result, in the expansive case we obtain a worse explicit dependence of the mistake bound on  $\epsilon$ , as follows:

$$\text{PM}(\epsilon) \leq C' (1/\epsilon)^{n\xi} |\mathbb{A}| (2\mathcal{N} + 1),$$

where  $C'$  is polynomial in  $\alpha, \beta, 1/(1-\gamma)$ , and is exponential in  $\xi$ ; compare this bound to (18).

## 9 Conclusion

We presented a model-based learning algorithm that aims to solve the online, continuous state space reinforcement learning problem in deterministic domain, under continuity assumption of model parameters. We note that we did not address at all the issue of the computational complexity. The goal of the analysis was to show feasibility in the sense of sample efficiency.

Some ideas for improvement of the proposed algorithm and its analysis follow. First, it would be interesting from computational perspective, to formulate an on-line asynchronous variant, that will perform only one back-up of Value Iteration each time-step, instead of exact calculation, and analyze its performance. Also, the explicit dependence of the posterior number of cells on the number of cells needed on the *optimal trajectory* remains an open and difficult question which requires new tools for its analysis. In addition, more elaborate splitting rules and possible merging schemes should be considered. Finally, evaluation of the algorithm using simulations would be interesting from the practical point of view.

The extension of similar ideas to the stochastic domains seems possible, under a different continuity assumption (namely, under continuity of *transition density* as in [9]). Preliminary results for this case can be found in [3]. A possible future direction here is to formulate an algorithm that will work for both the stochastic and deterministic cases, under a unified continuity assumption. Another possible extension is to the case of continuous action space  $\mathbb{A}$ , which can be approached using aggregation, similarly to the state space. Finally, other reward criteria should be considered – average reward (with associated loss bounds), and shortest path problems (total reward). In particular, the shortest path formulation is more natural in such deterministic problems, as navigation in maze.

## References

- [1] A. Antos, R. Munos, and C. Szepesvari. Fitted Q-iteration in continuous action-space MDPs. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2007.
- [2] P. Auer and R. Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2006.
- [3] A. Bernstein. Adaptive state aggregation for reinforcement learning. Master's thesis, Technion – Israel Institute of Technology, 2007. URL: [http://tx.technion.ac.il/~andreyb/MSc\\_Thesis\\_final.pdf](http://tx.technion.ac.il/~andreyb/MSc_Thesis_final.pdf).
- [4] D. P. Bertsekas. *Dynamic Programming and Optimal Control, vol. 2*. Athena Scientific, Belmont, MA, third edition, 2007.
- [5] A. Bonarini, A. Lazaric, and M. Restelli. LEAP: an adaptive multi-resolution reinforcement learning algorithm. *To appear*.
- [6] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.
- [7] R. I. Brafman and M. Tennenholtz. R-MAX - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.
- [8] H. Chapman. Global confidence bound algorithms for the exploration-exploitation tradeoff in reinforcement learning. Master's thesis, Technion – Israel Institute of Technology, 2007.
- [9] C.-S. Chow and J.N. Tsitsiklis. An optimal one-way multigrid algorithm for discrete-time stochastic control. *IEEE Transactions on Automatic Control*, 36(8):898–914, 1991.
- [10] C. Diuk, A. L. Strehl, and M. L. Littman. A hierarchical approach to efficient reinforcement learning in deterministic domains. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 313–319, 2006.
- [11] G. J. Gordon. Reinforcement learning with function approximation converges to a region. In *Advances in Neural Information Processing Systems (NIPS) 12*, pages 1040–1046, 2000.
- [12] S. M. Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, UK, 2003.
- [13] M. Kearns and S. P. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49:209–232, 2002.
- [14] I. Menache, S. Mannor, and N. Shimkin. Q-Cut – dynamic discovery of sub-goals in reinforcement learning. In *Proceedings of the 13th European Conference on Machine Learning (ECML 2002)*, pages 187–195, 2002.
- [15] A. W. Moore and C. G. Atkeson. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Machine Learning*, 21:199–233, 1995.
- [16] R. Munos and A. W. Moore. Variable resolution discretization in optimal control. *Machine Learning*, 49:291–323, 2002.
- [17] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994.
- [18] A. L. Strehl and M. L. Littman. A theoretical analysis of model-based interval estimation. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 857–864, 2005.
- [19] A. L. Strehl, E. Wiewiora, J. Langford, and M. L. Littman. PAC model-free reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 881–888, 2006.
- [20] A. Tewari and P. L. Bartlett. Optimistic linear programming gives logarithmic regret for irreducible MDPs. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2007.
- [21] W. Whitt. Approximations of dynamic programs, I. *Mathematics of Operations Research*, 3(3):231–243, 1978.

---

# High-Probability Regret Bounds for Bandit Online Linear Optimization

---

Peter L. Bartlett \*

UC Berkeley

bartlett@cs.berkeley.edu

Varsha Dani

University of Chicago

varsha@cs.uchicago.edu

Thomas P. Hayes

TTI Chicago

hayest@tti-c.org

Sham M. Kakade

TTI Chicago

sham@tti-c.org

Alexander Rakhlin \*

UC Berkeley

rakhlin@cs.berkeley.edu

Ambuj Tewari \*

TTI Chicago

tewari@tti-c.org

## Abstract

We present a modification of the algorithm of Dani et al. [8] for the online linear optimization problem in the bandit setting, which *with high probability* has regret at most  $O^*(\sqrt{T})$  against an *adaptive* adversary. This improves on the previous algorithm [8] whose regret is bounded *in expectation* against an *oblivious* adversary. We obtain the same dependence on the dimension ( $n^{3/2}$ ) as that exhibited by Dani et al. The results of this paper rest firmly on those of [8] and the remarkable technique of Auer et al. [2] for obtaining high-probability bounds via optimistic estimates. This paper answers an open question: it eliminates the gap between the high-probability bounds obtained in the full-information vs bandit settings.

## 1 Introduction

In the online linear optimization problem, there is a fixed decision set  $D \in \mathbb{R}^n$  and the player (or decision maker) makes a decision  $x_t$  at time  $t \in \{1, \dots, T\}$ . Simultaneously, an adversary chooses a loss vector  $L_t$  and the player suffers loss  $L_t^\dagger x_t$ . The goal is to minimize *regret* which measures how much worse the player did as compared to any fixed decision, even one chosen with complete knowledge of the sequence  $L_1, \dots, L_T$ ,

$$R = \sum_{t=1}^T L_t^\dagger x_t - \min_{x \in D} \sum_{t=1}^T L_t^\dagger x.$$

The adversary can be *oblivious* to the player's moves in which case it chooses the entire sequence  $L_1, \dots, L_T$  in advance of the player's moves. An *adaptive* adversary can, however, choose  $L_t$  based on the player's moves  $x_1, \dots, x_{t-1}$  up to that point.

In the full information version of the problem, the loss vector  $L_t$  is revealed to the player at the end of round  $t$ . For this case, Kalai and Vempala [12] gave an efficient algorithm assuming that the *offline* problem (given  $L$  minimize  $L^\dagger x$  over  $x \in D$ ) can be solved efficiently. Note that the standard

“experts” problem is a special case of this problem because we can choose the set  $D$  to be  $\{e_1, \dots, e_n\}$ , the unit vectors forming the standard basis of  $\mathbb{R}^n$ . Kalai and Vempala separated the issue of the number of available decisions from the dimensionality of the problem and gave an algorithm with expected regret  $O(\text{poly}(n)\sqrt{T})$ . In many important cases, for example the *online shortest path problem* [15], the size of the decision set can be exponential in the dimensionality. So, it is important to design algorithms that have polynomial dependence on the dimension.

In the partial information or “bandit” version of the problem, the only feedback that the player receives at the end of round  $t$  is its own loss  $L_t^\dagger x_t$ . The bandit version of the experts problem was considered by Auer et al. [2] who gave a number of algorithms for the problem. Their **Exp3** algorithm achieves  $O(\sqrt{T})$  expected regret against oblivious adversaries. However, due to the large variance of the estimates kept by **Exp3** it fails to enjoy a similar regret bound with high probability. To address this issue, the authors used the idea of *high confidence upper bounds* to derive the **Exp3.P** algorithm which achieves  $O(\sqrt{T})$  regret with high probability. The regret of these algorithms also has a  $\sqrt{|D|}$  dependence on the number  $|D|$  of available actions. Hence, these cannot be used directly if  $|D|$  is large.

Awerbuch and Kleinberg [4] were the first to consider the general online linear optimization problem in the bandit setting. For oblivious adversaries, they proved a regret bound of  $O^*(\text{poly}(n)T^{2/3})$ . The case of a general adaptive adversary was handled by McMahan and Blum [14] but they could only prove a regret bound of  $O^*(\text{poly}(n)T^{3/4})$ . Dani and Hayes [7] later showed that McMahan and Blum's algorithm actually enjoys a regret bound of  $O^*(\text{poly}(n)T^{2/3})$ . However, the known lower bound for the bandit problem was the same as that in the full information case, namely  $\Omega(\sqrt{T})$ . Therefore, it was an important open question if there is an algorithm with a regret bound of  $O(\text{poly}(n)\sqrt{T})$  for the bandit online linear optimization problem. An affirmative answer was recently given by Dani et al. [8]. Their algorithm has expected regret at most  $O^*(\text{poly}(n)\sqrt{T})$  against an oblivious adversary. It was still not known if the same bounds could be achieved with high probability and against adaptive adversaries as well. In this paper, we show how to do this by combining Dani et al.'s techniques with those of Auer et al. [2]. Like **Exp3.P**, our **GEOMETRICEDGE.P** algorithm

---

\*PB, AR and AT gratefully acknowledge the support of DARPA under grant FA8750-05-2-0249.

keeps biased estimates of the losses of different actions such that, with high probability, the sums of these estimates are *lower bounds* (because we use losses not gains) on the actual unknown cumulative losses (Lemma 5).

The bandit version of the online shortest path problem has recently received a lot of attention. It can be used to model, for example, routing in ad hoc wireless networks. If we want to make our routing algorithm secure against adversarial attacks, it is necessary to design algorithms that work against adaptive adversaries [3, 13]. Therefore, obtaining low regret against adaptive adversaries is not only an important theoretical problem but it also has practical implications. The algorithm with the best regret guarantee so far is by György et al. [11]. There the authors consider a number of feedback models. Our feedback model in this paper corresponds to what they call the “path-bandit” model. For this model, they give an efficient algorithm specially designed for the bandit online shortest path problem that achieves  $O^*(\text{poly}(n)T^{2/3})$  regret with high probability against an adaptive adversary where  $n$  is the number of edges in the graph. Our results imply that it is actually possible to achieve  $O^*(n^{3/2}\sqrt{T})$  regret with high probability. However, since our algorithm is not efficient, the quest for an efficient algorithm with the same regret, even for this special problem, is still on.

The key tools from probability theory that we use in our proofs are Bernstein-type inequalities, such as Freedman’s. These provide sharper concentration bounds for martingales in the presence of variance information. There is a simple corollary of Freedman’s inequality that we think is useful not just in our setting but more generally. We state it as Lemma 2 in Section 4.

The present work closes the gap between full information and bandit online optimization against the adaptive adversary in terms of the growth of regret with  $T$ . As we said above, our algorithm is not necessarily efficient, because the decision space might need to be discretized to a fine level. We mention that a parallel work by Abernethy, Hazan, and Rakhlin [1] provides an efficient algorithm for the setting; however, their result holds in expectation only (against an oblivious adversary). The present paper and [1] are addressing disparate aspects of the problem and neither result can be concluded from the other. It remains an open question whether there exists an efficient algorithm which enjoys high probability bounds on the regret.

## 2 Preliminaries

Let  $D \subset [-1, 1]^n$  denote the decision space. At each  $t$  of  $T$  time steps, the environment selects a cost vector  $L_t$ , and simultaneously, the player (decision maker) selects  $x_t \in D$ . The loss incurred by the decision maker for this prediction is  $L_t^\dagger x_t$ . Let

$$L_{\min} := \min_{x \in D} \sum_{t=1}^T L_t^\dagger x$$

be the loss of the best single decision in hindsight. The goal of the decision maker is to minimize the *regret*,

$$R = \sum_{t=1}^T L_t^\dagger x_t - L_{\min}.$$

We assume that  $L_t^\dagger x \in [0, 1]$  for all  $x \in D$ . We also assume that the environment is *adaptive*, i.e., the cost vector  $L_t$  selected by the environment at time  $t$  may depend arbitrarily on the history  $(L_1, x_1, \dots, L_{t-1}, x_{t-1})$  (note that without loss of generality this dependence may be assumed to be deterministic.) We show that even against such a powerful environment, it is possible to ensure that  $R$  is small with high probability.

As in [8], we will require a barycentric spanner for  $D$ . Recall that a *barycentric spanner* for  $D$  is a set

$$\{y_1, \dots, y_n\} \subseteq D$$

such that every  $x \in D$  can be written as a linear combination of  $y_i$ ’s with coefficients in  $[-1, 1]$ . A  $c$ -barycentric spanner is defined similarly where we allow coefficients to be in  $[-c, c]$ . For  $c > 1$ ,  $c$ -barycentric spanners for  $D$  may be found efficiently (see [4].) However, for ease of exposition we’ll assume that we have an actual barycentric spanner. (Using a  $c$ -barycentric spanner instead will only affect the constants.) Finally, if the set  $D$  is too large (for example if it is infinite) we can replace it by a cover of size at most  $(4nT)^{n/2}$ , as the loss of the optimal decision in this cover is within an additive  $\sqrt{nT}$  of the optimal loss in  $D$ ; see [8][Lemma 3.1] for details. Accordingly, after doing this transformation if necessary, we may assume that  $D$  is finite and  $\ln |D| = O(n \ln T)$ . Only the logarithm of the cardinality of the set will enter in our bounds.

## 3 Algorithm and Main Result

The algorithm presented below is a modification of the algorithm in [8]. Note that the difference is in the way we update weights  $w_t$ , using lower confidence intervals. This idea of using confidence intervals is motivated by the **Exp3.P** algorithm of Auer et al. [2]. Feeding in confidence bounds, as opposed to unbiased estimates of the losses, to the exponential updates is the crucial change we make to the algorithm of Dani et al [8]. Lemma 5 below shows that, with high probability, for any  $x \in D$ ,  $\sum_t \tilde{L}_t(x)$  lower bounds  $\sum_t L_t^\dagger x$  (up to an additive  $O(\sqrt{T})$  term). Our algorithm reduces to **Exp3.P** in the special case of the  $n$ -armed bandit problem (when  $D = \{e_1, \dots, e_n\}$ ). As we point out in the next section, Auer et al.’s proof can be simplified by using the simple corollary of Freedman’s inequality [10] that we state as Lemma 2 below.

The main result of this paper is the following guarantee on the algorithm.

**Theorem 1** *Let  $T \geq 4$ ,  $n \geq 2$  and  $\delta \leq \frac{1}{e}$ . If we set  $\gamma = \frac{n^{3/2}}{\sqrt{T}}$ ,  $\delta' = \frac{\delta}{|D| \log_2 T}$ , and  $\eta = \frac{1}{\sqrt{nT+2\sqrt{\ln(1/\delta')}}}$ , then against any adaptive adversary with probability at least  $1 - 4\delta$ ,*

$$R = O(n^{3/2}\sqrt{T} \ln(nT/\delta)).$$

The dependence on  $T$  is optimal (up to logarithmic factors). We get the same dependence on  $n$  as Dani et al. [8]. The lower bound known for this problem is  $\Omega(n\sqrt{T})$  [8]. Recently,  $O(n\sqrt{T})$  regret bounds have been obtained for the stochastic version of the problem [9]. This leads us to conjecture that the lower bound is tight and it remains an open

**Algorithm 3.1:** GEOMETRICHEDGE.P( $D, \gamma, \eta, \delta'$ )

$\forall x \in D, w_1(x) := 1$   
 $W_1 := |D|$   
**for**  $t = 1$  **to**  $T$   
 $\forall x \in D,$   
 $p_t(x) = (1 - \gamma) \frac{w_t(x)}{W_t} + \frac{\gamma}{n} \mathbf{I}\{x \in \text{spanner}\}$   
 Sample  $x_t$  according to distribution  $p_t$   
 Incur and observe loss  $\ell_t := L_t^\dagger x_t$   
 $\mathbf{C}_t := \mathbb{E}_{p_t}[xx^\dagger]$   
 $\hat{L}_t := \ell_t \mathbf{C}_t^{-1} x_t$   
 $\forall x \in D, \tilde{L}_t(x) := \hat{L}_t^\dagger x - 2x^\dagger \mathbf{C}_t^{-1} x \sqrt{\frac{\ln(1/\delta')}{nT}}$   
 $\forall x \in D, w_{t+1}(x) := w_t(x) \exp\{-\eta \tilde{L}_t(x)\}$   
 $W_{t+1} = \sum_{x \in D} w_{t+1}(x)$

question to close the gap (for the dependence on  $n$ ) between upper and lower bounds. We also note here that although the analysis we provide is for losses, essentially the same algorithm, with a similar analysis, works for gains. We just have to make a few obvious changes to the algorithm: instead of subtracting, we *add* the correction term to the gain estimates and replace  $-\eta$  with  $\eta$  in the exponential update.

#### 4 Concentration for Martingales

In this section we derive a concentration inequality for martingale difference sequences. It is a direct application of Freedman's inequality.

**Lemma 2** Suppose  $X_1, \dots, X_T$  is a martingale difference sequence with  $|X_t| \leq b$ . Let

$$\text{Var}_t X_t = \mathbf{Var}(X_t | X_1, \dots, X_{t-1}).$$

Let  $V = \sum_{t=1}^T \text{Var}_t X_t$  be the sum of conditional variances of  $X_t$ 's. Further, let  $\sigma = \sqrt{V}$ . Then we have, for any  $\delta < 1/e$  and  $T \geq 4$ ,

$$\begin{aligned} \text{Prob} \left( \sum_{t=1}^T X_t > 2 \max \left\{ 2\sigma, b\sqrt{\ln(1/\delta)} \right\} \sqrt{\ln(1/\delta)} \right) \\ \leq \log_2(T) \delta. \end{aligned}$$

**Proof:** Note that a crude upper bound on  $\text{Var}_t X_t$  is  $b^2$ . Thus,  $\sigma \leq b\sqrt{T}$ . We choose a discretization  $0 = \alpha_{-1} < \alpha_0 < \dots < \alpha_l$  such that  $\alpha_{i+1} = 2\alpha_i$  for  $i \geq 0$  and  $\alpha_l \geq b\sqrt{T}$ . We will specify the choice of  $\alpha_0$  shortly. We then have,

$$\begin{aligned} \text{Prob} \left( \sum_t X_t > 2 \max \{ 2\sigma, \alpha_0 \} \sqrt{\ln(1/\delta)} \right) \\ = \sum_{j=0}^l \text{Prob} \left( \sum_t X_t > 2 \max \{ 2\sigma, \alpha_j \} \sqrt{\ln(1/\delta)} \right. \\ \left. \& \alpha_{j-1} < \sigma \leq \alpha_j \right) \end{aligned}$$

$$\begin{aligned} &\leq \sum_{j=0}^l \text{Prob} \left( \sum_t X_t > 2\alpha_j \sqrt{\ln(1/\delta)} \right. \\ &\quad \left. \& \alpha_{j-1}^2 < V \leq \alpha_j^2 \right) \\ &\leq \sum_{j=0}^l \text{Prob} \left( \sum_t X_t > 2\alpha_j \sqrt{\ln(1/\delta)} \& V \leq \alpha_j^2 \right) \\ &\stackrel{(*)}{\leq} \sum_{j=0}^l \exp \left( \frac{-4\alpha_j^2 \ln(1/\delta)}{2\alpha_j^2 + \frac{2}{3} (2\alpha_j \sqrt{\ln(1/\delta)}) b} \right) \\ &= \sum_{j=0}^l \exp \left( \frac{-2\alpha_j \ln(1/\delta)}{\alpha_j + \frac{2}{3} (\sqrt{\ln(1/\delta)}) b} \right) \end{aligned}$$

where the inequality  $(*)$  follows from Freedman's inequality (Theorem 9). If we now choose  $\alpha_0 = b\sqrt{\ln(1/\delta)}$  then  $\alpha_j \geq b\sqrt{\ln(1/\delta)}$  for all  $j$  and hence every term in the above summation is bounded by  $\exp\left(\frac{-2\ln(1/\delta)}{1+2/3}\right) < \delta$ . Choosing  $l = \log_2(\sqrt{T})$  ensures that  $\alpha_l \geq b\sqrt{T}$ . Thus we have

$$\begin{aligned} &\text{Prob} \left( \sum_{t=1}^T X_t > 2 \max \{ 2\sigma, b\sqrt{\ln(1/\delta)} \} \sqrt{\ln(1/\delta)} \right) \\ &= \text{Prob} \left( \sum_t X_t > 2 \max \{ 2\sigma, \alpha_0 \} \sqrt{\ln(1/\delta)} \right) \\ &\leq (l+1)\delta = (\log_2(\sqrt{T}) + 1)\delta \leq \log_2(T)\delta. \end{aligned}$$

■

This inequality says that, roughly speaking,  $\sum_t X_t$  is of the order of  $\sigma\sqrt{\ln(1/\delta)}$  which is a central limit theorem-like behavior except that  $\sigma$  here is not fixed but is the actual sum of conditional variances, a random quantity. The overall constant in front of  $\sigma$  is 4. This can be improved to 2 by a slightly more careful analysis. We already know of two instances in the literature where Lemma 2 can be used to give shorter proofs of certain probabilistic upper bounds.

1. The first is in the proof of **Exp3.P**'s regret bound itself. To show that the estimates are upper bounds on the actual losses of an action, the authors explicitly use the exponential moment method in the proof of their Lemma 6.1. Essentially the same lemma can be proved by a direction application of the above lemma.
2. The other instance is in Cesa-Bianchi and Gentile's paper [5] on online to batch conversions. When an online algorithm is run on i.i.d. data with a non-negative and bounded loss function, the conditional variance of the loss at time  $t$  can immediately be bounded by the risk of the hypothesis at time  $t-1$ . The authors use this fact along with an application of Freedman's inequality to prove a sharp upper bound (Proposition 2 in their paper) on the average risk of the hypotheses generated by the online algorithm in terms of its actual cumulative loss. The same result can be quickly derived by an application of the above lemma.

## 5 Analysis

The remainder of the paper is devoted to the proof of Theorem 1. We first state several results obtained in Dani et al [8] which will be important in our proofs.

**Lemma 3** For any  $x \in D$  and  $t \in \{1, \dots, T\}$ , it holds that

1.  $|\widehat{L}_t^\dagger x| \leq n^2/\gamma$
2.  $x^\dagger C_t^{-1} x \leq n^2/\gamma$ .
3.  $\sum_{x \in D} p_t(x) x^\dagger C_t^{-1} x = n$ .
4.  $\mathbb{E}_t \left( \widehat{L}_t^\dagger x \right)^2 \leq x^\dagger C_t^{-1} x$ .

We now prove a bound on the perturbed estimated costs,  $\widetilde{L}_t$ , which are used to update the distribution.

**Lemma 4** For all  $x \in D$ ,  $|\widetilde{L}_t(x)| \leq \sqrt{nT} + 2\sqrt{\ln(1/\delta')}$ .

**Proof:** For each  $x \in D$ ,

$$\begin{aligned} |\widetilde{L}_t(x)| &\leq |\widehat{L}_t \cdot x| + \left| 2x^\dagger C_t^{-1} x \sqrt{\frac{\ln(1/\delta')}{nT}} \right| \\ &\leq \frac{n^2}{\gamma} + 2\frac{n^2}{\gamma} \sqrt{\frac{\ln(1/\delta')}{nT}} \\ &\leq \sqrt{nT} + 2\sqrt{\ln(1/\delta')} \end{aligned}$$

using Lemma 3 and the choice of  $\gamma = \frac{n^{3/2}}{\sqrt{T}}$ .  $\blacksquare$

### 5.1 High Confidence Bounds

Let  $\mathbb{E}_t[\cdot]$  denote  $\mathbb{E}[\cdot|x_1, \dots, x_{t-1}]$ . Since we are considering adaptive (but deterministic) adversaries,  $L_t$  is not random given  $x_1, \dots, x_{t-1}$ . Observe that  $\mathbb{E}_t[x_t x_t^\dagger] = \mathbb{E}_{x \sim p_t}[xx^\dagger]$  and thus,  $\mathbb{E}_t[\widehat{L}_t] = L_t$ . However, the fluctuations of the random variable  $\widehat{L}_t$  are very large. The following lemma provides a bound on these fluctuations.

**Lemma 5** Assume  $T \geq 4$ . Let  $\delta' = \frac{\delta}{|D|^{\log_2 T}}$ . Then with probability at least  $1 - \delta$ , simultaneously for all  $x \in D$ ,

$$\sum_t \widetilde{L}_t(x) \leq \sum_t L_t^\dagger x + 2 \left(1 + \sqrt{nT}\right) \ln(1/\delta')$$

**Proof:** Fix  $x \in D$ . Let  $M_t = M_t(x) = \widehat{L}_t^\dagger x - L_t^\dagger x$ . Then  $(M_t)$  is a martingale difference sequence. Using Lemma 3,  $|M_t| \leq \frac{n^2}{\gamma} + 1 = \sqrt{nT} + 1$ . Let  $V = \sum_t \text{Var}_t(M_t)$  and let  $\sigma = \sqrt{V}$ . Using Lemma 2, we have that with probability at least  $1 - \delta' \log_2 T$ ,

$$\begin{aligned} \sum_t \widehat{L}_t^\dagger x &\leq \sum_t L_t^\dagger x + 2 \max\{2\sigma, \\ &\quad (1 + \sqrt{nT})\sqrt{\ln(1/\delta')}\} \sqrt{\ln(1/\delta')} \end{aligned} \quad (1)$$

Now note that

$$\sigma \leq \sqrt{\sum_t x^\dagger C_t^{-1} x} \leq \frac{1}{2} \left( \frac{\sum_t x^\dagger C_t^{-1} x}{\sqrt{nT}} + \sqrt{nT} \right),$$

by the arithmetic mean-geometric mean inequality.

Substituting this into (1), we have

$$\begin{aligned} \sum_t \widehat{L}_t^\dagger x &\leq \sum_t L_t^\dagger x + 2 \max \left\{ \left(1 + \sqrt{nT}\right) \sqrt{\ln(1/\delta')}, \right. \\ &\quad \left. \left( \frac{\sum_t x^\dagger C_t^{-1} x}{\sqrt{nT}} + \sqrt{nT} \right) \right\} \sqrt{\ln(1/\delta')} \end{aligned}$$

with probability at least  $1 - \delta' \log_2 T$ .

Finally, taking a union bound over all  $x \in D$  and rearranging (using the fact that  $\max\{a+b, c\} \leq a + \max\{b, c\}$  if  $a \geq 0$ ) gives the required result.  $\blacksquare$

### 5.2 Potential Function Analysis

By Lemma 4 and our choice of  $\eta = \frac{1}{\sqrt{nT+2\sqrt{\ln(1/\delta')}}}$ , we have

$$|\eta \widetilde{L}_t(x)| \leq 1.$$

In the following computation, we will use the facts that  $e^{-a} \leq 1 - a + a^2$  whenever  $|a| \leq 1$ .

$$\begin{aligned} \frac{W_{t+1}}{W_t} &= \sum_{x \in D} \frac{w_t(x) \exp(-\eta \widetilde{L}_t(x))}{W_t} \\ &\leq \sum_{x \in D} \frac{w_t(x)}{W_t} (1 - \eta \widetilde{L}_t(x) + \eta^2 (\widetilde{L}_t(x))^2) \\ &\leq 1 + \frac{\eta}{1 - \gamma} \left( - \sum_{x \in D} p_t(x) \widetilde{L}_t(x) \right. \\ &\quad \left. + \sum_{x \in \text{spanner}} \frac{\gamma}{n} \widetilde{L}_t(x) + \sum_{x \in D} p_t(x) \eta (\widetilde{L}_t(x))^2 \right) \end{aligned}$$

since by definition of  $p_t$ ,

$$\frac{w_t(x)}{W_t} = \frac{p_t(x) - \frac{\gamma}{n} \mathbf{I}\{x \in \text{spanner}\}}{1 - \gamma}.$$

Note that we have,

$$\begin{aligned} & - \sum_{x \in D} p_t(x) \widetilde{L}_t(x) \\ &= - \sum_{x \in D} p_t(x) \widehat{L}_t^\dagger x + 2 \sum_{x \in D} p_t(x) x^\dagger C_t^{-1} x \sqrt{\frac{\ln(1/\delta')}{nT}} \\ &= - \sum_{x \in D} p_t(x) \widehat{L}_t^\dagger x + 2n \sqrt{\frac{\ln(1/\delta')}{nT}} \end{aligned}$$

where the last step is by Lemma 3.

Further, since  $(b+c)^2 \leq 2(b^2+c^2)$  for every  $b, c$ , apply-

ing the definition of  $\tilde{L}_t(x)$ , we also have

$$\begin{aligned}
& \sum_{x \in D} p_t(x) \eta(\tilde{L}_t(x))^2 \\
& \leq 2\eta \sum_{x \in D} p_t(x) \left( (\hat{L}_t^\dagger x)^2 + (2x^\dagger \mathbf{C}_t^{-1} x)^2 \frac{\ln(1/\delta')}{nT} \right) \\
& \leq 2\eta \sum_{x \in D} p_t(x) \left( (\hat{L}_t^\dagger x)^2 + 4x^\dagger \mathbf{C}_t^{-1} x \frac{n^2 \ln(1/\delta')}{\gamma nT} \right) \\
& = 2\eta \left[ \sum_{x \in D} p_t(x) (\hat{L}_t^\dagger x)^2 + \frac{4 \ln(1/\delta')}{\sqrt{nT}} \sum_{x \in D} p_t(x) x^\dagger \mathbf{C}_t^{-1} x \right] \\
& = 2\eta \left[ \sum_{x \in D} p_t(x) (\hat{L}_t^\dagger x)^2 + \frac{4\sqrt{n} \ln(1/\delta')}{\sqrt{T}} \right]
\end{aligned}$$

by successive applications of Lemma 3.

Putting these together, we have

$$\begin{aligned}
\frac{W_{t+1}}{W_t} & \leq 1 + \frac{\eta}{1-\gamma} \left( - \sum_{x \in D} p_t(x) \hat{L}_t^\dagger x \right. \\
& \quad + 2\sqrt{\frac{n \ln(1/\delta')}{T}} \\
& \quad + \sum_{x \in \text{spanner}} \frac{\gamma}{n} \tilde{L}_t(x) \\
& \quad + 2\eta \sum_{x \in D} p_t(x) (\hat{L}_t^\dagger x)^2 \\
& \quad \left. + 8\eta \frac{\sqrt{n} \ln(1/\delta')}{\sqrt{T}} \right)
\end{aligned}$$

Taking logs, using the fact that  $\ln(1+x) \leq x$ , and summing over  $t$ , we have

$$\begin{aligned}
\ln \left( \frac{W_{T+1}}{W_1} \right) & \leq \frac{\eta}{1-\gamma} \left[ - \sum_{t=1}^T \sum_{x \in D} p_t(x) \hat{L}_t^\dagger x \right. \\
& \quad + 2\sqrt{nT \ln(1/\delta')} \\
& \quad + \sum_{t=1}^T \sum_{x \in \text{spanner}} \frac{\gamma}{n} \tilde{L}_t(x) \\
& \quad + 2\eta \sum_{t=1}^T \sum_{x \in D} p_t(x) (\hat{L}_t^\dagger x)^2 \\
& \quad \left. + 8\eta \ln(1/\delta') \sqrt{nT} \right] \quad (2)
\end{aligned}$$

The next three lemmas will bound the three summations that appear on the right hand side above.

**Lemma 6** *With probability at least  $1 - \delta$ ,*

$$\begin{aligned}
& \sum_{t=1}^T L_t^\dagger x_t - \sum_{t=1}^T \sum_x p_t(x) \hat{L}_t^\dagger x \\
& \leq (\sqrt{n} + 1) \sqrt{2T \ln(1/\delta)} + \frac{4}{3} \ln(1/\delta) \left( \frac{n^2}{\gamma} + 1 \right).
\end{aligned}$$

**Proof:** Let us define  $\bar{x} := \mathbb{E}_{x \sim p_t} x = \sum_{x \in D} p_t(x) x$  and  $Y_t := \ell_t - \hat{L}_t^\dagger \bar{x}$ . Note that  $\mathbb{E}_t \hat{L}_t^\dagger \bar{x} = \mathbb{E}_t \ell_t$  and therefore  $Y_t$  is a martingale difference sequence.

We bound the conditional variance of  $Y_t$  as follows.

$$\begin{aligned}
\sqrt{\text{Var}_t Y_t} & = \sqrt{\mathbb{E}_t(Y_t^2)} \\
& = \sqrt{\mathbb{E}_t \left( (\hat{L}_t^\dagger \bar{x} - \ell_t)^2 \right)} \\
& \leq \sqrt{\mathbb{E}_t (\hat{L}_t^\dagger \bar{x})^2} + \sqrt{\mathbb{E}_t (\ell_t^2)} \quad \text{by Cauchy-Schwarz} \\
& \leq \sqrt{\mathbb{E}_t (\hat{L}_t^\dagger \bar{x})^2} + 1 \quad \text{since } |\ell_t| \leq 1 \\
& \leq \sqrt{\bar{x}^\dagger \mathbf{C}_t^{-1} \bar{x}} + 1 \quad \text{by Lemma 3} \\
& \leq \sqrt{\mathbb{E}_{x \sim p_t} x^\dagger \mathbf{C}_t^{-1} x} + 1 \quad \text{by Jensen's inequality} \\
& = \sqrt{n} + 1 \quad \text{by Lemma 3.}
\end{aligned}$$

Moreover,  $|Y_t| \leq n^2/\gamma + 1$  by Lemma 3. Applying Bernstein's inequality for martingale differences (see Appendix) to the sequence  $Y_t$ , we obtain that with probability at least  $1 - \delta$ ,

$$\sum_{t=1}^T Y_t \leq (\sqrt{n} + 1) \sqrt{2T \ln(1/\delta)} + \frac{4}{3} \ln(1/\delta) \left( \frac{n^2}{\gamma} + 1 \right),$$

which is the desired bound.  $\blacksquare$

**Lemma 7** *With probability at least  $1 - \delta$ ,*

$$\sum_{t=1}^T \sum_{x \in \text{spanner}} \frac{\gamma}{n} \tilde{L}_t(x) \leq \gamma T + 2\gamma \left( 1 + \sqrt{nT} \right) \ln(1/\delta').$$

**Proof:** Using Lemma 5, with probability at least  $1 - \delta$ , we have, for all  $x \in \text{spanner}$ ,

$$\begin{aligned}
\frac{\gamma}{n} \sum_t \tilde{L}_t(x) & \leq \frac{\gamma}{n} \sum_t L_t^\dagger x + \frac{2\gamma}{n} \left( 1 + \sqrt{nT} \right) \ln(1/\delta') \\
& \leq \frac{\gamma T}{n} + \frac{2\gamma}{n} \left( 1 + \sqrt{nT} \right) \ln(1/\delta'),
\end{aligned}$$

because  $L_t^\dagger x$ , being the loss of an element of the spanner, is bounded by 1. Summing over the  $n$  elements of the spanner, we get the desired bound.  $\blacksquare$

**Lemma 8** *With probability at least  $1 - \delta$ ,*

$$\sum_{t=1}^T \sum_x p_t(x) (\hat{L}_t^\dagger x)^2 \leq nT + T \sqrt{2n \ln(1/\delta)}.$$

**Proof:** First we observe that for  $1 \leq t \leq T$ ,

$$\begin{aligned}
\sum_x p_t(x) (\hat{L}_t^\dagger x)^2 & = \sum_x p_t(x) \hat{L}_t^\dagger x x^\dagger \hat{L}_t \\
& = \hat{L}_t^\dagger \left( \sum_x p_t(x) x x^\dagger \right) \hat{L}_t \\
& = \ell_t^2 x_t^\dagger \mathbf{C}_t^{-1} \mathbf{C}_t \mathbf{C}_t^{-1} x_t \\
& \leq x_t^\dagger \mathbf{C}_t^{-1} x_t
\end{aligned}$$

Summing over  $t$ ,

$$\sum_{t=1}^T \sum_x p_t(x) (\hat{L}_t^\dagger x)^2 \leq \sum_{t=1}^T x_t^\dagger \mathbf{C}_t^{-1} x_t.$$

Lemma 3 tells us that, on the one hand, the summands  $x_t^\dagger \mathbf{C}_t^{-1} x_t$  are uniformly bounded by  $n^2/\gamma = \sqrt{nT}$ , and on the other hand, that each one has expectation  $n$ , even conditioned on the previous ones.

Applying the Hoeffding-Azuma inequality to the martingale difference sequence

$$x_t^\dagger \mathbf{C}_t^{-1} x_t - \mathbb{E}_{x \sim p_t} x_t^\dagger \mathbf{C}_t^{-1} x$$

it follows that, with probability at least  $1 - \delta$ ,

$$\sum_{t=1}^T x_t^\dagger \mathbf{C}_t^{-1} x_t \leq nT + T \sqrt{2n \ln(1/\delta)},$$

completing the proof.  $\blacksquare$

Substituting the bounds of Lemmas 6, 7 and 8 into (2), we obtain that with probability at least  $1 - 3\delta$ ,

$$\begin{aligned} \ln \left( \frac{W_{T+1}}{W_1} \right) &\leq \frac{\eta}{1-\gamma} \left[ - \sum_{t=1}^T L_t^\dagger x_t \right. \\ &\quad + (\sqrt{n} + 1) \sqrt{2T \ln(1/\delta)} \\ &\quad + \frac{4}{3} \ln(1/\delta) \left( \frac{n^2}{\gamma} + 1 \right) \\ &\quad + 2\sqrt{nT \ln(1/\delta')} + \gamma T \\ &\quad + 2\gamma \left( 1 + \sqrt{nT} \right) \ln(1/\delta') \\ &\quad + 2\eta n T + 2\eta T \sqrt{2n \ln(1/\delta)} \\ &\quad \left. + 8\eta \ln(1/\delta') \sqrt{nT} \right] \quad (3) \end{aligned}$$

On the other hand, using Lemma 5, we have with probability at least  $1 - \delta$ , for all  $x \in D$ ,

$$\begin{aligned} \ln \frac{W_{T+1}}{W_1} &\geq -\eta \left( \sum_{t=1}^T \tilde{L}_t(x) \right) - \ln |D| \\ &\geq -\eta \sum_{t=1}^T L_t^\dagger x - 2\eta(1 + \sqrt{nT}) \ln(1/\delta') - \ln |D|. \quad (4) \end{aligned}$$

Combining (3) with (4), we have that with probability at least

$1 - 4\delta$ , for every  $x \in D$ ,

$$\begin{aligned} \sum_{t=1}^T L_t^\dagger x_t &\leq \sum_{t=1}^T L_t^\dagger x \\ &\quad + 2(1 + \sqrt{nT}) \ln(1/\delta') \\ &\quad + \frac{1}{\eta} \ln |D| \\ &\quad + (\sqrt{n} + 1) \sqrt{2T \ln(1/\delta)} \\ &\quad + \frac{4}{3} \ln(1/\delta) \left( \frac{n^2}{\gamma} + 1 \right) \\ &\quad + 2\sqrt{nT \ln(1/\delta')} + \gamma T \\ &\quad + 2\gamma \left( 1 + \sqrt{nT} \right) \ln(1/\delta') \\ &\quad + 2\eta n T + 2\eta T \sqrt{2n \ln(1/\delta)} \\ &\quad + 8\eta \ln(1/\delta') \sqrt{nT} \end{aligned}$$

Recall that  $\eta = \frac{1}{\sqrt{nT} + 2\sqrt{\ln(1/\delta')}}$ ,  $\gamma = \frac{n^{3/2}}{\sqrt{T}}$ ,  $\delta' = \delta/(|D| \log_2 T)$ , and  $\ln |D| = O(n \ln T)$ . Plugging in these values yields

$$\sum_{t=1}^T L_t^\dagger x_t \leq L_{\min} + O(n^{3/2} \sqrt{T} \ln(nT/\delta)),$$

completing the proof of Theorem 1.

## 6 Conclusions and Open Problems

We presented an algorithm that achieves the desired regret bound of  $O^*(\sqrt{T})$  with high probability. However, the quest for an *efficient* algorithm with the same high-probability guarantee, even for the special case of bandit online shortest paths, is still open. Achieving similar results for general convex functions is also an intriguing open question.

### A Concentration Inequalities

The following inequalities are well known. Theorem 9 is from [10]. Lemmas 10 and 11 can be found, for instance, in [6], Appendix A.

**Theorem 9 (Freedman)** *Suppose  $X_1, \dots, X_T$  is a martingale difference sequence, and  $b$  is an uniform upper bound on the steps  $X_i$ . Let  $V$  denote the sum of conditional variances,*

$$V = \sum_{i=1}^n \mathbf{Var}(X_i | X_1, \dots, X_{i-1}).$$

*Then, for every  $a, v > 0$ ,*

$$\text{Prob} \left( \sum X_i \geq a \text{ and } V \leq v \right) \leq \exp \left( \frac{-a^2}{2v + 2ab/3} \right).$$

**Lemma 10 (Bernstein's inequality for martingales)** *Let  $Y_1, \dots, Y_T$  be a martingale difference sequence. Suppose that  $Y_t \in [a, b]$  and*

$$\mathbb{E}[Y_t^2 | X_{t-1}, \dots, X_1] \leq v \text{ a.s.}$$

for all  $t \in \{1, \dots, T\}$ . Then for all  $\delta > 0$ ,

$$\Pr \left( \sum_{t=1}^T Y_t > \sqrt{2Tv \ln(1/\delta)} + 2 \ln(1/\delta)(b-a)/3 \right) \leq \delta$$

**Lemma 11 (Hoeffding-Azuma inequality)** Let  $Y_1, \dots, Y_T$  be a martingale difference sequence. Suppose that  $|Y_t| \leq c$  almost surely for all  $t \in \{1, \dots, T\}$ . Then for all  $\delta > 0$ ,

$$\Pr \left( \sum_{t=1}^T Y_t > \sqrt{2Tc^2 \ln(1/\delta)} \right) \leq \delta$$

## References

- [1] Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, 2008. to appear.
- [2] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.
- [3] Baruch Awerbuch, David Holmer, Herb Rubens, and Robert Kleinberg. Provably competitive adaptive routing. In *Proceedings of the 31st IEEE INFOCOM*, volume 1, pages 631–641, 2005.
- [4] Baruch Awerbuch and Robert Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC)*, 2004.
- [5] Nicolò Cesa-Bianchi and Claudio Gentile. Improved risk tail bounds for on-line algorithms. *IEEE Transactions on Information Theory*, 54(1):386–39, Jan 2008.
- [6] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [7] Varsha Dani and Thomas P. Hayes. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.
- [8] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. The price of bandit information for online optimization. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20 (NIPS 2007)*. MIT Press, 2008.
- [9] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, 2008. to appear.
- [10] David A. Freedman. On tail probabilities for martingales. *The Annals of Probability*, 3(1):100–118, Feb 1975.
- [11] András György, Tamás Linder, Gábor Lugosi, and György Ottucsák. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 8:2369–2403, 2007.
- [12] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.
- [13] Robert Kleinberg. *Online decision problems with large strategy sets*. PhD thesis, MIT, 2005.
- [14] H. Brendan McMahan and Avrim Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT)*, 2004.
- [15] Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. *Journal of Machine Learning Research*, 4:773–818, 2003.



---

# Adapting to a Changing Environment: the Brownian Restless Bandits

---

Aleksandrs Slivkins\* and Eli Upfal†

## Abstract

In the multi-armed bandit (MAB) problem there are  $k$  distributions associated with the rewards of playing each of  $k$  strategies (slot machine arms). The reward distributions are initially unknown to the player. The player iteratively plays one strategy per round, observes the associated reward, and decides on the strategy for the next iteration. The goal is to maximize the reward by balancing *exploitation*: the use of acquired information, with *exploration*: learning new information.

We introduce and study a *dynamic* MAB problem in which the reward functions stochastically and gradually change in time. Specifically, the expected reward of each arm follows a Brownian motion, a discrete random walk, or similar processes. In this setting a player has to continuously keep exploring in order to adapt to the changing environment. Our formulation is (roughly) a special case of the notoriously intractable *restless MAB problem*.

Our goal here is to characterize the cost of learning and adapting to the changing environment, in terms of the stochastic rate of the change. We consider an infinite time horizon, and strive to minimize the average cost per step which we define with respect to a hypothetical algorithm that at every step plays the arm with the maximum expected reward at this step. A related line of work on the *adversarial MAB problem* used a significantly weaker benchmark, the best *time-invariant* policy.

The dynamic MAB problem models a variety of practical online, game-against-nature type optimization settings. While building on prior work, algorithms and steady-state analysis for the dynamic setting require a novel approach based on different stochastic tools.

## 1 Introduction

The multi-armed bandit (MAB) problem [27, 5, 12] has been studied extensively for over 50 years in Operations Research, Economics and Computer Science literature, modeling on-line decisions under uncertainty in a setting in which an agent simultaneously attempts to acquire new knowledge and to optimize its decisions based on the existing knowledge. In the basic MAB setting, which we term the *static MAB problem*, there are  $k$  time-invariant probability distributions associated with the rewards of playing each of the  $k$  strategies (slot machine arms). The distributions are initially unknown to the player. The player iteratively plays one strategy per round, observes the associated reward, and decides on the strategy for the next iteration. The goal of a MAB algorithm is to optimize the total reward<sup>1</sup> by balancing *exploitation*: the use of acquired information, with *exploration*: learning new information. For several algorithms in the literature (e.g. see [5, 2]) as the number of rounds goes to infinity the expected total reward asymptotically approaches that of playing a strategy with the highest expected reward. The quality of an algorithm for the static MAB problem is therefore measured by the expected cost, or *regret*, incurred during an initial finite time interval. The regret in the first  $t$  steps is defined as the expected gap between the total reward collected by the algorithm and that collected by playing an optimal strategy in these  $t$  steps.

The MAB problem models a variety of practical online optimization problems. As an example consider a packet routing network where a router learns about delays on routes by measuring the time to receive an acknowledgment for a packet sent on that route [4, 16]. The delay for one packet on a given route is a random value drawn from some distribution. The router must try various routes in order to learn about the delays. Trying a loaded route adds unnecessary delay to the routing of one packet, while discovering a route with low delay can improve the routing of the future packets.

Another application is in marketing and advertising. A store would like to display and advertise the products that sell best, but it needs to display and advertise various products to learn how good they sell. Similarly, a web search engine tries to optimize its revenue by displaying advertise-

---

\*Microsoft Research, Mountain View CA. E-mail: slivkins at microsoft.com. Parts of this work has been completed while A. Slivkins was a postdoc at Brown University.

†Computer Science Department, Brown University, Providence RI. E-mail: eli at cs.brown.edu. Supported in part by NSF awards CCR-0121154 and DMI-0600384, and ONR Award N000140610607.

---

<sup>1</sup>In this paper the *total reward* is simply the sum of the rewards, following the line of work in [21, 2, 3] and many other papers. Alternatively, many papers consider the *time-discounted* sum of rewards, e.g. see [5, 12, 29] and references therein.

ments that would bring the largest number of clicks for a given web content. The company needs to experiment with various combinations of advertisements and page contents in order to find the best matches. The cost of these experiments is the loss of advertisement clicks when trying unsuccessful matches [25].

The above examples demonstrate the practical applications of the "explore and exploit" paradigm captured in the MAB model. These examples also point out the limitation of the static approach to the problem. The delay on a route is gradually changing over time, and the router needs to continuously adapt its routing strategy to the changes in route delays. Taste and fashion change over time. A store cannot completely rely on information collected in the previous season to optimize for the next one. Similarly, a web search engine continually updates their content matching strategies to account for the changing customers' response.

A number of models have been proposed for capturing the dynamic aspect of the MAB problem. Motivated by task scheduling, Gittins [13] considered the case where only the state of the active arm (the arm currently being played) can change in a given step, giving an optimal policy for the Bayesian formulation with time discounting. This seminal result gave rise to a rich line of work (e.g. [11, 12, 32, 31, 30, 6, 29]), a proper review of which is beyond the scope of this paper. In particular, Whittle [33] introduced an extension termed *restless bandits* [33, 7, 24], where the states of all arms can change in each step according to a known (but arbitrary) stochastic transition function. Restless bandits are notoriously intractable: e.g. even with deterministic transitions the problem of computing an (approximately) optimal strategy is PSPACE-hard [26]. Guha et al. [14, 15] have recently made a progress on some tractable special cases of the restless MAB problem.<sup>2</sup> Their motivations, the actual problems they considered, and the techniques they used, are very different from ours. In [14] they gave a constant-factor approximation for the special case of the problem in which arms move stochastically between two possible states. This result was improved to a 2-approximation in [15], and extended to arms assuming a number of possible states, but with a very strict set of transition probabilities that are not compatible with the stochastic processes discussed here.

Auer et al. [3] adopted an adversarial approach: they defined the *adversarial MAB problem* where the reward distributions are allowed to change arbitrarily in time, and the goal is to approach the performance of the best *time-invariant* policy. This formulation has been further studied in [1, 20, 17, 22, 10, 9, 19, 8]. Auer et al. [3, 1] also considered a more general definition of regret, where the comparison is to the best policy that can change arms a limited number of times. Due to the overwhelming strength of the adversary, the guarantees obtained in this line of work are relatively weak when applied to the setting that we consider in this paper.

We propose and study here a somewhat different approach to addressing the dynamic nature of the MAB problem. We note that in a variety of practical applications the time evolution of the system, in particular of the reward functions, is *gradual*. Obvious examples are price, supply and demand

<sup>2</sup>These papers were published after the initial technical report version of this paper appeared.

in economics, load and delay in networks, etc. A gradual stochastic evolution is traditionally modeled via a random walk or a Brownian motion; for instance, in Mathematical Finance the (geometric) Brownian motion (Wiener process) is the standard model for continuous-time evolution of a stock price. In line with this approach, we describe the *state* of each arm – its expected reward at time  $t$  – via a Brownian motion.<sup>3</sup> The actual reward at a given time is an independent random sample from the reward distribution parameterized by the current state of this arm, e.g. a 0-1 random variable with an expectation given by the state of the arm (in the web advertising setting this corresponds to a user clicking or not clicking on an ad).

We are interested in systems that exhibit a stationary, steady-state behavior. For this reason instead of the usual Brownian motion on a real line (which diverges to infinity) we consider a Brownian motion on an interval with reflecting bounds. Following the bulk of the stochastic MAB literature, we assume that the evolution of each arm is independent (in fact, we conjecture that regret is maximized in the case of independently evolving arms).

Our goal here is to characterize the long-term average cost of adapting to such changing environment in terms of the stochastic rate of change – the *volatility* of Brownian motion. The paradigmatic setting for us is one in which each arm's state has the same stationary distribution and, therefore, all arms are essentially equivalent in the long term. In such setting the standard benchmark – the *best* time-invariant policy – is uninformative. Instead, we optimize with respect to a more demanding (and also more natural) benchmark – a policy that at each step plays an arm with the currently maximal expected reward.

We consider two versions of the *dynamic MAB problem* described above. In the *state-informed* version an algorithm not only receives a reward of the chosen arm but also finds out the current state of this arm. This is the setting in the restless MAB problem as defined in Whittle [33] and the follow-up literature. In the second, *state-oblivious*, version an algorithm receives its reward and no other information. This formulation generalizes the static MAB problem to stochastically changing expected rewards.

## 1.1 The Dynamic MAB problem

Let  $\{\mathcal{D}(\mu) : \mu \in [0; 1]\}$  be a fixed family of probability distributions on  $[0; 1]$  such that  $\mathcal{D}(\mu)$  has expectation  $\mu$ . Time proceeds in rounds. Each arm  $i$  at each round  $t$  has a *state*  $\mu_i(t) \in [0; 1]$  such that the reward from playing arm  $i$  in round  $t$  is an independent random sample from  $\mathcal{D}(\mu_i(t))$ . At each round  $t$  an algorithm chooses one of the  $k$  alternative strategies ("arms") and receives a reward. In the *state-oblivious* version, the reward is the only information that the algorithm receives in a given round. In the *state-informed* version, the algorithm also finds out the current state of the arm that it has chosen. The distributions  $\mathcal{D}(\cdot)$  are not revealed to the algorithm (and are not essential to the analysis).

<sup>3</sup>As we only sample arms at integer time points, we can equivalently describe the state as a sum of  $t$  i.i.d. normal increments. In fact, we allow the increments to come from a somewhat more general class of distributions.

The state  $\mu_i(\cdot)$  varies in an interval with reflecting boundaries. To clarify the concept of reflecting boundaries, consider an object that starts moving on an interval  $I = [0; 1]$ , reversing direction every time it hits a boundary. If the object starts at 0 and traverses distance  $x \geq 0$ , its position is

$$f_I(x) = \begin{cases} x', & x' \leq 1 \\ 1 - (x' - 1), & x' > 1, \end{cases} \quad (1)$$

where  $x' = x \pmod{2} = x - 2 \lfloor x/2 \rfloor$ . Similarly, we define  $f_I(x)$ ,  $x < 0$  as the position of an object that starts moving from 1 and traverses distance  $|x|$ .

For concreteness we focus here on the case when each arm's state follows a Brownian motion. Similar results hold for related stochastic processes such as discrete random walks (see the Extensions Section).

The state of each arm  $i$  undergoes an independent Brownian motion on an interval with reflecting boundaries. Specifically, we define  $\mu_i(t) = f_I(B_i(t))$  where  $I = [0; 1]$  is the *fundamental interval* and  $B_i$  is an independent Brownian motion with volatility  $\sigma_i$ . Since we only sample  $\mu_i(\cdot)$  at integer times, we can also define it as a Markov chain:

$$\mu_i(t) = f_I(\mu_i(t-1) + X_i(t)), \quad (2)$$

where each  $X_i(t)$  is an i.i.d. sample from  $\mathcal{N}(0, \sigma_i)$ . The stochastic rate of change is thus given by  $\sigma_i$ , which we term the *volatility* of arm  $i$ .

We assume that for each arm  $i$  the initial state  $\mu_i(0)$  is an independent uniformly random sample from  $I$ . This is a reasonable assumption given our goal to study the stationary behavior of the system. Indeed, the uniform distribution on  $I$  is the stationary distribution of the Markov chain 2 to which this Markov chain eventually converges.<sup>4</sup>

In the dynamic MAB problem, we measure the performance of a MAB algorithm with respect to a policy that at every step chooses a strategy with the highest expected reward. This policy changes in time, and thus it is a more demanding benchmark than the *time-invariant regret* that is often used in the MAB literature.

**Definition 1.1.** Consider an instance of the dynamic MAB problem. For a given MAB algorithm  $\mathcal{A}$ , let  $W_{\mathcal{A}}(t)$  be the reward received by algorithm  $\mathcal{A}$  in round  $t$ . Let  $\emptyset$  be an algorithm that in every round chooses a strategy with the highest expected reward. The *dynamic regret* in round  $t$  is

$$R_{\mathcal{A}}(t) = W_{\emptyset}(t) - W_{\mathcal{A}}(t).$$

Define the *steady-state regret* as

$$\bar{R}_{\mathcal{A}} = \limsup_t \sup_{t_0} E \left[ \frac{1}{t} \sum_{s=t_0+1}^{t_0+t} R_{\mathcal{A}}(s) \right]. \quad (3)$$

<sup>4</sup>The convergence follows from the ergodic theorem. It should be noted that the *rate* of convergence for Markov chains with infinite state spaces is a rather delicate matter, e.g. see Rosenthal [28]. In this paper the rate of convergence is non-essential. Moreover, the convergence itself does not appear in the proofs: it is used only as intuition and an (additional) justification for assuming the uniform distribution of the initial state.

Thus, for any fixed  $R > \bar{R}_{\mathcal{A}}$  the expected average dynamic regret of algorithm  $\mathcal{A}$  over any sufficiently large interval is at most  $R$ , and it is the best possible upper bound of this form. Our goal is to bound  $\bar{R}_{\mathcal{A}}$  in terms of the arms' volatility.

We use the following notation throughout the paper. The state of arm  $i$  at time  $t$  is  $\mu_i(t)$ . The maximal state at time  $t$  is  $\mu^*(t) = \max_{i \in [k]} \mu_i(t)$ . An arm  $i$  is *maximal* in round  $t$  if  $\mu_i(t) = \mu^*(t)$ .

## 1.2 Results: the state-informed case

We present an algorithm whose steady-state regret is optimal up to a poly-log factor.

**Theorem 1.2.** Consider the state-informed dynamic MAB problem with  $k$  arms, each with volatility at most  $\sigma$ . Assume that  $k < \sigma^{-\gamma}$  for some  $\gamma < \frac{1}{2}$ . Then there exists a MAB algorithm whose steady-state regret is at most  $\tilde{O}(k\sigma^2)$ .

The algorithm is very intuitive. An arm with the highest last-observed state is called a *leader* and is played often, e.g. at least every other round. Suppose the last time some other arm  $i$  was observed was  $t$  rounds ago. By Azuma inequality the state of this arm changed by at most  $\Delta\mu = \tilde{O}(\sigma\sqrt{t})$  since then, with high probability. If  $\mu_i(t) + \Delta\mu$  is smaller than the state of the leader, then there is no point yet in trying arm  $i$  again. Else, we mark this arm *suspicious* and enqueue it to be played soon.

The main technical contribution here is the analysis, which is quite delicate since we need to deal with the complicated dependencies in the algorithm's behavior induced by the stochastically changing environment. Essentially, we manage to reduce the stochastic aspect of the problem to simple events in the state space. We achieve it as follows. Every time each arm is played, we spread the corresponding dynamic regret evenly over the corresponding idle time. This way we express the cumulative dynamic regret as a sum over the contributions of each arm in each round. We prove a uniform bound on the expectation of each such contribution. To this end, we identify a useful high-probability behavior of the system, derive deterministic guarantees conditional on this behavior (which is the tricky part), and then argue in terms of the corresponding conditional expectations.

Surprisingly, the steady-state regret of our algorithm essentially matches a lower bound based on a very simple idea: if in a given round the states of the best two arms are within  $\frac{\sigma}{4}$  from one another, then in the next round with constant probability either one of them can be  $\frac{\sigma}{4}$  above another, so any algorithm incurs expected dynamic regret  $\Omega(\sigma)$ .<sup>5</sup>

**Theorem 1.3.** Consider the state-informed dynamic MAB problem with  $k$  arms of volatility  $\sigma$ . Then the steady-state regret of any MAB algorithm is at least  $\Omega(k\sigma^2)$ .

## 1.3 Results: the state-oblivious case

Our algorithm for the state-oblivious case builds on an algorithm from [2] for the static MAB problem. That algorithm implicitly uses a simple "padding" function that for a given

<sup>5</sup>The former event happens with probability  $\Omega(k\sigma)$ , so the steady-state regret is  $\Omega(k\sigma^2)$ . This is the entire proof!

arm bounds the drift of an average reward from its (static) expected value. We design a new algorithm  $UCB_f$  which relies on a novel "padding" function  $f$  that accounts for the changing expected rewards. The analysis is quite technical: the specific results from [2] do not directly apply to our setting; instead, we need to "open up the hood" and combine the technique from [2] with some new ideas.

**Theorem 1.4.** *Consider the state-oblivious dynamic MAB problem with  $k$  arms such that each arm  $i$  has volatility at most  $\sigma_i$ . Then there exists a MAB algorithm whose steady-state regret is  $\tilde{O}(k\sigma_{av})$ , where  $\sigma_{av}^2 = \frac{1}{k} \sum_{i=1}^k \sigma_i^2$ .*

Note that (unlike the guarantee in Theorem 1.2), the guarantee here is in terms of an average volatility rather than the maximal one.

#### 1.4 Using off-the-shelf MAB algorithms?

We ask whether similar results can be obtained using off-the-shelf MAB algorithms. Specifically, we investigate the following idea: take an off-the-shelf algorithm, run it and restart it every fixed number of rounds.

For the state-informed version we consider the obvious "greedy" approach: probe each arm, choose the best one, play it for a fixed number  $m$  of rounds, restart. The greedy algorithm is parameterized by the *phase length*  $m$  which can be tuned depending on the number of arms and their volatility. We show that the greedy algorithm is indeed suboptimal as compared to Theorem 1.2: the dependence on volatility (which is smaller than one) is linear rather than quadratic; we provide both upper and lower bounds.

For the state-oblivious version one can leverage on the existing work for the adversarial MAB problem [3]. This work assumes no restrictions on the state evolution, but provides guarantees only with respect to the best time-invariant policy, or a policy that switches arms a bounded number of times. We consider the following algorithm: run a fresh instance of algorithm EXP3 from [3] for a fixed number  $m$  of rounds, then restart. Using the off-the-shelf performance guarantees for EXP3 and fine-tuning  $m$ , one can (only) bound the steady-state regret by  $\tilde{O}((k\sigma_{av})^{2/3})$ , which is inferior to the result in Theorem 1.4. It is an open question whether one can obtain improved guarantees by tailoring the analysis in [3] to our setting.

#### 1.5 Extensions and open questions

We extend our results in several directions. First, we generalize the Markov-chain formulation (2) to allow the random increments  $X_i(t)$  to come from other distributions which has a certain "light-tailed" property, such as the discrete random walk. Second, we consider the setting in which each arm has a distinct fundamental interval. Third, we relax the assumption that the upper bound(s) on volatilities are known to the algorithm.

The main question left open by this paper is to close the gap between the upper and lower bounds for the state-oblivious dynamic MAB problem. The only lower bound we have is Theorem 1.3. We conjecture that one may obtain a better bound based on the relative entropy-based technique from [3]. It is also possible that the algorithmic result can

be improved, possibly via a more refined mechanism for discounting information with time.

Another open question is whether one can obtain the optimal  $\tilde{O}(k\sigma^2)$  steady-state regret for the state-informed version in the case when  $k \geq \sigma^{-1/2}$ . Note that the greedy algorithm mentioned in Section 1.4 achieves steady-state regret  $\tilde{O}(k\sigma)$  which is non-trivial for any  $k \leq \sigma^{-1}$ .

#### 1.6 Organization of the paper

In Sections 2 and 3 we present our main results for the state-informed and the state-oblivious versions, respectively. Section 4 discusses using off-the-shelf MAB algorithms. Section 5 covers the extensions.

### 2 The state-informed dynamic MAB problem

We consider the state-informed dynamic MAB problem where the volatility of each arm is at most  $\sigma$ . Recall that the state of arm  $i$  at time  $t$  is denoted  $\mu_i(t)$ .

For arm  $i$  and time  $t$ , the *last-seen time*  $\tau_i(t)$  is the last time this arm has been played strictly before time  $t$ ; the *last-seen state*  $\nu_i(t) = \mu_i(\tau_i(t))$  is the corresponding state.

**Definition 2.1.** The *leader* in round  $t$  is the arm with a larger last-seen state, among the arms played in rounds  $t-1$  and  $t-2$ ; break ties in favor of the arm played in round  $t-1$ .

In our algorithm, the leader is our running estimate for an arm with the maximal state. We alternate rounds in which we always *exploit* – play the leader, with rounds in which we may explore other options. Since we define the leader in terms of the last two rounds only, our knowledge of its state is essentially up-to-date.

Let  $\nu^*(t)$  be the last-seen state of the leader in round  $t$ . Let  $c_{\text{susp}} = \Theta(\log \frac{1}{\sigma})^{1/2}$  be the factor to be defined later.

**Definition 2.2.** An arm  $i$  is called *suspicious* at time  $t$  if

$$\nu^*(t) - \nu_i(t) \leq c_{\text{susp}} \sigma \sqrt{t - \tau_i(t)}. \quad (4)$$

If an arm  $i$  is not suspicious at time  $t$ , then with high probability its current reward is less than  $\nu^*(t)$ . If no arm is suspicious then, intuitively, the best bet is to play the leader. Roughly, our algorithm behaves as follows: if the time is even it plays the current leader, and if the time is odd it plays a suspicious arm if one exists, and the leader otherwise. To complete the description of the algorithm, we need to specify what it does when there are multiple suspicious arms. In particular, we need to guarantee that after an arm becomes suspicious, it is played eventually (and preferably soon).

**Definition 2.3.** An arm  $i$  is *active* at time  $t$  if it is not the leader and it has been suspicious at some time  $t' > \tau_i(t)$ . The *activation time*  $\tau_i^{\text{act}}(t)$  is the earliest such time  $t'$ .

An arm becomes active when it becomes suspicious. It stays active until it is played. The idea is to play an active arm with the earliest activation time.

**Algorithm 2.4.** *For bootstrapping, each arm is played once. At any later time  $t$  do the following. If  $t$  is even, play the current leader. If  $t$  is odd play an active arm (with the earliest activation time) if one exists, else play the leader.*

We will use a slightly more refined algorithm which allows for a more efficient analysis. Essentially, we give priority to arms whose state is close to the leader's.

**Definition 2.5.** Arm  $i$  is *high-priority* at time  $t$  if it is active at this time and moreover  $\tau_i^{\text{act}}(t) - \tau_i(t) \leq 4k$ .

**Algorithm 2.6.** For bootstrapping, each arm is played once. At any later time  $t$  do the following. If  $t$  is even, play the current leader. If  $t$  is odd play an active arm if one exists, else play the leader. If there are multiple active arms:

- if  $t \equiv 1 \pmod{4}$  then play an active arm with the earliest activation time; break ties arbitrarily
- if  $t \equiv 3 \pmod{4}$  then play a high-priority arm with the earliest activation time if one exists; else, play any active arm; break ties arbitrarily.

The analysis of these two algorithms are very similar, except that Algorithm 2.4 has inefficiencies which lead to an extra  $k^2$  factor in its regret. We focus on Algorithm 2.6.

**Theorem 2.7.** Consider the state-informed dynamic MAB problem with  $k$  arms, each with volatility at most  $\sigma$ . Assume that  $k < \sigma^{-\gamma}$  for some  $\gamma < \frac{1}{2}$ . Then Algorithm 2.6 achieves steady-state regret  $O(k \sigma^2 \log^2 1/\sigma)$ .

In the rest of this section we prove Theorem 2.7.

Let  $\bar{R}_A(t)$  be the average dynamic regret up to time  $t$ . Then, letting  $T_i(t)$  be the set of times arm  $i$  was played before and including time  $t$ , we have

$$E[\bar{R}_A(t)] = \frac{1}{t} \sum_{i \in [k]} \sum_{t' \in T_i(t)} E[\mu^*(t') - \mu_i(t')]. \quad (5)$$

Let us spread contributions of individual arms evenly over the corresponding idle time. Specifically, let us define

$$\begin{aligned} \Delta\mu_i(t) &= \mu^*(t) - \mu_i(t), \\ \Delta\tau_i(t) &= \tau_i^+(t) - \tau_i(t), \end{aligned}$$

where  $\tau_i^+(t)$  is the next time arm  $i$  is played after time  $\tau_i(t)$ .<sup>6</sup> Then we can re-write (5) as follows:

$$E[\bar{R}_A(t)] = \frac{1}{t} \sum_{i \in [k]} \sum_{t' \in [t]} E\left[\frac{\Delta\mu_i(\tau_i(t'))}{\Delta\tau_i(t')}\right]. \quad (6)$$

We define the *contribution* of arm  $i$  in round  $t$  as

$$C_i(t) = \frac{\Delta\mu_i(\tau_i(t))}{\Delta\tau_i(t)}.$$

A crucial idea is that we upper-bound  $E[C_i(t)]$  for each round  $t$  separately. Namely, we will prove that

$$E[C_i(t)] < O(\sigma^2 \log^2 \frac{1}{\sigma}). \quad (7)$$

We identify the high-probability behavior of the processes  $\{\mu_i(t)\}_{i \in [k]}$ . Specifically, we consider the  $\tilde{O}(\sqrt{t})$  bound on deviations, and an  $O(1)$  bound on the number of near-optimal arms. A large portion of our analysis is deterministic conditional on such behavior.

<sup>6</sup>In other words,  $\tau_i(t)$  and  $\tau_i^+(t)$  are the two consecutive times arm  $i$  is played such that  $\tau_i(t) < t \leq \tau_i^+(t)$ .

**Definition 2.8.** A real-valued function  $f$  is *well-behaved* on an interval  $[t_1; t_2]$  if for any  $t, t + \Delta t \in [t_1; t_2]$  we have

$$|f(t + \Delta t) - f(t)| < c_{\text{well}} \sigma \sqrt{\Delta t}. \quad (8)$$

where  $c_{\text{well}} = \Theta(\log \frac{1}{\sigma})^{1/2}$  will be chosen later.

**Definition 2.9.** An instance of the dynamic MAB problem is *well-behaved* on a time interval  $I$  if

- functions  $\mu_1(t), \dots, \mu_k(t)$  are well-behaved on  $I$ ;
- at each time  $t \in I$  there are at most  $c_{\text{near}} = O(1)$  arms  $i$  such that  $\Delta\mu_i(t) < (8k + 15\sqrt{k}) c_{\text{well}} \sigma$ .<sup>7</sup>

A problem instance is *well-behaved near time  $t$*  if it is well-behaved on the time interval  $[t - 3\sigma^{-2}; t + \sigma^{-2}]$ .

**Choosing the parameters.** The factors  $c_{\text{well}}$  and  $c_{\text{near}}$  are chosen so that for any fixed  $t$  a problem instance is well-behaved near time  $t$  with probability at least  $1 - \sigma^{-3}$ . In Definition 2.1, define  $c_{\text{susp}} = 5 c_{\text{well}}$ .

Our conditionally deterministic guarantees (conditional on the problem instance being well-behaved) are expressed by the following lemma.

**Lemma 2.10** (The Deterministic Lemma). *Suppose a problem instance is well-behaved near time  $t$ . Fix arm  $i$  and let  $\delta = \Delta\mu_i(t)$ . Then:*

- If  $\delta = 0$  and  $C_i(t) > 0$  then

$$C_i(t) \leq O(\sigma \log \frac{1}{\sigma}) / \sqrt{t - \tau_i(t)}, \quad (9)$$

and moreover for some arm  $j \neq i$  we have

$$\Delta\mu_j(t) < O(\sigma \log \frac{1}{\sigma}) \sqrt{t - \tau_i(t)}. \quad (10)$$

- If  $\delta > 0$  then  $C_i(t) \leq O(\sigma^2/\delta) \log^2 \frac{1}{\sigma}$ .

Let us use Lemma 2.10 to derive the main result.

**Proof of Theorem 2.7:** It suffices to prove (7). Let  $\mathcal{E}(t)$  denote the event that the problem instance is well-behaved near time  $t$ . By Lemma 2.10(a), letting  $x = \sqrt{t - \tau_i(t)}$  and suppressing the  $\log \frac{1}{\sigma}$  factors under the  $\tilde{O}(\cdot)$  notation,

$$\begin{aligned} E[C_i(t) \mid \Delta\mu_i(t) = 0, \mathcal{E}(t)] \\ \leq \tilde{O}(\sigma/x) \Pr[\exists j \neq i : \Delta\mu_j(t) < \tilde{O}(\sigma x)] \\ \leq \tilde{O}(\sigma^2). \end{aligned} \quad (11)$$

By Lemma 2.10(b) for any  $\delta > 0$  we have

$$E[C_i(t) \mid \Delta\mu_i(t) \geq \delta, \mathcal{E}(t)] \leq \tilde{O}(\sigma^2/\delta) \quad (12)$$

$$\Pr[\Delta\mu_i(t) \leq \delta \mid \Delta\mu_i(t) > 0, \mathcal{E}(t)] \leq \tilde{O}(\delta). \quad (13)$$

Now (7) follows from (11-13) via a simple computation.  $\square$

In the rest of this section we prove Lemma 2.10.

<sup>7</sup>This expression is tailored to (16) in the subsequent analysis. The event in question happens with probability at least  $1 - O(c_{\text{well}} k^2 \sigma)^{c_{\text{near}}}$ . Recall that we assume  $k < \sigma^{-\gamma}$  for some  $\gamma < \frac{1}{2}$ . Thus, a (large enough) constant  $c_{\text{near}}$  suffices to guarantee a sufficiently low failure probability.

## 2.1 Deterministic bounds for the leader

We will argue deterministically assuming that the problem instance is well-behaved. We split our argument into a chain of claims and lemmas. The proofs are quite detailed; one can skip them for the first reading. For shorthand, let  $\mathcal{E}[t_1; t_2]$  denote the event that the (fixed) problem instance is well-behaved on the time interval  $[t_1; t_2]$ .

First, we argue that the leader's last-seen state,  $\nu^*(\cdot)$ , does not decrease too much in one round.

**Claim 2.11.** *If  $\mathcal{E}[t-2; t]$  then*

$$\nu^*(t+1) \geq \nu^*(t) - 2c_{\text{well}}\sigma.$$

*Proof.* Assume that  $t$  is even (if  $t$  is odd the proof proceeds similarly). Recall that the leader in round  $t$  is some arm  $i$  played in one of the previous two rounds. It follows that

$$\nu^*(t) = \nu_i(t) \leq \mu_i(t) + 2c_{\text{well}}\sigma.$$

Moreover, the leader (i.e. arm  $i$ ) is played in round  $t$  and therefore  $\nu^*(t+1) \geq \mu_i(t)$ , claim proved.  $\square$

Second, each arm becomes active eventually.

**Claim 2.12.** *Any arm  $i$  becomes active at most  $\sigma^{-2}$  rounds after it is played:  $\tau_i^{\text{act}}(t) - \tau_i(t) \leq \sigma^{-2}$  for any time  $t$ .*

*Proof.* If  $t - \tau_i(t) \geq \sigma^{-2}$  then (4) is trivially true.  $\square$

Third, we show that a currently maximal arm has been activated within  $4k$  rounds from its last-seen time, and therefore it has been played in the previous  $8k$  rounds. The proof of this lemma is one of the crucial arguments in our analysis.

**Lemma 2.13.** *Suppose  $\mathcal{E}[t - \sigma^{-2}; t]$  and arm  $i$  is maximal at time  $t$ . Then*

$$\tau_i^{\text{act}}(t) - \tau_i(t) \leq t - \tau_i^{\text{act}}(t) \leq 4k.$$

*Proof.* Note that  $t - \tau_i^{\text{act}}(t) \leq 4k$ , since otherwise after becoming active at time  $\tau_i^{\text{act}}(t)$  arm  $i$  would have been played strictly before round  $t$ , contradiction.

Let  $\tau = \tau_i(t)$ . For the sake of contradiction assume that

$$\tau_i^{\text{act}}(t) - \tau > t - \tau_i^{\text{act}}(t). \quad (14)$$

Since arm  $i$  is not suspicious at time  $t' = \tau_i^{\text{act}}(t) - 1$ , by Definition 2.2 we have

$$\nu^*(t') - \nu_i(t') \geq c_{\text{susp}}\sigma\sqrt{t' - \tau}. \quad (15)$$

By Claim 2.12 the problem instance is well-behaved on  $[\tau; t]$ . It follows that

$$\begin{aligned} \nu_i(t') &= \mu_i(\tau) \geq \mu_i(t) - c_{\text{well}}\sigma\sqrt{t - \tau} \\ \nu^*(t') &= \mu_j(t'') \leq \mu_j(t) + c_{\text{well}}\sigma\sqrt{t - t''}, \end{aligned}$$

where arm  $j$  is the leader in round  $t'$ , and  $t''$  is one of the two rounds preceding  $t'$ . Plugging this into (15) and using (14), we see that  $\mu_j(t) > \mu_i(t)$ , contradiction.  $\square$

Fourth, we show that the leader's last-seen state is not much worse than the maximal state.

**Claim 2.14.** *If  $\mathcal{E}[t - \sigma^{-2}; t]$  then*

$$\mu^*(t) - \nu^*(t) \leq (8k + \sqrt{8k})c_{\text{well}}\sigma.$$

*Proof.* Let  $\mu^*(t) = \mu_i(t)$  for some arm  $i$ , and let  $\tau = \tau_i(t)$  be the last time this arm was played. By Lemma 2.13 we have  $t - \tau \leq 8k$ . Therefore

$$\nu^*(\tau + 1) \geq \mu_i(\tau) \geq \mu_i(t) - c_{\text{well}}\sigma\sqrt{8k},$$

and the claim follows by Claim 2.11.  $\square$

Fifth, we show that high-priority arms are played very soon after they become active.

**Claim 2.15.** *Suppose arm  $i$  is a high-priority active arm at time  $t$ . Assume  $\mathcal{E}[t - \sigma^{-2}; t]$ . Then  $t - \tau_i^{\text{act}}(t) \leq 4c_{\text{near}}$ .*

*Proof.* Fix time  $t$  and let  $t' = \tau_i^{\text{act}}(t)$  be the activation time of arm  $i$ . Then by Definition 2.4 and Definition 2.3

$$\nu^*(t') - \nu_i(t') \leq c_{\text{susp}}\sigma\sqrt{t - t'} \leq c_{\text{susp}}\sigma\sqrt{4k}.$$

Using Claim 2.14 to relate  $\nu^*(t')$  and  $\mu^*(t')$ , and using the fact that  $\nu_i(t') = \mu_i(\tau)$  and that  $\mu_i(\cdot)$  is well-behaved, we obtain

$$\Delta\mu_i(t') \leq (8k + 15\sqrt{k})c_{\text{well}}\sigma. \quad (16)$$

Lemma follows by Definition 2.9(ii) which is, in fact, tailored to (16).  $\square$

Now we have the tools needed to prove a stronger version of Claim 2.14:  $\mu^*(t) - \nu^*(t) \leq \tilde{O}(\sigma)$ .

**Lemma 2.16.** *If the problem instance is well-behaved on  $[t - \sigma^{-2}; t]$  then  $\mu^*(t) - \nu^*(t) \leq O(c_{\text{well}}\sigma)$ .*

*Proof.* Let  $i$  be an active arm at time  $t$ . By Lemma 2.13  $\tau_i^{\text{act}}(t) - \tau_i(t) \leq 4k$ , so at time  $\tau_i^{\text{act}}(t)$  arm  $i$  is a high-priority active arm. By Claim 2.15  $t - \tau_i^{\text{act}}(t) \leq 4c_{\text{near}} = O(1)$ . By Lemma 2.13 it follows that  $t - \tau_i(t) \leq O(1)$ .

Now  $\nu^*(\tau + 1) \geq \mu_i(\tau)$  by definition of the leader;  $\nu^*(t) \geq \nu^*(\tau + 1) - O(c_{\text{well}}\sigma)$  by Claim 2.11; and also  $\mu^*(t) \leq \mu^*(\tau) + O(c_{\text{well}}\sigma)$  since the problem instance is well-behaved. Putting it together, we obtain the lemma.  $\square$

## 2.2 Proof of The Deterministic Lemma

Let  $\tau = \tau_i(t)$  and recall that we denote  $\delta = \Delta\mu_i(t)$ .

By Lemma 2.13 we have  $t - \tau \leq 8k$ . Since the problem instance is well-behaved on  $[t - 8k; t]$ , it follows that  $\mu^*(\cdot)$  is well-behaved, too, and therefore

$$|\Delta\mu_i(t) - \Delta\mu_i(\tau)| \leq 2c_{\text{well}}\sigma\sqrt{t - \tau}, \quad (17)$$

which immediately implies (9). To obtain (10) note that (17) in fact applies to any arm  $j$ , in particular to an arm  $j$  that is maximal at time  $\tau$ .

To prove Lemma 2.10(b), it suffices to prove the following two inequalities:

$$\Delta\tau_i(t) \geq \Omega(\delta/\sigma)^2 / \log \frac{1}{\sigma}, \quad (18)$$

$$\Delta\mu_i(\tau) \leq O(\delta + \sigma \log \frac{1}{\sigma}). \quad (19)$$

**Proof of (18):** We consider two cases.

First, if we have  $\Delta\mu_i(\tau) < \delta/2$  then by (17) we obtain

$$2c_{\text{well}}\sigma\sqrt{t - \tau} \geq |\Delta\mu_i(t) - \Delta\mu_i(\tau)| \geq \delta/2,$$

and (18) follows since  $\Delta\tau_i(t) \geq t - \tau$ .

Second, assume  $\Delta\mu_i(\tau) \geq \delta/2$ . Then by Lemma 2.16 for any time  $t' \in (\tau; t + \sigma^{-2})$  we have

$$\begin{aligned} \nu^*(t') - \mu_i(\tau) &\geq \mu^*(t') - O(c_{\text{well}}\sigma) + \Delta\mu_i(\tau) - \mu^*(\tau) \\ &\geq \delta/2 - c_{\text{well}}\sigma\sqrt{t' - \tau} + O(1). \end{aligned}$$

This is at least  $\geq c_{\text{susp}}\sigma\sqrt{t' - \tau}$  as long as it is the case that  $t' - \tau \leq (12c_{\text{well}}\sigma/\delta)^{-2}$ . So for any such  $t'$  arm  $i$  is not suspicious, proving (18).  $\square$

**Proof of (19):** First, note that if  $\tau_i^{\text{act}}(t) - \tau_i(t) \leq 4k$  then by Definition 2.5 arm  $i$  is a high-priority active arm at time  $\tau_i^{\text{act}}(t)$ , so by Claim 2.15 we have  $t - \tau_i^{\text{act}}(t) \leq O(1)$  and so  $t - \tau_i(t) \leq O(1)$  by Lemma 2.13. It follows by (17) that

$$\Delta\mu_i(\tau) \leq \Delta\mu_i(t) + O(\sigma),$$

and we are done. In what follows we will assume that

$$\tau_i^{\text{act}}(t) - \tau_i(t) > 4k. \quad (20)$$

Note that for any time  $t'$  we have

$$\begin{aligned} \nu^*(t') &\leq \max(\mu^*(t' - 1), \mu^*(t' - 2)) \\ &\leq \mu^*(t') + 2c_{\text{well}}\sigma. \end{aligned}$$

Let  $t' = \tau_i^{\text{act}}(t) - 1$  be the round immediately preceding the activation time. Since arm  $i$  is not suspicious at time  $t'$ ,

$$\begin{aligned} c_{\text{susp}}\sigma\sqrt{t' - \tau} &\leq \nu^*(t') - \mu_i(\tau) \\ &\leq \mu^*(t') - \mu_i(\tau) + 2c_{\text{well}}\sigma \\ &\leq \Delta\mu_i(t') + c_{\text{well}}\sigma(2 + \sqrt{t' - \tau}). \end{aligned}$$

Since  $c_{\text{susp}} = 5c_{\text{well}}$ , it follows that

$$\Delta\mu_i(t') + 2c_{\text{well}}\sigma \geq 4c_{\text{well}}\sigma\sqrt{t' - \tau}. \quad (21)$$

Combining (17) and (21), we obtain

$$\begin{aligned} \Delta\mu_i(\tau) &\leq \Delta\mu_i(t') + 2c_{\text{well}}\sigma\sqrt{t' - \tau} \\ &\leq \frac{3}{2}\Delta\mu_i(t') + 2c_{\text{well}}\sigma. \end{aligned}$$

Finally, by (17), (20) and (21) we obtain

$$\begin{aligned} \Delta\mu_i(t') &\leq \Delta\mu_i(t) + 2c_{\text{well}}\sigma\sqrt{t - t'} \\ &\leq \Delta\mu_i(t) + \frac{1}{2}\Delta\mu_i(t') + 2c_{\text{well}}\sigma \\ \Delta\mu_i(t') &\leq 2\Delta\mu_i(t) + 4c_{\text{well}}\sigma \\ \Delta\mu_i(\tau') &\leq 3\Delta\mu_i(t) + 6c_{\text{well}}\sigma. \quad \square \end{aligned}$$

### 3 The state-oblivious dynamic MAB problem

We consider the state-oblivious dynamic MAB problem with  $k$  arms where the volatility of each arm  $i$  is at most  $\sigma_i$ .

**Definition 3.1.** For each arm  $i$ ,  $N_i(t)$  is the number of times it has been played in the first  $t - 1$  rounds, and  $\bar{W}_i(t)$  is the corresponding average reward. Let  $\bar{W}_i(0) = 0$  if  $N_i(t) = 0$ . For shorthand, let  $\mu_i = \mu_i(0)$  be the initial state.

**Definition 3.2.** Consider an instance of the state-oblivious dynamic MAB problem. A function  $f_i : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}_+$  is a *padding* for arm  $i$  if the following two properties hold:

- $f_i(t, t_i)$  is increasing in  $t$  and decreasing in  $t_i$ ,

- for any time  $t$ , letting  $t_i = N_i(t)$  we have

$$\Pr [ |\bar{W}_i(t) - \mu_i(0)| > f_i(t, t_i) ] < O(t^{-4}). \quad (22)$$

The family  $\{f_i\}_{i \in [k]}$  is a *padding* for the problem instance.

We build on an algorithm UCB1 from [2] for the static MAB problem. We define a generalization of UCB1, which we call  $\text{UCB}_f$ , which is parameterized by a padding  $f = \{f_i\}_{i \in [k]}$ .

**Algorithm 3.3** ( $\text{UCB}_f$ ). *In each round  $t$  play any arm*

$$i \in \operatorname{argmax}_{i \in [k]} [\bar{W}_i(t) + f_i(t, N_i(t))].$$

The original UCB1 algorithm is defined for a specific padding  $f$ , and in fact does not explicitly uses the notion of a padding. We introduce this notion here in order to extend the ideas from [2] to our setting.

We incorporate the analysis from [2] via the following lemma which, essentially, bounds the number of times a sub-optimal arm is played by the algorithm.

**Lemma 3.4** (Auer et al. [2]). *Consider an instance of the state-oblivious MAB problem with a padding  $f = \{f_i\}_{i \in [k]}$ . Consider the behavior of algorithm  $\text{UCB}_f$  in the first  $t$  rounds. Then for each arm  $i$  and any  $t_i < t$  we have*

$$f_i(t, t_i) \leq \frac{1}{2}\Delta\mu_i(t) \Rightarrow E[N_i(t)] \leq t_i + O(1). \quad (23)$$

This lemma is implicit in Auer et al. [2], where it is the crux of the main proof. That proof considers the static MAB problem and (implicitly) a specific padding  $f$ .

We will use  $\text{UCB}_f$  where  $f = \{f_i\}_{i \in [k]}$  is defined by

$$f_i(t, t_i) = \sqrt{2\ln(t)/t_i} + \sigma_i\sqrt{8t\log t}. \quad (24)$$

Define the *average dynamic regret* of an algorithm  $\mathcal{A}$   $\bar{R}_{\mathcal{A}}(t) = \frac{1}{t} \sum_{s \in [t]} R_{\mathcal{A}}(s)$ . We prove the following guarantee for algorithm  $\text{UCB}_f$ :

**Theorem 3.5.** *Consider the state-oblivious dynamic MAB problem with  $k$  arms. Suppose the volatility of each arm  $i$  is at most  $\sigma_i$ . Then there exists time  $t_0$  such that*

$$E[\bar{R}_{\text{UCB}_f}(t_0)] \leq O(k\sigma_{\text{av}})\log^{3/2}(\sigma_{\text{av}}^{-1}), \quad (25)$$

where  $\sigma_{\text{av}}^2 = \frac{1}{k} \sum_{i=1}^k \sigma_i^2$ .

To obtain Theorem 1.4 from Theorem 3.5, we start a fresh instance of algorithm  $\text{UCB}_f$  after every  $t_0$  steps. We take advantage of the facts that (i) the "restarting times" are deterministic and, in particular, independent of the past history, and (ii) in any fixed round each  $\mu_i(t)$  is distributed independently and uniformly on  $[0; 1]$ .

In the rest of this section we prove Theorem 3.5. We start with a very useful fact about the state evolution  $\mu_i(t)$ . In general, if  $\mu_i(0) > \frac{1}{2}$  then due to the influence of the upper boundary the expected state  $E[\mu_i(\cdot)]$  drifts down from its initial value. The following claim upper-bounds such drift.

Let us use a shorthand for the second summand in (24):

$$\delta_i(t) = \sigma_i\sqrt{8t\log t}.$$

**Claim 3.6.** Fix arm  $i$  and integer times  $t \leq t_*$ . Then

$$\Pr[|\mu_i(t) - \mu_i| > \delta_i] < t_*^{-3} \quad (26)$$

where  $\mu_i = \mu_i(0)$  and  $\delta_i = \delta_i(t_*)$ , and therefore

$$E[\mu_i(t) | \mu_i] \geq \min(\mu_i, 1 - \delta_i) - t_*^{-2}. \quad (27)$$

*Proof.* Recall that the state  $\mu_i(t)$  is defined as  $f_I(B_i(t))$  where  $B_i$  is a Brownian motion with volatility  $\sigma_i$ , and  $f_I$  is the "projection" (1) into the interval  $I = [0; 1]$  with reflective boundaries. Note that  $\mu_i = \mathcal{B}_i(0)$ .

It follows that  $|\mu_i(t) - \mu_i| > \delta_i$  only if  $|\mathcal{B}_i(t) - \mu_i| > \delta_i$ . We know that for any  $c > 1$  we have

$$\Pr[|\mathcal{B}_i(t) - \mu_i| > c \sigma_i \sqrt{t}] < 2e^{-c^2/2}.$$

We obtain (26) setting  $c = \sqrt{6 \log t_*}$ .

Now let us prove (27). Define

$$f(\mu) = E[\mu_i(t) | \mu_i].$$

Note that if  $\mu < \frac{1}{2}$  then  $f(\mu) > \mu$ . Also, note that  $f(\mu)$  is increasing and  $f(\frac{1}{2}) = \frac{1}{2}$  by symmetry. Therefore, it suffices to prove (27) under the assumption that  $\frac{1}{2} < \mu_i \leq 1 - \delta_i$ .

Consider  $T = \min(t, T_B)$ , where

$$T_B = \min\{s \in \mathbb{N} : B_i(s) \notin (0; 1)\}.$$

Then  $Z_s = \mu_i(\min(s, T))$  is a martingale such that  $Z_0 = \mu$  and  $T$  is a bounded stopping time. By the Optional Stopping Theorem it follows that  $E[Z_T] = \mu$ . By (26) we have  $T_B \geq t$  with probability at least  $1 - t_*^{-2}$ , in which case  $T = t$  and  $\mu = Z_T = \mu_i(t)$ . Thus (27) follows.  $\square$

Using Claim 3.6, let us argue that (24) is indeed a padding. Essentially, the first summand in (24) is tuned for an application of Chernoff-Hoeffding Bounds, whereas the second one corrects for the drift.

**Lemma 3.7.** The family  $f$  defined by (24) is a padding.

*Proof.* We need to prove (22). Fix arm  $i$  and time  $t$ . Let  $\{t_j\}_{j=1}^\infty$  be the enumeration of all times when arm  $i$  is played. Let  $X_j = \mu_i(t_j)$  be the state of arm  $i$  in round  $t$ . Let  $\hat{X}_j$  be the actual reward collected by the algorithm from arm  $i$  in round  $t_j$ . Let us define the sums  $S = \sum_{j \in [n]} X_j$  and  $S^* = \sum_{j \in [n]} \hat{X}_j$ , where  $n = N_i(t)$  is the number of times arm  $i$  is played before time  $t$ . Let  $\mu = \mu_i(0)$  and  $\delta = \delta_i(t)$ .

We can rewrite (22) as follows:

$$\Pr[|S^* - \mu n| > \sqrt{2n \ln t} + \delta n] < O(t^{-4}). \quad (28)$$

Let  $F$  be the failure event when  $|\mu_i(s) - \mu| > \delta$  for some  $s \in [t]$ . Recall that by Claim 3.6 the probability of  $F$  is at most  $t^{-4}$ . In the probability space induced by conditioning on  $\hat{X}_1, \dots, \hat{X}_{j-1}$  and the event  $\bar{F}$ , we have

$$\begin{aligned} E[\hat{X}_j] &= E[E[\hat{X}_j | t_j, X_j]] = E[E[X_j | t_j]] \\ &= E[X_j] \in [\mu - \delta, \mu + \delta]. \end{aligned}$$

Going back to the original probability space,

$$E[\hat{X}_j | \hat{X}_1 \dots \hat{X}_{j-1}, \bar{F}] \in [\mu - \delta, \mu + \delta]. \quad (29)$$

The Chernoff-Hoeffding bounds (applied to the probability space induced by conditioning on  $\bar{F}$ ) say precisely that the condition (29) implies the following tail inequality:

$$\Pr[|\hat{S} - \mu n| > \delta m + a | \bar{F}] \leq 2e^{-2a^2/m}$$

for any  $a \leq 0$ . We obtain (28) by taking  $a = \sqrt{2m \ln T}$ .  $\square$

To argue about algorithm  $\text{UCB}_f$ , we will use the following notation:

**Definition 3.8.** We will use the following notation:

$$\begin{cases} \rho_i(t) &= \min(\mu_i, 1 - \delta_i(t)), & \mu_i &= \mu_i(0), \\ \Delta_i &= \mu^* - \mu_i, & \mu^* &= \mu^*(0) \\ S(t) &= \{\text{arms } i : \Delta_i \geq 4\delta_i(t)\}. \end{cases}$$

**Lemma 3.9.** Consider any algorithm for the state-oblivious dynamic MAB problem. Then for each arm  $i$  and time  $t \geq k$

$$E[N_i(t) \bar{W}_i(t) | \mu_i] \geq \rho_i(t) E[N_i(t)] - t^{-2}. \quad (30)$$

The left-hand side of (30) is the total winnings collected by arm  $i$  up to time  $t$ . If the bandit algorithm always plays arm  $i$ , then  $N_i(t) = t$  and the left-hand side of (30) is simply equal to  $\sum_s E[\mu_i(s)]$ , so the lemma follows from Claim 3.6. In this sense, Lemma 3.9 is an extension of Claim 3.6. The proof of (30) is a rather intricate exercise in conditional expectations and martingales. We defer it to Section 3.1.

We combine Lemma 3.9 and Lemma 3.4 to derive a conditional bound on  $\bar{R}_{\text{UCB}_f}(t)$ :

**Corollary 3.10.** For any time  $t$  we have

$$\begin{aligned} E[\bar{R}_{\text{UCB}_f}(t) | \mu_1, \dots, \mu_k] &\leq \frac{k}{t^2} + O(1) \left[ \sum_{i \notin S(t)} \mu^* - \rho_i(t) \right] \\ &\quad + O\left(\frac{1}{t} \log t\right) \left[ \sum_{i \in S(t)} \frac{1}{\Delta_i} \right]. \quad (31) \end{aligned}$$

*Proof.* Fix time  $t$  and let  $\bar{W}_i = \bar{W}_i(t)$ ,  $\rho_i = \rho_i(t)$  and  $N_i = N_i(t)$ . Let  $R(t)$  be the left-hand side of (31). Using (30),

$$\begin{aligned} t R(t) &= \sum_{i \in [k]} E[(\mu^* - \bar{W}_i) N_i] \\ &\leq \sum_{i \in [k]} E[N_i] (\mu^* - \rho_i) + t^{-2}. \end{aligned}$$

For each  $i \in S(t)$  we have  $\mu^* - \rho_i \leq 2\Delta_i$  and by Lemma 3.4

$$E[N_i(t)] \leq 32 \ln(m) / \Delta_i^2 + O(1). \quad \square$$

We obtain Theorem 3.5 by integrating both sides of (31) with respect to  $\mu_1 \dots \mu_k$ .

**Proof of Theorem 3.5:** Fix time  $t$  and let  $\delta_i = \delta_i(t)$  and  $\rho_i = \rho_i(t)$ . Note that (31) is, essentially, the sum over all arms. We partition the arms into three sets and bound the three corresponding sums separately.

Note that the following holds for any fixed  $\gamma > 0$ : given  $\mu^*$  and the event  $\{\Delta_i > \gamma\}$ , the random variable  $\mu_i$  is distributed uniformly on the interval  $[0; \mu^* - \gamma]$ . We will use this property in the forthcoming integrations.

First, we consider the set  $S = S(t)$ . Conditional on  $\mu^*$ ,

$$\begin{aligned} E \left[ \sum_{i \in S} \Delta_i^{-1} \right] &= \sum_{i \in [k]} E [\Delta_i^{-1} | \Delta_i > 4\delta_i] \Pr[\Delta_i > 4\delta_i] \\ &\leq \sum_{i \in [k]} \ln \sigma_i^{-1} \leq O(k \ln \sigma_{\text{av}}^{-1}). \end{aligned} \quad (32)$$

Second, let us consider the set  $S^+$  of all arms  $i$  such that  $0 < \Delta_i < 4\delta_i$ . Conditional on  $\mu^*$ , we obtain

$$\begin{aligned} E \left[ \sum_{i \in S^+} \mu^* - \rho_i \right] &\leq \sum_{i \in [k]} O(\delta_i) \Pr[\Delta_i < 4\delta_i | \Delta_i > 0] \\ &\leq \sum_{i \in [k]} O(\delta_i) \min(1, \delta_i / \mu^*). \end{aligned}$$

Integrating over  $\mu^*$ , we obtain

$$\begin{aligned} E \left[ \sum_{i \in S^+} \mu^* - \rho_i \right] &\leq \sum_{i \in [k]} O(\delta_i^2) \\ &\leq O(k \sigma_{\text{av}}^2 t \log t). \end{aligned} \quad (33)$$

Third, we consider the set  $S^*$  of all maximal arms, i.e. the set of all arms  $i$  such that  $\Delta_i = 0$ . We show the main steps of the argument, omitting the details of some straightforward integrations:

$$\begin{aligned} Z_i &:= \mathbb{I}_{\{\Delta_i=0\}} (\mu^* - \rho_i) \\ E[Z_i] &= E[E[Z_i | \mu^*]] = \frac{1}{k} E[\mu^* - \rho_i] = O(\delta_i^2) \\ E \left[ \sum_{i \in S^*} \mu^* - \rho_i \right] &= \sum_{i \in [k]} E[Z_i] \leq O(k \sigma_{\text{av}}^2) (t \log t). \end{aligned} \quad (34)$$

Finally, using (32-34), we take expectations in (31):

$$E[\bar{R}_{\text{UCB}_f}(t)] = O\left(\frac{k}{t} \log t\right) ((\sigma_{\text{av}} t)^2 + \log \sigma_{\text{av}}^{-1}).$$

The theorem follows if we take  $t_0 = \sigma_{\text{av}} \sqrt{\log \sigma_{\text{av}}^{-1}}$ .  $\square$

### 3.1 Proof of Lemma 3.9: conditional expectations

Fix arm  $i$  and time  $t$ . Let us introduce a more concise notation which gets rid of the subscript  $i$ . Let  $\mu = \mu_i(0)$  and  $\delta = \delta_i(t)$ , and denote  $N = N_i(t)$ . For every time  $s$ , let  $Y_s = \mu_i(s)$ , and let  $X_s$  be the winnings from arm  $i$  at time  $s$  if it is played by the algorithm.<sup>8</sup> Let  $\zeta_s$  be equal to 1 if arm  $i$  is played at time  $s$ , and 0 otherwise.

To prove (30), we will show that

$$\begin{aligned} E \left[ \sum_{s \in [t]} \zeta_s X_s \right] &= E \left[ \sum_{s \in [t]} \zeta_s Y_s \right] \\ &\geq \min(\mu, 1 - \delta) E[N] + t^{-2}. \end{aligned} \quad (35)$$

Note that  $\zeta_s$  and  $X_s$  are conditionally independent given  $Y_s$ . It follows that

$$\begin{aligned} E[\zeta_s X_s | Y_s] &= E[\zeta_s | Y_s] E[X_s | Y_s] = E[\zeta_s | Y_s] Y_s \\ &= E[\zeta_s Y_s | Y_s]. \end{aligned}$$

<sup>8</sup>That is,  $X_s$  is an independent random sample from distribution  $\mathcal{D}(Y_s)$ , as defined in Section 1.1.

Taking expectations on both sides, we obtain

$$E[\zeta_s X_s] = E[\zeta_s Y_s],$$

which proves (35).

Going from (35) to (36) is somewhat more complicated. In what follows we denote  $S = \sum_{t \in [m]} \zeta_s Y_s$ .

**Claim 3.11.** *If  $\mu \leq 1 - \delta$  then  $E[S] \geq \mu E[N] - t^{-2}$ .*

*Proof.* As in Claim 3.6, we recall the definition  $\mu_i(s) = f_I(B_i(s))$  where  $B_i$  is a Brownian motion with volatility  $\sigma_i$ , and  $f_I$  is the "projection" (1) into the interval  $I = [0; 1]$  with reflective boundaries. Note that  $\mu_i = B_i(0)$ .

For brevity, denote  $\hat{Y}_s = B_i(s)$ , and define the corresponding shorthand  $\hat{S} = \sum_{s \in [t]} \zeta_s \hat{Y}_s$ . Let  $F$  be the failure event when  $\hat{Y}_s \geq 1$  for some  $t \leq m$ . Note that if this event does not occur, then  $Y_s \geq \hat{Y}_s$  for every time  $t \in [m]$  and therefore  $S \geq \hat{S}$ . We use this observation to express  $E[S]$  in terms of  $E[\hat{S}]$ . Let  $p := \Pr[F]$  and note that it is at most  $m^{-4}$ . Then:

$$\begin{aligned} E[\hat{S}] &= (1 - p) E[\hat{S} | \text{not } F] + p E[\hat{S} | F] \\ &\leq (1 - p) E[\hat{S} | \text{not } F] + p(\mu + t\sigma_i) \\ E[S] &\geq (1 - p) E[S | \text{not } F] + p E[S | F] \\ &\geq (1 - p) E[\hat{S} | \text{not } F] \\ &\geq E[\hat{S}] - pt\sigma_i - p. \end{aligned}$$

To prove the claim, it remains to bound  $E[\hat{S}]$ .

Let  $\{s_j\}_{j=1}^\infty$  be the enumeration of all times when arm  $i$  is played. Note that  $N = \max\{j : s_j \leq t\}$ . Define  $\hat{Z}_j = \hat{Y}_{s_j}$  for each  $j$ . We would like to argue that  $\{\hat{Z}_j\}_{j=1}^\infty$  is a martingale and  $N$  is a stopping time. More precisely, claim that this is true for some common filtration. Indeed, one way to define such filtration  $\{\mathcal{F}_j\}_{j=1}^\infty$  is to define  $\mathcal{F}_j$  as the  $\sigma$ -algebra generated by  $s_{j+1}$  and all tuples  $(s_l, Z_l, Z_l^*, \hat{Z}_l)$  such that  $l \leq j$ . Now using the Optional Stopping Theorem one can show that

$$E[\hat{S}] = \sum_{j \in [N]} Z_j = E[N] E[\hat{Z}_0],$$

which proves the claim since  $\hat{Z}_0 = \mu$ .  $\square$

To prove (36), it remains to consider the case  $\mu > 1 - \delta$ .

**Claim 3.12.** *if  $\mu > 1 - \delta$  then*

$$E[S] \geq (1 - \delta) E[N] - t^{-2}.$$

*Proof.* Let  $T$  be the smallest time  $s$  such that  $Y_s \leq 1 - \delta$ . Let  $\{s_j\}_{j=1}^\infty$  be the enumeration of all times when arm  $i$  is played, and let  $J = \max j : t_j \leq T$ . Conditioning on  $T$  and  $J$ , consider the entire problem starting from time  $T+1$ . Then by Claim 3.11 we have:

$$E \left[ \sum_{s=T+1}^m \zeta_s Y_s | T, J \right] \geq (1 - \delta) (E[N] - J) - t^{-2}.$$

Let  $S_T = \sum_{s=T+1}^t \zeta_s Y_s$ . It follows that

$$\begin{aligned} S &= S_T + \sum_{t \in [T]} \zeta_s Y_s \geq S_T + (1 - \delta) J \\ E[S] &= E[S_T] + (1 - \delta) E[J] \\ &\geq (1 - \delta) E[N - J] - t^{-2} + (1 - \delta) E[J] \\ &\geq (1 - \delta) E[N] - t^{-2}. \quad \square \end{aligned}$$

## 4 Using off-the-shelf algorithms

In this section we investigate the following idea: take an off-the-shelf MAB algorithm, run it, and restart it every fixed number of rounds. We consider both the state-informed and state-oblivious versions of the dynamic MAB problem.

We use the following notation: there are  $k$  arms, each arm  $i$  has volatility  $\sigma_i$ , and the average volatility  $\sigma_{\text{av}}$  is defined by  $\sigma_{\text{av}}^2 = \frac{1}{k} \sum_{i=1}^k \sigma_i^2$ . We rely on the following lemma:

**Lemma 4.1.** *Let  $\mu^* = \mu^*(0)$  and let  $i^* \in \operatorname{argmax} \mu_i(0)$ , ties broken arbitrarily. Then for any times  $t \leq t_*$*

$$E[\mu^* - \mu_{i^*}(t)] \leq O(k)(t_*^{-4} + \sigma_{\text{av}}^2 t_* \log t_*). \quad (37)$$

More generally, we can consider arbitrary fixed times

$$0 \leq t_1 \leq t_2 \leq \dots \leq t_k \leq t \leq t_*$$

and define  $\mu^* = \max \mu_i(t_i)$  and  $i^* \in \operatorname{argmax} \mu_i(t_i)$ .

The lemma is obtained, essentially, by combining Claim 3.6 and (34); we omit the details of the proof.

*Remark.* The intuition is that each arm  $i$  is probed in round  $t_i$ , so that  $\mu_i(t_i)$  is the expected value of the corresponding probe. This lemma is similar to Claim 3.6 in that it bounds the downwards drift of  $E[\mu_i(\cdot)]$  which is caused by the proximity of the upper boundary. The difference is that here we specifically consider a "maximal" arm, e.g. when  $t_i \equiv 0$  we consider an arm which is maximal at time 0.

### 4.1 State-informed version: greedy algorithm

For the state-informed version we consider a very simple, "greedy" approach: probe each arm once, choose one with the largest state, play it for a fixed number  $m - k$  of rounds, restart. Call this a *greedy algorithm* with phase length  $m$ .

**Theorem 4.2.** *Consider the state-informed dynamic MAB problem with  $k$  arms such that the volatility of each arm  $i$  is  $\sigma_i$ . With phase length  $m = \sigma_{\text{av}} \sqrt{\log \sigma_{\text{av}}^{-1}}$ , the steady-state regret of the greedy algorithm is at most*

$$O(k \sigma_{\text{av}} \log \sigma_{\text{av}}^{-1}), \text{ where } \sigma_{\text{av}}^2 = \frac{1}{k} \sum_{i=1}^k \sigma_i^2.$$

*Proof.* For the algorithmic result, fix phase length  $m > k$  and consider a single phase of the greedy algorithm. Assume without loss of generality that in the first  $k$  rounds of the phase our algorithm plays arm  $i$  in step  $i$ . Let  $\mu_i = \mu_i(i)$  be the corresponding rewards, and let  $\mu^*$  be the largest of them. Then the greedy algorithm chooses arm  $i^* \in \operatorname{argmax}_{i \in [k]} \mu_i$  and plays it for  $m - k$  rounds. Consider the  $t$ -th of these  $m - k$  rounds and let  $Y_t = \mu_{i^*}(t + k)$  be the state of arm  $i^*$  in this round. By Lemma 4.1 we have  $E[Y_t] \geq E[\mu^*] - z$ , where  $z$  is the right-hand side of (37). Therefore, letting  $\bar{W}$  be the per-round average reward in this phase, we have

$$\begin{aligned} E[\bar{W}] &\geq \frac{1}{m} \sum_{t=1}^{m-k} Y_t \geq \frac{m-k}{m} (E[\mu^*] - z) \\ E[\mu^* - \bar{W}] &\leq z + \frac{k}{m} E[\mu^*] \\ &\leq O(km \sigma_{\text{av}}^2 \log m) + \frac{k}{m} (1 + \frac{1}{m}) \\ &= O(k \sigma_{\text{av}}) \sqrt{\log \sigma_{\text{av}}^{-1}} \end{aligned}$$

for  $m = \sigma_{\text{av}} \sqrt{\log \sigma_{\text{av}}^{-1}}$ .  $\square$

We provide a matching lower bound.

**Theorem 4.3.** *Consider the setting in Theorem 4.2. Then the steady-state regret of the greedy algorithm is  $\tilde{\Omega}(k \sigma_{\text{av}})$ .*

*Proof Sketch.* For simplicity assume  $\sigma_i \equiv \sigma$ . It is known that in time  $t$  a Brownian motion with volatility  $\sigma$  drifts by at least  $\Delta = \tilde{\Omega}(\sigma \sqrt{t})$  with high probability. Thus for each arm  $i$  with high probability  $\mu_i(t) \leq 1 - \Delta/2$ , regardless of the initial value  $\mu_i(0)$ . Now we can obtain a lower bound that corresponds to Lemma 4.1: letting  $\mu^* = \max \mu_i(i)$  and  $i^* \in \operatorname{argmax} \mu_i(i)$  be the arm chosen by the greedy algorithm,

$$E[\mu^* - \mu_{i^*}(t)] \geq \tilde{\Omega}(k \sigma^2 t), \quad (38)$$

for any  $t > k$ . Now consider a given phase of the greedy algorithm. In the first  $k$  rounds the algorithm accumulates regret  $\Omega(k)$ , and in each subsequent round  $t$  the regret is the left-hand side of (38). The theorem follows easily.  $\square$

### 4.2 State-oblivious version via adversarial MAB

For the state-oblivious dynamic MAB problem, we use a very general result of Auer et al. [3] for the adversarial MAB problem. For simplicity, here we only state this result in terms of the present setting.

Let  $\bar{W}_{\mathcal{A}}(t)$  be the average reward collected by algorithm  $\mathcal{A}$  during the time interval  $[1; t]$ .

**Theorem 4.4** (Auer et al.[3]). *Consider the state-oblivious dynamic MAB problem with  $k$  arms. Let  $\mathcal{A}_i$  be an algorithm that plays arm  $i$  at every step. Then there exists an algorithm, call it EXP3, such that for any arm  $i$  and any time  $t$*

$$E[\bar{W}_{\text{EXP3}}(t)] \geq E[\bar{W}_{\mathcal{A}_i}(t)] - O(\frac{k}{t} \log t)^{1/2}.$$

For our problem, we restart EXP3 every  $m$  steps, for some fixed  $m$ ; call this algorithm EXP3( $m$ ).

**Theorem 4.5.** *Consider the state-informed dynamic MAB problem with  $k$  arms such that the volatility of each arm  $i$  is at most  $\sigma_i$ . Then there exists  $m$  such that algorithm EXP3( $m$ ) has steady-state regret*

$$O(k \sigma_{\text{av}} \log \sigma_{\text{av}}^{-1})^{2/3}, \text{ where } \sigma_{\text{av}}^2 = \frac{1}{k} \sum_{i=1}^k \sigma_i^2.$$

*Proof.* Let use shorthand  $\mathcal{A} = \text{EXP3}(m)$ . Let  $\mu^*$  be the maximal expected reward at time 0, and suppose it is achieved by some arm  $i^*$ . Let  $\mathcal{A}^*$  be the algorithm that plays this arm at every step. Let  $Y_t = \mu_{i^*}(t)$  the state of arm  $i^*$  in round  $t$ . Then by Lemma 4.1 we have  $E[Y_t] \geq E[\mu^*] - z(m)$ , where  $z(m)$  is the right-hand side of (37). Therefore:

$$\begin{aligned} E[\bar{W}_{\mathcal{A}^*}(m)] &= E[E[\bar{W}_{\mathcal{A}^*}(m) | Y_1, \dots, Y_m]] \\ &= \frac{1}{m} E[\sum_{t=1}^m Y_t] \\ &\geq \mu^* - z(m) \\ E[\bar{R}_{\mathcal{A}}(m)] &= E[\mu^* - \bar{W}_{\mathcal{A}^*}(m)] \\ &\quad + E[\bar{W}_{\mathcal{A}^*}(m) - \bar{W}_{\mathcal{A}}(m)] \end{aligned} \quad (39)$$

Now using (39) and Theorem 4.4 we obtain

$$E[\bar{R}_{\mathcal{A}}(m)] \leq z(m) + O(\frac{k}{m} \log m)^{1/2}. \quad (40)$$

We choose  $m$  that minimizes the right-hand side of (40).  $\square$

We note in passing that we can also get non-trivial (but worse) guarantees for the state-oblivious dynamic MAB problem using two other off-the-shelf approaches:

- a version of the greedy algorithm which probes each arm a few times in the beginning of each phase,
- a version of Theorem 4.4 in which the benchmark algorithm is allowed to switch arms a few times [3].

Essentially, the first approach is too primitive, while the second one makes overly pessimistic assumptions about the environment. In both cases we obtain guarantees of the form  $\tilde{O}(k\sigma_{\text{av}})^\gamma$ ,  $\gamma < \frac{2}{3}$ , which are inferior to Theorem 4.5.

## 5 Extensions

Recall that the state evolution of arm  $i$  in the dynamic MAB problem is described by (2), where the i.i.d. increments  $\mu_i(t)$  are distributed with respect to some fixed distribution  $\mathcal{X}_i$ . Can we relax the assumption that  $\mathcal{X}_i$  is normal?

**Definition 5.1.** Random variable  $X$  is *stochastically*  $(\rho, \sigma)$ -bounded if its moment-generating function satisfies

$$E[e^{r(X-E[x])}] \leq e^{r^2\sigma^2/2} \text{ for } |r| \leq \rho.$$

This is precisely the condition needed to establish an Azuma-type inequality: if  $S$  is the sum of  $t$  independent stochastically  $(\rho, \sigma)$ -bounded random variables with zero mean, then with high probability  $S \leq \tilde{O}(\sigma\sqrt{t})$ . Specifically, for any  $\lambda \leq \frac{1}{2}\rho\sigma\sqrt{t}$  we have

$$\Pr[S > \lambda\sigma\sqrt{t}] \leq \exp(-\lambda^2/2). \quad (41)$$

Note that a normal distribution  $\mathcal{N}(0, \sigma)$  is  $(\infty, \sigma)$ -bounded, and any distribution with support  $[-\sigma, \sigma]$  is  $(1, \sigma)$ -bounded.

We can recover all of our algorithmic results if we assume that each distribution  $\mathcal{X}_i$  has zero mean and is stochastically  $(\rho, \sigma_i)$ -bounded for some  $\sigma_i$ , where  $\rho > 0$  is a fixed absolute constant. We re-define the *volatility* of arm  $i$  as the infimum of all  $\sigma$  such that  $\mathcal{X}_i$  is  $(\rho, \sigma)$ -bounded.

It is appealing to tackle a more general setting when the only restriction on each distribution  $\mathcal{X}_i$  is that it has mean 0 and variance  $\sigma_i^2$ . We can extend our analysis (at the cost of somewhat weaker guarantees) if we further assume that, essentially, the absolute third moment of  $\mathcal{X}_i$  is comparable to  $\sigma_i^3$ . Then instead of (41) we can use a weaker inequality called the *non-uniform Berry-Esseen theorem* [23]:

$$\Pr\left[\sum_{s=1}^t \mu_i(s) > \sigma_i t^\gamma\right] \leq O\left(\left(\frac{\rho_i}{\sigma_i}\right)^3 t^{1-3\gamma}\right), \quad (42)$$

for any  $\gamma > 1/2$ , where  $\rho_i^3 = E[|\mu_i(s)|^3]$ . We omit further discussion of this extension from the present version.

Let us discuss one other direction in which our setting can be generalized. Recall that in the dynamic MAB problem the state of each arm evolves on the same interval  $I = [0; 1]$  (see Section 1.1) which we term the *fundamental interval*. What if we allow each arm to have a distinct fundamental interval? All our algorithms fit this extended setting with little or no modification. The performance guarantees

should look like a weighted sum of contributions from different arms, where the weights depend (perhaps in rather complicated way) on the respective fundamental intervals. To illustrate this point, we worked out the guarantees for the two algorithms discussed in Section 4, see Appendix A for details. It is an open question to derive similar closed-form guarantees for the other algorithms in this paper.

Recall that in all our results we assumed that the volatilities are known to the algorithm. In fact, this assumption is not necessary: we are interested in the stationary performance of our algorithms and, as it turns out, we can afford to learn the static parameters of the model. Roughly, the argument goes as follows. It suffices for our analysis if for each arm an algorithm knows a 2-approximate upper bound on volatility  $\sigma_i$ , rather than the exact value. One can learn such bound by playing arm  $i$  for  $O(\log^2 \sigma_i)$  rounds, with failure probability as low as  $O(\sigma_i^{-10})$ , and repeat this learning phase every  $\sigma_i^{-1}$  rounds (we omit the details).

**Acknowledgments.** The first author would like to thank Bobby Kleinberg for many stimulating conversations about multi-armed bandits.

## References

- [1] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Machine Learning Research*, 3:397–422, 2002. Preliminary version in *41st IEEE FOCS*, 2000.
- [2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002. Preliminary version in *15th ICML*, 1998.
- [3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002. Preliminary version in *36th IEEE FOCS*, 1995.
- [4] B. Awerbuch and R. D. Kleinberg. Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In *36th ACM Symp. on Theory of Computing (STOC)*, pages 45–53, 2004.
- [5] D. A. Berry and B. Fristedt. *Bandit problems: sequential allocation of experiments*. Chapman and Hall, 1985.
- [6] D. Bertsimas and J. Nino-Mora. Conservation laws, extended polymatroids and multi-armed bandit problems: A unified polyhedral approach. *Math. of Oper. Res.*, 21(2):257–306, 1996.
- [7] D. Bertsimas and J. Nino-Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Operations Research*, 48(1):80–90, 2000.
- [8] V. Dani and T. P. Hayes. How to beat the adaptive multi-armed bandit. Technical report. Available from arXiv at <http://arxiv.org/cs.DS/0602053>, 2006.
- [9] V. Dani and T. P. Hayes. Robbing the bandit: less regret in online geometric optimization against an adaptive adversary. In *17th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 937–943, 2006.
- [10] A. Flaxman, A. Kalai, and H. B. McMahan. Online Convex Optimization in the Bandit Setting: Gradient Descent without a Gradient. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 385–394, 2005.
- [11] J. C. Gittins. Bandit processes and dynamic allocation indices (with discussion). *J. Roy. Statist. Soc. Ser. B*, 41:148–177, 1979.
- [12] J. C. Gittins. *Multi-Armed Bandit Allocation Indices*. John Wiley & Sons, 1989.

- [13] J. C. Gittins and D. M. Jones. A dynamic allocation index for the sequential design of experiments. In J. G. et al., editor, *Progress in Statistics*, pages 241–266. North-Holland, 1974.
- [14] S. Guha and K. Munagala. Approximation algorithms for partial-information based stochastic control with Markovian rewards. In *48th Symp. on Foundations of Computer Science (FOCS)*, 2007.
- [15] S. Guha, K. Munagala, and P. Shi. On Index Policies for Restless Bandit Problems. arXiv:0711.3861v1 [cs.DS], 2007.
- [16] F. Heidari, S. Mannor, and L. Mason. Reinforcement learning-based load shared sequential routing. In *IFIP Networking*, 2007.
- [17] R. D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*, 2004. Full version appeared as Chapters 4-5 in [18].
- [18] R. D. Kleinberg. *Online Decision Problems with Large Strategy Sets*. PhD thesis, MIT, Boston, MA, 2005.
- [19] R. D. Kleinberg. Anytime algorithms for multi-armed bandit problems. In *17th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 928–936, 2006. Full version appeared as Chapter 6 in [18].
- [20] R. D. Kleinberg and F. T. Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *44th Symp. on Foundations of Computer Science (FOCS)*, pages 594–605, 2003.
- [21] T. Lai and H. Robbins. Asymptotically efficient Adaptive Allocation Rules. *Advances in Applied Mathematics*, 6:4–22, 1985.
- [22] H. B. McMahan and A. Blum. Online Geometric Optimization in the Bandit Setting Against an Adaptive Adversary. In *17th Conference on Learning Theory (COLT)*, pages 109–123, 2004.
- [23] K. Neammanee. On the constant in the nonuniform version of the Berry-Esseen theorem. *Intl. J. of Mathematics and Mathematical Sciences*, 2005:12:1951–1967, 2005.
- [24] J. Nino-Mora. Restless bandits, partial conservation laws and indexability. *Advances in Applied Probability*, 33:76–98, 2001.
- [25] S. Pandey, D. Agarwal, D. Chakrabarti, and V. Josifovski. Bandits for Taxonomies: A Model-based Approach. In *SIAM Intl. Conf. on Data Mining (SDM)*, 2007.
- [26] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of optimal queueing network control. In *Structure in Complexity Theory Conference*, pages 318–322, 1994.
- [27] H. Robbins. Some Aspects of the Sequential Design of Experiments. *Bull. Amer. Math. Soc.*, 58:527–535, 1952.
- [28] J. S. Rosenthal. Markov chain convergence: From finite to infinite. *Stochastic Processes Appl.*, 62(1):55–72, 1996.
- [29] R. K. Sundaram. Generalized Bandit Problems. In D. Austen-Smith and J. Duggan, editors, *Social Choice and Strategic Decisions: Essays in Honor of Jeffrey S. Banks (Studies in Choice and Welfare)*, pages 131–162. Springer, 2005. First appeared as *Working Paper, Stern School of Business*, 2003.
- [30] J. N. Tsitsiklis. A short proof of the Gittins index theorem. *Annals of Applied Probability*, 4(1):194–199, 1994.
- [31] G. Weiss. Branching bandit processes. *Probab. Engng. Inform. Sci.*, 2:269–278, 1988.
- [32] P. Whittle. Arm acquiring bandits. *Ann. Probab.*, 9:284–292, 1981.
- [33] P. Whittle. Restless bandits: Activity allocation in a changing world. *J. of Appl. Prob.*, 25A:287–298, 1988.

## A Distinct fundamental intervals

Recall that in the dynamic MAB problem the state of each arm evolves on the same interval  $I = [0; 1]$  (see Section 1.1)

which we term the *fundamental interval*. In this section we consider a generalization in which we allow each arm to have a distinct fundamental interval. We work out the guarantees for the two algorithms discussed in Section 1.4.

The main contribution of this appendix is that we find a way to upper-bound the steady-state regret of the respective algorithms in terms of reasonably defined averages of the arms’ properties. The actual derivations are rather tedious but not that illuminating; we omit them from this version.

### A.1 The setting and notation

We consider the following setting. There are  $k$  arms. Each arm has volatility  $\sigma_i$  and fundamental interval  $[a_i; b_i]$ . Without loss of generality we assume that  $b_1 \leq \dots \leq b_k$  and that  $\max a_i < \min b_i$ . (If the latter fails then we can always ignore the arm with the smallest upper boundary  $b_i$ .) To simplify the derivation we assume that  $\max \sigma_i \leq \frac{1}{3}$ .

Define the *weight* of arm  $i$  as

$$w_i = \prod_{l=1}^k \frac{b_l - a_l}{b_l - a_l},$$

Define the *average volatility*  $\sigma_{av}$  by

$$\sigma_{av}^2 = \frac{\sum_{i \in [k]} w_i (b_i - a_i) \sigma_i^2}{\sum_{i \in [k]} w_i (b_i - a_i)}$$

Define the *average length* as

$$d_{av} = \frac{1}{k} \sum_{i \in [k]} w_i (b_i - a_i).$$

To see that the quantities we defined above are reasonable as *averages*, note that if all arms have the same fundamental interval  $[a; b]$  then all weights are 1 and  $d_{av} = b - a$  and, moreover, the average volatility  $\sigma_{av}$  coincides with the one defined in the body of the paper.

### A.2 Results

We present two results that extend, respectively, Theorem 4.2 and Theorem 4.5 to the setting from Section A.1. In both cases the algorithms are exactly the same. The main tool is a version of Lemma 37, where the guarantee (37) looks exactly the same in our notation, except the right-hand side is multiplied by  $d_{av}$ .

**Theorem A.1.** *Consider the deterministic dynamic MAB problem in the setting from Section A.1. Let  $a_{min} = \min a_i$ . Then for phase length*

$$m = \sigma_{av}^{-1} \sqrt{(b_k - a_{min}) / \log \sigma_{av}^{-1}}$$

*the greedy algorithm has steady-state regret*

$$O(k \sigma_{av}) \sqrt{(b_k - a_{min}) d_{av} \log \sigma_{av}^{-1}}.$$

**Theorem A.2.** *Consider the state-informed dynamic MAB problem in the setting from Section A.1. Then there exists  $m$  such that algorithm EXP3( $m$ ) has steady-state regret*

$$O(d_{av})^{1/3} (k \sigma_{av} \log \sigma_{av}^{-1})^{2/3}.$$

---

# Stochastic Linear Optimization under Bandit Feedback

---

Varsha Dani\* and Thomas P. Hayes† and Sham M. Kakade†

## Abstract

In the classical stochastic  $k$ -armed bandit problem, in each of a sequence of  $T$  rounds, a decision maker chooses one of  $k$  arms and incurs a cost chosen from an unknown distribution associated with that arm. The goal is to minimize *regret*, defined as the difference between the cost incurred by the algorithm and the optimal cost.

In the linear optimization version of this problem (first considered by Auer [2002]), we view the arms as vectors in  $\mathbb{R}^n$ , and require that the costs be linear functions of the chosen vector. As before, it is assumed that the cost functions are sampled independently from an unknown distribution. In this setting, the goal is to find algorithms whose running time and regret behave well as functions of the number of rounds  $T$  and the dimensionality  $n$  (rather than the number of arms,  $k$ , which may be exponential in  $n$  or even infinite).

We give a nearly complete characterization of this problem in terms of both upper and lower bounds for the regret. In certain special cases (such as when the decision region is a polytope), the regret is  $\text{polylog}(T)$ . In general though, the optimal regret is  $\Theta^*(\sqrt{T})$  — our lower bounds rule out the possibility of obtaining  $\text{polylog}(T)$  rates in general.

We present two variants of an algorithm based on the idea of “upper confidence bounds.” The first, due to Auer [2002], but not fully analyzed, obtains regret whose dependence on  $n$  and  $T$  are both essentially optimal, but which may be computationally intractable when the decision set is a polytope. The second version can be efficiently implemented when the decision set is a polytope (given as an intersection of half-spaces), but gives up a factor of  $\sqrt{n}$  in the regret bound.

Our results also extend to the setting where the set of allowed decisions may change over time.

## 1 Introduction

The seminal work of Robbins [1952] introduced a formalism for studying the sequential design of experiments, which is now referred to as the *multi-armed bandit* problem. In this foundational paradigm, at each time step a decision maker chooses one of  $K$  decisions or “arms” (e.g. treatments, job schedules, manufacturing processes, etc) and receives some feedback loss only for the chosen decision. In the most unadorned model, it is assumed that the cost for each decision is independently sampled from some fixed underlying (and unknown) distribution (that is different for each decision). The goal of the decision maker is to minimize the average loss over some time horizon. This basic model of decision making under uncertainty already typifies the conflict between minimizing the immediate loss and gathering information that will be useful in the long-run. This sequential design problem — often referred to as the *stochastic multi-armed bandit* problem — and a long line of successor bandit problems have been extensively studied in the statistics community (see, e.g., [Berry and Fristedt, 1985]), with close attention paid to obtaining sharp convergence rates.

While this paradigm offers a formalism to a host of natural decision problems (e.g. clinical treatment, manufacturing processes, job scheduling), a vital issue to address for applicability to modern problems is how to tackle a set of feasible decisions that is often large (or infinite). For example, the classical bandit problem of clinical treatments (often considered in statistics) — where each decision is a choice of one of  $K$  treatments — is often better modelled by choosing from some (potentially infinite) set of *mixed* treatments subject to some budget constraint (where there is a cost per unit amount of each of drug). In manufacturing problems, often the goal is to maximize revenue subject to choosing among some large set of decisions that satisfy certain manufacturing constraints (where the revenue from each decision may be unknown). A modern variant of this problem that is receiving increasing attention is the routing problem where the goal is to send packets from  $A$  to  $B$  and the cost of each route is unknown (see, e.g., [Awerbuch and Kleinberg, 2004]).

We study a natural extension of the stochastic multi-armed bandit problem to linear optimization — a problem first considered in Auer [2002]. Here, we assume the decision space is an arbitrary subset  $D \subset \mathbb{R}^n$  and that there is fixed distribution  $\pi$  over cost functions. At each round, the learner chooses a decision  $x \in D$ , then a cost function  $f(\cdot) : D \rightarrow [0, 1]$  is

---

\*Department of Computer Science, University of Chicago, varsha@cs.uchicago.edu

†Toyota Technological Institute at Chicago, {hayest, sham}@tti-c.org

sampled from  $\pi$ . Only the loss  $f(x)$  is revealed to the learner (and not the function  $f(\cdot)$ ). We assume that the expected loss is a fixed linear function, i.e. that  $\mathbb{E}[f(x)] = \mu \cdot x$ , where the expectation is with respect to  $f$  sampled from  $\pi$  (technically, we make a slightly weaker assumption, precisely stated in the next section). The goal is to minimize the total loss over  $T$  steps. As is standard, success is measured by the regret — the difference between the performance of the learner and that of the optimal algorithm which has knowledge of  $\pi$ . Note that the optimal algorithm here simply chooses the best decision with respect to the linear mean vector  $\mu$ .

Perhaps the most important and natural example in this paradigm is the (stochastic) online linear programming problem. Here,  $D$  is specified by linear inequality constraints. If the mean  $\mu$  were known, then this is simply a linear programming problem. Instead, at each round, the learner only observes noisy feedback of the chosen decision, with respect to the underlying linear cost function.

### 1.1 Summary of Our Results and Related Work

Auer [2002] provides the first analysis of this problem. This paper builds and improves upon the work of Auer [2002] in a number of ways. A related model was considered by Abe and Long [1999], where the decision sets are allowed to vary as a function of the time. Our results can be extended to this more general model, which we discuss in Section 7.

While Auer [2002] provides an elegant deterministic algorithm, based on upper confidence bounds of  $\mu$ , an analysis of the performance of this algorithm was not provided, due to rather subtle independence issues (though it was conjectured that this simple algorithm was sufficient). Instead, a more complicated master algorithm was analyzed — this master algorithm called the simpler upper confidence algorithm as a subroutine. In this work, we directly analyze the simpler upper confidence algorithm. Unfortunately, implementing this algorithm in certain cases (when  $D$  is large or infinite) may be inefficient. However, we also provide a modification to this algorithm (that uses a different confidence region based on the  $L_1$ -norm), which may be implemented efficiently for the case when  $D$  is an (infinite) convex set, given certain oracle optimization access to  $D$ .

The analysis of Auer [2002] achieves a regret bound of  $O^*((\log |D|)^{3/2} \text{poly}(n) \sqrt{T})$  where  $n$  is dimension of the decision space,  $T$  is the time horizon, and  $|D|$  is the number of feasible decisions. For the simpler upper confidence algorithm, we show that it enjoys a bound of  $O^*(n\sqrt{T})$ , which does not depend on the cardinality of the decision region,  $|D|$ . While this algorithm may be inefficient in some cases, we also provide an efficient algorithm (that uses a slightly different confidence region), which achieves a slightly worse bound of  $O^*(n^{3/2}\sqrt{T})$ . Using the result in Auer [2002], one can also derive a bound of the form  $O(\text{poly}(n)\sqrt{T})$  for infinite decision sets by appealing to a naive (inefficient) covering argument (where the algorithm is run on an appropriately fine cover of  $D$ ). However, this argument results in a less sharp bound in terms of  $n$ <sup>1</sup>, though a better reduction to

<sup>1</sup> Using Auer [2002], one can derive the less sharp bound of  $O^*(n^{5/2}\sqrt{T})$  for arbitrary compact decision sets with two observations. First, through a covering argument, we need only consider

Auer [2002] may be possible.

For the case of finite decision sets, such as the  $K$ -arm bandit case, a regret that is only logarithmic in the time horizon is achievable. In particular, in a different line of work, Auer et al. [2002] showed that the optimal regret for the  $K$ -arm bandit case was characterized as  $\frac{K}{\Delta} \log T$ , where  $\Delta$  is the “gap” between the performance of the best arm and the second best arm. This result is stated in terms of the problem dependent constant  $\Delta$ , so one can view it as the asymptotic regret for a given problem. In fact, historically, there is long line of work in the  $K$ -arm bandit literature (e.g. [Lai and Robbins., 1985, Agrawal, 1995]) concerned with obtaining optimal rates for a fixed problem, which are often logarithmic in  $T$  when stated in terms of some problem dependent constant.

Hence, in our setting, in the case where  $|D|$  is finite, we know that a log rate in the time is achievable by a direct reduction to the  $K$ -arm bandit case (though this naive reduction results in an exponentially worse dependence in terms of  $|D|$ ). Our work shows that a regret of  $\frac{n^2}{\Delta} \text{polylog}(T)$  can be achieved, where  $\Delta$  is a generalized definition of the gap that is appropriate for a potentially infinite  $D$ . Hence, a polylogarithmic rate in  $T$  is achievable with a constant that is only polynomial in  $n$  and has *no dependence* on the size of the (potentially infinite) decision region. Here,  $\Delta$  can be thought of as the gap between the values of the best and second best extremal points of the decision set (which we define precisely later). For example, if  $D$  is a polytope, then  $\Delta$  is the gap in value between the first and second best corner decisions. For the case where  $D$  is finite,  $\Delta$  is exactly the same as in the  $K$  arm case. However, for some natural decision regions, such as a sphere,  $\Delta$  is 0 so this (problem dependent) bound is not applicable. Note that  $\Delta$  is *never* 0 for the  $K$ -arm case (unless there is effectively one arm), so a logarithmic rate in  $T$  is always possible in the  $K$ -arm case.

Note that this set of results still raises the question of whether there is an algorithm achieving polylogarithmic regret (as a function of  $T$ ) for the case when  $\Delta = 0$ , which could be characterized in terms of some different, more appropriate problem dependent constant. Our final contribution answers this question in the negative. We provide a lower bound showing that the regret of any algorithm on a particular problem (which we construct with  $\Delta = 0$ ) is  $\Omega(n\sqrt{T})$ . In addition to showing that a polylogarithmic rate is not achievable in general, it also shows our upper bound is tight in terms of  $n$  and  $T$ . Note this result is in stark contrast to the  $K$ -arm case where the optimal asymptotic regret for any given problem is always logarithmic in  $T$ .

We should also note that the lower bound in this paper is significantly stronger than the bound provided in Dani et al. [2008], which is also  $\Omega(n\sqrt{T})$ . In this latter lower bound, the decision problem the algorithm faces is chosen as a function of the time  $T$ . In particular, the construction in Dani et al. [2008] used a decision region which was a hypercube

$D$  to be exponential in  $n$ . Second, Auer [2002] assumes that  $D$  is a subset of the sphere, which leads to an additional  $\sqrt{n}$  factor. To see this, note the comments in the beginning of Section 5 essentially show that a general decision region can be thought of as living in a hypercube (due to the barycentric spanner property), so the additional  $\sqrt{n}$  factor comes from rescaling the cube into a sphere.

(so  $\Delta > 0$  as this a polytope) — in fact,  $\Delta$  actually scaled as  $1/\sqrt{T}$ . In order to negate the possibility of a polylogarithmic rate for a particular problem, we must hold  $\Delta = 0$  as we scale the time, which we accomplish in this paper with a more delicate construction using an  $n$ -dimensional decision space constructed out of a Cartesian product of 2-dimensional spheres.

## 1.2 The Price of Bandit Information

It is natural to ask how much worse the regret is in the bandit setting as compared to a setting where we received full information about the complete loss function  $f(\cdot)$  at the end of each round. In other words, what is the *price of bandit information*?

For the full information case, Dani et al. [2008] showed the regret is  $O^*(\sqrt{nT})$  (which is tight up to log factors). In fact, in the stochastic case considered here, it is not too difficult to show that, in the full information case, the algorithm of “do the best in the past” achieves this rate. Hence, as the regret is  $O^*(n\sqrt{T})$  in the bandit case and  $O^*(\sqrt{nT})$  (both of which are tight up to log factors), we have characterized the price of bandit information as  $\sqrt{n}$ , which is a rather mild dependence on  $n$  for having such limited feedback.

We should also note that the work in Dani et al. [2008] considers the adversarial case, where the cost functions are chosen in an arbitrary manner rather than stochastically. Here, it was shown that the regret in the bandit setting is  $O(n^{3/2}\sqrt{T})$  (ignoring polylogarithmic factors), though it was conjectured that this bound was loose and the optimal rate should be identical to rate for the stochastic case, considered here.

It is striking that the convergence rate for the bandit setting is only a factor of  $\sqrt{n}$  worse than in the full information case — in stark contrast to the  $K$ -arm bandit setting, where the gap in the dependence on  $K$  is exponential ( $\sqrt{TK}$  vs.  $\sqrt{T \log K}$ ). See Dani et al. [2008] for further discussion.

## 2 Preliminaries

Let  $D \subset \mathbb{R}^n$  be a compact (but otherwise arbitrary) set of decisions. Without loss of generality, assume this set is of full rank. On each round, we must choose a decision  $x_t \in D$ . Each such choice results in a cost  $\ell_t = c_t(x_t) \in [-1, 1]$ .

We assume that, regardless of the history  $\mathcal{H}_t$ , the conditional expectation of  $c_t$  is a fixed linear function, *i.e.*, for all  $x \in D$ ,

$$\mathbb{E}(c_t(x) \mid \mathcal{H}_t) = \mu \cdot x = \mu^\dagger x \in [-1, 1].$$

where  $x \in D$  is arbitrary, and we denote the transpose of any column vector  $v$  by  $v^\dagger$ . (Naturally, the vector  $\mu$  is unknown, though fixed.) Under these assumptions, the *noise sequence*,

$$\eta_t = c_t(x_t) - \mu \cdot x_t$$

is a martingale difference sequence.

[We remark here that that our earlier assumption that  $D$  was compact was actually unnecessary, in light of our further assumptions that the cost functions are bounded and linear in expectation.]

A special case of particular interest is when the cost functions  $c_t$  are themselves linear functions sampled independently from some fixed distribution. Note, however, that

our assumptions are also met under the addition of any time-dependent unbiased random noise function.

In this paper we address the bandit version of the geometric optimization problem, where the decision maker’s feedback on each round is only the actual cost  $\ell_t = c_t(x_t)$  received on that round, *not* the entire cost function  $c_t(\cdot)$ .

If  $x_1, \dots, x_T$  are the decisions made in the game, then define the *cumulative regret* by

$$R_T = \sum_{t=1}^T (\mu^\dagger x_t - \mu^\dagger x^*)$$

where  $x^* \in D$  is an optimal decision for  $\mu$ , *i.e.*,

$$x^* \in \operatorname{argmin}_{x \in D} \mu^\dagger x$$

which exists since  $D$  is compact. Observe that if the mean  $\mu$  were known, then the optimal strategy would be to play  $x^*$  every round. Since the expected loss for each decision  $x$  equals  $\mu^\dagger x$ , the cumulative regret is just the difference between the expected loss of the optimal algorithm and the expected loss for the actual decisions  $x_t$ . Since the sequence of decisions  $x_1, \dots, x_T$  may depend on the particular sequence of random noise encountered,  $R_T$  is a random variable. Our goal in designing an algorithm is to keep  $R_T$  as small as possible.

It is also important for us to make use of a *barycentric spanner* for  $D$  as defined in Awerbuch and Kleinberg [2004]. A *barycentric spanner* for  $D$  is a set of vectors  $b_1, \dots, b_n$ , all contained in  $D$ , such that every vector in  $D$  can be expressed as a linear combination of the spanner with coefficients in  $[-1, 1]$ . Awerbuch and Kleinberg [2004] showed that such a set exists for compact sets  $D$ . We assume we have access to such a spanner of the decision region, though an approximate spanner would suffice for our purposes (Awerbuch and Kleinberg [2004] provide an efficient algorithm for computing an approximate spanner).

Let  $A$  be a positive definite  $n \times n$  matrix, and let  $\nu \in \mathbb{R}^n$ . We will use the following notation for the 1- and 2-norms based on  $A$ .

$$\begin{aligned} \|\nu\|_{2,A} &:= \|A^{1/2}\nu\|_2 = \sqrt{\nu^\dagger A \nu}. \\ \|\nu\|_{1,A} &:= \|A^{1/2}\nu\|_1 = \sum_{i=1}^n |A^{1/2}\nu|_i. \end{aligned}$$

Here  $A^{1/2}$  is the unique positive definite  $n \times n$  matrix whose square is  $A$ .

## 3 Main Results

### 3.1 Algorithms

We now present our main algorithms, ConfidenceBall<sub>2</sub> and ConfidenceBall<sub>1</sub>. The subscripts on the names refer to the type of norm used in the algorithm; apart from scaling the radius differently, which we do only for convenience, this is the sole difference between the algorithm statements. As we shall discuss later, we are able to prove better regret guarantees for ConfidenceBall<sub>2</sub>, matching the lower bound, up to log factors.

Both algorithms can be efficiently implemented in the simplistic case when the decision set is a small finite set.

**Algorithm 3.1:** CONFIDENCEBALL<sub>2</sub>( $D, \delta$ )**Initialization:**Find a barycentric spanner  $b_1, \dots, b_n$  for  $D$ 

$$A_1 = \sum_{i=1}^n b_i b_i^\dagger$$

$$\hat{\mu}_1 = 0$$

for  $t \leftarrow 1$  to  $\infty$ 

$$\beta_t = \max \left( 128n \ln t \ln(t^2/\delta), \left( \frac{8}{3} \ln \left( \frac{t^2}{\delta} \right) \right)^2 \right)$$

$$B_t^2 = \{ \nu: \|\nu - \hat{\mu}_t\|_{2, A_t} \leq \sqrt{\beta_t} \}$$

$$x_t = \operatorname{argmin}_{x \in D} \min_{\nu \in B_t^2} (\nu^\dagger x)$$

Incur and observe loss  $\ell_t := c_t(x_t)$ 

$$A_{t+1} = A_t + x_t x_t^\dagger$$

$$\hat{\mu}_{t+1} = A_{t+1}^{-1} \sum_{\tau=1}^t \ell_\tau x_\tau$$

However, in the important special case when the decision set is a polytope presented as the intersection as halfspaces,<sup>2</sup> ConfidenceBall<sub>1</sub> can be implemented in polynomial time, while ConfidenceBall<sub>2</sub> is NP-hard to implement, at least for some decision sets. More generally, ConfidenceBall<sub>1</sub> can be implemented efficiently given oracle access to an algorithm which can find a decision in  $\operatorname{argmin}_{x \in D} \nu \cdot x$  (where  $\nu$  is the input). We discuss these issues further in Subsection 3.4.

**ConfidenceBall<sub>2</sub>**

Algorithm 3.1 is due to Auer [2002], who called it the LinRel algorithm. We have generalized the statement slightly so that it can be applied in settings where  $D$  is not necessarily stored in enumerated form, and indeed, may not even be finite. We have renamed the algorithm ConfidenceBall<sub>2</sub> to emphasize its key feature of maintaining an  $\ell_2$  ball,  $B_t^2$ , which contains  $\mu$  with high probability.

The algorithm is motivated as follows. Suppose decisions  $x_1, \dots, x_{t-1}$  have been made, incurring corresponding losses  $\ell_1, \dots, \ell_{t-1}$ . Then a reasonable estimate  $\hat{\mu}$  to the true mean cost vector  $\mu$  can be constructed by minimizing the square loss:

$$\hat{\mu} := \operatorname{argmin}_{\nu} \mathcal{L}(\nu), \text{ where } \mathcal{L}(\nu) := \sum_{\tau < t} (\nu^\dagger x_\tau - \ell_\tau)^2.$$

Defining  $A = \sum x_\tau x_\tau^\dagger$ , we have that the least squares estimator is

$$\hat{\mu} = A^{-1} \sum_{\tau < t} \ell_\tau x_\tau.$$

A natural confidence region for  $\mu$  is the set of  $\nu$  for which  $\mathcal{L}(\nu)$  exceeds  $\mathcal{L}(\hat{\mu})$  by at most some amount  $\beta$ , i.e. the set

$$\{ \nu: \mathcal{L}(\nu) - \mathcal{L}(\hat{\mu}_t) \leq \beta \}$$

It is straightforward to see that:

$$\mathcal{L}(\nu) - \mathcal{L}(\hat{\mu}) = (\nu - \hat{\mu})^\dagger A (\nu - \hat{\mu})$$

Thus the confidence region proposed above has the shape of an ellipsoid centered on  $\hat{\mu}$ , where the axes are defined

<sup>2</sup>Note that the number of vertices of a polytope may be exponential in the number of defining half-spaces.

**Algorithm 3.2:** CONFIDENCEBALL<sub>1</sub>( $D, \delta$ )**Initialization:**Find a barycentric spanner  $b_1, \dots, b_n$  for  $D$ 

$$A_1 = \sum_{i=1}^n b_i b_i^\dagger$$

$$\hat{\mu}_1 = 0$$

for  $t \leftarrow 1$  to  $\infty$ 

$$\beta_t = \max \left( 128n \ln t \ln(t^2/\delta), \left( \frac{8}{3} \ln \left( \frac{t^2}{\delta} \right) \right)^2 \right)$$

$$B_t^1 = \{ \nu: \|\nu - \hat{\mu}_t\|_{1, A_t} \leq \sqrt{n\beta_t} \}$$

$$x_t = \operatorname{argmin}_{x \in D} \min_{\nu \in B_t^1} (\nu^\dagger x)$$

Incur and observe loss  $\ell_t := c_t(x_t)$ 

$$A_{t+1} = A_t + x_t x_t^\dagger$$

$$\hat{\mu}_{t+1} = A_{t+1}^{-1} \sum_{\tau=1}^t \ell_\tau x_\tau$$

through  $A$ . This set is commonly referred to as the set of vectors  $\nu$  with bounded Mahalanobis distance with respect to mean  $\hat{\mu}$  and covariance matrix  $A^{-1}$ .

A difficulty with the above reasoning is that we have implicitly assumed that  $A$  is invertible, which is clearly false for  $t < n$ . Under a slight alteration, define the estimator  $\hat{\mu}_t$  at time  $t$  by

$$\hat{\mu}_t = A_t^{-1} \sum_{\tau < t} \ell_\tau x_\tau.$$

where  $A_t$  is now defined as

$$A_t = \sum_{i=1}^n b_i b_i^\dagger + \sum_{\tau < t} x_\tau x_\tau^\dagger$$

where  $b_1, \dots, b_n$  is the barycentric spanner (see Preliminaries for the definition). It is easily seen that  $A_t$  is positive definite (and hence invertible), since the spanner is linearly independent. Intuitively, the first term in  $A_t$  (the sum of outerproducts of the spanner vectors) is a natural initialization of the confidence region, as it imposes uncertainty along the directions in which  $D$  varies most (namely the spanner directions). Our proofs effectively show that an approximate spanner would suffice instead. Note that  $\hat{\mu}_t$  is the least squares estimator for the sampled data if we pretend that decisions  $b_1, \dots, b_n$  were selected on fictitious rounds  $t = -n + 1, \dots, t = 0$  and all incurred loss 0.

Now define the confidence region at time  $t$  to be the ellipsoid

$$B_t^2 := \{ \nu: \|\nu - \hat{\mu}_t\|_{2, A_t} \leq \sqrt{\beta_t} \}$$

In the proofs, we show that, with our choice of  $\beta_t$ ,  $\mu$  always remains inside this ellipsoid for all times  $t$ , with high probability.

The decision at the next round is then the greedy optimistic decision:

$$x_t = \operatorname{argmin}_{x \in D} \min_{\nu \in B_t^2} (\nu^\dagger x).$$

Again, this exists since  $D$  is compact.

It should be remarked that although the linear function  $x \mapsto \mu \cdot x$  is a feasible cost function, and  $\hat{\mu}_t$  is an approximation to  $\mu$ , the function  $x \mapsto \hat{\mu}_t \cdot x$  may be far from being a

feasible (i.e.  $[-1, 1]$ -valued) cost function on  $D$  — however, it is bounded in  $[-n, n]$ .

### ConfidenceBall<sub>1</sub>

ConfidenceBall<sub>1</sub>, Algorithm 3.2, uses a (skewed) octahedron,  $B_t^1$ , as its confidence region, rather than the ellipsoid,  $B_t^2$ . The radius of  $B_t^1$  has been set just large enough that it contains the ellipsoid  $B_t^2$  as an inscribed subset.

The cost of this enlarged confidence region is a slightly worse regret (in terms of  $n$ ). The benefit we get in exchange is that balls in the 1-norm have only  $2n$  extremal points, rather than the infinitely many that balls in the 2-norm have. This leads to a more computationally efficient algorithm, as we discuss in Section 3.4.

## 3.2 Upper Bounds

In the traditional  $K$ -arm bandit literature, the regret is often characterized for a particular problem in terms of  $T$ ,  $K$ , and problem dependent constants. In the  $K$ -arm bandit results of Auer et al. [2002], this problem dependent constant is the “gap” between the loss of the best arm and the second best arm.

We cannot naively use the same definition since if the decision space is, say a convex set, then there is no well defined notion of second best arm. Instead, we define the gap as follows. Let  $\mathcal{E}$  denote the set of extremal points of the decision set  $D$ , where an *extremal point* of  $D$  is defined as a point which is not a proper convex combination of points in  $D$ . It is easy to see that any linear loss function on  $D$  always attains its minimum value at a point in  $\mathcal{E}$ . It is not too difficult to show that ConfidenceBall<sub>2</sub> always plays extremal points, due to the strict convexity of the confidence region. Similarly, although ConfidenceBall<sub>1</sub> can potentially play non-extremal points  $x_t$ , it can easily be implemented so that it only plays extremal points (see Section 3.4 for further discussion of implementation issues.)

Now define the set of suboptimal extremal points as:

$$\mathcal{E}_- = \{x \in \mathcal{E} : \mu \cdot x > \mu \cdot x^*\},$$

and note that  $\mathcal{E}_-$  is non-empty (unless  $\mu = 0$ , in which case there is nothing to prove). Define the gap,  $\Delta$ , as

$$\Delta = \inf_{x \in \mathcal{E}_-} \mu \cdot x - \mu \cdot x^*$$

so the  $\Delta$  is just the difference in costs between the optimal and next to optimal decision among the extremal points. Note that if  $D$  is a fixed polytope then  $\Delta > 0$ . However, if  $D$  is a ball then  $\Delta = 0$ , as all points on the surface (a sphere) are extremal — so  $\inf_{x \in \mathcal{E}_-} \mu \cdot x = \mu \cdot x^*$  (and no point in  $\mathcal{E}_-$  achieves this value).

We now state the first upper bound, which is a problem dependent bound stated in terms of  $\Delta$ .

**Theorem 1 (Problem Dependent Upper Bound)** Recall that  $\beta_T = \max \left( 128n \ln T \ln(T^2/\delta), \left( \frac{8}{3} \ln \left( \frac{T^2}{\delta} \right) \right)^2 \right)$ . Let  $0 < \delta < 1$ . Suppose the decision set  $D$  and the true mean  $\mu$  have a gap  $\Delta > 0$ . We have:

- **ConfidenceBall<sub>2</sub>**: For all sufficiently large  $T$ , the cumulative regret  $R_T$  of ConfidenceBall<sub>2</sub>( $D, \delta$ ) is with high

probability at most  $O(\frac{n^2}{\Delta} \log^3 T)$ . More precisely,

$$\text{Prob} \left( \forall T, R_T \leq \frac{8n\beta_T \ln(T)}{\Delta} \right) \geq 1 - \delta,$$

- **ConfidenceBall<sub>1</sub>**: If ConfidenceBall<sub>1</sub> is implemented so that it only chooses extremal points  $x_t \in D$  (which is always possible) then, for all sufficiently large  $T$ , the cumulative regret  $R_T$  of ConfidenceBall<sub>1</sub>( $D, \delta$ ) is with high probability at most  $O(\frac{n^3}{\Delta} \log^3 T)$ . More precisely,

$$\text{Prob} \left( \forall T, R_T \leq \frac{8n^2\beta_T \ln(T)}{\Delta} \right) \geq 1 - \delta,$$

Analogous to the  $K$ -arm case, when  $\Delta > 0$ , a polylogarithmic rate in  $T$  is achievable with a constant that is only polynomial in  $n$  and has *no dependence* on the size of the decision region.

The following upper bound is stated without regard to the specific parameter  $\Delta$  for a given problem. Furthermore, it also holds for the case when  $\Delta = 0$ .

**Theorem 2 (Problem Independent Upper Bound)** Recall that  $\beta_T = \max \left( 128n \ln T \ln(T^2/\delta), \left( \frac{8}{3} \ln \left( \frac{T^2}{\delta} \right) \right)^2 \right)$ . Let  $0 < \delta < 1$ . We have:

- **ConfidenceBall<sub>2</sub>**: For all sufficiently large  $T$ , the cumulative regret  $R_T$  of ConfidenceBall<sub>2</sub>( $D, \delta$ ) is with high probability at most  $O^*(n\sqrt{T})$ , where the  $O^*$  notation hides a polylogarithmic dependence on  $T$ . More precisely,

$$\text{Prob} \left( \forall T, R_T \leq \sqrt{8nT\beta_T \ln T} \right) \geq 1 - \delta.$$

- **ConfidenceBall<sub>1</sub>**: For all sufficiently large  $T$ , the cumulative regret  $R_T$  of ConfidenceBall<sub>1</sub>( $D, \delta$ ) is with high probability at most  $O^*(n^{3/2}\sqrt{T})$ , where the  $O^*$  notation hides a polylogarithmic dependence on  $T$ . More precisely,

$$\text{Prob} \left( \forall T, R_T \leq \sqrt{8n^2T\beta_T \ln T} \right) \geq 1 - \delta.$$

The following subsection shows our bound of  $O^*(n\sqrt{T})$  is tight, in terms of both  $n$  and  $T$ . Also, as mentioned in the Introduction, tightly characterizing the dimensionality dependence allows us to show that the price of bandit information is  $\Theta^*(\sqrt{n})$ .

## 3.3 Lower Bounds

Note that our upper bounds still leave open the possibility that there is a polylogarithmic regret (as a function of  $T$ ) for the case when  $\Delta = 0$ , which could be characterized in terms of some different, more appropriate problem dependent constant. Our next result is a lower bound of  $\Omega(n\sqrt{T})$  on the expected regret, showing that no such improvement is possible.

For the lower bound, we must consider a decision region with  $\Delta = 0$ , which rules out polytopes and finite sets (so the decision region of a hypercube, used by Dani et al. [2008],

is not appropriate here. See Introduction for further discussion). The decision region is constructed as follows. Assume  $n$  is even. Let  $D_n = (S^1)^{n/2}$  be the Cartesian product of  $n/2$  circles. That is,  $D_n = \{(x_1, \dots, x_n) : x_1^2 + x_2^2 = x_3^2 + x_4^2 = \dots = x_{n-1}^2 + x_n^2 = 1\}$ . Observe that  $D_n$  is a subset of the intersection of the cube  $[-1, 1]^n$  with the sphere of radius  $\sqrt{n/2}$  centered at the origin.

Our cost functions take values in  $\{-1, +1\}$ , and for every  $x \in D_n$ , the expected cost is  $\mu \cdot x$ , where  $n\mu \in D_n$ . Since each cost function is only evaluated at one point, any two distributions over  $\{-1, +1\}$ -valued cost functions with the same value of  $\mu$  are equivalent for the purposes of our model.

**Theorem 3 (Lower Bound)** *If  $\mu$  is chosen uniformly at random from the set  $D_n/n$ , and the cost for each  $x \in D_n$  is in  $\{-1, +1\}$  with mean  $\mu \cdot x$ , then, for every algorithm, for every  $T \geq 1$ ,*

$$\mathbb{E} R = \mathbb{E}_\mu \mathbb{E}(R \mid \mu) \geq \frac{1}{10} n \sqrt{T}.$$

where the inner expectation is with respect to observed costs.

In addition to showing that a polylogarithmic rate is not achievable in general, this bound shows our upper bound is tight in terms of  $n$  and  $T$ . Again, contrast this with the  $K$ -arm case where the optimal asymptotic regret for any given problem is always logarithmic in  $T$ .

### 3.4 Computational Efficiency

We now turn our attention to the computational complexity of implementing the ConfidenceBall algorithms.

As discussed in Section 2, it is easy to find an approximate barycentric spanner in  $O(n^2)$  time. Of all the other steps in the algorithm, the only one which poses serious difficulties is the selection of the decision  $x_t$ :

$$x_t := \operatorname{argmin}_{x \in D} \min_{\nu \in B_t} (\nu^\dagger x)$$

where  $B_t$  is the confidence ball.

Now, if  $|D|$  is small, we can enumerate all choices for  $x$ , and the inner minimization is easy for both norms. This shows that an implementation in time  $\operatorname{poly}(n)|D|$  is possible. There are also some special cases, such as when  $D$  is the unit ball, when the algorithm can be implemented in time  $\operatorname{poly}(n)$  using a little calculus, despite  $|D|$  being infinite. We leave the details as an exercise to the interested reader.

The most practically relevant setting is when  $D$  is (the vertex set of) a polytope defined by a system of linear inequalities (or equivalently, the intersection of a given set of halfspaces). In this case, the number of vertices of  $D$  may be exponential in the number of inequalities.

In this setting (and others), we can assume oracle access to an algorithm which can efficiently find a decision in  $\operatorname{argmin}_{x \in D} \nu \cdot x$  (where  $\nu$  is the input). Here, in the case of ConfidenceBall<sub>1</sub>, we can enumerate over the  $2n$  vertices of  $B_t$  to find the optimum. For each such  $\nu \in B_t$ , we can call this oracle to find the optimal  $x \in D$ , and then we can choose the appropriate decision out of these  $2n$  decisions. Thus, the decision can be found in  $O(n)$  calls to this oracle.

On the other hand, for ConfidenceBall<sub>2</sub>, the minimization problem can easily be seen as polynomial-time equivalent to the negative definite linearly constrained quadratic programming problem

$$\begin{aligned} & \text{minimize} && -\|\nu - \hat{\mu}_t\|_{2, A_t}^2 \\ & \text{subject to} && Mx \leq b \text{ and } \nu^\dagger x \geq C, \end{aligned}$$

where  $Mx \leq b$  is the system defining the decision set  $D$ , and  $C$  is a real parameter. Since Sahni [1974] proved that solving such programs is NP-hard, ConfidenceBall<sub>2</sub> may not be computationally practical for large  $n$ .

## 4 Concentration of Martingales

In our analysis, we use the following Bernstein-type concentration inequality for martingale differences, due to Freedman [1975] (see also [McDiarmid, 1998, Theorem 3.15]).

**Theorem 4 (Freedman)** *Suppose  $X_1, \dots, X_T$  is a martingale difference sequence, and  $b$  is an uniform upper bound on the steps  $X_i$ . Let  $V$  denote the sum of conditional variances,*

$$V = \sum_{i=1}^n \operatorname{Var}(X_i \mid X_1, \dots, X_{i-1}).$$

Then, for every  $a, v > 0$ ,

$$\operatorname{Prob}\left(\sum X_i \geq a \text{ and } V \leq v\right) \leq \exp\left(\frac{-a^2}{2v + 2ab/3}\right).$$

## 5 Upper Bound Analysis

Throughout the proof, without loss of generality, assume that the barycentric spanner is the standard basis  $\vec{e}_1 \dots \vec{e}_n$  (this just amounts to a choice of a coordinate system, where we identify the spanner with the standard basis). Hence, the decision set  $D$  is a subset of the cube  $[-1, 1]^n$ . In particular, this implies  $\|x\| \leq \sqrt{n}$  for all  $x \in D$ . This is really only a notational convenience; the problem is stated in terms of decisions in an abstract vector space, and expected costs in its dual, with no implicit standard basis.

In establishing the upper bounds there are two main theorems from which the upper bounds follow. The first is in showing that the confidence region is appropriate. Let  $E$  be the event that for every time  $t \leq T$ , the true mean  $\mu$  lies in the confidence region,  $B_t^2$  or  $B_t^1$ . The following shows that event  $E$  occurs with high probability. More precisely,

**Theorem 5 (Confidence)** *Let  $\delta > 0$ .*

- For ConfidenceBall<sub>2</sub>,

$$\operatorname{Prob}(\forall t, \mu \in B_t^2) \geq 1 - \delta.$$

- For ConfidenceBall<sub>1</sub>,

$$\operatorname{Prob}(\forall t, \mu \in B_t^1) \geq 1 - \delta.$$

Section 5.2 is devoted to establishing this confidence bound. In essence, the proof seeks to understand the growth of the quantity  $(\hat{\mu}_t - \mu)^\dagger A_t (\hat{\mu}_t - \mu)$ , which involves a rather technical construction of a martingale (using the matrix inversion

lemma) along with a careful application of Freedman’s inequality (Theorem 4).

The second main step in analyzing ConfidenceBall<sub>2</sub> is to show that as long as the aforementioned high-probability event holds, we have some control on the growth of the regret. The following bounds the sum of the squares of instantaneous regret.

**Theorem 6** (*Sum of Squares Regret Bound*) *Let*

$$r_t = \mu \cdot x_t - \mu \cdot x^*$$

*denote the instantaneous regret acquired by the algorithm on round  $t$ .*

- For ConfidenceBall<sub>2</sub>, if  $\mu \in B_t^2$  for all  $t \leq T$ , then

$$\sum_{t=1}^T r_t^2 \leq 8n\beta_T \ln T$$

- For ConfidenceBall<sub>1</sub>, if  $\mu \in B_t^1$  for all  $t \leq T$ , then

$$\sum_{t=1}^T r_t^2 \leq 8n^2\beta_T \ln T$$

This is proven in Section 5.1. The idea of the proof involves a potential function argument on the log volume (i.e. the log determinant) of the “precision matrix”  $A_t$  (which tracks how accurate our estimates of  $\mu$  are in each direction). The proof involves relating the growth of this volume to the regret.

At this point the proofs of Theorems 1 and 2 diverge. To show the former, we use the gap to bound the regret in terms of  $\sum_{t=1}^T r_t^2$ . For the latter, we simply appeal to the Cauchy-Schwarz inequality.

Using these two results we are able to prove our upper bounds as follows.

**Proof:**[Proof of Theorem 1] We only prove the result for ConfidenceBall<sub>2</sub>, as the proof for ConfidenceBall<sub>1</sub> is analogous. Let us analyze  $r_t = \mu \cdot x_t - \mu \cdot x^*$ , the regret on round  $t$ . Since ConfidenceBall<sub>2</sub> always chooses a decision from  $\mathcal{E}$ , either  $\mu \cdot x_t = \mu \cdot x^*$  or  $x_t \in \mathcal{E}_-$ , so that  $\mu \cdot x_t - \mu \cdot x^* \geq \Delta$ . Since  $\Delta > 0$  it follows that either  $r_t = 0$  or  $r_t/\Delta \geq 1$  and in either case,

$$r_t \leq \frac{r_t^2}{\Delta}$$

By Theorem 6, we see that if  $\mu \in B_t^2$ , then

$$\begin{aligned} R_T &= \sum_{t=1}^T r_t \\ &\leq \sum_{t=1}^T \frac{r_t^2}{\Delta} \\ &\leq \frac{8n\beta_T \ln T}{\Delta} \end{aligned}$$

Applying Theorem 5, we see that this occurs with probability at least  $1 - \delta$ , which completes the proof. ■

**Proof:**[Proof of Theorem 2] We only prove the result for ConfidenceBall<sub>2</sub>, as the proof for ConfidenceBall<sub>1</sub> is analogous. By Theorems 5 and 6, we know that with probability

at least  $1 - \delta$ ,  $\sum_{t=1}^T r_t^2 \leq 8n\beta_T \ln T$ . Applying the Cauchy-Schwarz inequality, we have, with probability at least  $1 - \delta$

$$\begin{aligned} R_T &= \sum_{t=1}^T r_t \\ &\leq \left( T \sum_{t=1}^T r_t^2 \right)^{1/2} \\ &\leq \sqrt{8nT\beta_T \ln T} \end{aligned}$$

which completes the proof. ■

We now provide the proofs of these two theorems.

### 5.1 Proof of Theorem 6

In this section, we prove Theorem 6, which says that the sum of the squares of the instantaneous regrets of the algorithm is small, assuming the evolving confidence balls always contain the true mean  $\mu$ . A key insight is that on any round  $t$  in which  $\mu \in B_t^2$ , the instantaneous regret is at most the “width” of the ellipsoid in the direction of the chosen decision. Moreover, the algorithm’s choice of decisions forces the ellipsoids to shrink at a rate that ensures that the sum of the widths is small. We now formalize this.

**Lemma 7** *Let  $x \in D$ . Then*

- For ConfidenceBall<sub>2</sub>, if  $\nu \in B_t^2$  and  $x \in D$ . Then

$$|(\nu - \hat{\mu}_t)^\dagger x| \leq \sqrt{\beta_t x^\dagger A_t^{-1} x}$$

- For ConfidenceBall<sub>1</sub>, if  $\nu \in B_t^1$  and  $x \in D$ . Then

$$|(\nu - \hat{\mu}_t)^\dagger x| \leq \sqrt{n\beta_t x^\dagger A_t^{-1} x}$$

**Proof:** Unless explicitly stated, all norms refer to the  $\ell_2$  norm. For ConfidenceBall<sub>2</sub>,

$$\begin{aligned} |(\nu - \hat{\mu}_t)^\dagger x| &= |(\nu - \hat{\mu}_t)^\dagger A_t^{1/2} A_t^{-1/2} x| \\ &= |(A_t^{1/2}(\nu - \hat{\mu}_t))^\dagger A_t^{-1/2} x| \\ &\leq \|A_t^{1/2}(\nu - \hat{\mu}_t)\| \|A_t^{-1/2} x\| \\ &\quad \dots \text{ by Cauchy-Schwarz} \\ &= \|A_t^{1/2}(\nu - \hat{\mu}_t)\| \sqrt{x^\dagger A_t^{-1} x} \\ &\leq \sqrt{\beta_t x^\dagger A_t^{-1} x} \end{aligned}$$

where the last inequality holds since  $\nu \in B_t^2$ .

For ConfidenceBall<sub>1</sub>,

$$\begin{aligned} |(\nu - \hat{\mu}_t)^\dagger x| &\leq \|A_t^{1/2}(\nu - \hat{\mu}_t)\|_1 \|A_t^{-1/2} x\|_\infty \\ &\quad \dots \text{ by Holder's Inequality} \\ &\leq \|A_t^{1/2}(\nu - \hat{\mu}_t)\|_1 \|A_t^{-1/2} x\|_2 \\ &\leq \sqrt{n\beta_t x^\dagger A_t^{-1} x} \end{aligned}$$

where the last inequality holds since  $\nu \in B_t^1$ . ■

Define

$$w_t := \sqrt{x_t^\dagger A_t^{-1} x_t}$$

which we interpret as the “normalized width” at time  $t$  in the direction of the chosen decision. The true width,  $2\sqrt{\beta_t} w_t$ , turns out to be an upper bound for the instantaneous regret.

**Lemma 8** Fix  $t$ .

- For ConfidenceBall<sub>2</sub>, if  $\mu \in B_t^2$ , then

$$r_t \leq 2 \min(\sqrt{\beta_t} w_t, 1)$$

- For ConfidenceBall<sub>1</sub>, if  $\mu \in B_t^1$ , then

$$r_t \leq 2 \min(\sqrt{n\beta_t} w_t, 1)$$

**Proof:** Let  $\tilde{\mu} \in B_t^2$  denote the vector which minimizes the dot product  $\tilde{\mu}^\dagger x_t$ . By choice of  $x_t$ , we have

$$\tilde{\mu}^\dagger x_t = \min_{\nu \in B_t^2} \min_{x \in D} \nu^\dagger x \leq \mu^\dagger x^*,$$

where the inequality used the hypothesis  $\mu \in B_t^2$ . Hence,

$$\begin{aligned} r_t &= \mu^\dagger x_t - \tilde{\mu}^\dagger x_t \\ &\leq (\mu - \tilde{\mu})^\dagger x_t \\ &= (\mu - \hat{\mu}_t)^\dagger x_t + (\hat{\mu}_t - \tilde{\mu})^\dagger x_t \\ &\leq 2\sqrt{\beta_t} w_t \end{aligned}$$

where the last step follows from Lemma 7 since  $\tilde{\mu}$  and  $\mu$  are in  $B_t^2$ . Since  $\ell_t \in [-1, 1]$ ,  $r_t$  is always at most 2 and the result follows. The proof for ConfidenceBall<sub>1</sub> is analogous. ■

Next we show that the sum of the squares of the widths does not grow too fast.

**Lemma 9** We have for all  $t$

$$\sum_{\tau=1}^t \min(w_\tau^2, 1) \leq 2n \ln t.$$

The following two facts prove useful to this end.

**Lemma 10** For every  $t \leq T$ ,

$$\det A_{t+1} = \prod_{\tau=1}^t (1 + w_\tau^2).$$

**Proof:** By the definition of  $A_{t+1}$ , we have

$$\begin{aligned} \det A_{t+1} &= \det(A_t + x_t x_t^\dagger) \\ &= \det(A_t^{1/2} (I + A_t^{-1/2} x_t x_t^\dagger A_t^{-1/2}) A_t^{1/2}) \\ &= \det(A_t) \det(I + A_t^{-1/2} x_t (A_t^{-1/2} x_t)^\dagger) \\ &= \det(A_t) \det(I + v_t v_t^\dagger), \end{aligned}$$

where  $v_t := A_t^{-1/2} x_t$ . Now observe that  $v_t^\dagger v_t = w_t^2$  and

$$(I + v_t v_t^\dagger) v_t = v_t + v_t (v_t^\dagger v_t) = (1 + w_t^2) v_t$$

Hence  $(1 + w_t^2)$  is an eigenvalue of  $I + v_t v_t^\dagger$ . Since  $v_t v_t^\dagger$  is a rank one matrix, all the other eigenvalues of  $I + v_t v_t^\dagger$  equal 1. It follows that  $\det(I + v_t v_t^\dagger)$  is  $(1 + w_t^2)$ , and so

$$\det A_{t+1} = (1 + w_t^2) \det A_t.$$

Recalling that  $A_1$  is the identity matrix, the result follows by induction. ■

**Lemma 11** For all  $t$ ,  $\det A_t \leq t^n$ .

**Proof:** The rank one matrix  $x_t x_t^\dagger$  has  $x_t^\dagger x_t = \|x_t\|^2$  as its unique non-zero eigenvalue. Also, since we have identified the spanner with the standard basis, we have  $\sum_{i=1}^n b_i b_i^\dagger = I$ . Since the trace is a linear operator, it follows that

$$\begin{aligned} \text{trace } A_t &= \text{trace} \left( I + \sum_{\tau < t} x_\tau x_\tau^\dagger \right) \\ &= n + \sum_{\tau < t} \text{trace}(x_\tau x_\tau^\dagger) \\ &= n + \sum_{\tau < t} \|x_\tau\|^2 \\ &\leq nt. \end{aligned}$$

Now, recall that  $\text{trace } A_t$  equals the sum of the eigenvalues of  $A_t$ . On the other hand,  $\det(A_t)$  equals the product of the eigenvalues. Since  $A_t$  is positive definite, its eigenvalues are all positive. Subject to these constraints,  $\det(A_t)$  is maximized when all the eigenvalues are equal; the desired bound follows. ■

**Proof:**[Proof of Lemma 9]

Using the fact that for  $0 \leq y \leq 1$ ,  $\ln(1 + y) \geq y/2$ , we have

$$\begin{aligned} \sum_{\tau=1}^t \min(w_\tau^2, 1) &\leq \sum_{\tau=1}^t 2 \ln(1 + w_\tau^2) \\ &= 2 \ln(\det A_{t+1}) \\ &\leq 2n \ln t \end{aligned}$$

by Lemmas 10 and 11 ■

Finally, we are ready to prove that if  $\mu$  always stays within the evolving confidence region, then our regret is under control.

**Proof:**[Proof of Theorem 6] Assume that  $\mu \in B_t^2$  for all  $t$ . Then

$$\begin{aligned} \sum_{t=1}^T r_t^2 &\leq \sum_{t=1}^T 4\beta_t \min(w_t^2, 1) && \text{by Lemma 8} \\ &\leq 4\beta_T \sum_{t=1}^T \min(w_t^2, 1) && \text{since } 1 < \beta_1 < \dots < \beta_T \\ &\leq 8\beta_T n \ln T && \text{by Lemma 9.} \end{aligned}$$

The proof for Confidenceball<sub>1</sub> is analogous. ■

## 5.2 Proof of Theorem 5

In this section, we prove Theorem 5, which states that with high probability, for all  $t$ , the true mean  $\mu$  lies in the confidence ball  $B_t$ .

Recall that

$$\eta_t := c_t(x_t) - \mu^\dagger x_t = \ell_t - \mathbb{E}(\ell_t \mid \mathcal{H}_t)$$

where  $\mathcal{H}_t$  denotes the complete history of the game on rounds  $1, \dots, t-1$ , that is, the  $\sigma$ -algebra generated by  $\ell_1, \dots, \ell_{t-1}$ .

For either algorithm, we will analyze the quantity:

$$Z_t := (\hat{\mu}_t - \mu)^\dagger A_t (\hat{\mu}_t - \mu)$$

which measures the error of  $\widehat{\mu}_t$  as an approximation to the true mean,  $\mu$ , under the norm induced by  $A_t$ .

We will show that, with probability greater than  $1 - \delta$ ,  $Z_t \leq \beta_t$  for all  $t$  for either algorithm. For ConfidenceBall<sub>2</sub>, this directly implies that  $\mu \in B_t^2$ . For ConfidenceBall<sub>1</sub>, note that

$$\|A_t^{1/2}(\widehat{\mu}_t - \mu)\|_1 \leq \sqrt{n} \|A_t^{1/2}(\widehat{\mu}_t - \mu)\|_2 = \sqrt{nZ_t}$$

so if  $Z_t \leq \beta_t$  then  $\mu \in B_t^1$ .

The next lemma bounds the growth of  $Z_t$ .

**Lemma 12** For all  $t$ ,

$$Z_t \leq n + 2 \sum_{\tau=1}^{t-1} \eta_\tau \frac{x_\tau^\dagger (\widehat{\mu}_\tau - \mu)}{1 + w_\tau^2} + \sum_{\tau=1}^{t-1} \eta_\tau^2 \frac{w_\tau^2}{1 + w_\tau^2}.$$

**Proof:** For notational convenience, define:

$$Y_t = A_t(\widehat{\mu}_t - \mu)$$

We have the following relations:

$$Z_t = Y_t^\dagger A_t^{-1} Y_t$$

$$Y_t = \sum_{\tau < t} \eta_\tau x_\tau - \mu$$

$$Y_{t+1} = Y_t + \eta_t x_t$$

which are immediate from the definitions of  $A_t$ ,  $\widehat{\mu}_t$ , and  $\eta_t$ .

Now examining the growth of  $Z_t$ , we have:

$$\begin{aligned} Z_{t+1} &= Y_{t+1}^\dagger A_{t+1}^{-1} Y_{t+1} \\ &= (Y_t + \eta_t x_t)^\dagger A_{t+1}^{-1} (Y_t + \eta_t x_t) \\ &= Y_t^\dagger A_{t+1}^{-1} Y_t + 2\eta_t x_t^\dagger A_{t+1}^{-1} Y_t + \eta_t^2 x_t^\dagger A_{t+1}^{-1} x_t \quad (1) \end{aligned}$$

Applying the matrix inversion lemma to  $A_{t+1}^{-1}$ , we note that:

$$\begin{aligned} A_{t+1}^{-1} &= (A_t + x_t x_t^\dagger)^{-1} \\ &= A_t^{-1} - \frac{A_t^{-1} x_t x_t^\dagger A_t^{-1}}{1 + x_t^\dagger A_t^{-1} x_t} \\ &= A_t^{-1} - \frac{A_t^{-1} x_t x_t^\dagger A_t^{-1}}{1 + w_t^2} \end{aligned}$$

We can use this to bound the three terms of (1) as follows. For the first term,

$$\begin{aligned} Y_t^\dagger A_{t+1}^{-1} Y_t &= Y_t^\dagger A_t^{-1} Y_t - \frac{(Y_t^\dagger A_t^{-1} x_t)^2}{1 + w_t^2} \\ &\leq Z_t. \end{aligned}$$

For the second term,

$$\begin{aligned} 2\eta_t x_t^\dagger A_{t+1}^{-1} Y_t &= 2\eta_t x_t^\dagger A_t^{-1} Y_t - 2\eta_t \frac{x_t^\dagger A_t^{-1} x_t x_t^\dagger A_t^{-1} Y_t}{1 + w_t^2} \\ &= 2\eta_t x_t^\dagger A_t^{-1} Y_t - 2\eta_t \frac{w_t^2 x_t^\dagger A_t^{-1} Y_t}{1 + w_t^2} \\ &= 2\eta_t \frac{x_t^\dagger A_t^{-1} Y_t}{1 + w_t^2} \\ &= 2\eta_t \frac{x_t^\dagger (\widehat{\mu}_t - \mu)}{1 + w_t^2} \end{aligned}$$

For the third term,

$$\begin{aligned} \eta_t^2 x_t^\dagger A_{t+1}^{-1} x_t &= \eta_t^2 w_t^2 - \eta_t^2 \frac{w_t^4}{1 + w_t^2} \\ &= \eta_t^2 \frac{w_t^2}{1 + w_t^2} \end{aligned}$$

Putting these together, we have shown

$$Z_{t+1} \leq Z_t + 2\eta_t \frac{x_t^\dagger (\widehat{\mu}_t - \mu)}{1 + w_t^2} + \eta_t^2 \frac{w_t^2}{1 + w_t^2}.$$

By induction, it follows that

$$Z_t \leq Z_1 + 2 \sum_{\tau=1}^{t-1} \eta_\tau \frac{x_\tau^\dagger (\widehat{\mu}_\tau - \mu)}{1 + w_\tau^2} + \sum_{\tau=1}^{t-1} \eta_\tau^2 \frac{w_\tau^2}{1 + w_\tau^2}.$$

Finally, we check that  $Z_1 \leq n$ . To see this, recall from the algorithm that  $A_1 = I$  and  $\widehat{\mu}_1 = 0$ . Also, since  $e_1, \dots, e_n \in D$ , by assumption,  $\mu \cdot e_j \in [-1, 1]$ .

$$\begin{aligned} Z_1 &= (\widehat{\mu}_1 - \mu)^\dagger A_1 (\widehat{\mu}_1 - \mu) \\ &= \|\mu\|^2 \\ &= \sum_{j=1}^n (\mu^\dagger e_j)^2 \\ &\leq n. \end{aligned}$$

This completes the proof.  $\blacksquare$

We now define a useful martingale difference sequence. First, it is convenient to define an ‘‘escape event’’  $E_t$  as:

$$E_t = \mathbf{I}\{Z_\tau \leq \beta_\tau \text{ for all } \tau \leq t\} = \mathbf{I}\{\mu \in B_\tau \text{ for all } \tau \leq t\}$$

where  $\mathbf{I}\{\cdot\}$  is the indicator function.

**Lemma 13** Define a random variable  $M_t$  by

$$M_t = 2\eta_t E_t \frac{x_t^\dagger (\widehat{\mu}_t - \mu)}{1 + w_t^2}.$$

Then  $M_t$  is a martingale difference sequence with respect to the sequence of game histories  $\mathcal{H}_t$ .

**Proof:** To see that  $M_t$  is a martingale difference sequence, note that:

$$\begin{aligned} \mathbb{E}(M_t \mid \mathcal{H}_t) &= 2E_t \frac{x_t^\dagger (\widehat{\mu}_t - \mu)}{1 + w_t^2} \mathbb{E}(\eta_t \mid \mathcal{H}_t) \\ &= 0 \end{aligned}$$

since the history  $\mathcal{H}_t$  fully determines  $x_1, \dots, x_t, \widehat{\mu}_1, \dots, \widehat{\mu}_t, Z_1, \dots, Z_t$ , and  $E_1, \dots, E_t$ , and since the noise functions  $\eta_t$  are a martingale difference sequence with respect to  $\mathcal{H}_t$ .  $\blacksquare$

We show that with high probability, the associated martingale,  $\sum_{\tau=1}^t M_\tau$ , never grows too large.

**Lemma 14** Given  $\delta < 1$ ,

$$\text{Prob} \left( \forall t, \sum_{\tau=1}^{t-1} M_\tau \leq \beta_t/2 \right) \geq 1 - \delta,$$

We defer the proof to Section 5.2.1. Equipped with this lemma, we can prove Theorem 5.

**Proof:**[Proof of Theorem 5] It suffices to show that the high-probability event described in Lemma 14 is contained in the support of  $E_t$  for every  $t$ . We prove the latter by induction on  $t$ .

By Lemma 12 and the definition of  $\beta_1$ , we know that  $Z_1 \leq n < \beta_1$ . Hence  $E_1$  is always 1 (equivalently,  $\mu$  is always in  $B_1$ ).

Now suppose the high-probability event of Lemma 14 holds, so in particular,

$$\sum_{\tau=1}^{t-1} M_\tau \leq \beta_t/2.$$

By inductive hypothesis,  $E_\tau = 1$  for  $\tau \leq t-1$ . Hence by Lemma 12 we have

$$\begin{aligned} Z_t &\leq n + 2 \sum_{\tau=1}^{t-1} \eta_\tau \frac{x_\tau^\dagger (\hat{\mu}_\tau - \mu)}{1 + w_\tau^2} + \sum_{\tau=1}^{t-1} \eta_\tau^2 \frac{w_\tau^2}{1 + w_\tau^2} \\ &= n + \sum_{\tau=1}^{t-1} M_\tau + \sum_{\tau=1}^{t-1} \eta_\tau^2 \frac{w_\tau^2}{1 + w_\tau^2} \\ &\leq n + \beta_t/2 + \sum_{\tau=1}^{t-1} \eta_\tau^2 \frac{w_\tau^2}{1 + w_\tau^2} \\ &\leq n + \beta_t/2 + \sum_{\tau=1}^{t-1} \min(w_\tau^2, 1) \quad \text{since } |\eta_\tau| \leq 1 \\ &\leq n + \beta_t/2 + 2n \ln t \quad \text{by Lemma 9} \\ &\leq \beta_t. \end{aligned}$$

Thus we have shown  $E_t = 1$ , completing the induction. ■

### 5.2.1 Concentration

All that remains to complete the proof now is to show that our martingale  $\sum_1^t M_\tau$  has good concentration properties. As we show, the step sizes  $|M_t|$  are uniformly bounded so that an application of the Hoeffding-Azuma inequality would bound the probability that  $\sum_1^t M_\tau$  grows too large. Unfortunately, the bound thus obtained translates into a regret bound of  $T^{3/4}$ , which is not good enough for our purpose.

Instead we use Theorem 4, which allows us to bound the step sizes in terms of random variables, as long as the conditional variances remain under control.

**Proof:**[Proof of Lemma 14] Let us first obtain upper bounds on the step sizes of our martingale.

$$\begin{aligned} |M_t| &= 2|\eta_t| E_t \frac{|x_t^\dagger (\hat{\mu}_t - \mu)|}{1 + w_t^2} \\ &\leq 2|\eta_t| E_t \frac{\sqrt{\beta_t x_t^\dagger A_t^{-1} x_t}}{1 + w_t^2} \\ &= 2|\eta_t| E_t \frac{w_t \sqrt{\beta_t}}{1 + w_t^2} \\ &\leq 2|\eta_t| E_t \sqrt{\beta_t} \min(w_t, 1/2) \end{aligned} \quad (2)$$

where the first inequality follows trivially when  $E_t = 0$ , and by Lemma 7 when  $E_t = 1$ . Additionally this gives a family of uniform upper bounds:

$$|M_\tau| \leq \sqrt{\beta_t} \text{ for all } \tau \leq t$$

since  $|\eta_t| \leq 1$  and (by choice)  $\beta_\tau$  is a non-decreasing sequence.

Next we bound the sum of the conditional variances of our martingale. Note that  $(\min(w_t, 1/2))^2 = \min(w_t^2, 1/4)$

$$\begin{aligned} V_t &:= \sum_{\tau=1}^t \text{Var}(M_\tau | M_1 \dots M_{\tau-1}) \\ &\leq \sum_{\tau=1}^t 4|\eta_\tau|^2 E_\tau \beta_\tau \min(w_\tau^2, 1/4) \quad \text{by (2)} \\ &\leq 4(\max_{\tau \leq t} \beta_\tau) \sum_{\tau=1}^t E_\tau \min(w_\tau^2, 1) \quad \text{since } |\eta_\tau| \leq 1 \\ &\leq 4\beta_t \sum_{\tau \leq t} E_\tau \min(w_\tau^2, 1) \\ &\leq 8\beta_t n \ln(\max\{\tau \leq t : E_\tau = 1\}) \quad \text{by Lemma 9} \\ &\leq 8\beta_t n \ln t \end{aligned}$$

Since we have established that the sum of conditional variances,  $V_t$ , is *always* bounded by  $8\beta_t n \ln t$ , we can apply Theorem 4 with parameters  $a = \beta_t/2$ ,  $b = \sqrt{\beta_t}$  and  $v = 8n\beta_t \ln t$ , to get

$$\begin{aligned} &\text{Prob} \left( \sum_{\tau=1}^{t-1} M_\tau \geq \beta_t/2 \right) \\ &= \text{Prob} \left( \sum_{\tau=1}^{t-1} M_\tau \geq \beta_t/2 \text{ and } V_t \leq 8n\beta_t \ln t \right) \\ &\leq \exp \left( \frac{-(\beta_t/2)^2}{2(8n\beta_t \ln t) + \frac{2}{3}(\beta_t/2)(\sqrt{\beta_t})} \right) \\ &= \exp \left( \frac{-\beta_t}{64n \ln t + \frac{4}{3}\sqrt{\beta_t}} \right) \\ &\leq \max \left\{ \exp \left( \frac{-\beta_t}{128n \ln t} \right), \exp \left( \frac{-3\sqrt{\beta_t}}{8} \right) \right\} \\ &\leq \frac{\delta}{t^2} \end{aligned}$$

where the last inequality follows from the definition of  $\beta_t$ . Finally, we apply a union bound to get

$$\begin{aligned} &\text{Prob} \left( \sum_{\tau=1}^{t-1} M_\tau \geq \frac{\beta_t}{2} \text{ for some } t \right) \\ &\leq \sum_{t=1}^{\infty} \text{Prob} \left( \sum_{\tau=1}^{t-1} M_\tau \geq \frac{\beta_t}{2} \right) \\ &\leq \sum_{t=2}^{\infty} \frac{\delta}{t^2} \\ &\leq \delta \left( \frac{\pi^2}{6} - 1 \right) \\ &\leq \delta \end{aligned}$$

completing the proof of Lemma 14. ■

## 6 Lower Bound Analysis

We start with the 2-dimensional case. The extension to the general case is provided in the next Subsection 6.2.

### 6.1 $n = 2$ case

Assume  $n = 2$ . Recall from Section 3.3 that in the  $n = 2$  case our decision set  $D$  is the unit circle.

Let us condition on the event that  $\mu \in \{\mu_1, \mu_2\}$ , where  $\mu_1, \mu_2 \in \mathbb{R}^2$  such that  $\|\mu_1\| = \|\mu_2\| = 1/2$  and  $\|\mu_1 - \mu_2\| = \varepsilon$ .

Note that  $\mu$  is uniform over  $\{\mu_1, \mu_2\}$  in this event. We show that, even conditioned on this additional information, the expected regret is  $\Omega(\sqrt{T})$ . The conclusion of Theorem 3 then follows by an averaging argument.

Let

$$b_t := \Pr(\mu = \mu_1 \mid \mathcal{H}_t) - \Pr(\mu = \mu_2 \mid \mathcal{H}_t)$$

be the bias towards  $\mu_1$  at time  $t$ . Note that  $b_0 = 0$ , and that the sequence  $(b_t)$  is a martingale with respect to  $(\mathcal{H}_t)$ . Our next Lemma, whose proof is somewhat technical, gives a lower bound on regret in terms of the martingale differences  $b_{t+1} - b_t$ .

**Lemma 15** *For all  $\varepsilon > 0$  and  $t \geq 1$ , for any sequence of decisions  $x_1, \dots, x_t$  and outcomes  $\ell_1, \dots, \ell_{t-1}$ , the regret from round  $t$  satisfies*

$$\mathbb{E}_{\mu}(r_t \mid \mathcal{H}_t) \geq \frac{1}{16} \left( \varepsilon^2 + \frac{|b_{t+1} - b_t|^2}{\varepsilon^2} \right) \mathbf{1}\{|b_t| \leq 1/2\}$$

**Proof:** Let  $v_1$  be the unit vector in the direction of  $\mu_1 - \mu_2$ , and let  $v_2$  be the unit vector in the direction of  $\mu_1 + \mu_2$ . Note that  $v_1, v_2$  is an orthonormal basis for the plane. Decompose  $x_t = \alpha v_1 + \beta v_2$ , and  $\mathbb{E}(\mu \mid \mathcal{H}_t) = \gamma v_1 + \delta v_2$ . Since  $\mathbb{E}(\mu \mid \mathcal{H}_t) = \frac{\mu_1 + \mu_2}{2} + b_t \frac{\mu_1 - \mu_2}{2}$ , we have  $\gamma = \varepsilon b_t / 2$  and  $\delta = \frac{\sqrt{1 - \varepsilon^2}}{2}$ .

Let  $p = \Pr(\mu = \mu_1 \mid \mathcal{H}_t)$ . Then  $b_t = 2p - 1$ . Observe that

$$\begin{aligned} b_{t+1} - b_t &= \frac{p(1 + \ell_t \mu_1^\dagger x_t) - (1-p)(1 + \ell_t \mu_2^\dagger x_t)}{p(1 + \ell_t \mu_1^\dagger x_t) + (1-p)(1 + \ell_t \mu_2^\dagger x_t)} - 2p + 1 \\ &= \frac{(2p-1) + p\ell_t \mu_1^\dagger x_t - (1-p)\ell_t \mu_2^\dagger x_t}{1 + p\ell_t \mu_1^\dagger x_t + (1-p)\ell_t \mu_2^\dagger x_t} - 2p + 1 \\ &= \frac{2p(1-p)\ell_t \mu_1^\dagger x_t - 2p(1-p)\ell_t \mu_2^\dagger x_t}{1 + p\ell_t \mu_1^\dagger x_t + (1-p)\ell_t \mu_2^\dagger x_t} \\ &= \frac{2p(1-p)\ell_t (\mu_1 - \mu_2)^\dagger x_t}{1 + p\ell_t \mu_1^\dagger x_t + (1-p)\ell_t \mu_2^\dagger x_t} \end{aligned}$$

Since  $|\mu_i^\dagger x_t| \leq 1/2$ , the denominator of the above expression is at least  $1/2$ . Since  $p(1-p) \leq 1/4$ , it follows that

$$|b_{t+1} - b_t| \leq |(\mu_1 - \mu_2)^\dagger x_t| = \varepsilon |\alpha|. \quad (3)$$

Assume the game history is such that  $|b_t| \leq 1/2$ . Otherwise, since the regret is non-negative, there is nothing to

prove. Now we calculate

$$\begin{aligned} \mathbb{E}_{\mu}(r_t \mid \mathcal{H}_t) &= \frac{1}{2} + x_t \cdot \mathbb{E}(\mu \mid \mathcal{H}_t) \\ &= \frac{1}{2} + \alpha\gamma + \beta\delta \\ &= \frac{1}{2}(1 + \alpha\varepsilon b_t + \beta\sqrt{1 - \varepsilon^2}) \\ &\geq \frac{1}{2} \left( 1 + \alpha\varepsilon b_t + \left( \frac{\alpha^2}{2} - 1 \right) \sqrt{1 - \varepsilon^2} \right) \quad (4) \\ &\geq \frac{1}{2} \left( 1 + \alpha\varepsilon b_t + \left( \frac{\alpha^2}{2} - 1 \right) \left( 1 - \frac{\varepsilon^2}{2} \right) \right) \\ &= \frac{1}{16}(\alpha^2 + \varepsilon^2) + \frac{1}{8}(\alpha^2 + 4b_t\alpha\varepsilon + \varepsilon^2) \\ &\quad + \frac{1}{16}(\alpha^2 + \varepsilon^2 - 2\alpha^2\varepsilon^2) \\ &\geq \frac{1}{16}(\alpha^2 + \varepsilon^2) \quad (5) \\ &\geq \frac{1}{16} \left( \varepsilon^2 + \frac{|b_{t+1} - b_t|^2}{\varepsilon^2} \right) \quad (6) \end{aligned}$$

Here (4) follows because  $\alpha^2 + \beta^2 = 1$  implies that  $1 + \beta \geq \alpha^2/2$ , with equality iff  $\beta = -1$ . Inequality (5) follows because  $|b_t| \leq 1/2$  and  $|\alpha|, |\varepsilon| \leq 1$ . Inequality (6) follows from (3), which completes the proof.  $\blacksquare$

We are now ready to prove Theorem 3 in the  $n = 2$  case. We generalize the argument to  $n$ -dimensions in Section 6.2.

**Proof:**[Proof of Theorem 3 for  $n = 2$ ] Let  $\varepsilon = T^{-1/4}$ . First, observe that, by Fubini's theorem and linearity of expectation,

$$\begin{aligned} \mathbb{E} R &= \sum_{t=1}^T \mathbb{E}_{\mathcal{H}_t} \mathbb{E}_{\mu} (r_t \mid \mathcal{H}_t) \\ &\geq \frac{1}{16} \sum_{t=1}^T \mathbb{E}_{\mathcal{H}_t} \left( \left( \varepsilon^2 + \frac{|b_{t+1} - b_t|^2}{\varepsilon^2} \right) \mathbf{1}\{|b_t| \leq 1/2\} \right) \\ &\quad \dots \text{by Lemma 15} \\ &\geq \frac{1}{16} \varepsilon^2 T \text{Prob}(\text{for all } t, |b_t| \leq 1/2) \\ &\quad + \frac{1}{16} \sum_{t=1}^T \mathbb{E}_{\mathcal{H}_t} \left( \frac{|b_{t+1} - b_t|^2}{\varepsilon^2} \mathbf{1}\{|b_t| \leq 1/2\} \right) \\ &= \frac{\sqrt{T}}{16} \left( \text{Prob}(\text{for all } t, |b_t| \leq 1/2) \right. \\ &\quad \left. + \sum_{t=1}^T \mathbb{E}_{\mathcal{H}_t} (|b_{t+1} - b_t|^2 \mathbf{1}\{|b_t| \leq 1/2\}) \right) \end{aligned}$$

Thus, if  $\text{Prob}(\text{for all } t \leq T |b_t| \leq 1/2) \geq 1/2 - 1/e$ , then we are done by the first term on the right-hand side. Otherwise, with probability at least  $1/2 + 1/e$ , there exists  $t \leq T$  such that  $|b_t| \geq 1/2$ . By Freedman's Bernstein-type inequality for martingales (Theorem 4) applied to the martingale

$b_{t \wedge \sigma}$ , where  $\sigma = \min\{\tau : |b_\tau| \geq 1/2\}$ , we have

$$\begin{aligned} \text{Prob} \left( (\exists t \leq T) |b_t| \geq \frac{1}{2} \text{ and } V \leq \frac{1}{32} \right) \\ \leq 2 \exp \left( \frac{-1/4}{1/8 + \varepsilon/3} \right) \leq \frac{2}{e^2} < \frac{1}{e} \end{aligned}$$

where

$$V = \sum_{t=1}^T \mathbf{1}\{\forall \tau \leq t, |b_\tau| \leq 1/2\} \mathbb{E} (|b_{t+1} - b_t|^2 \mid \mathcal{H}_t).$$

It follows that

$$\text{Prob} \left( V > \frac{1}{32} \right) \geq 1/2.$$

In particular,

$$\sum_{t=1}^T \mathbb{E}_{\mathcal{H}_t} (|b_{t+1} - b_t|^2 \mathbf{1}\{|b_t| \leq 1/2\}) \geq \mathbb{E} V \geq \frac{1}{64}.$$

completing the proof.  $\blacksquare$

## 6.2 General Case

Now suppose  $n > 2$  is even. Fix an index  $1 \leq i \leq n/2$ , and consider the contribution to the total expected regret from the choice of  $(x_{2i-1}, x_{2i})$ , i.e., the component from the  $i$ 'th circle.

Analogously to the 2-dimensional case, we condition on the  $i$ 'th component of  $\mu$  being one of two vectors,  $\nu_1, \nu_2 \in S^1/n$ . We further condition on the exact values of the other  $n/2 - 1$  components of  $\mu$ . We denote  $\varepsilon = \|\nu_1 - \nu_2\|$

Let  $b_t$  denote the bias toward  $\nu_1$ , given the history  $\mathcal{H}_t$  of the game on rounds  $1, \dots, t - 1$ . That is,

$$b_t = \Pr(\mu_i = \nu_1 \mid \mathcal{H}_t) - \Pr(\mu_i = \nu_2 \mid \mathcal{H}_t)$$

Then we have the following analog of Lemma 15.

**Lemma 16** *For all  $t$ , for any sequence of decisions  $x_1, \dots, x_t$  and outcomes  $\ell_1, \dots, \ell_{t-1}$ , the regret from round  $t$  due to the  $i$ th component of  $x_t$  satisfies*

$$\mathbb{E}_{\mu} (r_t^{(i)} \mid \mathcal{H}_t) \geq \frac{1}{64} \left( \varepsilon^2 + \frac{|b_{t+1} - b_t|^2}{\varepsilon^2} \right) \mathbf{1}\{|b_t| \leq 1/2\}$$

It follows along the same lines as before that the expected total regret from the  $i$ th component is  $\Omega(\sqrt{T})$ . Summing over the  $n/2$  possible values of  $i$  completes the proof.

## 7 Extension: time-varying decision sets

Our techniques also apply to the setting when only a subset of the full decision set  $D$  is available in each round. Suppose, at time  $t$ , only a subset of decisions  $D_t \subset D$  are available. In this case, the correct notion of regret is to compare each chosen decision  $x_t$ , not with the global optimum  $x^*$ , but with the best choice from the available subset  $D_t$ . Thus

$$R_T = \sum_{t=1}^T (\mu^\dagger x_t - \mu^\dagger x_t^*)$$

where  $x_t^* \in D_t$  is an optimal decision for  $\mu$ , i.e.,

$$x_t^* \in \operatorname{argmin}_{x \in D_t} \mu^\dagger x$$

The only change that needs to be made to our algorithm is that now  $x_t$  is chosen from  $D_t$  instead of  $D$ .

With these changes in definitions, all of our numbered Theorems and Lemmas still hold, with  $D$  replaced by  $D_t$  and  $x^*$  replaced by  $x_t^*$  where they appear. (This is trivial in the case of the lower bounds.) The changes to the proofs are minimal.

We note that a very similar model was considered by Abe and Long [1999], who proved a lower bound of  $\Omega(T^{3/4})$  in their setting. However, this does not contradict our results, because their lower bound requires the dimension  $n$  to be a function of  $T$ .

## References

- Naoki Abe and Philip M. Long. Associative reinforcement learning using linear probabilistic concepts. In *Proc. 16th International Conf. on Machine Learning*, pages 3–11. Morgan Kaufmann, San Francisco, CA, 1999.
- R. Agrawal. Sample mean based index policies with  $O(\log n)$  regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27:1054–1078, 1995.
- P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3): 235–256, 2002. ISSN 0885-6125.
- Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, 2002. ISSN 1533-7928.
- B. Awerbuch and R. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC)*, 2004.
- Donald A. Berry and Bert Fristedt. *Bandit Problems: Sequential Allocation of Experiments*. Springer, October 1985.
- V. Dani, T. P. Hayes, and S. M. Kakade. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, 2008. To appear. Available online at <http://books.nips.cc/>.
- David A. Freedman. On tail probabilities for martingales. *The Annals of Probability*, 3(1):100–118, Feb. 1975.
- T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4, 1985.
- Colin McDiarmid. *Concentration. In Probabilistic Methods for Algorithmic Discrete Mathematics*. Springer, 1998.
- H. Robbins. Some aspects of the sequential design of experiments. In *Bulletin of the American Mathematical Society*, volume 55, 1952.
- Sartaj Sahni. Computationally related problems. *SIAM J. Comput.*, 3(4):262–279, 1974.

---

# Model Selection and Stability in $k$ -means Clustering

---

Ohad Shamir<sup>†</sup> and Naftali Tishby<sup>†‡</sup>

<sup>†</sup> School of Computer Science and Engineering

<sup>‡</sup> Interdisciplinary Center for Neural Computation

The Hebrew University, Jerusalem 91904, Israel

{ohadsh, tishby}@cs.huji.ac.il

## Abstract

Clustering Stability methods are a family of widely used model selection techniques applied in data clustering. Their unifying theme is that an appropriate model should result in a clustering which is robust with respect to various kinds of perturbations. Despite their relative success, not much is known theoretically on why or when do they work, or even what kind of assumptions they make in choosing an 'appropriate' model. Moreover, recent theoretical work has shown that they might 'break down' for large enough samples. In this paper, we focus on the behavior of clustering stability using  $k$ -means clustering. Our main technical result is an exact characterization of the distribution to which suitably scaled measures of instability converge, based on a sample drawn from any distribution in  $\mathbb{R}^n$  satisfying mild regularity conditions. From this, we can show that clustering stability does not 'break down' even for arbitrarily large samples, in the  $k$ -means framework that we study. Moreover, it allows us to identify the factors which influence the behavior of clustering stability for any sample size. This leads to some interesting preliminary observations about what kind of assumptions are made when using these methods. While often reasonable, these assumptions might also lead to unexpected consequences.

## 1 Introduction

The important and difficult problem of model selection in data clustering has been the focus of an extensive literature spanning several research communities in the natural and social sciences. Since clustering is often used as a first step in the data analysis process, the questions of what type of clusters or how many clusters are in the data can be crucial.

An important family of model selection methods, whose popularity has grown in the past few years, is based on clustering stability. The unifying theme of these methods is that an appropriate model for the data should result in a clustering which is robust with respect to various kinds of perturbations. In other words, if we choose an appropriate clustering algorithm, and feed it with the 'correct' parameters (such as

the number of clusters, the metric used, etc.), the clustering returned by the algorithm should not be overly sensitive to the exact structure of the data.

In particular, we will focus on clustering stability methods which compare the discrepancy or 'distance' between clusterings of different random subsets of our data. These methods seek a 'stable' model, in the sense that the value of such distance measures should tend to be small.

Although these methods have been shown to be rather effective in practice (cf. [2],[4],[7],[9]), little theory exists so far to explain their success, or for which cases are they best suited for. Over the past few years, a theoretical study of these methods has been initiated, in a framework where the data are assumed to be an i.i.d sample. However, a fundamental hurdle was the observation [1] that under mild conditions and for any model choice, the clustering algorithm should tend to converge to a single solution which is optimal with respect to the underlying distribution. As a result, clustering stability might 'break down' for large enough samples, since we get approximately the same clustering hypothesis based on each random subsample, and thus achieve stability regardless of whether the model fits the data or not (this problem was also pointed out in [6]). A possible solution to this difficulty was proposed in [15]. In a nutshell, that paper showed that the important factor in the way these clustering stability methods work may not be the asymptotic stability of the model, but rather *how fast exactly does it converge to this stability*. With this more refined analysis, it was argued that differences in the stability of different models should usually be discernible for any sample size, no matter how large, despite the universal convergence to absolute stability. Although it provided the necessary groundwork, that paper only rigorously proved this assertion for a single toy example, as a proof-of-concept.

In this paper, we formally investigate the application of clustering stability to the well known and popular  $k$ -means clustering framework, when the goal is to determine the value of  $k$ , or the number of clusters in the data. Assuming an algorithm which minimizes the  $k$ -means objective function, we consider arbitrary distributions in  $\mathbb{R}^n$  satisfying certain mild regularity conditions, and analyze the behavior of the clustering distance measure, scaled by the square root of the sample size. Rather than converging to zero in probability as the sample size increases to infinity, this scaled measure converges to a non-degenerate distribution which depends on the choice of  $k$ . From this we can show that clustering stabil-

ity does not 'break down' even for arbitrarily large samples, in the sense described earlier, at least for the  $k$ -means framework that we study.

The asymptotic distribution is also interesting for two additional reasons. The first is that it can be seen as an approximation which improves as the sample size increases. The second and more profound reason is that if we are interested in discovering what fundamental assumptions are implicit in performing model selection with clustering stability, these should not be overly dependent on the sample size used. Therefore, as we look at larger samples, noisy and hard to analyze finite sample effects diminish, and what remains are the fundamental characteristics, which should be relevant for *any* sample size. As a result, the analysis leads to some preliminary observations about the factors influencing clustering stability in  $k$ -means, of both theoretical and practical interest.

## 2 Problem Setting and Notation

We refer the reader to Fig. 1 for a graphical illustration of the basic setting, and some of the notation introduced below.

Denote  $\{1, \dots, k\}$  as  $[k]$ . Vectors will be denoted by bold-face characters.  $\|\cdot\|$  will denote the Euclidean norm unless stated otherwise.  $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$  denotes the multivariate normal distribution with mean  $\boldsymbol{\mu}$  and covariance matrix  $\Sigma$ .

We will use the stochastic order notation  $O_p(\cdot)$  and  $o_p(\cdot)$  (cf. [18]). Let  $\{X_m\}$  and  $\{Y_m\}$  be sequences of random vectors, defined on the same probability space. We write  $X_m = O_p(Y_m)$  to mean that for each  $\epsilon > 0$  there exists a real number  $M$  such that  $\Pr(\|X_m\| \geq M\|Y_m\|) < \epsilon$  if  $m$  is large enough. We write  $X_m = o_p(Y_m)$  to mean that  $\Pr(\|X_m\| \geq \epsilon\|Y_m\|) \rightarrow 0$  for each  $\epsilon > 0$ . Notice that  $\{Y_m\}$  may also be non-random. For example,  $X_m = o_p(1)$  means that  $X_m \rightarrow 0$  in probability.

Let  $\mathcal{D}$  be a probability distribution on  $\mathbb{R}^n$ , with a bounded probability density function  $p(\cdot)$  which is continuous as a function on  $\mathbb{R}^n$ . Assume that the following two regularity conditions hold:

- $\int_{\mathbb{R}^n} p(\mathbf{x}) \|\mathbf{x}\|^2 d\mathbf{x} < \infty$  (in words,  $\mathcal{D}$  has bounded variance).
- There exists a bounded, monotonically decreasing function  $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ , such that  $p(\mathbf{x}) \leq g(\|\mathbf{x}\|)$  for all  $\mathbf{x} \in \mathbb{R}^n$ , and  $\int_{r=0}^{\infty} r^n g(r) < \infty$ .

The second requirement is needed in order to apply the main theorem of [13] (it is a slightly stronger version of condition (iv) there), and can probably be improved. Nevertheless, it is quite mild, and holds in particular for any distribution that is not heavy-tailed or has bounded support. As to the continuity requirement of  $p(\cdot)$ , it should be noted that our results hold even if we assume continuity solely in some neighborhood of the optimal cluster boundaries, but we will take this stronger assumption for simplicity.

Let  $\mathbf{A}_k$  denote an 'ideal' version of the standard  $k$ -means algorithm, which is given a sample  $S = \{\mathbf{x}_i\}_{i=1}^m \subseteq \mathbb{R}^n$ , sampled i.i.d from  $\mathcal{D}$ , and a required number of clusters  $k$ ,

and returns a set of centroids  $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_k) \in \mathbb{R}^{nk}$ , which are a global minimum of the objective function:

$$\hat{W}(\mathbf{c}) := \frac{1}{m} \sum_{i=1}^m \min_{j \in [k]} \|\mathbf{c}_j - \mathbf{x}_i\|^2.$$

Let  $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k) \in \mathbb{R}^{nk}$  be an optimal  $k$ -means solution with respect to  $\mathcal{D}$ , defined as a minimizer of

$$W(\mathbf{c}) := \int_{\mathbb{R}^n} p(\mathbf{x}) \min_{j \in [k]} \|\mathbf{c}_j - \mathbf{x}_i\|^2 d\mathbf{x}.$$

We assume that such a minimizer exists, is unique up to permutation of the centroids, and that all centroids are distinct (for all  $i \neq j$ ,  $\boldsymbol{\mu}_i \neq \boldsymbol{\mu}_j$ ). To avoid ambiguities involving permutation of the centroids, we assume that the numbering of the centroids is by some uniform canonical ordering (for example, by sorting with respect to the coordinates).

For some set of centroids  $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_k)$ , and for each cluster centroid  $\mathbf{c}_i$ , we denote the interior of its corresponding cluster as  $C_{\mathbf{c},i}$ , defined as:

$$C_{\mathbf{c},i} := \left\{ \mathbf{x} \in \mathbb{R}^n : \arg \min_{j \in [k]} \|\mathbf{c}_j - \mathbf{x}\|^2 = i \right\}.$$

From the continuity assumptions on  $p$ , we may assume that the set of points not in the interior of some cluster has zero measure with respect to  $p$ . We can therefore neglect the issue of how points along cluster boundaries are assigned.

The (scaled) distance between two clusterings  $\mathbf{A}_k(S_1)$  and  $\mathbf{A}_k(S_2)$ , where  $S_1, S_2$  are samples of size  $m$ , is defined as:

$$d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2)) := \sqrt{m} \Pr_{\mathbf{x}_1, \mathbf{x}_2 \sim \mathcal{D}} (\mathbf{A}_k(S_1)(\mathbf{x}_1, \mathbf{x}_2) \neq \mathbf{A}_k(S_2)(\mathbf{x}_1, \mathbf{x}_2)),$$

where  $\mathbf{A}_k(S)(\mathbf{x}_1, \mathbf{x}_2)$  is an indicator function of whether the instances  $\mathbf{x}_1, \mathbf{x}_2$  are in the same cluster according to the clustering given by  $\mathbf{A}_k(S)$ . This definition follows that of [1] and [15], with the additional scaling by  $\sqrt{m}$  (the 'correct' scaling factor as will become evident later on). A typical way to measure instability in practice is to cluster independent subsamples of the data, and empirically estimate the distance between the resulting clusterings. Thus, understanding the behavior of  $d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$  (over drawing and clustering independent samples) is of much interest in analyzing the behavior of clustering stability.

Any choice of cluster centroids  $\mathbf{c}$  induces a Voronoi partition on  $\mathbb{R}^n$ . We will denote  $F_{\mathbf{c},i,j}$ , for  $i \neq j$ , as the boundary face between clusters  $i$  and  $j$ . Namely, the points in  $\mathbb{R}^n$  whose two closest cluster centroids are  $\mathbf{c}_i$  and  $\mathbf{c}_j$ , and are equidistant from them:

$$F_{\mathbf{c},i,j} := \left\{ \mathbf{x} \in \mathbb{R}^n : \arg \min_{a \in [k]} \|\mathbf{c}_a - \mathbf{x}\|^2 = \{i, j\} \right\}.$$

Assuming  $\mathbf{c}_i, \mathbf{c}_j$  are distinct,  $F_{\mathbf{c},i,j}$  is a (possibly empty) subset of the hyperplane  $H_{\mathbf{c},i,j}$ , defined as

$$H_{\mathbf{c},i,j} := \left\{ \mathbf{x} \in \mathbb{R}^n : \left( \mathbf{x} - \frac{\mathbf{c}_i + \mathbf{c}_j}{2} \right)^\top \cdot (\mathbf{c}_1 - \mathbf{c}_2) = 0 \right\}.$$

In our discussion, we use integrals with respect to both the  $n$ -dimensional Lebesgue measure, as well as the  $(n-1)$ -dimensional Lebesgue measure. The type of integral we are

using should be clear from the context, depending on the set over which we are integrating. For example, integrals over some  $C_{c,i}$  are of the first type, while integrals over some  $F_{c,i,j}$  are of the second type.

Let  $\Gamma$  be the  $kn \times kn$  matrix, which is the Hessian of the mapping  $W(\cdot)$  at the optimal solution  $\mu$ . This matrix is composed of  $k \times k$  blocks  $\Gamma_{i,j}$  for  $i, j \in [k]$ . Each block  $\Gamma_{i,j}$  can be shown to be equal to<sup>1</sup>

$$\Gamma_{i,j} := 2 \left[ \int_{C_{\mu,i}} p(\mathbf{x}) d\mathbf{x} \right] I_n - 2 \sum_{a \neq i} \frac{\int_{F_{\mu,i,a}} p(\mathbf{x})(\mathbf{x} - \mu_i)(\mathbf{x} - \mu_a)^\top d\mathbf{x}}{\|\mu_i - \mu_a\|}$$

if  $i = j$ , and for  $i \neq j$  it is defined as

$$\Gamma_{i,j} := \frac{2}{\|\mu_i - \mu_j\|} \int_{F_{\mu,i,j}} p(\mathbf{x})(\mathbf{x} - \mu_i)(\mathbf{x} - \mu_j)^\top d\mathbf{x}$$

We will use the same block notation later for its inverse  $\Gamma^{-1}$ . The existence of these integrals can be shown to follow from the assumptions on  $p(\cdot)$ . We assume that the matrix  $\Gamma$  is positive definite. This is in fact an almost redundant requirement, since the optimality of  $\mu$  entails that  $\Gamma$  is always positive semidefinite. Therefore, cases where  $\Gamma$  is not positive definite correspond to singularities which are apparently pathological (for more discussion on this, see [14]).

Let  $V$  be a  $kn \times kn$  matrix, which represents (up to a constant) the covariance matrix of  $\mathcal{D}$  with respect to each cluster, assuming the optimal clustering induced by  $\mu$ . More specifically,  $V$  is composed of  $k$  diagonal blocks  $V_i$  of size  $n \times n$  for  $i \in [k]$  (all other elements of  $V$  are zero), where

$$V_i := 4 \int_{C_{\mu,i}} p(\mathbf{x})(\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^\top d\mathbf{x}.$$

We shall assume that  $V_i \neq 0$  for any  $i$ .

### 3 Main Results

In this section, we present the main results of our paper, and discuss observations that might be drawn from them about the use of clustering stability in the  $k$ -means framework. All the detailed proofs are presented in Sec. 4.

#### 3.1 Statement of Technical Results

Our main technical result is the following theorem, which characterizes the exact distribution to which  $d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$  converges for any appropriate underlying distribution  $\mathcal{D}$ , and its expected value.

**Theorem 1.** *Assume  $\mathcal{D}$  has a bounded probability density function  $p(\cdot)$ , which is continuous as a function on  $\mathbb{R}^n$  and fulfills the two regularity conditions specified in Sec. 2. Let  $\mathbf{A}_k$  be an algorithm which returns a global minimizer  $\mathbf{c}$  of*

<sup>1</sup>This is proven in [13]. The definition of  $\Gamma$  there differs from ours in one of the signs, apparently due to a small error in that paper [12].

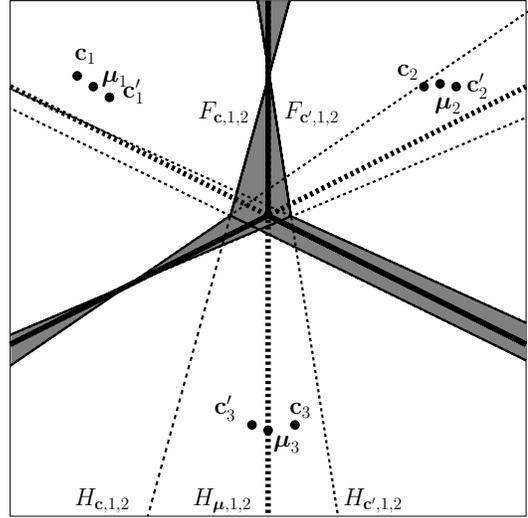


Figure 1: An illustrative drawing of the setting and notation used. Thicker lines represent the optimal  $k$ -means clustering partition (for  $k = 3$  clusters) with respect to the underlying distribution. Clustering two independent random samples gives us two random centroid sets  $\mathbf{c}$  and  $\mathbf{c}'$ . These induce two different Voronoi partitions of  $\mathbb{R}^n$ , and the distance measure is intimately related to the probability mass in the area which switches between clusters, when we compare these two partitions (gray area).

$\hat{W}(\cdot)$  for any  $k$  of interest, and assume that  $\mathbf{c}$  converges in probability to some set of  $k$  distinct centroids  $\mu$  which are the unique global minimizer of  $W(\cdot)$ . Furthermore, assume that  $\Gamma$  is invertible and that  $V_i \neq 0$  for any  $i \in [k]$ . Then we have that  $d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$  converges in distribution to that of

$$2\sqrt{2} \sum_{1 \leq i < j \leq k} \left[ \left( \int_{C_{\mu,i} \cup C_{\mu,j}} p(\mathbf{x}) d\mathbf{x} \right) \times \left( \int_{F_{\mu,i,j}} p(\mathbf{x}) \frac{\left| \begin{pmatrix} \mu_i - \mathbf{x} \\ \mathbf{x} - \mu_j \end{pmatrix}^\top \begin{pmatrix} \mathbf{c}_i - \mu_i \\ \mathbf{c}_j - \mu_j \end{pmatrix} \right|}{\|\mu_i - \mu_j\|} d\mathbf{x} \right) \right],$$

where  $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_k)^\top \sim \mathcal{N}(\mu, \Gamma^{-1}V\Gamma^{-1})$ .

Denoting the expected value of this distribution as  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$ , we have that it is equal to

$$\frac{4}{\sqrt{\pi}} \sum_{1 \leq i < j \leq k} \left[ \left( \int_{C_{\mu,i} \cup C_{\mu,j}} p(\mathbf{x}) d\mathbf{x} \right) \times \left( \int_{F_{\mu,i,j}} p(\mathbf{x}) \frac{\Psi(\mathbf{x}, i, j)}{\|\mu_i - \mu_j\|} d\mathbf{x} \right) \right],$$

where  $\Psi(\mathbf{x}, i, j)$  is defined as

$$\left\| \begin{pmatrix} V_i^{1/2} & 0 \\ 0 & V_j^{1/2} \end{pmatrix} \begin{pmatrix} (\Gamma^{-1})_{i,i} & (\Gamma^{-1})_{i,j} \\ (\Gamma^{-1})_{j,i} & (\Gamma^{-1})_{j,j} \end{pmatrix} \begin{pmatrix} \mu_i - \mathbf{x} \\ \mathbf{x} - \mu_j \end{pmatrix} \right\|.$$

All the integrals can be shown to exist by the assumptions on  $p(\cdot)$ . It should be emphasized that  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$  is *not* necessarily the same as  $\lim_{m \rightarrow \infty} \mathbb{E} d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$ . This is because our convergence result does not necessarily imply convergence of expectations. Thus, formally speaking, the result above does not deal directly with the limit of  $\mathbb{E} d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$ , which has been used in [1],[15] as the theoretical definition of clustering stability. However, it turns out that for our purposes this is not too significant. It seems to be the asymptotic distribution and  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$ , rather than the asymptotic expectation, which determine the asymptotic behavior of clustering stability.

The following theorem exemplifies this on a simple empirical estimator of clustering stability. The main difference between the following estimator and those proposed in the literature is that it measures the distance between just a single pair of clusterings from a pair of independent samples, rather than averaging over several pairs based on subsampling the data. This just makes our result stronger, because these kind of bootstrap procedures should only increase the reliability of the estimator, whereas here we are interested in a 'lower bound' on reliability.

**Theorem 2.** *Define a clustering stability estimator,  $\hat{\theta}_{k,4m}$ , as follows: Given a sample of size  $4m$ , split it randomly into 3 disjoint subsets  $S_1, S_2, S_3$  of size  $m, m$  and  $2m$  respectively. Estimate  $d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2)) / \sqrt{m}$  by computing*

$$\frac{1}{m} \sum_{x_i, x_{m+i} \in S_3} \mathbf{1}(\mathbf{A}_k(S_1)(x_i, x_{m+i}) \neq \mathbf{A}_k(S_2)(x_i, x_{m+i})),$$

where  $(x_1, \dots, x_m)$  is a random permutation of  $S_3$ . For any distribution  $\mathcal{D}$  satisfying the conditions of Thm. 1, assume that for some two values of  $k$ ,  $k_s \neq k_u$ , the ratio of  $\widehat{\text{instab}}(\mathbf{A}_{k_u}, \mathcal{D})$  and  $\widehat{\text{instab}}(\mathbf{A}_{k_s}, \mathcal{D})$  (as defined in Thm. 1) is  $\infty > R > 3$ . Then we have that:

$$\Pr\left(\hat{\theta}_{k_s, 4m} \geq \hat{\theta}_{k_u, 4m}\right) \leq \frac{0.3 + 3 \log(R)}{R} + o(1),$$

where the probability is over a sample of size  $4m$  used for both estimators, and  $o(1)$  converges to 0 as  $m \rightarrow \infty$ .

The theorem implies the following: Suppose we are considering two possible values for  $k$ , designated as  $k_s$  and  $k_u$ , such that the ratio between  $\widehat{\text{instab}}(\mathbf{A}_{k_u}, \mathcal{D})$  and  $\widehat{\text{instab}}(\mathbf{A}_{k_s}, \mathcal{D})$  is some reasonably large constant (one can think of it as a relatively unstable model corresponding to  $k_u$ , vs. a relatively stable model corresponding to  $k_s$ ). Then the probability of *not* empirically detecting  $k_s$  as the most stable model has an upper bound which actually decreases with the sample size, converging to a constant value dependent on the ratio of  $\widehat{\text{instab}}(\mathbf{A}_{k_s}, \mathcal{D})$  and  $\widehat{\text{instab}}(\mathbf{A}_{k_u}, \mathcal{D})$ . In this sense, according to the bound, clustering stability does not 'break down' in the large sample regime, and the asymptotic reliability of its empirical estimation is determined by  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$ . We emphasize that the theorem deals with the reliability of detecting the most stable model, not whether a stable model is really a 'good' model in any other sense.

We note that our proof actually produces an entire range of bounds, which provides a trade off for the minimality requirement on  $R$  with the tightness in terms of the constants.

See the proof for further details. Also, if  $\widehat{\text{instab}}(\mathbf{A}_{k_s}, \mathcal{D}) = 0$ , while  $\widehat{\text{instab}}(\mathbf{A}_{k_u}, \mathcal{D}) > 0$  (corresponding to  $R = \infty$ ), it is easy to show that the probability of detecting  $k_s$  as the most stable model converges to 1 as  $m \rightarrow \infty$ .

### 3.2 Factors Influencing Stability of Clustering Models

According to Thm. 1, for any distribution satisfying the necessary conditions, the distance between clusterings (after scaling by  $\sqrt{m}$ ) converges to a generally non-degenerate distribution, which depends on the underlying distribution and the number of clusters  $k$ . As Thm. 2 shows, this implies that clustering stability does not 'break down' in the large sample regime, and its choice of the most 'appropriate' value of  $k$  seems to depend essentially on  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$ .

Thm. 1 provides an explicit formula for  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$ . Although one can always calculate it for specific cases, it is of much more interest to try and understand what are the governing factors influencing its value. These factors eventually determine what is considered by clustering stability as the 'correct' model, with a low value for  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$ . Therefore, analyzing these factors can explain what sample-size-free assumptions correspond to the use of clustering stability, at least in the  $k$ -means setting that we study. Since a rigorous analysis is a complex endeavor in itself, we will limit ourselves to some preliminary and non-formal observations, which should be taken as such.

According to Thm. 1, the value of  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$  is asymptotically determined by three factors:

- The probability density along the cluster boundaries.
- The Hessian  $\Gamma$  of the objective function  $W(\cdot)$  at  $\boldsymbol{\mu}$ .
- The variance  $V$  and mass of the clusters with respect to the underlying distribution.

A fourth factor appearing in the formula is  $\boldsymbol{\mu}_i - \boldsymbol{\mu}_j$ , but this can be seen simply as a normalization term, eliminating the dependence on the norm of  $\mathbf{x}$ .

The probability density along the cluster boundaries seems to play an interesting role. For example, when the density at the boundaries is exactly 0, we get that  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D}) = 0$ . Although this density is multiplied by  $\Psi(\mathbf{x}, i, j)$ , note that  $\Psi(\mathbf{x}, i, j)$  actually becomes 'nicer' when the boundary density is lower (since  $\Gamma^{-1}$  approaches a diagonal matrix with entries proportional to the inverse of the mass of the clusters, hence having well-controlled eigenvalues assuming reasonably balanced clusters). Therefore, we might expect low instability even when the boundary density is low but not exactly 0.

As to the Hessian  $\Gamma$ , an exact analysis of its influence on  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$  is problematic in the general case, but a useful rough characterization is the spectrum of  $\Gamma$ . If all the eigenvalues of  $\Gamma^{-1}$  are 'large', then we might expect  $\Psi(\mathbf{x}, i, j) / \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|$  to be relatively large as well, leading to a higher value for  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$ . On the other hand, small eigenvalues might lead to lower values of  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$ . Thus, we see that a small spectral radius of the Hessian  $\Gamma$ , representing a 'locally shallow' optimal solution, may result in

more instability. It is interesting to note that shallow, ill-defined minima in terms of the objective function are often a sign of a mismatch between the model and the data, and therefore clustering stability seems to be doing a good thing on that regard.

When will the spectral radius of  $\Gamma$  be small, contributing to instability? By inspecting the formula for  $\Gamma$ , and assuming all clusters have equal sizes, we see that the diagonal elements of  $\Gamma$  are at most  $2/k$ , and can become smaller if the density along the boundary points is larger. Since the main diagonal majorizes the spectrum of the symmetric matrix  $\Gamma$  (cf. [5]), it seems that a small spectral radius might correspond to larger values of  $k$ , as well as high density along the cluster boundaries. A similar analysis for  $V$  seems to indicate that high cluster variance increases instability as well.

These observation also imply that clustering instability might tend to be larger for higher values of  $k$ . As  $k$  becomes larger,  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$  is the result of integrating over a larger area (all cluster boundaries), and the Hessian  $\Gamma$  might tend to have a smaller spectral radius, especially if the boundaries have high density. This is somewhat compensated in the formula by the mass and variance of each cluster becoming smaller, but these seem to scale down more slowly than the cluster boundaries area (and number) scaling up, especially in high dimensions. This matches a well known experimental phenomenon, in which clusterings tend to be less stable for higher  $k$ , even in hierarchical clustering settings where more than one value of  $k$  is acceptable. When the 'correct' model has a very low boundary density and nice structure compared to competing models, this might overcome the general tendency of instability to increase with  $k$ . However, when this is not the case, normalization procedures might be called for, as in [7].

### 3.3 Examples

To illustrate some of the observations from the previous subsection, we empirically evaluated the instability measure on a few simple toy examples, where everything is well controlled and easy to analyze. The results are displayed in Fig. 2. We emphasize that these are just simple illustrations of possible expected and unexpected characteristics of clustering stability in some very limited cases, which can be gleaned from the theoretical results above, and are not meant to represent more realistic or higher dimensional settings.

First of all, the average value of  $d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$  tends to converge to a constant value, which differs based on the choice of the model order  $k$ , and clustering stability does not seem to 'break down' as sample size increases. The three leftmost plots demonstrate how, for these particular examples, the density along the cluster boundaries seem to play an important role in determining  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$ . For each distribution,  $k = 3$  emerges as the most stable model, since the boundaries between the clusters with  $k = 3$  have low density. However,  $k = 3$  becomes less stable as the Gaussians get closer to each other, leading to higher densities in the boundaries between them. At some point, when the density along the cluster boundaries for  $k = 3$  becomes large enough,  $k = 2$  becomes more stable than  $k = 3$ .

A different manifestation of this behavior can be seen in the rightmost plot, which simulates a hierarchical clustering

setting. In this case, all three Gaussians are separated, but one of them is relatively more separated than the other two. As before,  $k = 4$  is less stable than  $k = 3$  and  $k = 2$ , but now  $k = 2$  is the most stable model. This is primarily because the sum of the boundary densities in  $k = 3$  is larger than the density at the boundary point for  $k = 2$ . Deciding on  $k = 2$  as the number of clusters in the data is not unreasonable (recall that clustering stability makes no explicit generative assumption on how the clusters look like). However, it can indicate that in a hierarchical clustering setting, clustering stability might prefer high levels in the hierarchy, which may or may not be what we want.

### 3.4 Convergence Rates

After establishing the asymptotic distribution of the clustering distance measures for  $k$ -means clustering, a reasonable next step is exploring what kind of guarantees can be made on the convergence rate to this asymptotic limit. As a first step, we establish the following negative result, which demonstrates that without additional assumptions, no universal guarantees can be given on the convergence rate. The theorem refers to the case  $k = 3$ , but the proof idea can easily be extended to other values of  $k$ .

**Theorem 3.** *For any positive integer  $m_0$ , there exists a distribution  $\mathcal{D}$  such that  $d_{\mathcal{D}}^m(\mathbf{A}_3(S_1), \mathbf{A}_3(S_2))$  converges in probability to 0 as  $m \rightarrow \infty$ , but  $\Pr(d_{\mathcal{D}}^m(\mathbf{A}_3(S_1), \mathbf{A}_3(S_2)) > \sqrt{m}/4)$  is at least  $1/3$  for some  $m \geq m_0$ .*

The theorem does not imply that the *asymptotic* convergence rate is arbitrarily bad. In fact, a complicated second-order analysis (omitted from this paper due to lack of space), seems to indicate a uniform power-law convergence rate for any distribution satisfying the conditions of Thm. 1, as well as a few other conditions such as Lipschitz-continuity and bounded third moment. However, the exact constants in this power law can be arbitrarily bad, depending on various characteristics of the distribution. Finding sufficient and empirically verifiable conditions which provide finite sample guarantees is therefore of much interest.

## 4 Proofs

### 4.1 Proof of Thm. 1

Before embarking on the proof, we briefly sketch its outline:

1. Using the central limit theorem for  $k$ -means due to Pollard [13], we can characterize the asymptotic Gaussian distribution of the cluster centroids  $\mathbf{c}$ , in terms of the underlying distribution  $\mathcal{D}$  (Lemma 1).
2. The cluster boundaries are determined by the positions of the centroids. Hence, we can derive the asymptotic distribution of these boundaries. In particular, for every boundary  $F_{\mathbf{c},i,j}$ , we characterize the asymptotic distribution of the pointwise Euclidean distance between two realizations of this boundary, over drawing and clustering two independent samples. This distance is defined relative to a projection on the hyperplane  $H_{\mu,i,j}$  (Lemma 2).

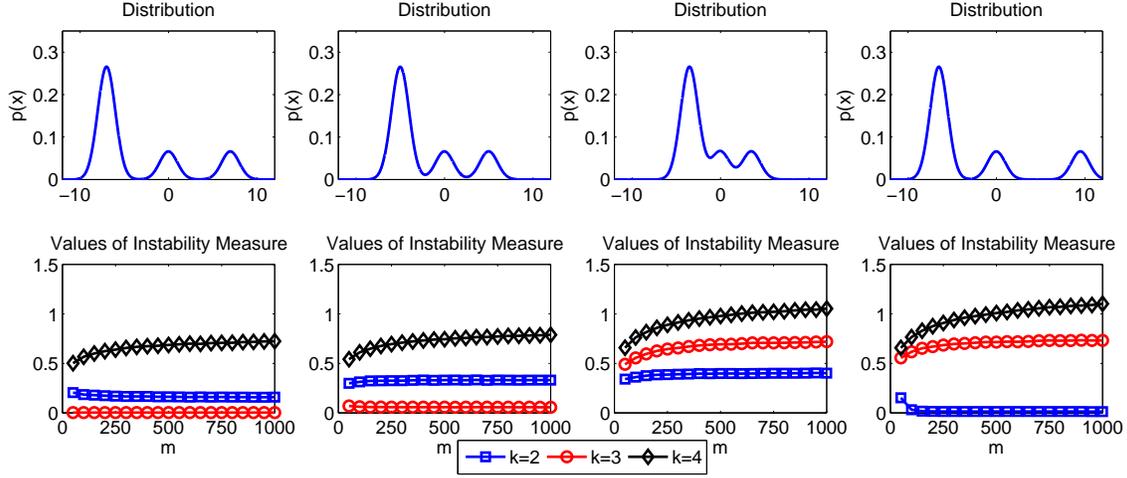


Figure 2: Illustrative examples of the behavior of clustering stability. In each column, the upper plot is the underlying distribution we sample from (a mixture of unit variance Gaussians on  $\mathbb{R}$ ), while the lower plot is an empirical average of  $d_{\mathcal{D}}^m(A_k(S_1), A_k(S_2))$  over 1000 trials, for different sample sizes  $m$ .

3. We show that the probability mass of  $\mathcal{D}$ , which switches between clusters  $i$  and  $j$  over the two independent clusterings, has an asymptotic distribution definable by an integral involving the distance function above, and the values of  $p(\cdot)$  on  $F_{\mu_i, i, j}$  (Lemma 3 and Lemma 4). This allows us to formulate the asymptotic distribution of  $d_{\mathcal{D}}^m(A_k(S_1), A_k(S_2))$ , and its expected value.

For convenience, we shall use  $\epsilon = (\epsilon_1, \dots, \epsilon_k)$  to denote the random element  $\mathbf{c} - \boldsymbol{\mu}$ .

**Lemma 1.** *Under the notation and assumptions of the theorem,  $\sqrt{m}\epsilon = \sqrt{m}(\mathbf{c} - \boldsymbol{\mu})$  converges in distribution to  $\mathbf{v}$ , where  $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \Gamma^{-1}V\Gamma^{-1})$ . As a result,  $\|\epsilon\| = O_p(1/\sqrt{m})$ .*

This lemma is a straightforward consequence of the main theorem in [13]. Notice that it allows us to assume that for large enough values of  $m$ , with arbitrarily high probability and for any  $i, j \in [k], i \neq j$ , the nearest centroid to  $\boldsymbol{\mu}_i$  is  $\mathbf{c}_i$ , all centroids are distinct,  $F_{\mathbf{c}_i, i, j}$  is non-orthogonal to  $F_{\boldsymbol{\mu}_i, i, j}$ , and  $\|\epsilon\|$  is arbitrarily small. We shall tacitly use these assumptions in the remainder of the proof.

**Lemma 2.** *For some  $i, j \in [k], i \neq j$ , assume that  $F_{\boldsymbol{\mu}_i, i, j} \neq \emptyset$ . For any  $\mathbf{x} \in H_{\boldsymbol{\mu}_i, i, j}$ , define the function:*

$$\ell(\mathbf{x}, \mathbf{c}_i, \mathbf{c}_j) = \frac{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| \left( \frac{\mathbf{c}_i + \mathbf{c}_j}{2} - \mathbf{x} \right) \cdot (\mathbf{c}_i - \mathbf{c}_j)}{(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \cdot (\mathbf{c}_i - \mathbf{c}_j)}.$$

Then if  $\|\epsilon\|$  is smaller than some positive constant which depends only on  $\boldsymbol{\mu}$ ,  $\ell(\mathbf{x}, \mathbf{c}_i, \mathbf{c}_j)$  can be rewritten as

$$\frac{1}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|} \left( \frac{\boldsymbol{\mu}_i - \mathbf{x}}{\mathbf{x} - \boldsymbol{\mu}_j} \right)^\top \begin{pmatrix} \boldsymbol{\mu}_i \\ \boldsymbol{\mu}_j \end{pmatrix} + O(\|\mathbf{x}\| + 1)\|\epsilon\|^2.$$

Considering the projection of  $H_{\mathbf{c}_i, i, j}$  to  $H_{\boldsymbol{\mu}_i, i, j}$ , we have that  $\ell(\mathbf{x}, \mathbf{c}_i, \mathbf{c}_j)$  is the signed Euclidean distance of  $\mathbf{x}$  from the point on  $H_{\boldsymbol{\mu}_i, i, j}$  which projects to it (see the left half of Fig. 3). This is because  $\ell(\mathbf{x}, \mathbf{c}_i, \mathbf{c}_j)$  must satisfy the equation:

$$\left( \left( \mathbf{x} + \ell(\mathbf{x}, \mathbf{c}_i, \mathbf{c}_j) \frac{\boldsymbol{\mu}_i - \boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|} \right) - \frac{\mathbf{c}_i + \mathbf{c}_j}{2} \right) \cdot (\mathbf{c}_i - \mathbf{c}_j) = 0.$$

*Proof.* We will separate the expression in the definition of  $\ell(\mathbf{x}, \mathbf{c}_i, \mathbf{c}_j)$  into 2 components and analyze them separately. We have that:

$$\begin{aligned} & \left( \frac{\mathbf{c}_i + \mathbf{c}_j}{2} - \mathbf{x} \right) \cdot (\mathbf{c}_i - \mathbf{c}_j) \\ &= \left( \frac{\boldsymbol{\mu}_i + \boldsymbol{\mu}_j + \boldsymbol{\epsilon}_i + \boldsymbol{\epsilon}_j}{2} - \mathbf{x} \right) \cdot ((\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + (\boldsymbol{\epsilon}_i - \boldsymbol{\epsilon}_j)) \\ &= \left( \frac{\boldsymbol{\mu}_i + \boldsymbol{\mu}_j}{2} - \mathbf{x} \right) \cdot (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \\ & \quad + \left( \frac{\boldsymbol{\mu}_i + \boldsymbol{\mu}_j}{2} - \mathbf{x} \right) \cdot (\boldsymbol{\epsilon}_i - \boldsymbol{\epsilon}_j) \\ & \quad + \left( \frac{\boldsymbol{\epsilon}_i + \boldsymbol{\epsilon}_j}{2} \right) \cdot (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + O(\|\epsilon\|^2). \end{aligned}$$

Notice that the first summand is exactly 0 (by definition of  $\mathbf{x}$  as lying on  $F_{\boldsymbol{\mu}_i, i, j}$ ), and can therefore be dropped. After expanding and simplifying, we get that the above is equal to

$$(\boldsymbol{\mu}_i - \mathbf{x}) \cdot \boldsymbol{\epsilon}_i - (\boldsymbol{\mu}_j - \mathbf{x}) \cdot \boldsymbol{\epsilon}_j + O(\|\epsilon\|^2) \quad (1)$$

As to the second component in the definition of  $\ell(\mathbf{x}, \mathbf{c}_i, \mathbf{c}_j)$ , we have that

$$\begin{aligned} & \frac{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|}{(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \cdot (\mathbf{c}_i - \mathbf{c}_j)} = \frac{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2 + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \cdot (\boldsymbol{\epsilon}_i - \boldsymbol{\epsilon}_j)} \\ &= \frac{1}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| \left( 1 + \frac{(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \cdot (\boldsymbol{\epsilon}_i - \boldsymbol{\epsilon}_j)}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2} \right)} \\ &= \frac{1}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| (1 + O(\|\epsilon\|))} \\ &= \frac{1}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|} \left( 1 - \frac{O(\|\epsilon\|)}{1 + O(\|\epsilon\|)} \right) = \frac{1 + O(\|\epsilon\|)}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|}, \quad (2) \end{aligned}$$

assuming  $\|\epsilon\|$  to be small enough. Multiplying Eq. (1) and Eq. (2) gives us the expression in the lemma.  $\square$

In order to calculate the asymptotic distribution of  $d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$ , we need to characterize the distribution of the probability mass of  $\mathcal{D}$  in the 'wedges' created between two boundaries for clusters  $i, j$ , based on two independent samples (see Fig. 1). For any two given boundaries, calculating the probability mass requires integration of the underlying density function  $p(\cdot)$  over these wedges, making it very hard to write the distribution of this probability mass explicitly. The purpose of the next two lemmas is to derive a more tractable, asymptotically exact approximation for each such wedge, which depends only on the values of  $p(\cdot)$  along the boundary  $F_{\mu, i, j}$ .

We begin with an auxiliary lemma, required for the main Lemma 4 which follows. To state these lemmas, we will need some additional notation. For some  $H_{\mu, i, j}$ , fix some (possibly unbounded) polytope  $F \subseteq H_{\mu, i, j}$ . For notational convenience, we shall assume w.l.o.g that  $H_{\mu, i, j}$  is aligned with the axes, in the sense that for all  $\mathbf{x} \in H_{\mu, i, j}$ , its last coordinate is 0 (it can be easily shown that the regularity conditions on  $p(\cdot)$  will still hold). Also, denote  $F' = \{\mathbf{y} \in \mathbb{R}^{n-1} : (\mathbf{y}, 0) \in F\}$ , which is simply the  $n - 1$  dimensional representation of  $F$  on the hyperplane. Finally, for ease of notation, denote  $\ell((\mathbf{y}, 0), \mathbf{c}_i, \mathbf{c}_j)$  for any  $\mathbf{y} \in F'$  as  $\tilde{\ell}_\epsilon(\mathbf{y})$ , where  $\epsilon = \mathbf{c} - \mu$ .

**Lemma 3.** *Let  $\epsilon, \epsilon'$  be two independent copies of  $\mathbf{c} - \mu$ , each induced by clustering an independent sample of size  $m$ . Let  $B = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq R\}$  be a ball of radius  $R$  centered at the origin. Then we have that*

$$\left| \int_{F' \cap B} \left| \int_{\tilde{\ell}_\epsilon(\mathbf{y})}^{\tilde{\ell}_{\epsilon'}(\mathbf{y})} p(\mathbf{y}, \xi) d\xi \right| d\mathbf{y} - \int_{F' \cap B} \left| \int_{\tilde{\ell}_\epsilon(\mathbf{y})}^{\tilde{\ell}_{\epsilon'}(\mathbf{y})} p(\mathbf{y}, 0) d\xi \right| d\mathbf{y} \right| = o_p(1/\sqrt{m}), \quad (3)$$

where the constants implicit in the r.h.s depend on  $R$ .

*Proof.* Since  $p(\cdot)$  is a non-negative function, we can rewrite the expression in the lemma as

$$\left| \int_{F' \cap B} \int_{\min\{\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})\}}^{\max\{\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})\}} p(\mathbf{y}, \xi) d\xi d\mathbf{y} - \int_{F' \cap B} \int_{\min\{\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})\}}^{\max\{\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})\}} p(\mathbf{y}, 0) d\xi d\mathbf{y} \right|,$$

or

$$\left| \int_{F' \cap B} \int_{\min\{\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})\}}^{\max\{\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})\}} p(\mathbf{y}, \xi) - p(\mathbf{y}, 0) d\xi d\mathbf{y} \right|.$$

By the integral mean value theorem, since  $p(\cdot)$  is continuous, we have that the expression above is equal to:

$$\left| \int_{F' \cap B} |\tilde{\ell}_{\epsilon'}(\mathbf{y}) - \tilde{\ell}_\epsilon(\mathbf{y})| (p(\mathbf{y}, \xi_{\mathbf{y}}) - p(\mathbf{y}, 0)) d\mathbf{y} \right|,$$

where  $\xi_{\mathbf{y}}$  is between the minimum and maximum of  $\{\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})\}$ . For simplicity of notation, we will write  $\xi_{\mathbf{y}} \in [\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})]$ .

The expression above is upper bounded in turn by:

$$\int_{F' \cap B} (|\tilde{\ell}_\epsilon(\mathbf{y})| + |\tilde{\ell}_{\epsilon'}(\mathbf{y})|) \sup_{\xi_{\mathbf{y}} \in [\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})]} |p(\mathbf{y}, \xi_{\mathbf{y}}) - p(\mathbf{y}, 0)| d\mathbf{y},$$

assuming the integral exists. Since  $\epsilon, \epsilon'$  have the same distribution, it is enough to show existence and analyze the convergence to zero in probability for

$$\int_{F' \cap B} |\tilde{\ell}_\epsilon(\mathbf{y})| \sup_{\xi_{\mathbf{y}} \in [\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})]} |p(\mathbf{y}, \xi_{\mathbf{y}}) - p(\mathbf{y}, 0)| d\mathbf{y}. \quad (4)$$

This integral can be upper bounded by

$$\sup_{\mathbf{y} \in F' \cap B} |\tilde{\ell}_\epsilon(\mathbf{y})| \sup_{\xi_{\mathbf{y}} \in [\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})]} |p(\mathbf{y}, \xi_{\mathbf{y}}) - p(\mathbf{y}, 0)| \int_{F' \cap B} 1 d\mathbf{y}. \quad (5)$$

Since  $B$  is bounded, we have according to Lemma 2 that if  $\|\epsilon\|$  is small enough,

$$\sup_{\mathbf{y} \in F' \cap B} |\tilde{\ell}_\epsilon(\mathbf{y})| = O(\|\epsilon\| + \|\epsilon\|^2), \quad (6)$$

and a similar equation holds for  $\tilde{\ell}_{\epsilon'}(\cdot)$  with  $\epsilon$  replaced by  $\epsilon'$  in the r.h.s. To make the equations less cumbersome, we will ignore the higher order term  $\|\epsilon\|^2$ , since  $\epsilon$  converges to 0 in probability anyway by Lemma 1 (it is straightforward to verify that the analysis below still holds). From Eq. (6) and the sentence which follows, we have that

$\sup_{\mathbf{y} \in F' \cap B, \xi_{\mathbf{y}} \in [\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})]} \xi_{\mathbf{y}} = O(\|\epsilon\|)$ . Since  $\|\epsilon\|$  converges to zero in probability, this implies that  $\xi_{\mathbf{y}}$  converges to zero in probability, uniformly for any  $\mathbf{y} \in F' \cap B$ . Moreover,  $p(\cdot)$  is uniformly continuous in the compact domain  $B$ , and thus  $p(\mathbf{y}, \xi_{\mathbf{y}})$  converges uniformly in probability to  $p(\mathbf{y}, 0)$ . As a result, we have that

$$\sup_{\mathbf{y} \in F' \cap B} \sup_{\xi_{\mathbf{y}} \in [\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})]} |p(\mathbf{y}, \xi) - p(\mathbf{y}, 0)| = o_p(1). \quad (7)$$

Substituting Eq. (6) and Eq. (7) into Eq. (5), and using the fact that  $\|\epsilon\| = O_p(1/\sqrt{m})$ , we get that the expression in Eq. (5) (and hence Eq. (4)) is  $o_p(1/\sqrt{m})$  as required.  $\square$

**Lemma 4.** *For some non-empty  $F_{\mu, i, j}$ , let  $t(\mathbf{c}, \mathbf{c}', i, j)$  be a random variable, defined as the probability mass of  $\mathcal{D}$  which switches between clusters  $i, j$  with respect to the two clusterings defined by  $\mathbf{c}, \mathbf{c}'$ , induced by independently sampling and clustering a pair of samples  $S_1, S_2$  each of size  $m$ . More formally, define the set-valued random variable*

$$Q(\mathbf{c}, \mathbf{c}', i, j) = \{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x} \in C_{\mathbf{c}, i} \wedge \mathbf{x} \in C_{\mathbf{c}', j}) \vee (\mathbf{x} \in C_{\mathbf{c}', i} \wedge \mathbf{x} \in C_{\mathbf{c}, j})\} \cup F_{\mathbf{c}, i, j} \cup F_{\mathbf{c}', i, j},$$

so that

$$t(\mathbf{c}, \mathbf{c}', i, j) = \int_{Q(\mathbf{c}, \mathbf{c}', i, j)} p(\mathbf{x}) d\mathbf{x}. \quad (8)$$

Then  $t(\mathbf{c}, \mathbf{c}', i, j)$  is distributed as

$$\int_{F_{\mu, i, j}} p(\mathbf{x}) |l(\mathbf{x}, \mathbf{c}_i, \mathbf{c}'_j)| d\mathbf{x} + o_p(1/\sqrt{m}),$$

where  $l(\mathbf{x}, \mathbf{c}_i, \mathbf{c}'_j)$  is distributed as

$$\frac{1}{\|\mu_i - \mu_j\|} \begin{pmatrix} \mu_i - \mathbf{x} \\ \mathbf{x} - \mu_j \end{pmatrix}^\top \begin{pmatrix} \epsilon_i - \epsilon'_i \\ \epsilon_j - \epsilon'_j \end{pmatrix}.$$

*Proof.* The right half of Fig. 3 should help to clarify the notation and the intuition of the following proof. Intuitively, the probability mass which switches between clusters  $i$  and  $j$  over the two samples is the probability mass of  $\mathcal{D}$  lying 'between'  $F_{c,i,j}$  and  $F_{c',i,j}$ . A potential problem is that this probability mass is also affected by the positions of other neighboring boundaries. However, the fluctuations of these additional boundaries decrease as  $m \rightarrow \infty$ , and their effect on the probability mass in question becomes negligible. Our goal is to upper and lower bound the integral in Eq. (8) by expressions which are identical up to  $o_p(1/\sqrt{m})$  terms, giving us the desired result.

As in Lemma 3, we assume that  $H_{\mu,i,j}$  is aligned with the axes, such that for any  $\mathbf{x} \in H_{\mu,i,j}$ , its last coordinate is 0. Define  $F_{\max}(\boldsymbol{\mu}, \mathbf{c}, \mathbf{c}', i, j) \subseteq H_{\mu,i,j}$  as the projection of  $Q(\mathbf{c}, \mathbf{c}', i, j)$  on  $H_{\mu,i,j}$ . By definition of  $\tilde{\ell}_\epsilon(\mathbf{y}), \tilde{\ell}_{\epsilon'}(\mathbf{y})$ , any point  $\mathbf{x} = (\mathbf{y}, 0)$  in  $F_{\max}(\boldsymbol{\mu}, \mathbf{c}, \mathbf{c}', i, j)$  has the property that the width of  $Q(\mathbf{c}, \mathbf{c}', i, j)$  relative to  $H_{\mu,i,j}$  at  $\mathbf{x}$  is at most  $|\tilde{\ell}_\epsilon(\mathbf{y}) - \tilde{\ell}_{\epsilon'}(\mathbf{y})|$ .

Define  $\delta F(\boldsymbol{\mu}, \mathbf{c}, \mathbf{c}', i, j) \subseteq H_{\mu,i,j}$  as the projection on  $H_{\mu,i,j}$  of  $\partial Q(\mathbf{c}, \mathbf{c}', i, j) \setminus (F_{c,i,j} \cup F_{c',i,j})$ , where  $\partial Q(\mathbf{c}, \mathbf{c}', i, j)$  is the boundary of  $Q(\mathbf{c}, \mathbf{c}', i, j)$ . In words, it is the projection of the boundaries of  $Q(\mathbf{c}, \mathbf{c}', i, j)$ , other than  $F_{c,i,j}, F_{c',i,j}$ , on  $H_{\mu,i,j}$ . Any point  $\mathbf{x} = (\mathbf{y}, 0)$  in  $\delta F(\boldsymbol{\mu}, \mathbf{c}, \mathbf{c}', i, j)$  has the property that the width of  $Q(\mathbf{c}, \mathbf{c}', i, j)$ , relative to  $H_{\mu,i,j}$  at  $\mathbf{x}$ , is less than  $|\tilde{\ell}_\epsilon(\mathbf{y}) - \tilde{\ell}_{\epsilon'}(\mathbf{y})|$ . This is because the segment of the normal to  $H_{\mu,i,j}$  at  $\mathbf{x}$ , between  $H_{c,i,j}$  and  $H_{c',i,j}$ , passes through other clusters besides clusters  $i, j$ .

For notational convenience, we will drop most of the parameters from now on, as they should be clear from the context. Let  $F_{\min} = F_{\max} \setminus \delta F$ . By the properties of  $F_{\max}, \delta F$ , any point  $\mathbf{x} = (\mathbf{y}, 0)$  in  $F_{\min}$  has the property that the width of  $Q$  relative to  $H_{\mu,i,j}$  at  $\mathbf{x}$  is exactly  $|\tilde{\ell}_\epsilon(\mathbf{y}) - \tilde{\ell}_{\epsilon'}(\mathbf{y})|$ .

Let  $F'_{\max}, F'_{\min}$  and  $F'$  be the  $n-1$  dimensional projections of  $F_{\max}, F_{\min}$  and  $F$  respectively, by removing the last zero coordinate which we assume to characterize  $H_{\mu,i,j}$ . As a result of the previous discussion, by Fubini's theorem, we have that:

$$\begin{aligned} \int_{F'_{\max}} \left| \int_{\tilde{\ell}_\epsilon(\mathbf{y})}^{\tilde{\ell}_{\epsilon'}(\mathbf{y})} p(\mathbf{y}, \xi) d\xi \right| d\mathbf{y} &\geq \int_Q p(\mathbf{x}) d\mathbf{x} \\ &\geq \int_{F'_{\min}} \left| \int_{\tilde{\ell}_\epsilon(\mathbf{y})}^{\tilde{\ell}_{\epsilon'}(\mathbf{y})} p(\mathbf{y}, \xi) d\xi \right| d\mathbf{y}, \end{aligned} \quad (9)$$

Assuming these integrals exist. Our goal will be to show that both the upper and lower bounds above are of the form

$$\int_{F_{\mu,i,j}} p(\mathbf{x}) |l(\mathbf{x}, \mathbf{c}_i, \mathbf{c}'_j)| d\mathbf{x} + o_p(1/\sqrt{m}),$$

which entails that the 'sandwiched' integral in Eq. (9) has the same form. We will prove this assertion for the upper bound only, as the proof for the lower bound is almost identical.

As in Lemma 3, we let  $B$  be a closed ball of radius  $R$  in  $\mathbb{R}^n$  centered on the origin, and separately analyze the integral in the upper bound of Eq. (9) with respect to what happens inside and outside this ball.

By Lemma 2, assuming  $\|\epsilon\|$  is small enough, there exists a constant  $a > 0$  dependent only on  $\boldsymbol{\mu}$ , such that

$$|\ell_\epsilon(\mathbf{y})| \leq a(\|\mathbf{y}\| + 1)(\|\epsilon\| + \|\epsilon'\|^2).$$

As before, to avoid making our equations too cumbersome, we shall ignore in the analysis below the higher order term  $\|\epsilon\|^2$ , since  $\epsilon$  converges to 0 in probability and therefore it becomes insignificant compared to  $\|\epsilon\|$ . Also, since we conveniently assume that  $H_{\mu,i,j}$  passes through the origin, then any normal to a point in  $H_{\mu,i,j} \cap B^c$  lies outside  $B$ . This is not critical for our analysis (in the general case, we could have simply defined  $B$  as centered on some point in  $H_{\mu,i,j}$ ), but does simplify things a bit. With these observations, we have that

$$\begin{aligned} &\int_{F'_{\max} \cap B^c} \left| \int_{\tilde{\ell}_\epsilon(\mathbf{y})}^{\tilde{\ell}_{\epsilon'}(\mathbf{y})} p(\mathbf{y}, \xi) d\xi \right| d\mathbf{y} \\ &\leq \int_{F'_{\max} \cap B^c} |\tilde{\ell}_\epsilon(\mathbf{y}) - \tilde{\ell}_{\epsilon'}(\mathbf{y})| \sup_{\xi \in \mathbb{R}} p(\mathbf{y}, \xi) d\mathbf{y} \\ &\leq \int_{F'_{\max} \cap B^c} (|\tilde{\ell}_\epsilon(\mathbf{y})| + |\tilde{\ell}_{\epsilon'}(\mathbf{y})|) \sup_{\xi \in \mathbb{R}} p(\mathbf{y}, \xi) d\mathbf{y} \\ &\leq a(\|\epsilon\| + \|\epsilon'\|) \int_{F'_{\max} \cap B^c} (\|\mathbf{y}\| + 1) \sup_{\xi \in \mathbb{R}} p(\mathbf{y}, \xi) d\mathbf{y} \\ &\leq a(\|\epsilon\| + \|\epsilon'\|) \int_{H_{\mu,i,j} \cap B^c} (\|\mathbf{x}\| + 1) g(\|\mathbf{x}\|) d\mathbf{x} \\ &\leq a(\|\epsilon\| + \|\epsilon'\|) \int_{r=R}^{\infty} (r+1)g(r) * e r^{n-1} dr, \end{aligned}$$

where  $g(\cdot)$  is the dominating function on  $p(\cdot)$  assumed to exist by the regularity conditions (see section 2), and  $e$  is the surface area of an  $n$  dimensional unit sphere. By the assumptions on  $g(\cdot)$  and the fact that  $\|\epsilon\|, \|\epsilon'\| = O_p(1/\sqrt{m})$ , we have that

$$\int_{F'_{\max} \cap B^c} \left| \int_{\tilde{\ell}_\epsilon(\mathbf{y})}^{\tilde{\ell}_{\epsilon'}(\mathbf{y})} p(\mathbf{y}, \xi) d\xi \right| d\mathbf{y} = O_p(h(R)/\sqrt{m}), \quad (10)$$

where  $h(R) \rightarrow 0$  as  $R \rightarrow \infty$ . Notice that to reach this conclusion, we did not use any characteristics of  $F'_{\max}$ , beside it being a subset of  $H_{\mu,i,j}$ . Therefore, since  $|l(\mathbf{x}, \mathbf{c}_i, \mathbf{c}'_j)| \leq a(\|\mathbf{x}\| + 1)(\|\epsilon\| + \|\epsilon'\|)/\sqrt{m}$  for some constant  $a > 0$ , a very similar analysis reveals that

$$\int_{F' \cap B^c} p(\mathbf{y}, 0) |l(\mathbf{x}, \mathbf{c}_i, \mathbf{c}'_j)| d\mathbf{y} = O_p(h(R)/\sqrt{m}). \quad (11)$$

We note for later that none of the constants implicit in the  $O_p(\cdot)$  notation, other than  $h(R)$ , depend on  $R$ . Turning now to what happens inside the ball, we have by Lemma 3 that

$$\begin{aligned} &\int_{F'_{\max} \cap B} \left| \int_{\tilde{\ell}_\epsilon(\mathbf{y})}^{\tilde{\ell}_{\epsilon'}(\mathbf{y})} p(\mathbf{y}, \xi) d\xi \right| d\mathbf{y} \\ &= \int_{F'_{\max} \cap B} |\tilde{\ell}_{\epsilon'}(\mathbf{y}) - \tilde{\ell}_\epsilon(\mathbf{y})| p(\mathbf{y}, 0) d\mathbf{y} + o_p(1/\sqrt{m}). \end{aligned} \quad (12)$$

Leaving this equation aside for later, we will now show

that

$$\left| \int_{F'_{\max} \cap B} |\tilde{\ell}_{\epsilon'}(\mathbf{y}) - \tilde{\ell}_{\epsilon}(\mathbf{y})| p(\mathbf{y}, 0) d\mathbf{y} - \int_{F' \cap B} |\tilde{\ell}_{\epsilon'}(\mathbf{y}) - \tilde{\ell}_{\epsilon}(\mathbf{y})| p(\mathbf{y}, 0) d\mathbf{y} \right| = o_p(1/\sqrt{m}). \quad (13)$$

The l.h.s can be upper bounded by

$$\begin{aligned} & \int_{(F'_{\max} \Delta F') \cap B} |\tilde{\ell}_{\epsilon}(\mathbf{y}) - \tilde{\ell}_{\epsilon'}(\mathbf{y})| p(\mathbf{y}, 0) d\mathbf{y} \\ & \leq \int_{(F'_{\max} \Delta F') \cap B} (|\tilde{\ell}_{\epsilon}(\mathbf{y})| + |\tilde{\ell}_{\epsilon'}(\mathbf{y})|) p(\mathbf{y}, 0) d\mathbf{y}. \end{aligned}$$

As  $\epsilon, \epsilon'$  have the same distribution, we just need to show that

$$\int_{(F'_{\max} \Delta F') \cap B} |\tilde{\ell}_{\epsilon}(\mathbf{y})| p(\mathbf{y}, 0) d\mathbf{y} = o_p(1/\sqrt{m}). \quad (14)$$

By Lemma 2, inside the bounded domain of  $B$ , we have that  $|\tilde{\ell}_{\epsilon}(\mathbf{y})| \leq a\|\epsilon\|$  for some constant  $a$  dependent solely on  $\mu$  and  $R$  (as before, to avoid making the equations too cumbersome, we ignore terms involving higher powers of  $\|\epsilon\|$ ). Moreover, since  $p(\mathbf{y}, 0)$  is bounded, we can absorb this bound into  $a$  and get that

$$\int_{(F'_{\max} \Delta F') \cap B} |\tilde{\ell}_{\epsilon}(\mathbf{y})| p(\mathbf{y}, 0) d\mathbf{y} \leq a\|\epsilon\| \int_{(F'_{\max} \Delta F') \cap B} 1 d\mathbf{y}, \quad (15)$$

Note that  $\int_{(F'_{\max} \Delta F') \cap B} 1 d\mathbf{y}$  is a continuous function of  $\epsilon, \epsilon'$  in some neighborhood of 0. Moreover, since  $F'_{\max} = F'$  when  $\epsilon = \epsilon' = 0$ , the integral above is 0 at  $\epsilon = \epsilon' = 0$ . Since  $\|\epsilon\|, \|\epsilon'\|$  converge to 0 in probability, it follows that

$$\int_{(F'_{\max} \Delta F') \cap B} 1 d\mathbf{y} = o_p(1).$$

Combining this with Eq. (15), and the fact that  $\|\epsilon\| = O_p(1/\sqrt{m})$ , justifies Eq. (14), and hence Eq. (13). Combining Eq. (10), Eq. (12) and Eq. (13), we get that

$$\begin{aligned} & \int_{F'_{\max}} \left| \int_{\tilde{\ell}_{\epsilon}(\mathbf{y})}^{\tilde{\ell}_{\epsilon'}(\mathbf{y})} p(\mathbf{y}, \xi) d\xi \right| d\mathbf{y} \\ & = \int_{F' \cap B} |\tilde{\ell}_{\epsilon'}(\mathbf{y}) - \tilde{\ell}_{\epsilon}(\mathbf{y})| p(\mathbf{y}, 0) d\mathbf{y} \\ & + o_p(1/\sqrt{m}) + O_p(h(R)/\sqrt{m}). \end{aligned} \quad (16)$$

By Lemma 2, definition of  $l(\mathbf{x}, \mathbf{c}_i, \mathbf{c}'_j)$ , and the fact that  $\|\epsilon\|, \|\epsilon'\| = O_p(1/\sqrt{m})$ , we have that  $\tilde{\ell}_{\epsilon}(\mathbf{y}) - \tilde{\ell}_{\epsilon'}(\mathbf{y})$  is equal to  $|l(\mathbf{x}, \mathbf{c}_i, \mathbf{c}'_j)| + o_p((\|\mathbf{y}\| + 1)/\sqrt{m})$ . This implies that the distribution of the r.h.s of Eq. (16) is equal to

$$\int_{F' \cap B} p(\mathbf{y}, 0) |l(\mathbf{x}, \mathbf{c}_i, \mathbf{c}'_j)| d\mathbf{y} + o_p(1/\sqrt{m}) + O_p(h(R)/\sqrt{m}).$$

By Eq. (11), this is equal in turn to

$$\int_{F'} p(\mathbf{y}, 0) |l(\mathbf{x}, \mathbf{c}_i, \mathbf{c}'_j)| d\mathbf{y} + o_p(1/\sqrt{m}) + O_p(h(R)/\sqrt{m}).$$

We now use the fact that  $R$  can be picked arbitrarily. Notice that the first remainder term has implicit constants which depend on  $R$ , but the second remainder term depends on  $R$  only through  $h(R)$  (recall the development leading to Eq. (10) and Eq. (11)). Therefore, the first remainder term converges to 0 at a rate faster than  $1/\sqrt{m}$  in probability for any  $R$ , and the second remainder term can be made arbitrarily smaller than  $1/\sqrt{m}$  in high probability by picking  $R$  to be large enough, since  $h(R) \rightarrow 0$  as  $R \rightarrow \infty$ . Thus, for any  $\delta > 0$ , we can pick  $R$  so that the remainder terms eventually become smaller than  $\delta/\sqrt{m}$  with arbitrarily high probability. As a result, we can replace the remainder terms by  $o_p(1/\sqrt{m})$ , with implicit constants not depending on  $R$ , and get that Eq. (16) can be rewritten as

$$\begin{aligned} & \int_{F'_{\max}} \left| \int_{\tilde{\ell}_{\epsilon}(\mathbf{y})}^{\tilde{\ell}_{\epsilon'}(\mathbf{y})} p(\mathbf{y}, \xi) d\xi \right| d\mathbf{y} \\ & = \int_{F'} p(\mathbf{y}, 0) |l(\mathbf{x}, \mathbf{c}_i, \mathbf{c}'_j)| d\mathbf{y} + o_p(1/\sqrt{m}). \end{aligned}$$

This gives us an equivalent formulation of the upper bound in Eq. (9). As discussed immediately after Eq. (9), an identical analysis can be performed for the lower bound appearing there, and this leads to the result of the lemma.  $\square$

We now turn to prove Thm. 1. Let  $t(\mathbf{c}, \mathbf{c}', i, j)$  be as defined in Lemma 4. Let  $\hat{C}_{\mathbf{c}, \mathbf{c}', i}$  denote the set of points in  $\mathbb{R}^n$  which remain in the same cluster  $i$  for both clusterings defined by  $\mathbf{c}, \mathbf{c}'$ . Then by definition,  $d_D^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$  is equal to

$$2 \sum_{1 \leq i < j \leq k} \left( \int_{\hat{C}_{\mathbf{c}, \mathbf{c}', i} \cup \hat{C}_{\mathbf{c}, \mathbf{c}', j}} p(\mathbf{x}) d\mathbf{x} \right) \sqrt{mt}(\mathbf{c}, \mathbf{c}', i, j). \quad (17)$$

As a straightforward consequence of  $\|\epsilon\| = O_p(1/\sqrt{m})$ , we have that

$$\int_{\hat{C}_{\mathbf{c}, \mathbf{c}', i} \cup \hat{C}_{\mathbf{c}, \mathbf{c}', j}} p(\mathbf{x}) d\mathbf{x} = \int_{C_{\mu, i} \cup C_{\mu, j}} p(\mathbf{x}) d\mathbf{x} + o_p(1). \quad (18)$$

By Lemma 4, we have that  $\sqrt{mt}(\mathbf{c}, \mathbf{c}', i, j)$  is of the form

$$\int_{F_{\mu, i, j}} \frac{\sqrt{m} p(\mathbf{x})}{\|\mu_i - \mu_j\|} \left| \begin{pmatrix} \mu_i - \mathbf{x} \\ \mathbf{x} - \mu_j \end{pmatrix}^\top \begin{pmatrix} \epsilon_i - \epsilon'_i \\ \epsilon_j - \epsilon'_j \end{pmatrix} \right| d\mathbf{x} + o_p(1). \quad (19)$$

By the continuous mapping theorem [18] and standard results on the difference of independent, identically distributed Gaussian vectors [17], we have that  $\sqrt{m}(\epsilon_i - \epsilon'_i, \epsilon_j - \epsilon'_j)^\top$  converges in distribution to  $\sqrt{2}(\mathbf{v}_i, \mathbf{v}_j)^\top$ , where  $\mathbf{v}$  is as defined in Lemma 1. Moreover, it is not difficult to show that Eq. (19), ignoring the remainder term, is a continuous function of  $(\epsilon_i - \epsilon'_i, \epsilon_j - \epsilon'_j)^\top$ . The idea is that it is obviously continuous with the integral restricted to some fixed ball around the origin, and the contributions outside the ball can be made arbitrarily small if the ball is large enough, by the assumptions on  $p(\mathbf{x})$  (a similar argument was made in the proof of Lemma 4). Thus, by the continuous mapping theorem,  $\sqrt{mt}(\mathbf{c}, \mathbf{c}', i, j)$  converges in distribution to

$$\int_{F_{\mu, i, j}} \frac{\sqrt{2} p(\mathbf{x})}{\|\mu_i - \mu_j\|} \left| \begin{pmatrix} \mu_i - \mathbf{x} \\ \mathbf{x} - \mu_j \end{pmatrix}^\top \begin{pmatrix} \mathbf{v}_i \\ \mathbf{v}_j \end{pmatrix} \right| d\mathbf{x}. \quad (20)$$

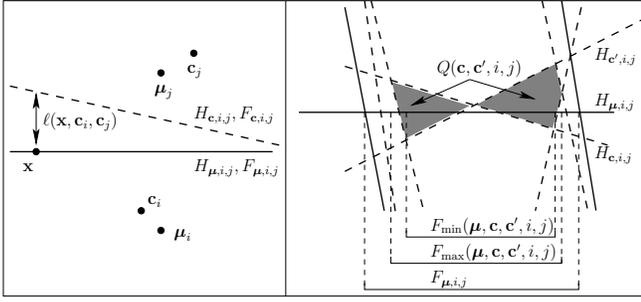


Figure 3: An illustrative drawing of some of the notation and geometrical constructs used in the proof of Thm. 1. Solid lines represent cluster boundaries with respect to the optimal cluster centroids  $\mu$ , while dashed lines represent cluster boundaries with respect to cluster centroids  $c$  or  $c'$  returned by the clustering algorithm based on an empirical sample. See the text for more details.

Substituting Eq. (18) and Eq. (20) into Eq. (17), we get convergence in distribution to the one specified in our theorem.

The only thing remaining is to derive the expected value of this distribution. For notational convenience, let  $\Sigma = \Gamma^{-1}V\Gamma^{-1}$ , and

$$\psi(\mathbf{x}, i, j) = \left| \begin{pmatrix} \mu_i - \mathbf{x} \\ \mathbf{x} - \mu_j \end{pmatrix}^\top \begin{pmatrix} \Sigma_{i,i} & \Sigma_{i,j} \\ \Sigma_{j,i} & \Sigma_{j,j} \end{pmatrix} \begin{pmatrix} \mu_i - \mathbf{x} \\ \mathbf{x} - \mu_j \end{pmatrix} \right|.$$

the expected value of the distribution is equal to:

$$\mathbb{E} \left[ 2\sqrt{2} \sum_{1 \leq i < j \leq k} \left( \int_{C_{\mu,i} \cup C_{\mu,j}} p(\mathbf{x}) d\mathbf{x} \right) \times \left( \int_{F_{\mu,i,j}} \frac{p(\mathbf{x})}{\|\mu_i - \mu_j\|} \left| \begin{pmatrix} \mu_i - \mathbf{x} \\ \mathbf{x} - \mu_j \end{pmatrix}^\top \begin{pmatrix} \mathbf{v}_i \\ \mathbf{v}_j \end{pmatrix} \right| d\mathbf{x} \right) \right].$$

By Fubini's theorem, this is equal to:

$$2\sqrt{2} \sum_{1 \leq i < j \leq k} \left( \int_{C_{\mu,i} \cup C_{\mu,j}} p(\mathbf{x}) d\mathbf{x} \right) \times \left( \int_{F_{\mu,i,j}} \frac{p(\mathbf{x})}{\|\mu_i - \mu_j\|} \mathbb{E} \left[ \left| \begin{pmatrix} \mu_i - \mathbf{x} \\ \mathbf{x} - \mu_j \end{pmatrix}^\top \begin{pmatrix} \mathbf{v}_i \\ \mathbf{v}_j \end{pmatrix} \right| \right] d\mathbf{x} \right).$$

The expression inside the expectation is normally distributed, as a linear transformation of a normal random vector. Using standard results on the distribution of such transformations [17], and since for any univariate  $a \sim \mathcal{N}(\mu, \sigma^2)$  it holds that  $\mathbb{E}[|a|] = \sigma\sqrt{2/\pi}$ , we can reduce the above to

$$\frac{4}{\sqrt{\pi}} \sum_{1 \leq i < j \leq k} \left[ \left( \int_{C_{\mu,i} \cup C_{\mu,j}} p(\mathbf{x}) d\mathbf{x} \right) \times \left( \int_{F_{\mu,i,j}} p(\mathbf{x}) \frac{\sqrt{\psi(\mathbf{x}, i, j)}}{\|\mu_i - \mu_j\|} d\mathbf{x} \right) \right].$$

The final form of  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$  is achieved by rewriting  $\Sigma$  as  $(V^{1/2}\Gamma^{-1})^\top V^{1/2}\Gamma^{-1}$ , substituting into the expression  $\psi(\mathbf{x}, i, j)$ , and simplifying.

## 4.2 Proof of Thm. 2

The proof is composed of several lemmas. The key insight is that the asymptotic distribution of  $d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$  can be viewed as a certain non-standard norm of a Gaussian random vector. Using theorems on Gaussian measures in Banach spaces allows us to bound the probability of  $d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$  being much larger or much smaller than its expectation, and thus bound the probability that the empirical clustering stability estimator will return deceiving results.

**Lemma 5.** *The asymptotic distribution of  $d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$  is equal to that of  $\|\mathbf{v}\|_*$ , where  $\mathbf{v} \sim \mathcal{N}(0, \Gamma^{-1}V\Gamma^{-1})$  and  $\|\mathbf{v}\|_*$  is a norm on  $\mathbb{R}^{nk}$ .*

*Proof.* Denote  $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_n)$  where  $\mathbf{v}_i \in \mathbb{R}^n$ . By Thm. 1, the asymptotic distribution of  $d_{\mathcal{D}}^m(\mathbf{A}_k(S_1), \mathbf{A}_k(S_2))$  is equal to

$$\sum_{1 \leq i < j \leq k} a_{i,j} \int_{F_{\mu,i,j}} p(\mathbf{x}) \left| \begin{pmatrix} \mu_i - \mathbf{x} \\ \mathbf{x} - \mu_j \end{pmatrix}^\top \begin{pmatrix} \mathbf{v}_i \\ \mathbf{v}_j \end{pmatrix} \right| d\mathbf{x}, \quad (21)$$

where  $\mathbf{v}$  is as defined in the lemma, and  $a_{i,j}$  are certain positive constants dependent on  $\mathcal{D}$ . Perhaps unexpectedly, it turns out that this expression defines a norm on  $\mathbf{v}$ : linearity and the triangle inequality are easy to show. Also, Eq. (21) is always non-negative. Finally, Eq. (21) is zero if and only if  $\mathbf{v} = \mathbf{0}$ . One direction is trivial. For the other direction, note that  $p(\cdot)$  must be strictly positive for some non-degenerate subset of some cluster boundary, in order that  $\widehat{\text{instab}}(\mathbf{A}_k, \mathcal{D})$  be positive (which is implied by the assumptions in the theorem). From this, it is straightforward to show that if  $\mathbf{v} \neq \mathbf{0}$  then Eq. (21) is larger than 0.  $\square$

**Lemma 6.** *Let  $\mathbf{v}$  be a non-degenerate normally distributed random vector in  $\mathbb{R}^n$ , let  $\|\cdot\|_*$  be a norm on  $\mathbb{R}^n$  as defined in Lemma 5, and let  $\theta \in (1/2, 1)$  be a free parameter. Introduce the following two parameters which depend on  $\theta$ :*

$$a_\theta = 1 + \frac{2(1-\theta)}{\log\left(\frac{\theta}{1-\theta}\right)}, \quad b_\theta = 1 - \theta + \frac{1 - \exp(-(\text{erf}^{-1}(\theta))^2)}{\sqrt{\pi}\text{erf}^{-1}(\theta)}.$$

*Then for any  $M, \epsilon$  such that  $Mb_\theta > 1$  and  $\epsilon a_\theta < 1$ , it holds that*

$$\Pr(\|\mathbf{v}\|_* > M\mathbb{E}\|\mathbf{v}\|_*) \leq \theta \left( \frac{1-\theta}{\theta} \right)^{(1+Mb_\theta)/2},$$

*and*

$$\Pr(\|\mathbf{v}\|_* \leq \epsilon\mathbb{E}\|\mathbf{v}\|_*) \leq \text{erf}(\text{erf}^{-1}(\theta)a_\theta\epsilon).$$

*Proof.* The distribution of a norm of a Gaussian random vector is continuous, except possibly at 0 (cf. [3]). For any  $\theta \in (1/2, 1)$ , let  $\text{med}_\theta$  be a positive number which satisfies:

$$\Pr(\|\mathbf{v}\|_* \leq \text{med}_\theta) = \theta.$$

Using two results from the literature on Gaussian measures in Banach spaces (theorem III.3 in [11], and theorem 1

from [8]), we have that for any  $M \geq 1$ , and for any  $\epsilon \in [0, 1]$ , it holds that:

$$\Pr(\|\mathbf{v}\|_* > M\text{med}_\theta) \leq \theta \left( \frac{1-\theta}{\theta} \right)^{(1+M)/2} \quad (22)$$

$$\Pr(\|\mathbf{v}\|_* \leq \epsilon\text{med}_\theta) \leq \text{erf}(\text{erf}^{-1}(\theta)\epsilon). \quad (23)$$

It remains to convert these bounds on the deviation from  $\text{med}_\theta$  to the deviation from  $\mathbb{E}\|\mathbf{v}\|_*$ . To achieve this, we need to upper and lower bound  $\mathbb{E}\|\mathbf{v}\|_*/\text{med}_\theta$ . By substitution of variables, we have that  $\mathbb{E}\|\mathbf{v}\|_*$  is equal to

$$\int_0^\infty \Pr(\|\mathbf{v}\|_* > t) dt = \text{med}_\theta \int_0^\infty \Pr(\|\mathbf{v}\|_* > M\text{med}_\theta) dM.$$

Using Eq. (22), this can be upper bounded by

$$\text{med}_\theta \left( 1 + \int_1^\infty \theta \left( \frac{1-\theta}{\theta} \right)^{(1+M)/2} dM \right),$$

which after straightforward computations leads to  $\mathbb{E}\|\mathbf{v}\|_* \leq \text{med}_\theta a_\theta$ , where  $a_\theta$  is as defined in the lemma.

In a similar manner, we can write  $\mathbb{E}\|\mathbf{v}\|_*$  as

$$\begin{aligned} \int_0^\infty 1 - \Pr(\|\mathbf{v}\|_* \leq t) dt \\ = \text{med}_\theta \int_0^\infty 1 - \Pr(\|\mathbf{v}\|_* \leq \epsilon\text{med}_\theta) d\epsilon, \end{aligned}$$

which is lower bounded in term, using Eq. (23), by

$$\text{med}_\theta \int_0^1 1 - \text{erf}(\text{erf}^{-1}(\theta)\epsilon) d\epsilon$$

Again by straightforward computations, we reach the conclusion that  $\mathbb{E}\|\mathbf{v}\|_* \geq \text{med}_\theta b_\theta$ , where  $b_\theta$  is as defined in the lemma.

Therefore, we have that if  $Mb_\theta > 1$ , then  $\Pr(\|\mathbf{v}\|_* > M\mathbb{E}\|\mathbf{v}\|_*)$  is upper bounded by

$$\Pr(\|\mathbf{v}\|_* > Mb_\theta\text{med}_\theta) \leq \theta \left( \frac{1-\theta}{\theta} \right)^{(1+Mb_\theta)/2}.$$

The other bound in the lemma is derived similarly.  $\square$

We can now turn to the proof of Thm. 2. By Lemma 5, both  $d_{\mathcal{D}}^m(\mathbf{A}_{\mathbf{k}_s}(S_1), \mathbf{A}_{\mathbf{k}_s}(S_2))$  and  $d_{\mathcal{D}}^m(\mathbf{A}_{\mathbf{k}_u}(S_1), \mathbf{A}_{\mathbf{k}_u}(S_2))$  converge in distribution to  $\|\mathbf{v}_{k_u}\|_*$  and  $\|\mathbf{v}_{k_s}\|_*$ , where  $\mathbf{v}_{k_u}, \mathbf{v}_{k_s}$  are Gaussian random variables (non-degenerate by the assumptions on  $\Gamma$  and  $V$ ). By Slutsky's theorem and the definition of convergence in distribution,

$$\begin{aligned} \Pr(d_{\mathcal{D}}^m(\mathbf{A}_{\mathbf{k}_u}(S_1), \mathbf{A}_{\mathbf{k}_u}(S_2)) \leq 1.1d_{\mathcal{D}}^m(\mathbf{A}_{\mathbf{k}_s}(S_1), \mathbf{A}_{\mathbf{k}_s}(S_2))) \\ = \Pr(\|\mathbf{v}_{k_u}\|_* \leq 1.1\|\mathbf{v}_{k_s}\|_*) + o(1). \end{aligned} \quad (24)$$

The combination of Lemma 5 and Lemma 6 allows us to upper bound the probability that  $\|\mathbf{v}_{k_u}\|_*$  is smaller than its expectation by a factor  $\epsilon < 1$ , and upper bound the probability that  $\|\mathbf{v}_{k_s}\|_*$  is larger than its expectation by some factor  $M > 1$ , provided that  $\epsilon, M$  satisfy the conditions specified in Lemma 6.

By a union bound argument, if we choose  $M$  and  $\epsilon$  so that  $1.1M/\epsilon \leq R$ , where  $R$  is as defined in the lemma, we get that  $\Pr(\|\mathbf{v}_{k_u}\|_* \leq 1.1\|\mathbf{v}_{k_s}\|_*)$  is upper bounded by

$$\theta_1 \left( \frac{1-\theta_1}{\theta_1} \right)^{((1+M)b_{\theta_1})/2} + \text{erf}(\text{erf}^{-1}(\theta_2)a_{\theta_2}\epsilon), \quad (25)$$

for any  $\theta_1, \theta_2 \in (1/2, 1)$ . Choosing different values for them (as well as the choice of appropriate  $M, \epsilon$ ) leads to different bounds, with a trade off between the tightness of the constants, and minimality requirements on  $R$  (which stem from the requirements on  $M, \epsilon$  by Lemma 6). Choosing  $\theta_1 = 0.9, \theta_2 = 0.8, M = 2 \log(R)/(b_{\theta_1} \log(\theta_1/(1-\theta_1)))$ ,  $\epsilon = 1.1M/R$ , and using the fact that  $\text{erf}(x) \leq (2/\sqrt{\pi})x$  for any  $x \geq 0$ , we get that Eq. (25) is upper bounded by  $(0.3 + 3 \log(R))/R$  for any  $R > 3$ , and therefore Eq. (24) is upper bounded by  $(0.3 + 3 \log(R))/R + o(1)$ .

Assume the event

$$d_{\mathcal{D}}^m(\mathbf{A}_{\mathbf{k}_u}(S_1), \mathbf{A}_{\mathbf{k}_u}(S_2)) > 1.1d_{\mathcal{D}}^m(\mathbf{A}_{\mathbf{k}_s}(S_1), \mathbf{A}_{\mathbf{k}_s}(S_2)), \quad (26)$$

occurs. Recall that the quantities in Eq. (26) depend on the unknown underlying distribution  $\mathcal{D}$ , and therefore cannot be calculated directly. Instead, we empirically estimate these quantities (divided by  $\sqrt{m}$  to be exact), as defined in the theorem statement, to get the stability estimators  $\hat{\theta}_{k_u, 4m}$  and  $\hat{\theta}_{k_s, 4m}$ . Thus, even if Eq. (26) occurs, it is still possible that  $\hat{\theta}_{k_u, 4m} \leq \hat{\theta}_{k_s, 4m}$ . Luckily, by Thm. 2 in [15], the probability for this, conditioned on the event in Eq. (26) is  $o(1)$  (namely, converges to 0 as  $m \rightarrow \infty$ ). Therefore, the probability that Eq. (26) does not occur, or that it does occur but the empirical comparison of these quantities fail, is  $(0.3 + 3 \log(R))/R + o(1)$  as required.

### 4.3 Proof of Thm. 3

To prove the theorem, we will borrow a setting discussed in [10] for a different purpose.

Let  $\Delta$  be some small positive constant (say  $\Delta < 0.1$ ). Consider the parameterized family of distributions  $\{D_\epsilon\}$  (where  $\epsilon \in (0, 1/4)$ ) on the real line, which assigns probability mass  $(1-\epsilon)/4$  to  $x = -1$  and  $x = -1 - \Delta$ , and  $(1+\epsilon)/4$  to  $x = 1$  and  $x = 1 + \Delta$ . Any such distribution satisfies the requirements of Thm. 1, except continuity. However, as mentioned in Sec. 2, the theorem only requires continuity in some region around the boundary points, so we may ignore this difficulty. Alternatively, we may introduce continuity by convolution with a small local smoothing operator. For any  $\epsilon$ , it is easily seen that  $d_{\mathcal{D}_\epsilon}^m(\mathbf{A}_{\mathbf{k}}(S_1), \mathbf{A}_{\mathbf{k}}(S_2))$  converges to 0 in probability, since the boundary points between the optimal clusters have zero density.

Let  $A_{m,\epsilon}^1$  denote the event where for a sample of size  $m$  drawn i.i.d from  $\mathcal{D}_\epsilon$ , there are more instances on  $\{-1 - \Delta, -1\}$  than on  $\{1, 1 + \Delta\}$ . Also, let  $A_{m,\epsilon}^2$  denote the event that for a sample of size  $m$  drawn i.i.d from  $\mathcal{D}_\epsilon$ , there are more instances on  $\{1, 1 + \Delta\}$  than on  $\{-1 - \Delta, -1\}$ . Finally, let  $B_{m,\epsilon}$  denote the event that every point in  $\{-1 - \Delta, -1, 1, 1 + \Delta\}$  is hit by at least one instance from the sample. Clearly, if  $A_{m,\epsilon}^1 \cap B_{m,\epsilon}$  occurs, then the optimal cluster centers for the sample are  $\{-1 - \Delta, -1, 1 + \Delta\}$  for some  $\Delta' \in [0, \Delta]$ , and if  $A_{m,\epsilon}^2 \cap B_{m,\epsilon}$  occurs, then the optimal

cluster centers for the sample are  $\{-1 - \Delta', 1, 1 + \Delta\}$  for some  $\Delta' \in [0, \Delta]$ .

By Thm. 2.1 in [16], for any Bernoulli random variable  $X$  such that  $\mathbb{E}[X] = p \leq 1/2$ , and any whole number  $a$  such that  $a/m \leq 1 - p$ , if  $X_1, \dots, X_m$  are  $m$  i.i.d copies of  $X$ , then

$$\Pr\left(\frac{1}{m} \sum_{i=1}^m X_i \geq \frac{a}{m}\right) \geq 1 - \Phi\left(\sqrt{\frac{m}{p(1-p)}}\left(\frac{a}{m} - p\right)\right),$$

where  $\Phi(\cdot)$  is the cumulative normal distribution function. The probability of the event  $A_{m,\epsilon}^1$  is equal to the probability of a success rate of more than half in  $m$  Bernoulli trials, whose probability of success is  $(1 - \epsilon)/2$ . Using the theorem above, we get after a few straightforward algebraic manipulations and relaxations that

$$\Pr(A_{m,\epsilon}^1) \geq 1 - \Phi\left(\frac{4}{\sqrt{m}} + 2\epsilon\sqrt{m}\right). \quad (27)$$

The probability of the event  $A_{m,\epsilon}^2$  is equal to the probability of a success rate of less than half in  $m$  Bernoulli trials, whose probability of success is  $(1 - \epsilon)/2$ . By a standard normal approximation argument, we have that for large enough values of  $m$ , and for any  $\epsilon \in (0, 1/4)$ , it holds that

$$\Pr(A_{m,\epsilon}^2) \geq 1/2. \quad (28)$$

Finally, it is straightforward to show that  $\Pr(B_{m,\epsilon})$  is arbitrarily close to 1 uniformly for any  $\epsilon$ , if  $m$  is large enough. Combining this with Eq. (27), Eq. (28) and the easily proven formula  $\Pr(A \cap B) \geq \Pr(A) - \Pr(B^c)$  for any two events  $A, B$ , we get that by choosing a large enough sample size  $m > m_0$ , and an appropriate value  $\epsilon$ , it holds that

$$\Pr(A_{m,\epsilon}^1 \cap B_{m,\epsilon}), \Pr(A_{m,\epsilon}^2 \cap B_{m,\epsilon}) \geq 1/2 - \nu$$

for an arbitrarily small  $\nu > 0$ . For that choice of  $m, \epsilon$ , if we draw and cluster two independent samples  $S_1, S_2$  of size  $m$  from  $\mathcal{D}_\epsilon$ , then the probability that event  $A_{m,\epsilon}^1 \cap B_{m,\epsilon}$  occurs for one sample, and  $A_{m',\epsilon}^2 \cap B_{m,\epsilon}$  occurs for the second sample, is at least  $2(1/2 - \nu)^2$ , or at least  $1/3$  for a small enough  $\nu$ . Note that in this case, we get the two different clusterings discussed above, and

$$d_{\mathcal{D}_\epsilon}^m(A_3(S_1), A_3(S_2)) = \frac{\sqrt{m}(1 + \epsilon^2)}{4} > \frac{\sqrt{m}}{4}.$$

So with a probability of at least  $1/3$  over drawing and clustering two independent samples, the distance between the clusterings is more than  $\sqrt{m}/4$ , as required.

## 5 Conclusions and Future Work

In this paper, we analyzed the behavior of clustering stability in the  $k$ -means framework. We were able to explicitly characterize its asymptotic behavior, concluded that it does not 'break down' in the large sample regime, and made some preliminary observations about the factors influencing it. These factors appear to be reasonable requirements from a 'correct' model, and accords with clustering stability working successfully in many situations. However, they also imply that clustering stability might sometimes behave unexpectedly, for example in hierarchical clustering situations, as illustrated in subsection 3.3.

There are several directions for future research. The most obvious perhaps is to extend our results and observations from the asymptotic domain to the finite sample size domain. Showing that clustering stability does not 'break down' in the large sample regime has theoretical and practical relevance, but leaves open the question of why clustering stability can work well for small finite samples. One route to achieve this might be through finite sample guarantees, but as demonstrated in Thm. 3, additional assumptions are needed for such results. Also, it would be interesting to perform a similar analysis for other clustering methods beyond the  $k$ -means framework.

**Acknowledgements:** The authors wish to thank Gideon Schechtman and Leonid Kontorovich for providing the necessary pointers for the proof of Thm. 2.

## References

- [1] Shai Ben-David, Ulrike von Luxburg, and Dávid Pál. A sober look at clustering stability. In *Proceedings of the Nineteenth Annual Conference on Computational Learning Theory*, pages 5–19, 2006.
- [2] Asa Ben-Hur, André Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, pages 6–17, 2002.
- [3] V.I. Bogachev. *Gaussian Measures*. American Mathematical Society, 1998.
- [4] S. Dudoit and J. Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7), 2002.
- [5] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [6] A. Krieger and P. Green. A cautionary note on using internal cross validation to select the number of clusters. *Psychometrika*, 64(3):341–353, 1999.
- [7] Tilman Lange, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6):1299–1323, June 2004.
- [8] Rafał Łatała and Krzysztof Oleszkiewicz. Gaussian measures of dilations of convex symmetric sets. *Annals of Probability*, 27(4):1922–1938, 1999.
- [9] Erel Levine and Eytan Domany. Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, 13(11):2573–2593, 2001.
- [10] T. Linder. *Principles of nonparametric learning*, chapter 4: Learning-theoretic methods in vector quantization. Number 434 in CISM Courses and Lecture Notes (L. Györfi ed.). Springer-Verlag, New York, 2002.
- [11] Vitali D. Milman and Gideon Schechtman. *Asymptotic Theory of Finite Dimensional Normed Spaces*. Springer, 1986.
- [12] David Pollard. Personal communication.
- [13] David Pollard. A central limit theorem for  $k$ -means clustering. *The Annals of Probability*, 10(4):919–926, November 1982.
- [14] Peter Radchenko. *Asymptotics Under Nonstandard Conditions*. PhD thesis, Yale University, 2004.
- [15] Ohad Shamir and Naftali Tishby. Cluster stability for finite samples. In *Advances in Neural Information Processing Systems 21*, 2007.
- [16] E. V. Slud. Distribution inequalities for the binomial law. *The Annals of Probability*, 5(3):402–412, June 1977.
- [17] Y.L. Tong. *The Multivariate Normal Distribution*. Springer, 1990.
- [18] Aad W. Van Der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes : With Applications to Statistics*. Springer, 1996.

---

# Relating clustering stability to properties of cluster boundaries

---

**Shai Ben-David**

David R. Cheriton School of Computer Science  
University of Waterloo  
Waterloo, Ontario, Canada  
shai@cs.uwaterloo.ca

**Ulrike von Luxburg**

Max Planck Institute for Biological Cybernetics  
Tübingen, Germany  
ulrike.luxburg@tuebingen.mpg.de

## Abstract

In this paper, we investigate stability-based methods for cluster model selection, in particular to select the number  $K$  of clusters. The scenario under consideration is that clustering is performed by minimizing a certain clustering quality function, and that a unique global minimizer exists. On the one hand we show that stability can be upper bounded by certain properties of the optimal clustering, namely by the mass in a small tube around the cluster boundaries. On the other hand, we provide counterexamples which show that a reverse statement is not true in general. Finally, we give some examples and arguments why, from a theoretic point of view, using clustering stability in a high sample setting can be problematic. It can be seen that distribution-free guarantees bounding the difference between the finite sample stability and the “true stability” cannot exist, unless one makes strong assumptions on the underlying distribution.

## 1 Introduction

In the domain of data clustering, the problem of model selection is one of the most difficult challenges. In particular the question of selecting the number of clusters has drawn a lot of attention in the literature. A very popular method to solve this problem is to use a stability-based approach. The overall idea is that a clustering algorithm with a certain setting of parameters is meaningful for a given input data if it produces “stable” results, that is, inputs similar to that data lead to similar clustering results. The other way round, an algorithm which is unstable cannot be trusted. This argument is then turned into a model selection criterion: to determine a “good” number  $K$  of clusters on a particular data set, one runs a clustering algorithm with different choices of  $K$  on many perturbed versions of that data set and selects the parameter  $K$  where the algorithm gives the most stable result.

This stability approach has been implemented in various different ways (e.g., Levine and Domany, 2001, Ben-Hur et al., 2002, Lange et al., 2004, Smolkin and Ghosh, 2003) and gains more and more influence in applications, for example in the domain of bioinformatics (Bittner et al.,

2000, Fridlyand and Dudoit, 2001, Kerr and Churchill, 2001, Bertoni and Valentini, 2007). However, its theoretical foundations are not yet well understood. While it is a reasonable requirement that an algorithm should demonstrate stability in general, it is not obvious that, among several stable algorithms, the one which is most stable leads to the best performance.

One important situation has been analyzed in Ben-David et al. (2006) and Ben-David et al. (2007). There it has been proved that in a setting where clustering is performed by globally minimizing an objective function, clustering stability can be characterized by simple properties of the underlying objective function. Namely, given a data set from some particular input distribution, a clustering algorithm is stable for this distribution for large sample sizes if and only if its objective function has a unique global minimizer for that input. As several counter-examples have shown, the latter property is not necessarily related to the fact that the algorithm constructs the correct number of clusters. Some examples for this behavior have also been given in Krieger and Green (1999) (but without rigorous analysis). The dilemma worked out by Ben-David et al. (2006) and Ben-David et al. (2007) is not so much that algorithms get unstable in case of multiple global optima, but the fact that all algorithms with unique global optima are stable. That is, for large sample size (in)stability converges to the same value 0, no matter what parameter  $K$  we choose. This result suggests that for large sample size, stability criteria are unsuitable for model selection.

While this looks like a very negative result on the first glance, recent follow-up work by Shamir and Tishby (2008b) and Shamir and Tishby (2008a) indicates a possible way out of this trap. In a simple situation where the data is distributed according to well separated, univariate Gaussians, the authors show that even though the  $K$ -means algorithm is stable for many values of  $K$ , the rate of convergence of a rescaled measure of stability behaves differently for different numbers of clusters. In this example, the authors show that a model selection criterion based on stability can be used to select the correct number of clusters. The difference to the approach considered in Ben-David et al. (2006) and Ben-David et al. (2007) is that the scaling constant in the definition of stability is chosen as  $1/\sqrt{n}$  rather than  $1/n$ . Hence, the authors consider a central limit theorem setting rather

than a law of large numbers. In the central limit theorem setting, they show that stability does not necessarily converge to 0, but to some normal distribution with particular parameters. Intuitively this means that stability behaves like  $c(K)/\sqrt{n}$  where the constant  $c(K)$  depends (in some complicated way) on the number  $K$  of clusters. In the simple univariate mixture of Gaussian settings studied in Shamir and Tishby (2008b) and Shamir and Tishby (2008a), this constant is higher for the "incorrect" parameter choice. This work indicates that even for large sample size, stability criteria might be useful for model selection after all. It remains to be seen whether this approach can successfully extended to more complex data scenarios reflecting real world data.

The work of Shamir and Tishby (2008b) shows how stability might be used to select the number of clusters in the setting of large sample size and unique global optimizer. However, one crucial question still remains unanswered: what is it really that stability reflects, how will stable clusterings look like in general, and what properties will they have? This is the direction we want to take in our current paper. The general setup is similar to the one discussed above, that is we study clustering algorithms which minimize a certain clustering quality function. As the other case has already been treated completely in Ben-David et al. (2006) and Ben-David et al. (2007), we are now solely concerned with the setting where the clustering quality function has one unique global optimizer. Our goal is to relate the stability of clustering algorithms (on finite sample sizes) to properties of the optimal data clustering itself.

One candidate for such a relation is the conjecture that in the large sample regime, differences in stability of clustering algorithms can be explained by whether the cluster boundaries of the optimal clustering of the underlying space lie in a low or a high density areas of the underlying space. The conjecture is that if the boundaries are in low density areas of the space, an algorithm which constructs clusterings sufficiently close to the optimal clustering will be stable. The other way round, we expect it to be more unstable if the decision boundaries of the optimal clustering are in a high density area. The intuition behind this conjecture is simple: if the decision boundary is in a low density area of the space, small perturbations of the samples might move the boundary a bit, but this movement of the boundary will only affect the cluster labels of very few points (as there are not many points close to the boundary). On the other hand, if the boundary is in a high density area, even small perturbations in the samples will change the cluster assignments of many data points. If this conjecture were true, it would have a very large impact on understanding the mechanism of stability-based model selection.

In this paper, we first prove one direction of this conjecture: the quantitative value of stability can be upper bounded by the mass in a small tube around the optimal clustering boundary. Such a statement has already been implicitly used in Shamir and Tishby (2008b), but only in a very simple one-dimensional setting where the cluster boundary just consists of one single point. The challenge is to prove this statement

in a more general, multidimensional setting.

Unfortunately, it turns out that the opposite direction of the conjecture does not hold. In general, there can be clusterings whose decision boundary lies in a high density area, but we have high stability. We demonstrate this fact with counterexamples which also shed light on the reasons for the failure of this direction of the conjecture.

Finally, we end our paper with a few cautionary thoughts about using stability in large sample scenarios. Essentially, we argue that even if one found satisfactory reasons which explain why a certain clustering tends to be more stable than an other one, such statements are not very useful for drawing conclusions about stability measures of any given *finite* sample size. The reason is that as opposed to the standard statistical learning theory settings, there cannot exist uniform convergence bounds for stability. Thus there is no way one can state any theoretical guarantees on the decisions based on stability for any fixed sample size, unless one makes very strong assumptions on the underlying data distributions.

## 2 Notation and ingredients

### 2.1 General setup

Let  $(\mathcal{X}, d)$  denote an arbitrary metric space. For convenience, in the following we will always assume that  $\mathcal{X}$  is compact. By  $\text{diam } \mathcal{X} := \max_{x,y \in \mathcal{X}} d(x, y)$  we denote the diameter of the space. The space of all probability measures on  $\mathcal{X}$  (with respect to the Borel  $\sigma$ -algebra) is denoted by  $M_1(\mathcal{X})$ . Let  $P$  be a fixed probability measure on  $\mathcal{X}$ , and  $X_1, \dots, X_n$  a sample of points drawn i.i.d. from  $\mathcal{X}$  according to  $P$ . The empirical measure of this sample will be denoted by  $P_n$ .

Let  $F$  be a set of admissible clustering functions of the form  $f : \mathcal{X} \rightarrow \{1, \dots, K\}$ , where  $K \in \mathbb{N}$  denotes the number of clusters. In the following, we will consider clusterings with respect to the equivalence relation of renaming the cluster labels. Namely, define the equivalence relation  $\sim$  on  $F$  by

$$f \sim g : \iff \exists \pi : f(x) = \pi(g(x))$$

where  $\pi$  is a permutation of the set  $\{1, \dots, K\}$ . Denote by  $\mathcal{F} := F/\sim$  the space of equivalence classes of this relation. This will be the space of clusterings we will work with. To perform clustering, we will rely on a clustering quality function  $Q : \mathcal{F} \times M_1(\mathcal{X}) \rightarrow \mathbb{R}$ . The optimal "true" clustering of  $\mathcal{X}$  with respect to  $P$  is defined as

$$f^* := \underset{f \in \mathcal{F}}{\text{argmin}} Q(f, P).$$

Throughout this paper we will assume that  $f^*$  is the unique global optimizer of  $Q$ . If this is not the case, it has already been proved that the corresponding clustering algorithm is not stable anyway (Ben-David et al., 2006, 2007).

When working on a finite sample, we will use an empirical quality function  $Q_n : \mathcal{F} \times M_1(\mathcal{X}) \rightarrow \mathbb{R}$ . We consider the clustering algorithm which, on any given sample, selects the clustering  $f_n$  by

$$f_n := \underset{f \in \mathcal{F}}{\text{argmin}} Q_n(f, P_n).$$

Note that implicit in this formulation, one makes the assumption that the clustering algorithm is able to detect the global minimum of  $Q_n$ . Of course, this is not the case for many commonly used clustering algorithms. For example, the standard  $K$ -means algorithm is not guaranteed to do so. Even though in applications, experience shows that the  $K$ -means algorithm is reasonably successful on “well-clustered” data sets, to get provable guarantees one has to revert to other algorithms, such as the nearest neighbor clustering introduced in von Luxburg et al. (2008) or approximation schemes such as the one introduced in Ostrovsky et al. (2006).

In the following, we will only deal with clustering algorithms which are statistically consistent, that is  $Q(f_n, P) \rightarrow Q(f^*, P)$  in probability. It has been proved that minimizing well-known objective functions such as the one used by  $K$ -means or the normalized cut used in spectral clustering can be performed consistently (von Luxburg et al., 2008).

For two independent samples  $\{X_1, \dots, X_n\}$  and  $\{X'_1, \dots, X'_n\}$  denote the clustering solutions based on minimizing a quality function  $Q_n$  by  $f_n$  and  $f'_n$ , respectively. For a given distance function  $D : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$  which measures some kind of distance between clusterings, the instability of the clustering algorithm minimizing the quality function  $Q$  based on sample size  $n$  is defined as

$$\text{InStab}_D(Q, n, P) := \mathbb{E}(D(f_n, f'_n))$$

where the expectation is over the random drawing of the two samples. So, the stability (or instability) is a function of several quantities: the input data distribution  $P$ , the clustering algorithm (defined by the quality function  $Q$  that the algorithm optimizes), the sample size  $n$ , and the clustering distance measure used. Unless otherwise mentioned, we shall be using the minimal matching distance (see below) for the definition of instability and drop the subscript  $D$  in the instability notation. Also, if it is clear which objective function  $Q$  we refer to, we drop the dependence on  $Q$ , too, and simply write  $\text{InStab}(n, P)$  for instability.

## 2.2 Distance functions between clusterings

Various measures of clustering distances have been used and analyzed in the literature (see for example Meila, 2005). We define below two measures that are most relevant to our discussion.

**Minimal matching distance.** This is perhaps the most widely used distance between clusterings. For two clusterings defined on a finite point set  $X_1, \dots, X_n$ , this distance is defined as

$$D_{\text{MinMatch}}(f_n, f'_n) := \min_{\pi} \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{f(X_i) \neq \pi(g(X_i))}$$

where the minimum is taken over all permutations  $\pi$  of the set  $\{1, \dots, K\}$ . This distance is close in spirit to the 0-1-loss used in classification. It is well known that  $D_{\text{MinMatch}}$  is a metric, and that it can be computed efficiently using a minimal bipartite matching algorithm.

**A distance based on cluster boundaries.** For our current work, we need to introduce a completely new distance between clusterings. Intuitively, this distance measures how far the class boundaries of two clusterings are away from each other. Let  $\mathcal{X}$  be a compact subset of  $\mathbb{R}^s$ ,  $d$  a metric on  $\mathbb{R}^s$  such as the Euclidean one, and  $\mathcal{F}$  the space of all clustering functions  $f : \mathcal{X} \rightarrow \{1, \dots, K\}$ , up to the equivalence relation  $\sim$ . For a given  $f \in \mathcal{F}$ , we define the **boundary**  $B(f)$  of  $f$  as the set

$$B(f) := \{x \in \mathcal{X} \mid f \text{ discontinuous at } x\}.$$

The distance of a point  $x$  to the boundary  $B(f)$  is defined as usual by

$$d(x, B(f)) := \inf\{d(x, y) \mid y \in B(f)\}.$$

For  $\gamma > 0$ , we then we define the **tube**  $T_\gamma(f)$  as the set

$$T_\gamma(f) := \{x \in \mathcal{X} \mid d(x, B(f)) \leq \gamma\}.$$

For  $\gamma = 0$  we set  $T_0(f) = B(f)$ .

We say that a clustering function  $g$  is in the  $\gamma$ -tube of  $f$ , written  $g \triangleleft T_\gamma(f)$ , if

$$\forall x, y \notin T_\gamma(f) : f(x) = f(y) \iff g(x) = g(y).$$

Finally, we define the distance function  $D_{\text{boundary}}$  on  $\mathcal{F}$  as

$$D_{\text{boundary}}(f, g) := \inf_{\gamma > 0} \{f \triangleleft T_\gamma(g) \text{ and } g \triangleleft T_\gamma(f)\}.$$

The distance  $D_{\text{boundary}}$  satisfies several nice properties:

**Proposition 1 (Properties of  $D_{\text{boundary}}$ )** Assume that the metric space  $\mathcal{X} \subset \mathbb{R}^s$  is compact. Let  $\mathcal{F}$  be the set of equivalence classes of clustering functions  $f : \mathcal{X} \rightarrow \{1, \dots, K\}$  as defined above. Then the following technical properties hold:

1.  $D_{\text{boundary}}$  is well-defined on the equivalence classes.
2. Let  $f, g \in \mathcal{F}$ . Then:  $g \triangleleft T_\gamma(f)$  implies that  $B(g) \subset T_\gamma(f)$ .
3. Let  $f, g$  two clusterings with  $D_{\text{boundary}}(f, g) \leq \gamma$ . Then there exists a permutation  $\pi$  such that for all  $x \in \mathcal{X}$ ,

$$f(x) \neq \pi(g(x)) \implies x \in T_\gamma(g).$$

Furthermore, the following fundamental properties hold:

5. The distance function  $D_{\text{boundary}}$  is a metric on  $\mathcal{F}$ .
6.  $\mathcal{F}$  is relatively compact under the topology induced by  $D_{\text{boundary}}$ .

*Proof.*

1. The definitions of all quantities above do not depend on the particular labeling of the clusters, but only on the positions of the cluster boundaries.
2. Let  $g \triangleleft T_\gamma(f)$ , but assume that  $B(g) \not\subset T_\gamma(f)$ . That is, there exists a point  $x \in B(g)$  with  $x \notin T_\gamma(f)$ . By definition of  $B(g)$ ,  $x$  is a point of discontinuity of  $g$ , thus the clustering  $g$  changes its label at  $x$ . On the other hand, by the definition of  $T_\gamma(f)$ ,  $f$  does not change its label at  $x$  (otherwise,  $x$  would be in  $B(f) \subset T_\gamma(f)$ ). But the latter contradicts the definition of  $g \triangleleft T_\gamma(f)$  which requires that  $f$  and  $g$  only change their labels at the same points outside of  $T_\gamma(f)$ . Contradiction.

3. Similar to Part 2.

4.  $D_{\text{boundary}}(f, g) \leq \text{diam } \mathcal{X} < \infty$ : As  $\mathcal{X}$  is compact, it has a finite diameter  $\text{diam } \mathcal{X}$ . Then for all  $f, g \in \mathcal{F}$  we have  $T_{\text{diam } \mathcal{X}}(f) = \mathcal{X}$  and  $T_{\text{diam } \mathcal{X}}(g) = \mathcal{X}$ . Thus, trivially  $f \triangleleft T_{\text{diam } \mathcal{X}}(g)$  and vice versa, that is  $D_{\text{boundary}}(f, g) \leq \text{diam } \mathcal{X}$ .

$D_{\text{boundary}}(f, g) \geq 0$ : clear.

$D_{\text{boundary}}(f, f) = 0$ : clear.

$D_{\text{boundary}}(f, g) = 0 \implies f = g$ :  $D_{\text{boundary}}(f, g) = 0$  implies that  $B(f) \subset T_0(g) = B(g)$  and vice versa, thus we have  $B(f) = B(g)$ . So the class boundaries of both clusterings coincide. Moreover, we have that for all  $x, y \notin B(g)$ ,  $f(x) = f(y) \iff g(x) = g(y)$ . Thus there exists a permutation of the labeling of  $g$  such that  $f(x) = \pi(g(x))$  for all  $x \notin B(g)$ . Thus  $f$  and  $g$  are in the same equivalence class with respect to  $\sim$ , that is  $f = g$  in the space  $\mathcal{F}$ .

Triangle inequality: assume that  $D_{\text{boundary}}(f, g) = \gamma_1$  and  $D_{\text{boundary}}(g, h) = \gamma_2$ , that is

$$\begin{aligned} \forall x, y \notin T_{\gamma_1}(f) : [f(x) = f(y) \iff g(x) = g(y)] \\ \forall x, y \notin T_{\gamma_1}(g) : [f(x) = f(y) \iff g(x) = g(y)] \\ \forall x, y \notin T_{\gamma_2}(g) : [h(x) = h(y) \iff g(x) = g(y)] \\ \forall x, y \notin T_{\gamma_2}(h) : [h(x) = h(y) \iff g(x) = g(y)]. \end{aligned} \quad (1)$$

Now define  $\gamma := \gamma_1 + \gamma_2$ . We first need to prove a small sub-statement, namely that

$$x \notin T_\gamma(f) \implies x \notin T_{\gamma_2}(g). \quad (2)$$

To this end, let  $x \in T_{\gamma_2}(g)$ , that is there exists some point  $y \in B(g)$  with  $d(x, y) \leq \gamma_2$ . As we know that  $g \triangleleft T_{\gamma_1}(f)$ , we also have  $B(g) \subset T_{\gamma_1}(f)$ , that is for all  $y \in B(g)$  exists  $z \in B(f)$  such that  $d(y, z) \leq \gamma_1$ . Combining those two statements and using the triangle inequality for the metric  $d$  on the original space  $\mathcal{X}$ , we can conclude that  $d(x, z) \leq d(x, y) + d(y, z) = \gamma_1 + \gamma_2 = \gamma$ , that is  $x \in T_\gamma(f)$ . This shows statement (2) by its contra-position. Now we can go ahead and prove the triangle inequality for  $D_{\text{boundary}}$ . Using the property (2) and the equations (1) we get that

$$\begin{aligned} x, y \notin T_\gamma(f) \implies x, y \notin T_{\gamma_2}(g) \\ \implies [g(x) = g(y) \iff h(x) = h(y)]. \end{aligned}$$

Moreover, by the definition of  $T_\gamma(f)$  and the fact that  $\gamma \geq \gamma_1$  we trivially have that  $x, y \notin T_\gamma(f)$  implies  $x, y \notin T_{\gamma_1}(f)$ . Together with equations (1) this leads to

$$\begin{aligned} x, y \notin T_\gamma(f) \implies x, y \notin T_{\gamma_1}(f) \\ \implies [g(x) = g(y) \iff f(x) = f(y)]. \end{aligned}$$

Combining those two statements we get

$$x, y \notin T_\gamma(f) \implies [f(x) = f(y) \iff h(x) = h(y)],$$

that is  $h \triangleleft T_\gamma(f)$ . Similarly we can prove that  $f \triangleleft T_\gamma(h)$ , that is we get  $D_{\text{boundary}}(f, h) \leq \gamma$ . This proves the triangle inequality.

All statements together prove that  $D_{\text{boundary}}$  is a metric.

5. By the theorem of Heine-Borel, a metric space is relatively compact if it is totally bounded, that is for any  $\gamma > 0$  it can be covered with finitely many  $\gamma$ -balls. By assumption, we know that  $\mathcal{X}$  is compact. Thus we can construct a finite covering of balls of size  $\gamma$  of  $\mathcal{X}$  (in the metric  $d$ ). Denote the centers of the covering balls as  $x_1, \dots, x_s$ . We want to use this covering to construct a finite covering of  $\mathcal{F}$ . To this end, let  $f \in \mathcal{F}$  be an arbitrary function (for now let us fix a labeling, we will go over to the equivalence class in the end). Given  $f$ , we reorder the centers of the covering balls such that all centers  $x_i$  with  $x_i \notin T_{2\gamma}(f)$  come in the ordering before the points  $x_j$  with  $x_j \in T_{2\gamma}(f)$ , that is:

$$x_i \notin T_{2\gamma}(f) \text{ and } x_j \in T_{2\gamma}(f) \implies i < j.$$

Now we construct a clustering  $\tilde{f}$  as follows: one after the other, in the ordering determined before, we color the balls of the covering according to the color  $f(x_i)$  of its center, that is we set:

$$\begin{aligned} \bullet x \in B(x_1) \implies \tilde{f}(x) := f(x_1) \\ \bullet x \in B(x_2) \setminus B(x_1) \implies \tilde{f}(x) := f(x_2) \\ \dots \\ \bullet x \in B(x_i) \setminus \cup_{t=1, \dots, i-1} B(x_t) : \tilde{f}(x) := f(x_i) \end{aligned}$$

By construction, for all points  $x \notin T_\gamma(f)$  we have  $\tilde{f}(x) = f(x)$ . Consequently,  $\tilde{f} \triangleleft T_\gamma(f)$ . Similarly, the other way round we have  $f \triangleleft T_\gamma(\tilde{f})$ . Thus,  $D_{\text{boundary}}(f, \tilde{f}) \leq \gamma$ . Note that given two representatives  $f, g$  of the same clustering in  $\mathcal{F}$  (that is, two functions such that  $f = \pi(g)$  for some permutation  $\pi$ ), the corresponding functions  $\tilde{f}$  and  $\tilde{g}$  are also representatives of the same clustering, that is  $\tilde{f} = \pi(\tilde{g})$ . Thus the whole construction is well-defined on  $\mathcal{F}$ .

Finally, it is clear that the set  $\tilde{\mathcal{F}} := \{\tilde{f} \mid f \in \mathcal{F}\}$  has finitely many elements: there only exist finitely many orderings of the  $s$  center points  $x_1, \dots, x_s$  and finitely many labelings of those center points using  $K$  labels. Hence, the set  $\tilde{\mathcal{F}}$  forms a finite  $\gamma$ -covering of  $\mathcal{F}$ . ☺

In the current paper, we will only use the distance  $D_{\text{boundary}}$  for clusterings of  $\mathbb{R}^s$ , but its construction is very general. The distance  $D_{\text{boundary}}$  can also be defined on more general metric spaces, and even discrete spaces. One just has to give up defining  $B(f)$  and directly define the set  $T_\gamma(f)$  as the set  $\{x \in \mathcal{X} \mid \exists y \in \mathcal{X} : f(x) \neq f(y) \text{ and } d(x, y) \leq \gamma\}$ . However, in that case, some care has to be taken when dealing with “empty regions” of the space.

### 3 Upper bounding stability by the mass in $\gamma$ -tubes

In this section we want to establish a simple, but potentially powerful insight: given any input data distribution,  $P$ , for large enough  $n$ , the stability of a quality-optimizing consistent clustering algorithm can be described in terms of the  $P$ -mass of along the decision boundaries of the optimal clustering. The intuition is as follows. The distance  $D_{\text{MinMatch}}$  counts the number of points for which two clusterings do not coincide, that is it counts the number of points which lie “between” the decision boundaries of the two clusterings. Stability is the expectation over  $D_{\text{MinMatch}}$ , computed on different random samples.

#### 3.1 Relation between stability and tubes

Let us first assume that we know that with high probability over the random drawing of samples, we have that  $D_{\text{boundary}}(f_n, f) \leq \gamma$  for some constant  $\gamma$ . Then the following proposition holds:

**Proposition 2 (Relating stability and mass in tubes)** *Let  $f$  be any fixed clustering, and  $f_n$  the clustering computed from a random sample of size  $n$ . Assume that with probability at least  $1 - \delta$  over the random samples, we have that  $D_{\text{boundary}}(f_n, f) \leq \gamma$ . Then the instability (based on distance  $D_{\text{MinMatch}}$ ) satisfies*

$$\text{InStab}(n, P) \leq 2\delta + 2P(T_\gamma(f)).$$

*Proof.* Denote the set of samples on which the event  $D_{\text{boundary}}(f_n, f) \leq \gamma$  is true by  $M$ . W.l.o.g. assume that for all  $n$ , the labels of the clustering  $f_n$  are chosen such that they already coincide with the ones of  $f$ , that is the permutation for which the minimum in  $D_{\text{MinMatch}}(f_n, f)$  is attained is the identity. Then we have:

$$\begin{aligned} \text{InStab}(n, P) &= \mathbb{E}(D_{\text{MinMatch}}(f_n, f'_n)) \\ &\leq \mathbb{E}(D_{\text{MinMatch}}(f_n, f) + D_{\text{MinMatch}}(f'_n, f)) \\ &= 2\mathbb{E}D_{\text{MinMatch}}(f_n, f) \\ &= 2 \int_M \mathbf{1}_{f_n(X) \neq f(X)} dP(X) + 2 \int_{M^c} \mathbf{1}_{f_n(X) \neq f(X)} dP(X) \\ &\quad (\text{on } M, f_n(x) \neq f(x) \implies x \in T_\gamma(f), \text{ see Prop. 1}) \\ &\leq 2 \int_M \mathbf{1}_{X \in T_\gamma(f)} dP(X) + 2P(M^c) \\ &= 2P(T_\gamma(f)) + 2\delta \end{aligned}$$

☺

Proposition 2 gives several very plausible reasons for why a clustering can be unstable:

- The decision boundaries themselves vary a lot (i.e.,  $\gamma$  is large). This case is pretty obvious.
- The decision boundaries do not vary so much (i.e.,  $\gamma$  is small), but lie in an area of high density. This is a more subtle reason, but a very valuable one. It suggests that if we compare two clusterings, one of them has its cluster boundary in a high density area and the other one in a low density area, then the first one tends to be more unstable

than the second one. However, to formally analyze such a comparison between stability values of different algorithms, one also has to prove a lower bound on stability, see later.

- The decision boundaries do not vary so much (i.e.,  $\gamma$  is small), are in a region of moderate density, but they are very long, so significant mass accumulates along the boundary.

#### 3.2 Determining the width $\gamma$ in terms of the limit clustering

Now we want to apply the insight from the last subsection to relate properties of the optimal clustering to stability. In this section, we still want to work in an abstract setting, without fixing a particular clustering objective function. In order to prove our results, we will have to make a few crucial assumptions:

- The objective function  $Q$  has a unique global minimum. Otherwise we know by Ben-David et al. (2006) and Ben-David et al. (2007) that the algorithm will not be stable anyway.
- The clustering algorithm is consistent, that is  $Q(f_n, P) \rightarrow Q(f^*, P)$  in probability. If this assumption is not true, any statement about the stability on a finite sample is pretty meaningless, as the algorithm can change its mind with the sample size. For example, consider the trivial algorithm which returns a fixed function  $f_1$  if the sample size  $n$  is even, and another fixed function  $f_2$  if the sample size is odd. This algorithm is perfectly stable for every  $n$ , but since the results do not converge, it is completely meaningless.
- The sample size  $n$  is sufficiently large so that  $Q(f_n) - Q(f^*)$  is sufficiently small:  $f_n$  is inside the region of attraction of the global minimum. With this assumption we want to exclude trivial cases where instability is induced due to too high sample fluctuations. See also Section 5 for discussion.

To state the following proposition, we recall the definition of a quasi-inverse of a function. The quasi-inverse of a function is a generalization of the inverse of a function to cases where the function is not injective. Let  $f : \mathcal{X} \rightarrow \mathcal{Y}$  be a function with range  $\text{rg}(f) \subset \mathcal{Y}$ . A function  $g : \text{rg}(f) \rightarrow \mathcal{X}$  which satisfies  $f \circ g \circ f = f$  is called a quasi-inverse of  $f$ . Note that quasi-inverses are not unique, unless the function  $f$  is injective.

#### Proposition 3 (Consequences of unique global optimum)

*Let  $(\mathcal{X}, d)$  a compact metric space with probability distribution  $P$ , and  $\mathcal{F}$  the space of  $P$ -measurable clusterings with  $K$  clusters on  $\mathcal{X}$ . As a topology on  $\mathcal{F}$ , consider the one induced by the distance  $D_{\text{boundary}}$ . Let  $Q := Q(\cdot, P) : \mathcal{F} \rightarrow \mathbb{R}$  be continuous and assume that it has a unique global minimizer  $f^*$ . Then, every quasi-inverse  $Q^{-1} : \text{rg}(Q) \subset \mathbb{R} \rightarrow \mathcal{F}$  is continuous at  $Q(f^*)$ . In particular, for all  $\gamma > 0$  there exists some  $\varepsilon(\gamma, f^*, P) > 0$  such that for all  $f \in \mathcal{F}$ ,*

$$|Q(f, P) - Q(f^*, P)| \leq \varepsilon \implies D_{\text{boundary}}(f, f^*) \leq \gamma. \quad (3)$$

*Proof.* Assume  $Q^{-1}$  is not continuous at  $Q(f^*)$ , that is there exists a sequence of functions  $(g_n)_n \subset \mathcal{F}$  such that  $Q(g_n) \rightarrow Q(f^*)$  but  $g_n \not\rightarrow f^*$ . By the compactness assumption, the sequence  $(g_n)_n$  has a convergent subsequence  $(f_{n_k})_k$  with  $f_{n_k} \rightarrow \tilde{f}$  for some  $\tilde{f} \in \mathcal{F}$ . Also by assumption, we can find such a subsequence such that  $\tilde{f} \neq f^*$ . By the continuity of  $Q$  we know that  $Q(f_{n_k}) \rightarrow Q(\tilde{f})$ , and by the definition of  $(g_n)_n$  we know also that  $Q(f_{n_k}) \rightarrow Q(f^*)$ . So we know that  $Q(f^*) = Q(\tilde{f})$ , and by the uniqueness of the optimum  $f^*$  this leads to  $f^* = \tilde{f}$ . Contradiction.  $\odot$

Note that the “geometry of  $Q$ ” plays an important role in this proposition. In particular, the size of the constant  $\varepsilon$  heavily depends on the “steepness” of  $Q$  in a neighborhood of the global optimum and on “how unique” the global optimum is. We formalize this by introducing the following quantity:

$$U_P^Q(\gamma) := \sup \left\{ \varepsilon > 0 : \right. \\ \left. |Q(f, P) - Q(f^*, P)| \leq \varepsilon \implies D_{\text{boundary}}(f, f^*) \leq \gamma \right\}.$$

One can think of  $U_P^Q$  as indicating how unique is the optimal clustering  $f^*$  of  $P$  is.

The following theorem bounds the stability of a clustering algorithm on a given input data distribution by the mass it has in the tube around the decision boundary. It replaces the assumption of uniform convergence of the empirical clusterings under the  $D_{\text{boundary}}$  metric of Proposition 2 by the more intuitive assumption that the underlying clustering algorithm is *uniformly consistent*. That is,  $Q(f_n, P) \rightarrow Q(f^*, P)$  in probability, uniformly over all probability distributions  $P$ :

$$\forall \varepsilon > 0 \forall \delta > 0 \exists n \in \mathbb{N} \forall P : \\ P(|Q(f_n, P) - Q(f^*, P)| > \varepsilon) \leq \delta.$$

In particular, for any positive  $\varepsilon$  and  $\delta$ , the required sample size  $n$  does not depend on  $P$ . Such an assumption holds, for example, for the algorithm constructing the global minimum of the  $K$ -means objective function, as shown by Ben-David (2007). For background reading on consistency of clustering algorithms and bounds for many types of objective function see von Luxburg et al. (2008). When such uniform consistency holds for  $Q$ , let us quantify the sample size by defining

$$C_Q(\varepsilon, \delta) := \min \left\{ m \in \mathbb{N} : \right. \\ \left. \forall P \forall n \geq m \ P(|Q(f_n, P) - Q(f^*, P)| > \varepsilon) \leq \delta \right\}.$$

We can now provide a bound on stability which refers to the following quantities: the uniqueness  $U_P^Q$  of the optimal clustering, the consistency  $C_Q$  of the quality measure, and the  $P$ -weight of the tubes around the optimal clustering of the input data distribution.

**Theorem 4 (High instability implies cut in high density region)** *Let  $\mathcal{X}$  be a compact subset of  $\mathbb{R}^s$ , Assume that the cluster quality function  $Q(\cdot, P) : \mathcal{F} \rightarrow \mathbb{R}$  is continuous with respect to the topology on  $\mathcal{F}$  induced by  $D_{\text{boundary}}$ . Let*

*$Q(\cdot, P)$  have a unique global minimizer  $f^*$ , and assume that  $Q(\cdot, P)$  can be minimized uniformly consistently, Then, for all  $\gamma > 0$  and for all  $\delta > 0$ , if*

$$n \geq C_Q(U_P^Q(\gamma), \delta)$$

*then*

$$\text{InStab}(n, P) \leq 2\delta + 2P(T_\gamma(f^*)).$$

*Proof.* By definition of  $C_Q$  we know that if  $n \geq C_Q(U_P^Q(\gamma), \delta)$  then we have that

$$P(|Q(f_n, P) - Q(f^*, P)| \leq U_P^Q(\gamma)) > 1 - \delta.$$

By definition of  $U_P^Q(\gamma)$  we know that if  $|Q(f_n, P) - Q(f^*, P)| \leq U_P^Q(\gamma)$ , then we have that  $D_{\text{boundary}}(f_n, f^*) \leq \gamma$ . Together this means that whenever  $n \geq C_Q(U_P^Q(\gamma), \delta)$  then with probability at least  $1 - \delta$  we have that  $D_{\text{boundary}}(f_n, f^*) \leq \gamma$ . Now the statement of the theorem follows by Proposition 2.  $\odot$

### 3.3 Application to particular objective functions

In this subsection we briefly want to show that the conditions in Theorem 4 are satisfied for many of the commonly used clustering quality functions. The major conditions to investigate are the consistency condition and the condition that  $Q$  is continuous with respect to  $D_{\text{boundary}}$  on  $\mathcal{F}$ .

**$K$ -means objective function.** The empirical  $K$ -means objective function  $Q_n$  on a finite sample of  $n$  points is defined as

$$Q_n(f) = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K \mathbf{1}_{f(X_i)=k} \|X_i - c_k\|^2$$

where  $c_i$  denote the cluster centers. Its continuous counterpart is the quality function

$$Q(f) = \int \sum_{k=1}^K \mathbf{1}_{f(X)=k} \|X - c_k\|^2 dP(X).$$

Assume that on any finite sample, the clustering algorithm returns the global optimizer of the empirical  $K$ -means function. Then it is known that this empirical optimizer converges to the true optimum uniformly over all probability distributions (e.g., Corollary 8 in Ben-David, 2007). (However, note that this guarantee does not apply to the standard  $K$ -means algorithm, which only constructs local optima of the empirical quality function.)

Moreover, the  $K$ -means objective function is continuous with respect to  $D_{\text{boundary}}$ , as can be seen by the following proposition:

**Proposition 5 (Continuity of  $K$ -means wrt.  $D_{\text{boundary}}$ )**

*Let  $\mathcal{X} \subset \mathbb{R}^s$  compact, and  $P$  a probability distribution on  $\mathcal{X}$  with a density with respect to the Lebesgue measure. Then the  $K$ -means quality function  $Q$  is continuous with respect to  $D_{\text{boundary}}$ .*

*Proof.* Assume  $f$  and  $g$  are two  $K$ -means clusterings with distance  $D_{\text{boundary}}(f, g) \leq \gamma$ . W.l.o.g. assume that the labeling of  $g$  is permuted such that outside of the  $\gamma$ -tubes, the labels of  $f$  and  $g$  coincide. Denote the complement of a set  $T$  by  $T^c$ . Then we can compute:

$$\begin{aligned}
Q(g) &= \int \sum_{k=1}^K \mathbf{1}_{g(X)=k} \|X - c_k(g)\|^2 dP(X) \\
&\leq \int \sum_{k=1}^K \mathbf{1}_{g(X)=k} \|X - c_k(f)\|^2 dP(X) \\
&= \int_{T_\gamma(f)^c} \sum_{k=1}^K \mathbf{1}_{g(X)=k} \|X - c_k(f)\|^2 dP(X) \\
&\quad + \int_{T_\gamma(f)} \sum_{k=1}^K \mathbf{1}_{g(X)=k} \|X - c_k(f)\|^2 dP(X) \\
&\quad \text{(now on } T_\gamma(f)^c : f(X) = k \iff g(X) = k) \\
&= \int_{T_\gamma(f)^c} \sum_{k=1}^K \mathbf{1}_{f(X)=k} \|X - c_k(f)\|^2 dP(X) \\
&\quad + \int_{T_\gamma(f)} \sum_{k=1}^K \mathbf{1}_{g(X)=k} \|X - c_k(f)\|^2 dP(X) \\
&\leq Q(f) + \text{diam}(\mathcal{X})^2 \cdot P(T_\gamma(f)).
\end{aligned}$$

By the symmetry in  $f$  and  $g$  this leads to

$$|Q(g) - Q(f)| \leq \text{diam}(\mathcal{X})^2 \cdot \max\{P(T_\gamma(f)), P(T_\gamma(g))\}.$$

Finally, the assumption  $g \triangleleft T_\gamma(f)$  implies that  $T_\gamma(g) \subset T_{2\gamma}(f)$ . Thus we finally get that

$$|Q(f) - Q(g)| \leq \text{diam}(\mathcal{X})^2 \cdot P(T_{2\gamma}(f)),$$

which shows the continuity of  $Q$  at function  $f$ , that is

$$\forall f \forall \gamma \exists \delta \forall g : D_{\text{boundary}}(f, g) \leq \delta \implies |Q(f) - Q(g)| \leq \gamma. \quad \odot$$

**Case of graph cut objective functions.** As an example, consider the normalized cut objective function, which is defined as follows. Let  $s : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$  be a similarity function which is upper bounded by a constant  $C$ . For a given cluster described by the cluster indicator function  $f_k : \mathbb{R}^d \rightarrow \{0, 1\}$ , we set

$$\begin{aligned} \text{cut}(f_k) &:= \text{cut}(f_k, P) := \mathbb{E} f_k(X_1)(1 - f_k(X_2))s(X_1, X_2) \\ \text{vol}(f_k) &:= \text{vol}(f_k, P) := \mathbb{E} f_k(X_1)s(X_1, X_2) \end{aligned}$$

For a clustering function  $f \in \mathcal{F}$  we can then define the normalized cut by

$$\text{Ncut}(f) := \text{Ncut}(f, P) := \sum_{k=1}^K \frac{\text{cut}(f_k)}{\text{vol}(f_k)}.$$

In Bubeck and von Luxburg (2007) it has been proved that there exists an algorithm such that Ncut can be minimized uniformly consistently. So it remains to be shown that Ncut is continuous with respect to  $D_{\text{boundary}}$ .

**Proposition 6 (Continuity of Ncut wrt.  $D_{\text{boundary}}$ )** *Let  $\mathcal{X} \subset \mathbb{R}^s$  compact, and  $P$  a probability distribution on  $\mathcal{X}$  with a density with respect to the Lebesgue measure. For a fixed constant  $C > 0$ , let  $\mathcal{F}_C$  be the space of all clusterings  $f : \mathcal{X} \rightarrow \{1, \dots, K\}$  such that all clusters have a minimal  $P$ -mass  $C$ . Then the Ncut objective function is continuous with respect to  $D_{\text{boundary}}$  on  $\mathcal{F}_C$ .*

*Proof.* The proof is very similar to the one for the  $K$ -means case, thus we just provide a sketch. We consider the numerator and denominator of Ncut separately. As for the  $K$ -means case, one splits the integrals over  $\mathcal{X}$  in a sum of the integrals over  $T_\gamma(f)$  and  $T_\gamma(f)^c$ . Both parts are dominated by the contributions from points in  $T_\gamma^c$ , and the contributions from inside the tubes can be bounded by some constant times the mass in the tubes. This leads to a similar argument as in the  $K$ -means case.  $\odot$

**Explicit form of the constant  $\gamma$ .** We have seen that Theorem 4 can be applied to several of the standard clustering objective functions, such as the  $K$ -means one and the normalized cut. What remains a bit vague is the exact functional form of the constant  $\gamma$  in this theorem. Essentially, this constant is the result of an existence statement in Proposition 3. For the case of  $K$ -means, it is possible to upper bound this constant by using the tools and methods from Meila (2006) and Meila (2007). There it has been proved in a finite sample setting that under certain conditions, if  $|Q(f) - Q(g)|$  is small, then also the  $D_{\text{MinMatch}}(f, g)$  is small. For  $K$ -means, one can show that small  $D_{\text{MinMatch}}(f, g)$  implies small  $D_{\text{boundary}}$ . Furthermore, all quantities used in the finite sample results of Meila (2006) need to be carried over to the limit setting. For example, the eigenvalues of the similarity matrices have to be replaced by eigenvalues of the corresponding limit operators, for example by using results from Blanchard et al. (2007). Combining all those arguments leads to an explicit upper bound for the constant  $\gamma$  in Theorem 4 for the  $K$ -means objective function. However, this upper bound became so technical that we refrain from deriving it in this paper. A similar argument might be possible for the normalized cut, as the results of Meila (2007) also cover this case. However, we have not worked out this case in detail, so we do not know whether it really goes through. If it does, the result is likely to look even more complicated than in the  $K$ -means case.

### 3.4 High-density boundaries do not imply instability

In the following example we demonstrate that, in some sense, the converse of Theorem 4 fails. We construct a data distribution over the two-dimensional plane for which the 2-means clustering has high probability mass in a narrow tube around the optimal clustering boundary, and yet the instability levels converge to zero fast (as a function of the sample sizes).

**Example 1** *Let  $P_\eta^\nu$  be a mixture distribution consisting of the following components (see Figure 1 for illustration). Define the sets  $A = \{-1\} \times [-1, 1]$ ,  $B = \{1\} \times [-1, 1]$ ,  $C = \{(-\eta, 0)\}$ , and  $D = \{(\eta, 0)\}$ . Let  $U_A$  and  $U_B$  be the uniform distributions on  $A$  and  $B$ , and  $\delta_C$  and  $\delta_D$  the probability distributions giving weight*

1 to the point  $C$  and  $D$ , respectively. Define  $P_\eta^\nu = \frac{1}{2}((1-\nu)(U_A + U_B) + \nu(\delta_C + \delta_D))$ . Namely, the distribution that allocates weight  $\nu/2$  to each of the singleton points  $C$  and  $D$ , and the rest of its weight is uniformly spread over the two vertical intervals at  $x = -1$  and at  $x = 1$ .

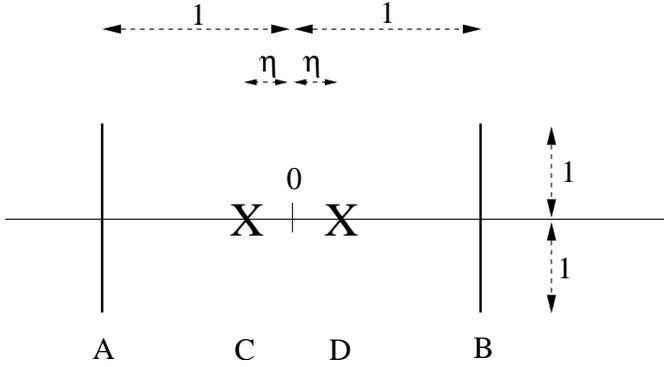


Figure 1: Illustration of Example 1

Clearly, the optimal 2-means clustering,  $f^*$ , divides the plane along the  $y$  axis. It is straight forward to see that if the parameters  $\eta$  and  $\nu$  are, say,  $\eta = 0.01$  and  $\nu = 0.2$ , then the following statements hold:

1. For  $\gamma$  comparable to the variance of  $D_{\text{boundary}}(f_n, f^*)$ , the  $\gamma$ -tube around this optimal boundary includes the points  $C$  and  $D$  and therefore has significant weight, namely  $P(T_\gamma(f^*)) = \nu$ .
2.  $\text{InStab}(n, P_\eta^\nu)$  goes to zero exponentially fast (as a function of  $n$ ).

To see this, note that as long as both of the cluster centers are outside the interval,  $[-1 + \eta, 1 - \eta]$ , the clustering will be fixed (cutting along the  $y$  axis). This condition, in turn, holds whenever the sample  $S$  satisfies

$$|S \cap A| > 20|S \cap C|$$

and

$$|S \cap B| > 20|S \cap D|$$

('20' here just stands for 'many times'). Note that these conditions are implied by having, for every  $T \in \{A, B, C, D\}$ ,

$$\left| \frac{|S \cap T|}{|S|} - P(T) \right| < 0.01$$

Finally, note that, by the Chernoff Bound, the probability (over samples  $S$  of size  $n$ ) that this condition fails is bounded by  $c'e^{-cn}$  for some constants  $c, c'$ . Consequently, for every sample size,  $n$ ,

$$\text{InStab}(n, P_\eta^\nu) \leq c'e^{-cn}$$

3. For any  $\varepsilon > 0$ ,  $P(|Q(f_n) - Q(f^*)| > \varepsilon)$  goes to zero exponentially fast with  $n$ .

Thus, while the preconditions of Theorem 4 hold, in spite of having  $\gamma$ -tubes with significant  $P$ -weight, the instability values are going to zero at an extremely fast rate. The reason is that the sample fluctuations will move the cluster centers up and down in a rather narrow tube around the two vertical intervals. The resulting fluctuations of the empirical clustering boundary will (with overwhelming probability) keep the boundary *between* the points  $C$  and  $D$ . Therefore the instability will practically be zero (no points change cluster membership). On the other hand, those up and down sample-based fluctuations of cluster centers cause the boundary between the two empirical clusters to rotate around the origin point (for example, if the cluster center corresponding to  $A$  sits above the  $x$ -axis, and the center corresponding to  $B$  sits below the  $x$ -axis). Such rotations result in relatively high expected value for the  $D_{\text{boundary}}$  distance between the sample based empirical clusterings and the optimal clustering. These fluctuations could even be made larger by concentrating the probability weight of the two vertical intervals at the end points of these intervals.

Furthermore, the phenomena of having significant weight in  $T_\gamma(f^*)$ , for small  $\gamma$  (i.e., comparable to the variance of the cluster centers) and yet retaining negligible instability can be shown for arbitrarily large sample sizes. Given any sample size  $n$ , one can choose  $\eta$  small enough so that, in spite of the decrease in the expected  $D_{\text{boundary}}$  empirical-to-optimal distances (due to having large samples), the points  $C$  and  $D$  will remain inside the  $T_\gamma(f^*)$ , for  $\gamma$  equal the variance of that  $D_{\text{boundary}}$  distance. Such a choice of parameters can be done while retaining the property that empirical clusterings are unlikely to move these points between clusters, and hence the stability.

**High boundary density version:** Example 1 has large weight on the  $\gamma$ -tube around the boundary of its optimal clustering partition. Yet, the value of the probability density function on the boundary is zero. One can construct a similar example, in which the probability density along the boundary itself is high, and yet the data has close-to-zero instability.

**Example 2** Similar to the example above, we consider a mixture distribution made up of three parts:  $S$  and  $T$  are the vertical intervals  $S = \{-1\} \times [-1/2, 1/2]$  and  $T = \{1\} \times [-1/2, 1/2]$ . However, now the third component is a the rectangle  $R = [-\eta, \eta] \times [-1, 1]$ . Our data space is then defined as  $\mathcal{X} := R \cup S \cup T$ , and as probability distribution we choose  $D_\eta^\nu = (1-\nu)/2 \cdot (U_S + U_T) + \nu \cdot U_R$ . Finally, we define a distance  $d_{\mathcal{X}}$  on this space by letting  $d_{\mathcal{X}}(a, b)$  be the usual Euclidean distance whenever  $a$  and  $b$  belong to the same component of  $\mathcal{X}$ , and  $d_{\mathcal{X}}(a, b)$  is defined as the distance between the projections of  $a$  and  $b$  on the  $x$ -axis whenever  $a$  and  $b$  belong to different components. Note that this metric is not Euclidean and that  $S \cup R \cup T$  is our full domain space, not the real plane.

Once again the optimal 2-means clustering splits the space along the  $y$ -axis. However, now this boundary has significantly high density. Yet, we claim that  $D_\eta^\nu$  instability goes to zero exponentially fast with the sample size. Intuitively, this is because the up and down fluctuations of the centers

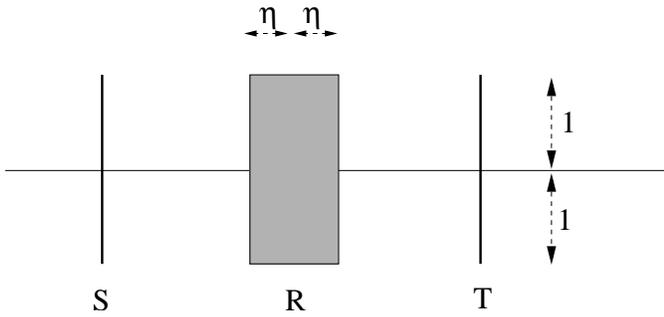


Figure 2: Illustration of Example 2

of the two clusters do not perturb the boundary between the two clusters.

More concretely, say we pick  $\eta < 0.01$  and  $\nu = 0.1$ . We wish to show that, with high probability over the choice of samples, the 2-means optimal sample clustering has its cluster centers in the sets  $S$  and  $T$ . Note that by our choice of distance function, if one center is in  $S$  and the other is in  $T$  then the clustering cuts our domain set along the  $y$  axis (regardless of the  $y$  coordinates of the centers).

Since our domain set equals  $S \cup R \cup T$  (there are no other points in our domain space), it suffices to show that it is unlikely that a sample based clustering will have a cluster center in the set  $R$ . To see that, note that if for some sample  $W$ , the 2-means cost clustering based on  $W$  has a cluster center, say of the left-hand cluster, is in  $R$  then the 2-means cost of that clustering is at least  $|S \cap W|0.99$ . On the other hand, if the center of that cluster is in  $S$  then the 2-means cost of that cluster is at most  $|S \cap W|0.25 + |W \cap R|(1.01)^2$ . It follows that, as long as  $|W \cap R| < 0.11|W|$  the optimal 2-means clustering of the sample  $W$  will have one cluster center in  $S$  and the other cluster center in  $T$ . We can now apply a similar argument to the one used for example 1. Namely, note that as long as the empirical weight of each of the three data components is within 0.01 of its true weight it will indeed be the case that  $|W \cap R| < 0.11|W|$ . It therefore follows, by the Chernoff Bound, that the probability of having a sample  $W$  violate this condition is bounded by  $c'e^{-c|W|}$  for some constants  $c, c'$ . Consequently, except for such minuscule probability, the clustering always splits our domain set along the  $y$  axis. Consequently the 2-means instability of our data distribution is exponentially small (in the sample size).

#### 4 Some inherent limitations of the stability approach in the large sample regime

We consider a setting in which one tries to gain insight into the structure of some unknown data set (or probability distribution over such a set) by sampling i.i.d. from that set. A major question is when can such samples be considered a reliable reflection of structure of that unknown domain. This is the typical setting in which notions of stability are applied. The most common use of stability is as a model selection tool. In that context stability is viewed as an indication that a clustering algorithm does the "right thing" and, in particular, that its choice of number of clusters is "correct". The work

of Shamir and Tishby (2008b) as well as the analysis in this paper claim that stability can be viewed as an indication that the clusters output by an algorithm are "correct" in the sense of having their boundaries pass through low-density data regions.

However, all such results relate the desired clustering properties to the eventual values of stability when the sample sizes grow unboundedly. Since in applications a user always examines finite size samples, the reliability of stability as a model selection tool requires the bound on the rate by which stabilities over  $n$ -size samples converge to their limit values to be uniform over the class of potential data distributions. We show below that no such bounds hold. Arbitrarily large sample sizes can have arbitrarily misleading stability values. The implications of stability values discussed in these papers kick in for sample sizes that depend upon the data distribution, and are therefore not available to the user in most practical applications. We are going to analyze this behavior based on the following example.

**Example 3** Consider the following probability distribution over the two dimensional plane (see Figure 3). Let  $B$  be the disk  $\{(x, y) : (x - 1)^2 + y^2 \leq 1/2\}$ , let  $C$  be the disk  $\{(x, y) : (x + 1)^2 + y^2 \leq 1/2\}$ . Let  $x_0$  be the point  $(0, M)$  for some large positive  $M$  (say,  $M = 100$ ). Given  $\varepsilon > 0$ , let  $P_\varepsilon^M$  be the probability distribution defined as  $P_\varepsilon^M = \varepsilon\delta_{x_0} + (1-\varepsilon)/2(U_B + U_C)$  (in the notation of the example in Section 3.4), where  $\varepsilon$  is some small number, say  $\varepsilon = 0.01$ .

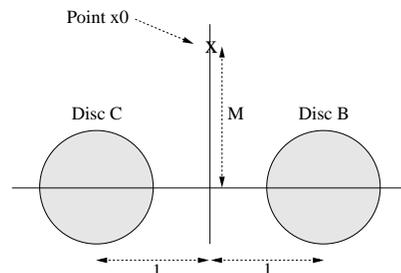


Figure 3: Illustration of Example 3

**No distribution-free stability convergence rates possible.** Consider the distribution of Example 3, and let  $\mathcal{A}$  be an algorithm that finds an optimal 2-means solution for every input data set. For  $n$  rather small, a sample of  $n$  points is rather unlikely to contain the point  $x_0$  as it has a very small mass on it. In those cases, the algorithm  $\mathcal{A}$  will cluster the data by vertically splitting between the disks  $B$  and  $C$ . Thus,  $\text{InStab}(n, P_\varepsilon^M)$  computed on such a data set is very low. However, as the sample size grows, the probability that a sample will contain the point  $x_0$  becomes significant. Now observe that as we chose  $M$  to be very large, then whenever  $x_0$  is a member of a sample  $S$  the optimal 2-clustering of  $S$  will have one of its center points at  $x_0$ . Consequently, as long as  $n$  is such that a significant fraction of the  $n$ -samples pick  $x_0$  and a significant fraction of the samples miss it,  $\text{InStab}(n, P_\varepsilon^M)$  is very high. Finally, when the sample size are large enough to guarantee that hardly any sample misses

$x_0$ , stability is regained.

All in all we have constructed an example of a probability distribution where the 2-means optimizing algorithm is very stable for sample size  $n$ , is very unstable for some sample size  $n' > n$  and converges to perfect stability as the sample sizes go to infinity. By playing with the parameters of  $P_\varepsilon^M$  one can in particular adjust the sample size  $n'$  for which the instable regime holds. As a consequence, there cannot be a distribution-free convergence rate for stability.

It is also worth while to note that throughout the above family of distributions (for all non-degenerate values of  $M$  and  $\varepsilon$ ), the optimal clustering has a wide tube of zero-density around it. Just the same, for arbitrarily large values of  $n'$ ,  $n'$ -size samples display large instability. In particular, this example shows that the assumption “ $D_{\text{boundary}}(f_n, f) \leq \gamma$ ” in Proposition 2, is indeed necessary.

### Stability does not imply close-to-optimal clustering cost.

Proposition 2 states that when sample sizes are such that the sample based clustering quality is close to its optimal value, and if that optimum is achieved with low-density tubes, then the value of instability is low. Example 3 shows that the converse of this statement does not always hold. For data distributions of the form  $P_\varepsilon^M$ , due to having a far outlier,  $x_0$ , when a sample misses that outlier point, the cost of the sample-based clusterings is at least  $M^2\varepsilon$ . On the other hand, the cost of the optimal clustering (that allocates a center to cover the outlier point) is less than  $3(1 - \varepsilon)$ . As long as  $\varepsilon \leq 1/n^2$ , samples are unlikely to hit  $x_0$  and therefore  $\text{InStab}(n, P_\varepsilon^M)$  is very low. However, if  $M$  is picked to be greater than, say  $10/\varepsilon$  we get a large gap between the cost of the sample based clustering and the cost of the distribution-optimal clustering.

**Stability does not imply proximity between the sample based clustering and the optimal clustering.** Again, it can be readily seen that the above family of  $P_\varepsilon^M$  data distributions demonstrates this point as well.

**Stability is not monotone as a function of the sample sizes.** Clearly Example 3 demonstrates such non-monotonicity. The values of  $\text{InStab}(n, P_\varepsilon^M)$  decrease with  $n$  for  $n < 1/\sqrt{\varepsilon}$ , they increase with  $n$  for values of  $n$  round  $1/\varepsilon$  and they decrease to zero for  $n \geq 1/\varepsilon^2$ .

We end this section with a few further observations demonstrating the somewhat “erratic behavior” of stability.

**No uniform convergence of cluster centers to a normal distribution.** Although Pollard (1982) has proved that as the sample sizes grow to infinity, the distribution of the empirical cluster centers converges to a normal distribution, there is no uniform bound on the rate of this convergence. For example, consider a two-mode probability distribution over the real line that has high peaks of its density function at the points  $(0, -\varepsilon)$  and  $(0, \varepsilon)$ , has 0 density for  $x = 0$ , and then tails off smoothly as  $|x|$  goes to infinity. Obviously, for every sample size,  $n$ , by choosing small enough  $\varepsilon$ , the

distribution of each of the cluster centers for 2-means of random  $n$ -samples drawn from this distribution is highly non-symmetric (it has higher variance in the direction away from the 0 than its variance towards 0), and therefore far from being a normal distribution.

### Arbitrarily slow convergence of stability for ‘nice’ data.

Even when data is stable and has a rather regular structure (no outliers like in the example discussed above), and the optimal boundaries pass through wide low-density data regions, the convergence to this stability, although asymptotically fast, is not uniformly bounded over different (well structured) data distributions. For every  $n$  there exists a data distribution  $D_n$  that enjoys the above properties, and yet  $\text{InStab}(n, D_n)$  is large. As an example of this type of non-uniformity, consider a planar distribution having its support on four small (say, of radius 0.1) discs centered on the four corners of the unit square. Assume the distribution is uniform over each of the discs, is symmetric around the  $x$  axis, but gives slightly more weight to the left hand side two disks than to the right hand side disks. For such a distribution, the optimal 2-means clustering is a unique partition along the  $x$  axis, and has wide 0-density margins around its boundary. Just the same, as long that the sample sizes are not big enough to detect the asymmetry of the distribution (around the  $y$  axis), a significant fraction of the sample based 2-means clustering will pick a partition along the  $y$  axis and a significant fraction of samples will pick a partition along the  $x$  axis, resulting in high instability. This instability can be made to occur for arbitrarily large sample sizes, by just making the asymmetry of the data sufficiently small.

## 5 Discussion

In this paper, we discuss the mechanism of stability-based model selection for clustering. The first part of the paper investigates a promising conjecture: in the large sample regime, the stability of a clustering algorithm can be described in terms of properties of the cluster boundary, particularly whether the boundary lies in a small or high density area. In the case of  $K$ -means, this would explain the success of stability-based methods by demonstrating that stability adds the “the missing piece” to the algorithm. As the  $K$ -means clustering criterion is only concerned by within-cluster similarity, but not with between-cluster dissimilarity, a model selection criterion based on low density areas would add a valuable aspect to the algorithm.

In parts, our results are promising: the conjecture holds at least in one direction. However, it is pretty discouraging that the conjecture does not hold the other way round, as we can show by a simple counterexample. This counterexample also indicates that a simple mechanism such as “low density” vs. “high density” does not exist. So, after all, the question which are the underlying geometric principles of stability-based model selection in the large sample regime remains unanswered.

On the other hand, we also provide some reasons why using stability-based methods in the large sample setting might be problematic in general. The reason is that it is impossible to

give global convergence guarantees for stability. Thus, while one can use stability criteria in practice, it is impossible to give distribution-free performance guarantees on any of its results. No matter how large our sample size  $n$  is, we can always find distributions where the stability evaluated on that particular sample size is misleading, in the sense that it is far from the “true stability”

Finally, we would like to put our results in a broader context and point out future research directions for investigating stability. In general, there are different reasons why cluster instability can arise:

**Instability due to multiple global optima.** If the global optimizer of the clustering objective function is not unique, this always leads to instability. However, this kind of instability is usually not related to the correct number of clusters, as has been proved in Ben-David et al. (2006), Ben-David et al. (2007). Instead, it might depend on completely unrelated criteria, for example symmetries in the data. In this situation, stability criteria are not useful for selecting the number of clusters.

**Geometric instability in the large sample setting.** This is the kind of instability we considered in this paper. Here one assumes that no issues with local optima exist, that is the algorithm always ends up in the global optimum, and that a unique global optimum exists (for all values of  $K$  under consideration). In this paper, we made an attempt to connect the mechanism behind stability-based model selection to geometric properties of the underlying distribution and clustering, but with moderate success only. On the other hand, we can demonstrate that using stability in the large sample setting has problems in general. While it might be possible that future work shows a tighter connection between geometric properties of the data space and stability issues, we are doubtful whether those methods can be applied successfully in practice, unless one makes strong assumptions on the underlying distributions.

**Instability due to too small sample size.** If the sample size is too small, and the cluster structure is not sufficiently well pronounced in the data set, we will observe instability. Here, clustering stability can be a useful criterion to detect whether the number of clusters is much too high. If this is the case, the algorithm will construct clusters which are mainly based on sampling artifacts, and those clusters will be rather unstable. Here, stability tells us whether we have enough data to support a given cluster structure. This is of course a useful thing to know. However, it is still not obvious whether stability can be used to detect the “best” number of clusters, as there might be several values of  $K$  which lead to stable results. We believe that it is a very important direction to investigate what guarantees can be given on stability-based methods in this scenario.

**Algorithmic instability.** This kind of instability occurs if the algorithm itself can converge to very different solutions, for example it ends up in different local optima, depending on starting conditions. Note that algorithmic instability

is rather a property of an algorithm than of an underlying distribution or sample. If we had a perfect algorithm which always found the global optimum, then this kind of instability would not occur. In our opinion, in a setting of algorithmic instability it is not clear that stability selects the “best” or “correct” number of clusters. Essentially, in this case stability simply detects whether there is a well-pronounced local optimum where the objective function has the shape of a “wide bowl” such that the algorithm gets trapped in this local optimum all the time. However, we find it unlikely that the conclusion “local optimum in wide bowl implies good  $K$ ” is true. It has been argued that the conclusion the other way round is true: “distribution with well-pronounced cluster structure implies global optimum in wide bowl” (e.g., Meila, 2006 or Srebro et al., 2006). However, this is not the direction which is needed to show that clustering stability is a good criterion to select the number of clusters. We conclude that in the “algorithmic instability” scenario, stability is not very well understood, and it would be very interesting to give conditions on distributions and algorithms in which this kind of stability can provably be useful for model selection.

In all settings discussed above, stability is useful in one respect: high instability can be used as an alarm sign to distrust the clustering result, be it for sampling, algorithmic or other reasons. However, the other way round, namely that the most stable algorithm leads to the best clustering result, so far has not been established for any of the settings above in a satisfactory way.

## Acknowledgments

We are grateful to Markus Maier who pointed out an error in an earlier version of this manuscript, and to Nati Srebro and David Pal for insightful discussions.

## References

- S. Ben-David. A framework for statistical clustering with constant time approximation algorithms for k-median and k-means clustering. *Machine Learning*, 66:243 – 257, 2007.
- S. Ben-David, U. von Luxburg, and D. Pál. A sober look on clustering stability. In G. Lugosi and H. Simon, editors, *Proceedings of the 19th Annual Conference on Learning Theory (COLT)*, pages 5 – 19. Springer, Berlin, 2006.
- S. Ben-David, D. Pál, and H.-U. Simon. Stability of k-means clustering. In N. Bshouty and C. Gentile, editors, *Conference on Learning Theory (COLT)*, pages 20–34. Springer, 2007.
- A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, pages 6 – 17, 2002.
- A. Bertoni and G. Valentini. Model order selection for bio-molecular data clustering. *BMC Bioinformatics*, 8(Suppl 2):S7, 2007.

- M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Bendor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, M. Hayward, and J. Trent. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, 406: 536 – 540, 2000.
- G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2-3):259–294, 2007.
- S. Bubeck and U. von Luxburg. Overfitting of clustering and how to avoid it. Preprint, 2007.
- J. Fridlyand and S. Dudoit. Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method. Technical Report 600, Department of Statistics, University of California, Berkeley, 2001.
- M. K. Kerr and G. A. Churchill. Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments. *PNAS*, 98(16):8961 – 8965, 2001.
- A. Krieger and P. Green. A cautionary note on using internal cross validation to select the number of clusters. *Psychometrika*, 64(3):341 – 353, 1999.
- T. Lange, V. Roth, M. Braun, and J. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6):1299 – 1323, 2004.
- E. Levine and E. Domany. Resampling Method for Unsupervised Estimation of Cluster Validity. *Neural Computation*, 13(11):2573 – 2593, 2001.
- M. Meila. Comparing clusterings: an axiomatic view. In *Proceedings of the International Conference of Machine Learning (ICML)*, pages 577–584, 2005.
- M. Meila. The uniqueness of a good optimum for K-means. In W. Cohen and A. Moore, editors, *Proceedings of the Twenty-Third International Conference on Machine Learning (ICML)*, pages 625–632. ACM, 2006.
- M. Meila. The stability of a good clustering. Manuscript in preparation, 2007.
- R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of Lloyd-type methods for the k-means problem. In *FOCS*, pages 165–176. IEEE Computer Society, 2006.
- D. Pollard. A central limit theorem for k-means clustering. *Annals of Probability*, 10(4):919 – 926, 1982.
- O. Shamir and N. Tishby. Model selection and stability in k-means clustering. In *Conference on Learning Theory (COLT)*, to appear, 2008a.
- O. Shamir and T. Tishby. Cluster stability for finite samples. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems (NIPS) 21*. MIT Press, Cambridge, MA, 2008b.
- M. Smolkin and D. Ghosh. Cluster stability scores for microarray data in cancer studies. *BMC Bioinformatics*, 36(4), 2003.
- N. Srebro, G. Shakhnarovich, and S. Roweis. An investigation of computational and informational limits in Gaussian mixture clustering. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 865 – 872. ACM Press, New York, 2006.
- U. von Luxburg, S. Bubeck, S. Jegelka, and M. Kaufmann. Consistent minimization of clustering objective functions. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems (NIPS) 21*, Cambridge, MA, 2008. MIT Press.

---

# Finding Metric Structure in Information Theoretic Clustering

---

**Kamalika Chaudhuri**

University of California, San Diego  
kamalika@soe.ucsd.edu

**Andrew McGregor**

University of California, San Diego  
andrewm@ucsd.edu

## Abstract

We study the problem of clustering discrete probability distributions with respect to the Kullback-Leibler (KL) divergence. This problem arises naturally in many applications. Our goal is to pick  $k$  distributions as “representatives” such that the average or maximum KL-divergence between an input distribution and the closest representative distribution is minimized. Unfortunately, no polynomial-time algorithms with worst-case performance guarantees are known for either of these problems.

The analogous problems for  $l_1$ ,  $l_2$  and  $l_2^2$  (i.e.,  $k$ -center,  $k$ -median and  $k$ -means) have been extensively studied and efficient algorithms with good approximation guarantees are known. However, these algorithms rely crucially on the (geo-)metric properties of these metrics and do not apply to KL-divergence. In this paper, our contribution is to find a “relaxed” metric-structure for KL-divergence. In doing so, we provide the first polynomial-time algorithm for clustering using KL-divergences with provable guarantees for general inputs.

## 1 Introduction

In this paper, we consider the problem of clustering discrete probability distributions with respect to the Kullback-Liebler (KL) divergence where, the KL-divergence from  $p = (p_1, \dots, p_d)$  to distribution  $q = (q_1, \dots, q_d)$  is defined as

$$\text{KL}(p, q) = \sum_{i \in [d]} p_i \ln \frac{p_i}{q_i} .$$

Specifically, we consider two problems that take  $n$  distributions  $p^1, \dots, p^n$  on  $[d]$  as input. In  $\text{MTC}_{\text{KL}}$  (minimum total cost), the goal is to find distributions  $c^1, \dots, c^k$  such that the total KL-divergence from each  $p^j$  to its closest  $c^i$ , i.e.,

$$\sum_{j \in [n]} \min_{i \in [k]} \text{KL}(p^j, c^i)$$

is minimized. In  $\text{MMC}_{\text{KL}}$  (minimum maximum cost), the goal is to find distributions  $c^1, \dots, c^k$  such that the maximum KL-divergence from each  $p^j$  to its closest  $c^i$ ,

$$\max_{j \in [n]} \min_{i \in [k]} \text{KL}(p^j, c^i)$$

is minimized. It turns out that polynomial time algorithms do not exist for either of these problems unless  $P = NP$ . Therefore, we are interested in  $\alpha$ -approximation algorithms, i.e., algorithms that find  $\tilde{c}^1, \dots, \tilde{c}^k$  satisfying the guarantee that

$$\frac{\sum_{j \in [n]} \min_{i \in [k]} \text{KL}(p^j, \tilde{c}^i)}{\min_{c^1, \dots, c^k} \sum_{j \in [n]} \min_{i \in [k]} \text{KL}(p^j, c^i)} \leq \alpha$$

for some  $\alpha \geq 1$ . The smaller the value of  $\alpha$ , the better the approximation.

Both problems have been studied extensively when the input is a set of arbitrary points (not necessarily distributions), and instead of KL, the measure of distance between two points is either a metric ( $l_1$  or  $l_2$  or an arbitrary metric), with symmetry and the triangle inequality, or a measure such as  $l_2^2$ . The problems are usually referred to as  $k$ -median if the measure is a metric or  $k$ -means if the measure is  $l_2^2$ . However, previous algorithms for these problems typically rely crucially on the (geo-)metric properties of these distances, which do not hold for the KL-divergence. For example, KL is not symmetric and does not satisfy the triangle inequality.

In the remainder of the introduction, we motivate the need to cluster distributions and the reason why KL is a natural measure in this context. We then review the related work and summarize our contributions.

**Why cluster distributions?** A natural application of distributional clustering is in clustering words for document classification by topic [4]. In document classification, we are given a training set of *documents* (or collections of words) whose labels indicate the topic they represent, and the goal is to classify other similar documents according to topic. A natural approach is to look at the words in a document as *features* that are somehow correlated with the document labels; each word is viewed as a frequency distribution over labels, and given a new document containing a set of words, the distributions corresponding to the words in it are used to find

the most-likely label for the new document. However, such data is typically very sparse, because each specific word occurs a few times in the document corpora. So a common approach is to cluster together similar word distributions, for more robust inference algorithms.

Other applications of distributional clustering include clustering words according to context for language modeling [24], information bottleneck techniques [27, 24, 25], and clustering users according to their preference for movies in collaborative filtering.

**Why KL-divergence?** KL-divergence arises as a natural measure of the dissimilarity between two distributions in numerous ways. We direct the interested reader to Pereira et al. [24] for a wider discussion on the motivations. In what follows, we describe the motivation in terms of compressibility.

Given an alphabet  $\Sigma$  of size  $d$  where the  $i$ -th symbol has relative frequency  $p_i$ , an important question is to find the binary encoding of the alphabet such the average number of bits required for an encoded symbol is minimized. This classic problem in information theory was essentially solved by Huffman who presented a simple encoding scheme that achieved the optimum value of

$$H(p) = - \sum_{i \in [d]} p_i \lg p_i$$

if all  $p_i$  were negative powers of two.

We consider an issue that arises when we have two or more distributions over  $\Sigma$ . Consider the problem of trying to encode multiple texts with different statistics such as texts written in different languages or magazine articles covering different topics. For example, the word “perforation” may be common in articles from *Gibbons Stamp Monthly Magazine*<sup>1</sup> whereas “peroxide” may be more frequent in issues of *Hairdressers Journal International*<sup>2</sup>. Hence, if the origin of the text is known it will make sense to tailor the encoding to the statistics of the the source. However, it is likely to be unfeasible to have a different scheme for every possible periodical. Rather, we consider the problem of designing  $k$  encoding schemes and assigning each of  $n$  periodicals to one of the encoding schemes. How should this be done such that extra cost of using  $k$  encoding schemes rather than  $n$  is minimized?

More formally, let  $p^j$  be distribution over symbols in the  $j$ -th periodical. We wish to design  $k$  encoding schemes  $E_1, \dots, E_k : \Sigma \rightarrow \{0, 1\}^*$  along with an assignment of distributions to encoding schemes  $f : [n] \rightarrow [k]$  such that the increase in average encoding length,

$$\sum_{j \in [n]} \sum_{i \in [d]} p_i^j |E_{f(j)}(i)| + \sum_{j \in [n]} \sum_{i \in [d]} p_i^j \lg p_i^j$$

is minimized. Each encoding scheme  $E_j$  can be characterized by a distribution  $q^j$  over  $[d]$  that will capture

<sup>1</sup><http://www.gibbonsstampmonthly.com/>

<sup>2</sup><http://www.hji.co.uk/>

the aggregate statistics of the distributions that use  $E_j$ . Hence we may rewrite the quantity to be minimized as

$$\begin{aligned} & - \sum_{j \in [n]} \sum_{i \in [d]} p_i^j \lg q_i^{f(j)} + \sum_{j \in [n]} \sum_{i \in [d]} p_i^j \lg p_i^j \\ & = \sum_{j \in [n]} \sum_{i \in [d]} p_i^j \lg \frac{p_i^j}{q_i^{f(j)}} = (\lg e) \sum_{j \in [n]} \text{KL}(p^j, q^{f(j)}) \end{aligned}$$

which is exactly the objective function to be minimized in  $\text{MTC}_{\text{KL}}$ .

## 1.1 Prior Work on Clustering

There has been a rich body of research on approximation algorithms for various forms of clustering. We restrict ourselves to those on hard-clustering, i.e., each input distribution is “assigned” to only the closest picked center. Even so, there is a considerable number of incomparable results in a variety of settings.

The common optimization measures when clustering points in general metrics are (a)  $k$ -median, in which the goal is to partition the input points into  $k$  sets, while minimizing the sum of the distances between each point and the center of the cluster it is assigned to, and (b)  $k$ -center, where the goal is to again partition the input points to  $k$  sets, while minimizing the maximum diameter of a cluster. When clustering in Euclidean spaces, an additional optimization measure which is commonly used is  $k$ -means, in which the goal is to partition the input points into  $k$  clusters, while minimizing the sum of the squares of the Euclidean distances between each point and the center of the cluster it is assigned to.

**General “Metrics”:** For metric  $k$ -center, the best approximation algorithm is due to [16], which achieves an approximation factor of 2 and this is the best possible in polynomial time unless  $P = NP$ . For asymmetric  $k$ -center, when the directed triangle inequality holds, the best known approximation algorithm is due to [23], which achieves a factor of  $O(\log^* n)$ , and this is also optimal in terms of hardness [7]. For metric  $k$ -median, the best known approximation algorithm is due to [2], which achieves an approximation factor of 3, when the distances between points are symmetric, and there is a triangle inequality.

**Euclidean Space:** When the input points lie in Euclidean space, two versions of the clustering problems have been studied. In the *restricted version*, we require the cluster centers to be input points, while in the *unrestricted version*, we allow the cluster centers to be any point in the Euclidean space. For more details about restricted and unrestricted versions of the problems, see Section 5. Most results for clustering in Euclidean space deal with the unrestricted version of the problem.

When the input points lie in  $d$ -dimensional Euclidean spaces, Kolliopoulos and Rao [21] showed an algorithm for  $k$ -median which provides a  $(1 + \epsilon)$  approximation, and runs in time

$$O(2^{(O(1+\epsilon^{-1} \log \epsilon^{-1}))^{d-1}} n \log k \log n) .$$

Har-Peled and Mazumdar [19] gave a  $(1 + \epsilon)$  approximation algorithm which runs in time

$$O(n + 2^{O(1+\epsilon^{-1} \log \epsilon^{-1})^{d-1}} k^{O(1)} \log^{O(1)} n).$$

A third algorithm was proposed by Badoiu et al. [3] with a running time of

$$O(d^{O(1)} n \log^{O(k)} n 2^{O(k/\epsilon)}).$$

For Euclidean  $k$ -means, Har-Peled and Mazumdar [19] provided an  $(1 + \epsilon)$  approximation algorithm with running time

$$O(n + (\epsilon^{-1})^{2d+1} k^{k+2} \log^{k+1} n \log^k \epsilon^{-1}).$$

A second  $(1 + \epsilon)$  approximation algorithm is due to Feldman et al. [15], which achieves a running time of

$$O(ndk + d(k\epsilon^{-1})^{O(1)} + 2^{O(k\epsilon^{-1})}).$$

Kumar et al. [22] provided a simple algorithm based on sampling for Euclidean  $k$ -means which gave a  $(1 + \epsilon)$ -approximation in

$$O(dn 2^{\text{poly}(k\epsilon^{-1})})$$

time. This was improved by Chen [6] to provide an algorithm which ran in

$$O(ndk + d^2 n^\sigma 2^{\text{poly}(k\epsilon^{-1})})$$

time, for any  $\sigma > 0$ . Kanungo et al. [20] gives a  $(9 + \epsilon)$ -approximation for  $k$ -means in time  $O(n^3/\epsilon^d)$ . For Euclidean  $k$ -center, Feder and Greene [13] show that it is NP-Hard to find an approximation-factor better than 1.822 for this problem.

**KL-clustering:** In this paper we are interested in KL-clustering on the probability simplex. We first note that algorithms that cluster distributions with respect to either  $\ell_1$  or  $\ell_2^2$  may give arbitrarily bad solutions for the KL-divergence. The following example shows this for  $\text{MTC}_{\text{KL}}$ .

**Example 1** Consider the following three distributions:

$$p = \left(\frac{1}{2}, \frac{1-\epsilon}{2}, \frac{\epsilon}{2}\right), \quad q = \left(\frac{1}{2}, \frac{1}{2}, 0\right), \quad r = \left(\frac{1}{2}+\epsilon, \frac{1}{2}-\epsilon, 0\right).$$

We consider the costs of all possible partitions of  $\{p, q, r\}$  into two groups.

Clustering	$\ell_2^2$ -cost	$\ell_1$ -cost	KL-cost
$\{p, q\}, \{r\}$	$\epsilon^2/4$	$\epsilon$	$\epsilon/2 + O(\epsilon^2)$
$\{p\}, \{q, r\}$	$\epsilon^2$	$2\epsilon$	$O(\epsilon^2)$
$\{p, r\}, \{q\}$	$3\epsilon^2/4$	$2\epsilon$	$\epsilon/2 + O(\epsilon^2)$

Note that the clustering  $\{\{p, q\}, \{r\}\}$  minimizes the  $\ell_2^2$  or  $\ell_1$  cost but that this clustering is a factor  $\Omega(1/\epsilon)$  from optimal in terms of  $\text{MTC}_{\text{KL}}$ . Since  $\epsilon$  may be made arbitrarily small, we conclude that clustering the distributions according to either  $\ell_2^2$  or  $\ell_1$  can lead to arbitrarily bad solutions.

There has been previous work on methods for KL-clustering [24, 4, 26, 5, 11]. However, none of these algorithms achieve guaranteed approximations in the worst case. The most directly relevant paper is a recent paper by Ackermann et al. [1]. They present a very nice algorithm that returns a good approximation for  $\text{MTC}_{\text{KL}}$  on the assumption that all distributions to be clustered have constant mass on each coordinate, i.e., for some constant  $\gamma$ ,  $p_i^j \geq \gamma$  for all  $j \in [t], i \in [d]$ . This implies that  $d \leq 1/\gamma$  is also constant and even for distributions with constant dimension, rules out any sparse data where some coordinates will have zero mass. Sparse data is common in many applications. In contrast, the algorithms we present are fully general and require no assumptions on the sparsity or the dimensionality of the input distributions.

## 1.2 Our Contributions

Our main contribution in this paper is to provide algorithms for clustering in the KL-divergence measure which achieve guaranteed approximations in the worst case. Our specific contributions are the following:

1. *Minimizing Average Distortion:* We provide the first guaranteed approximation algorithm for the problem of minimizing average distortion in the KL-divergence measure, when the input is a set of  $n$  arbitrary distributions. To show our result, we first provide constant factor approximation algorithms for the related divergences, Hellinger and Jensen-Shannon. These results exploit the fact that these divergences satisfy a relaxation of the triangle inequality and are closely related to the  $k$ -means problem on the sphere. We then show that although the KL-divergence between two distributions can be infinitely larger than the Jensen-Shannon or Hellinger divergence, one can relate the average clustering distortion in terms of the Hellinger cost to the average clustering distortion in terms of the KL-divergence. This yields an  $O(\log n)$ -approximation algorithm for  $\text{MTC}_{\text{KL}}$ .

We note that while a guarantee of  $O(\log n)$ -factor from optimality is weaker than we would like, this does not preclude the possibility that the algorithm achieves better results in practice. Furthermore, the clustering found could be used as a preprocessing step for an improvement heuristic for which there exist no guarantees. The most important contribution of a  $O(\log n)$ -factor approximation is to understanding the structure of the problem.

2. *Minimizing Maximum Distortion:* We provide the first guaranteed approximation algorithm for minimizing the maximum distortion, when the input is a set of  $n$  arbitrary distributions. To show our result, we relate the maximum clustering distortion in terms of the KL-divergence to the maximum diameter of a cluster measured in terms of the JS-divergence. We then show a constant factor

approximation to the problem of minimizing the JS diameter. This yields an  $O(\min(\log n, \log d))$ -approximation algorithm for  $\text{MMC}_{\text{KL}}$ .

3. *Hardness Results:* Finally, we provide hardness results for the above problems. First, we show that when we restrict the cluster centers to be in the set of input distributions, no polynomial-time approximation is possible, unless  $P \neq NP$ . In addition, when the centers are unrestricted, we show a hardness of approximation result for  $k$ -center by demonstrating that KL behaves like  $\ell_2^2$  near the middle of the probability simplex.

**Notation:** We denote the probability simplex over  $\mathbb{R}^d$  as  $\Delta$ . We write  $a = b \pm c$  as short hand for  $a \in [b - c, b + c]$ .

## 2 Information Geometry

In this section we review some known results about the geometry of KL and prove some new results. As we mentioned,  $\text{KL}(p, q)$  is asymmetric, does not satisfy a directed triangle inequality, and can be infinite even if  $p$  and  $q$  are on the probability simplex. (It is, however, at least always positive by Gibb's inequality.) Furthermore, KL does not even satisfy a relaxed directed triangle inequality, that is

$$\frac{\text{KL}(p, r) + \text{KL}(r, q)}{\text{KL}(p, q)}$$

can be made arbitrarily small with  $p, q, r \in \Delta$ .<sup>3</sup> The following example demonstrates this.

**Example 2** KL is not a relaxed metric. Consider

$$p = (1/2, 1/2), q = (e^{-c}, 1 - e^{-c}), r = (\epsilon, 1 - \epsilon)$$

where  $1/2 \geq \epsilon > e^{-c}$ . Then

$$\begin{aligned} \text{KL}(p, q) &\geq c/2 - \ln 2 \\ \text{KL}(p, r) &\leq (\ln \epsilon^{-1} - \ln 2)/2 \\ \text{KL}(r, q) &\leq \epsilon c - 1 \end{aligned}$$

Hence, by increasing  $c$  and decreasing  $\epsilon$ , the ratio

$$(\text{KL}(p, r) + \text{KL}(r, q))/\text{KL}(p, q)$$

can be made arbitrarily small.

Two other information divergences that will play an important role in our results are the Hellinger and Jensen-Shannon divergences. These are both divergences from the family of  $f$ -divergences [10].

**Definition 1** The Hellinger and Jensen-Shannon divergence between  $p, q \in \Delta$  are defined as

$$\begin{aligned} \text{He}(p, q) &= \sum_{i \in [d]} (\sqrt{p_i} - \sqrt{q_i})^2 \\ \text{JS}(p, q) &= \text{KL}(p, \frac{p+q}{2}) + \text{KL}(q, \frac{p+q}{2}). \end{aligned}$$

<sup>3</sup>We note that this ratio can be bounded below for some families of distributions in terms of the ratio of eigenvalues of a related Hessian matrix [9].

Both  $\text{JS}(p, q)$  and  $\text{He}(p, q)$  are symmetric and bounded: it can easily be shown that  $\text{JS}(p, q) \leq 2$  and  $\text{He}(p, q) \leq 2$  for all  $p, q \in \Delta$ . Note that since  $\text{KL}(p, q)$  may be infinite this rules out any multiplicative relationship in general.

Relationships between  $\text{JS}(p, q)$  and  $\text{He}(p, q)$  are given in the next lemma [18, 28].

**Lemma 2** For all distributions  $p$  and  $q$ ,

$$\text{He}(p, q)/2 \leq \text{JS}(p, q) \leq 2 \ln(2) \text{He}(p, q). \quad (1)$$

Unfortunately, neither JS or He are metrics but we can show that they are ‘‘almost metrics’’ in that they satisfy non-negativity, identity of indiscernibles, symmetry, and a relaxation of the triangle inequality. We say that a measure  $D$  satisfies an  $\alpha$ -relaxed triangle inequality if for all  $p, q, r \in \Delta$ ,

$$D(p, r) + D(r, q) \geq D(p, q)/\alpha.$$

(When  $\alpha = 1$ , this is the usual triangle inequality.)

**Lemma 3** He and JS obey the 2-relaxed triangle equality.

**Proof:** We note that He and JS are both the square of metrics: this is obvious for He and the result for JS was proved in [12]. Therefore, for all  $p, q, r \in \Delta$ ,

$$\sqrt{\text{He}(p, q)} + \sqrt{\text{He}(q, r)} \geq \sqrt{\text{He}(p, r)}$$

and hence

$$\text{He}(p, q) + \text{He}(q, r) + 2\sqrt{\text{He}(p, q)\text{He}(q, r)} \geq \text{He}(p, r).$$

By an application of the AM-GM inequality we deduce:

$$2(\text{He}(p, q) + \text{He}(q, r)) \geq \text{He}(p, r)$$

as required. The result for JS follows similarly. ■

The next lemma is a well-known identity (see, e.g., [8]) that relates the KL and JS divergence.

**Lemma 4** For all  $p, q, c \in \Delta$ :

$$\text{KL}(p, c) + \text{KL}(q, c) = \text{JS}(p, q) + 2\text{KL}((p+q)/2, c).$$

This is referred to as the parallelogram property.

Another useful property that we will exploit is that the He-balls are convex.

**Lemma 5**  $B_\ell(p) = \{p' : \text{He}(p, p') \leq \ell\}$  is convex for all  $\ell \geq 0$  and  $p \in \Delta$ . Furthermore, for all  $p, q, r \in \Delta$  and  $\alpha \in (0, 1)$ ,

$$\text{He}(p, \alpha q + (1 - \alpha)r) \leq \alpha \text{He}(p, q) + (1 - \alpha) \text{He}(p, r).$$

**Proof:** Consider any ball  $B_\ell(p) = \{p' : \text{He}(p, p') \leq \ell\}$  and let  $q, s \in B_\ell(p)$  and  $\alpha \in (0, 1)$ . Then it suffices to show that  $\alpha q + (1 - \alpha)r \in B_\ell(p)$ . Let  $\beta = 1 - \alpha$ . Note that

$$\frac{\alpha(\sqrt{p_i} - \sqrt{q_i})^2 + \beta(\sqrt{p_i} - \sqrt{r_i})^2}{(\sqrt{p_i} - \sqrt{\alpha q_i + \beta r_i})^2} \geq 1$$

$$\Leftrightarrow \alpha\sqrt{q_i} + \beta\sqrt{r_i} \leq \sqrt{\alpha q_i + \beta r_i}$$

$$\Leftrightarrow \alpha^2 q_i + \beta^2 r_i + 2\alpha\beta\sqrt{q_i r_i} \leq \alpha q_i + \beta r_i$$

$$\Leftrightarrow 2\alpha\beta\sqrt{q_i r_i} \leq \alpha\beta q_i + \alpha\beta r_i$$

$$\Leftrightarrow 2\sqrt{q_i r_i} \leq q_i + r_i$$

and this is true by the AM-GM inequality. ■

**Properties of Cluster Centers:** For the remaining result of this section we need to introduce some further notation. For any measure  $D : \Delta \times \Delta \rightarrow \mathbb{R}^+$ :

$$\text{SumCost}_D(p^1, \dots, p^t; c) = \sum_{j \in [t]} D(p^j, c)$$

$$\text{SumCost}_D(p^1, \dots, p^t) = \min_{c \in \Delta} \text{SumCost}_D(p^1, \dots, p^t; c)$$

$$\text{MaxCost}_D(p^1, \dots, p^t; c) = \max_{j \in [t]} D(p^j, c)$$

$$\text{MaxCost}_D(p^1, \dots, p^t) = \min_{c \in \Delta} \text{MaxCost}_D(p^1, \dots, p^t; c)$$

We denote the centroid of a set of distributions as

$$\text{cent}(p^1, \dots, p^t) = t^{-1} \sum p^i.$$

The next lemma (a special case of more general result for all Bregman divergences [5]) shows that the center that minimizes the average  $\ell_2^2$  or KL distortion is the centroid of the distributions being clustered.

**Lemma 6** For any distributions  $p^1, \dots, p^t$ ,

$$\begin{aligned} \text{cent}(p^1, \dots, p^t) &= \operatorname{argmin}_{q \in \Delta} \text{SumCost}_{\ell_2^2}(p^1, \dots, p^t; q) \\ &= \operatorname{argmin}_{q \in \Delta} \text{SumCost}_{\text{KL}}(p^1, \dots, p^t; q), \end{aligned}$$

i.e., the cluster centers for  $\ell_2^2$  and KL are at centroids.

The next lemma shows that when we are clustering distributions near the middle of the probability simplex, the centers that minimize either the maximum or average KL distortion also lie near the middle of the probability simplex. Define,

$$A(r) = \{p \in \Delta : p_j = \frac{1}{d} \pm r \text{ for all } j \in [d]\}. \quad (2)$$

**Lemma 7** Let  $p^1, \dots, p^t \in A(\epsilon/d)$  and  $0 < \epsilon < 1/10$ . Then,

$$\operatorname{argmin}_{c \in \Delta} \text{SumCost}_{\text{KL}}(p^1, \dots, p^t; c) \in A(\epsilon/d).$$

If

$$\frac{\text{MaxCost}_{\text{KL}}(p^1, \dots, p^t; c)}{\text{MaxCost}_{\text{KL}}(p^1, \dots, p^t)} \leq 10$$

then  $c \in A(10\sqrt{\epsilon})$ .

**Proof:** The first claim follows from Lemma 6 and the fact that  $\text{cent}(p^1, \dots, p^t)$  is a convex combination of  $p^1, \dots, p^t$ . For the second claim note that for  $i \in [t]$ ,

$$\text{KL}(p^i; p^1) \leq \ln \frac{1 + \epsilon}{1 - \epsilon} \leq 3\epsilon,$$

and hence  $\text{MaxCost}_{\text{KL}}(p^1, \dots, p^t) \leq 3\epsilon$ . Consider  $q \notin A(10\sqrt{\epsilon})$ . Then

$$\text{KL}(p^i; q) \geq \ell_1^2(p^i; q) \geq (10\sqrt{\epsilon} - \epsilon/d)^2 > 30\epsilon,$$

where the first inequality follows by Pinsker's inequality. Hence  $q$  does not satisfy,

$$\text{MaxCost}_{\text{KL}}(p^1, \dots, p^t; q) \leq 10 \cdot \text{MaxCost}_{\text{KL}}(p^1, \dots, p^t). \quad \blacksquare$$

### 3 Minimizing Average Distortion

In this section, we address the problem of computing a clustering of the input distributions which approximately minimizes the average Kullback-Liebler divergence between an input distribution and the center of the cluster it belongs to. We provide an algorithm that computes a clustering in which the average KL-divergence between an input distribution, and the center of the cluster it belongs to is at most  $O(\log n)$  times the optimal cost. The main theorem in this section is the following.

**Theorem 8** There exists a polynomial time  $O(\log n)$ -approximation algorithm for  $\text{MTC}_{\text{KL}}$ .

The main idea behind our algorithm is the observation that even though, in general, the KL-divergence between two distributions can be infinitely larger than the He-divergence between them, a clustering of the input distributions with low average distortion according to the He-divergence also has low average distortion by the KL-divergence. Therefore, our analysis proceeds in two steps. First, we show in Section 3.1 how to compute a clustering that approximately (within a factor of  $2 + \epsilon$ ) minimizes the average Hellinger divergence between input distributions and the closest cluster center. Then, we show in Section 3.2 how this leads to a clustering with low average distortion in the KL-divergence.

#### 3.1 Hellinger Clustering

In this section we present an algorithm for minimizing average He distortion. The main idea behind our algorithm is the simple observation that the Hellinger distance between two distributions  $p$  and  $q$  is the square of the Euclidean distance between the points  $\sqrt{p}$  and  $\sqrt{q}$  where  $\sqrt{p}$  is a shorthand for the vector in the positive quadrant of the unit sphere:

$$\sqrt{p} = (\sqrt{p_1}, \dots, \sqrt{p_d}).$$

Therefore, mapping each point  $p^i$  to  $\sqrt{p^i}$  and then computing a clustering that minimizes the average  $\ell_2^2$  measure between each transformed point and the center of the cluster it belongs to, should give us a good clustering for minimizing average Hellinger distortion. However, there will be a slight issue that arises because we insist that the cluster centers lie on the probability simplex.

Before we address the issue, we present the algorithm:

1. For each input distribution  $i \in [n]$ , compute  $\sqrt{p^i}$
2. Compute a  $(1 + \epsilon)$ -approximation to

$$\text{MTC}_{\ell_2^2}(\sqrt{p^1}, \dots, \sqrt{p^n}),$$

using any  $(1 + \epsilon)$ -approximation algorithm for  $k$ -means. Let the cluster centers be  $\tilde{c}^1, \dots, \tilde{c}^k$ . Note that in general  $\tilde{c}^j$  is not on the unit sphere.

3. Let  $\{p^{j_1}, \dots, p^{j_t}\}$  be the set of input distribution whose closest cluster center is  $\tilde{c}^j$ . Let the final center for this cluster be  $\text{cent}(p^{j_1}, \dots, p^{j_t})$ .

The issue we need to address is that the cluster center  $c$  that minimizes  $\text{SumCost}_{\text{He}}(p^1, \dots, p^t; c)$  need not lie on  $\Delta$ : this can be seen as a consequence of the fact that  $\tilde{c}^j$  is not on the unit sphere in general. Thus the actual average Hellinger divergence for the same clustering may be much higher than the  $k$ -means cost of the transformed points. However, the following lemma establishes that setting  $c = \text{cent}(p^1, \dots, p^t)$  (which necessarily lies on  $\Delta$ ) produces a clustering whose average Hellinger distortion is at most a factor 2 away from the  $k$ -means cost of the transformed points.

Before we state the lemma, we define some notation. For a vector  $p = (p_1, \dots, p_d)$  over  $d$  dimensions, we use  $p^2$  to denote the vector

$$p^2 = (p_1^2, \dots, p_d^2)$$

**Lemma 9** For  $p^1, \dots, p^t \in \Delta$ , for  $i \in [d]$ , define

$$a_i = \sum_{j \in [t]} p_i^j \text{ and } b_i = \sum_{j \in [t]} \sqrt{p_i^j}.$$

and let  $a = (a_1, \dots, a_d)$  and  $b = (b_1, \dots, b_d)$ .

$$\begin{aligned} & \text{SumCost}_{\text{He}}(p^1, \dots, p^t; a/t) \\ & \leq 2 \text{SumCost}_{\text{He}}(p^1, \dots, p^t; (b/t)^2) \end{aligned}$$

**Proof:**

$$\begin{aligned} \sum_j (\sqrt{p_i^j} - \sqrt{a_i/t})^2 &= a_i + \sum_j a_i/t - 2\sqrt{a_i/t} b_i \\ &= 2a_i - 2t^{-1/2} \sqrt{a_i} b_i \end{aligned}$$

and

$$2 \sum_j (\sqrt{p_i^j} - b_i/t)^2 \leq 2a_i - 2t^{-1} b_i^2.$$

Therefore it suffices to show that  $b_i \leq t^{1/2} \sqrt{a_i}$  this follows because

$$b_i^2 = a_i + \sum_{j \neq k} \sqrt{p_i^j p_i^k} \leq a_i + (t-1)a_i.$$

where the inequality follows by AM-GM inequality. ■

**Theorem 10** There exists a polynomial-time  $(2 + \epsilon)$ -approximation algorithm for  $\text{MTC}_{\text{He}}$ .

**Proof:** The result for  $\text{MTC}_{\text{He}}$  is achieved as described above: first we map each distribution from the probability simplex to the positive quadrant of the unit sphere:

$$\begin{aligned} f: \Delta &\rightarrow \{x \in \mathbb{R}^d : \ell_2(x) = 1, x_i \geq 1\} \\ (p_1, \dots, p_d) &\mapsto (\sqrt{p_1}, \dots, \sqrt{p_d}). \end{aligned}$$

We then run an algorithm for  $\text{MTC}_{\ell_2}$ . For each cluster formed, return the centroid of the original probability distributions. This is clearly a probability distribution. The cost of using this center rather than the center of

mass of the probability distributions once mapped to the sphere is a factor 2 as shown in Lemma 9. ■

We conclude this section by noting that our algorithm also leads to a good clustering for minimizing average distortion according to the Jensen-Shannon measure using Eq. 1.

**Lemma 11** There exists a polynomial-time  $(8 \ln 2 + \epsilon)$ -approximation algorithm for  $\text{MTC}_{\text{JS}}$ .

### 3.2 Kullback-Leibler Clustering

The following lemma relates  $\text{SumCost}_{\text{KL}}(p^1, \dots, p^t)$  and  $\text{SumCost}_{\text{He}}(p^1, \dots, p^t)$ . We note that that a later result in Section 4 could be used (in conjunction with Lemma 2) to achieve a result with that shows the ratio scales with  $\lg t$  in the worst case. However, the following proof establishes better constants and has the benefit that the proof is more geometric.

**Lemma 12** For any distributions  $p^1, \dots, p^t$ ,

$$1/2 \leq \frac{\text{SumCost}_{\text{KL}}(p^1, \dots, p^t)}{\text{SumCost}_{\text{He}}(p^1, \dots, p^t)} \leq \lceil \lg t \rceil (\ln 16).$$

**Proof:** The first inequality follows because for  $p, q \in \Delta$ ,  $\text{JS}(p, q) = \min_{c \in \Delta} (\text{KL}(p, c) + \text{KL}(q, c)) \leq \text{KL}(p, q)$  (this follows from e.g., Lemma 6) and Eq. 1.

We now prove the second inequality. Without loss of generality assume that  $t$  is a power of 2 (otherwise consider adding  $(2^{\lceil \lg t \rceil} - t)$  new points at He center of  $p^1, \dots, p^t$  – this can only increase the middle term of the equation.)

Consider a balanced binary tree on the nodes of the cluster. For an internal node at height  $j$ , associate a multi-set of distributions  $S(v)$  consisting of  $2^j$  copies  $p(u)$ , the center of mass of the  $2^j$  distributions at the leaves of the subtree rooted at  $v$ . Let  $S_j$  be the set of distributions at height  $j$ . Note that  $S_0 = \{p^1, \dots, p^t\}$ .

The lemma follows from the next three claims.

**Claim 13** For all  $j$ ,  $\text{SumCost}_{\text{He}}(S_j) \leq \text{SumCost}_{\text{He}}(S_0)$ .

**Proof:** Let  $c$  be an arbitrary distribution. By Lemma 5,

$$2^j \text{He}(p, c) + 2^j \text{He}(q, c) \geq 2^{j+1} \text{He}((p+q)/2, c)$$

and therefore  $\text{SumCost}_{\text{He}}(S_j; c)$  decreases as  $j$  increases and the result follows. ■

**Claim 14** For all  $j$ ,

$$\begin{aligned} & \sum_{v: \text{height}(v)=j+1} \text{SumCost}_{\text{KL}}(\cup_{u: u \in \text{ch}(v)} S(u)) \\ & \leq (\ln 16) \text{SumCost}_{\text{He}}(S_j). \end{aligned}$$

where  $\text{ch}(v)$  denotes the children of  $v$ .

**Proof:** Let  $u$  and  $w$  be the children of a node  $v$  at height  $j + 1$ . Let  $c = (p(u) + p(w))/2$ . Then,

$$\begin{aligned} & \text{SumCost}_{\text{KL}}(S(u), S(w)) \\ &= 2^j \text{JS}(p(u), p(w)) \\ &\leq 2^{j+1} (\ln 2) \text{He}(p(u), p(w)) \\ &\leq 2^{j+2} (\ln 2) (\text{He}(p(u), c) + \text{He}(p(w), c)) \\ &\leq (\ln 16) \text{SumCost}_{\text{He}}(S(u), S(w)) \end{aligned}$$

■

**Claim 15**

$$\begin{aligned} & \sum_j \sum_{v: \text{height}(v)=j+1} \text{SumCost}_{\text{KL}}(\cup_{u:u \in \text{ch}(v)} S(u)) \\ &= \text{SumCost}_{\text{KL}}(p^1, \dots, p^t) . \end{aligned}$$

**Proof:** Let  $v$  be at height  $j + 1$ . Let  $v$  have children  $u$  and  $w$  and grandchildren  $u_1, u_2, w_1, w_2$ . Then the result follows because

$$\begin{aligned} & \text{SumCost}_{\text{KL}}(S(u_1), S(u_2)) \\ & \quad + \text{SumCost}_{\text{KL}}(S(w_1), S(w_2)) \\ & \quad + \text{SumCost}_{\text{KL}}(S(u), S(w)) \\ &= 2^{j-1} (\text{KL}(p(u_1), p(u)) + \text{KL}(p(u_2), p(u)) \\ & \quad + \text{KL}(p(w_1), p(w)) + \text{KL}(p(w_2), p(w)) \\ & \quad + 2\text{KL}(p(u), p(v)) + 2\text{KL}(p(w), p(v))) \\ &= 2^{j-1} (\text{KL}(p(u_1), p(v)) + \text{KL}(p(u_2), p(v)) \\ & \quad + \text{KL}(p(w_1), p(v)) + \text{KL}(p(w_2), p(v))) \\ &= \text{SumCost}_{\text{KL}}(S(u_1), S(u_2), S(w_1), S(w_2)) \end{aligned}$$

where the second inequality follows from the parallelogram property and the fact that  $p(u) = (p(u_1) + p(u_2))/2$  and  $p(w) = (p(w_1) + p(w_2))/2$ . ■

■

We next show that the above lemma is nearly tight.

**Lemma 16** *There exists  $(p^i)_{i \in [t]}$  on  $d \geq t$  coordinates such that,*

$$\frac{\text{SumCost}_{\text{KL}}(p^1, \dots, p^t)}{\text{SumCost}_{\text{He}}(p^1, \dots, p^t)} = \Omega(\log t) .$$

**Proof:** Let  $(p^i)_{i \in [t]}$  be  $t$  distributions where  $p^i$  takes value  $i$  with probability 1. Then

$$\text{SumCost}_{\text{KL}}(p^1, \dots, p^t) = t \ln t$$

whereas

$$\begin{aligned} \text{SumCost}_{\text{He}}(p^1, \dots, p^t; c) &= t \left( \left(1 - \frac{1}{\sqrt{t}}\right)^2 + \frac{t-1}{t} \right) \\ &= 2t - 2\sqrt{t} , \end{aligned}$$

where  $c = t^{-1} \sum_i p^i$ . Then appeal to Lemma 9. ■

Then the proof of Theorem 8 follows immediately from Lemma 12 and Theorem 10.

## 4 Minimizing Maximum Distortion

In this section, we provide an algorithm for clustering the input distributions such that the maximum Kullback-Liebler divergence between an input distribution and the center of the cluster it belongs to is approximately minimized. In particular, our algorithm produces a clustering in which the maximum KL-divergence between an input distribution, and the closest center is at most a  $\min(O(\log d), O(\log n))$  factor greater than optimal.

Our algorithm is pleasantly simple: we use a variant of Gonzalez's algorithm [16] to cluster the input distributions such that the Jensen-Shannon divergence between any two points in the same cluster is minimized. We then show that although the KL-divergence between two distributions can be infinitely larger than their JS-divergence, this procedure still produces a good clustering according to the KL-divergence. The main theorem in this section can be stated as follows.

**Theorem 17** *There exists a polynomial-time*

$$O(\min(\log d, \log n))$$

*approximation for  $\text{MMC}_{\text{KL}}$ .*

Before proving the theorem, we show a lemma which establishes a general relationship between the KL-divergence and JS-divergence between two distributions, when the ratio of probability masses that the two distributions assign to any coordinate is bounded. This lemma may be of independent interest.

**Lemma 18** *Let  $p, q \in \Delta$  such that, for all  $i$ ,  $p_i/q_i \leq t$ , where  $t \geq e^2$ . Then,*

$$\text{KL}(p, q) \leq \frac{2 \ln t}{\ln(6/5)} \text{JS}(p, q)$$

**Proof:** For each  $i$ , let  $\delta_i = (p_i - q_i)/q_i$  so that  $p_i = (1 + \delta_i)q_i$ . Then,  $\sum_i \delta_i q_i = \sum_i p_i - q_i = 0$ ,

$$\text{KL}(p, q) = \sum_i q_i (1 + \delta_i) \ln(1 + \delta_i) ,$$

and

$$\text{JS}(p, q) = \sum_i q_i ((1 + \delta_i) \ln(1 + \delta_i) - (2 + \delta_i) \ln(1 + \frac{\delta_i}{2}))$$

Since  $p_i/q_i \leq t$ , and  $\delta_i \leq t - 1$ , from Lemma 19,

$$\text{KL}(p, q) \leq \Lambda \cdot \text{JS}(p, q) + \sum_i \delta_i q_i$$

where  $\Lambda = \frac{2 \ln t}{\ln(6/5)}$ . The lemma follows from the fact that  $\sum_i \delta_i q_i = 0$  and  $t \geq 4$ . ■

**Lemma 19** *For any  $x \in [-1, 2]$ ,*

$$\begin{aligned} (1 + x) \ln(1 + x) &\leq 4((1 + x) \ln(1 + x) \\ &\quad - 2(1 + x/2) \ln(1 + x/2)) + x . \end{aligned}$$

For any  $x \in (2, x^*]$ ,

$$(1+x) \ln(1+x) \leq \frac{2 \ln x^*}{\ln(6/5)} ((1+x) \ln(1+x) - 2(1+x/2) \ln(1+x/2)) + x .$$

**Proof:** Let  $\Lambda$  be a parameter and let

$$Y(x) = (1+x) \ln(1+x) - \Lambda((1+x) \ln(1+x) - 2(1+x/2) \ln(1+x/2)) - x .$$

Our goal is to show that  $Y(x) \leq 0$  for suitable values of the parameter  $\Lambda$ . The first and second order derivatives of  $Y$  can be written as follows:

$$Y'(x) = \Lambda \ln(1+x/2) - (\Lambda - 1) \ln(1+x)$$

and

$$Y''(x) = \frac{2 - \Lambda + x}{(1+x)(2+x)} .$$

We first consider  $x \in [-1, 2)$  and  $\Lambda = 4$ . If  $x < 2$ , then  $Y''(x) < 0$ . Therefore,  $Y'(x)$  is strictly decreasing in the range  $[-1, 2)$ . We note that  $Y'(-1) = \infty$  and  $Y'(0) = 0$ ; therefore  $Y$  is a strictly increasing function from  $[-1, 0)$  and strictly decreasing from  $(0, 2]$ . As  $Y(0) = 0$ ,  $Y(x) < 0$  for  $x < 0$  and  $Y(x) < 0$  for  $x > 0$ , and the first part of the lemma follows.

To prove the second part, we write the derivative  $Y'(x)$  as follows:

$$Y'(x) = \Lambda \cdot \ln \frac{1+x/2}{1+x} + \ln(1+x)$$

If  $x > 2$ , then  $\ln \frac{1+x/2}{1+x} < \ln(5/6)$ . By plugging in  $\Lambda = \frac{2 \ln x^*}{\ln 6/5}$ ,

$$Y'(x) < -2 \ln x^* + \ln(1+x) < 0$$

for  $x$  in  $(2, x^*]$ , which means that  $Y$  is strictly decreasing in this interval. As  $t \geq e^2$ , here  $\Lambda \geq 4$ . The previous part of the lemma implies that  $Y(2) < 0$ , for any  $\Lambda > 4$ , and hence the lemma follows. ■

**Lemma 20** Consider  $t$  distributions  $p^1, \dots, p^t$  such that  $\text{He}(p^i, p^j) \leq r$  for all  $i, j \in [t]$ . Then  $\text{He}(p^i, c) \leq r$  for all  $i \in [t]$  where  $c$  is any convex combination of  $p^1, \dots, p^t$ .

**Proof:** The result follows by Lemma 5: Consider distribution  $p^i$  and the set of distributions in  $B_r(p^i) = \{q : \text{He}(p^i, q) \leq r\}$ . By Lemma 5,  $B_r(p^i)$  is convex. Since  $p^j \in B_r(p^i)$  for all  $j \in [t]$  and  $c$  is a convex combination of  $\{p^j\}_{j \in [t]}$  we deduce that  $c \in B_r(p^i)$ . Hence  $\text{He}(p^i, c) \leq r$  as required. Since  $i$  was arbitrary the result follows. ■

**Lemma 21** Let  $p^1, \dots, p^t$  be  $t$  distributions over  $[d]$  and let  $c = \text{cent}(p^1, \dots, p^t)$ . Then,

$$\frac{\text{MaxCost}_{\text{KL}}(p^1, \dots, p^t; c)}{\max_{i,j} \text{JS}(p^i, p^j)} \leq O(\log t) .$$

Moreover, there exists some  $c^*$  which is a convex combination of  $p^1, \dots, p^t$  such that:

$$\frac{\text{MaxCost}_{\text{KL}}(p^1, \dots, p^t; c^*)}{\max_{i,j} \text{JS}(p^i, p^j)} \leq O(\log d) .$$

**Proof:** To show the first inequality, we observe that for any  $i \in [d], j \in [t]: p_i^j/c_i \leq t$ . Using this fact along with Lemma 18, we conclude that:

$$\text{MaxCost}_{\text{KL}}(p^1, \dots, p^t; c) \leq O(\log t) \cdot \max_i \text{JS}(p^i, c)$$

The rest of the inequality follows from the fact that JS is constant factor related to He (Lemma 2), followed by an application of Lemma 20.

To show the second inequality, let

$$q^1, \dots, q^d \subset \{p^1, \dots, p^t\}$$

be distributions such that for any  $i \in [d]$  and any  $j \in [n]$ ,  $q_i^i \geq p_i^j$ . We define

$$c^* = \text{cent}(q^1 + \dots + q^d) .$$

Observe that for any  $i \in [d], j \in [t]: p_i^j/c_i^* \leq d$ . Therefore, from Lemma 18, for any  $i$ ,

$$\text{MaxCost}_{\text{KL}}(p^1, \dots, p^t; c^*) \leq O(\log d) \text{JS}(p^i, c^*)$$

From Lemma 2,  $\text{JS}(p^i, c^*) \leq O(\text{He}(p^i, c^*))$ . As  $c^*$  is a convex combination of  $p^1, \dots, p^t$ , the rest of the lemma follows from an application of Lemma 20. ■

**Lemma 22** Let  $p^1, \dots, p^t$  be  $t$  distributions over  $[d]$ .

$$\frac{1}{2} \leq \frac{\text{MaxCost}_{\text{KL}}(p^1, \dots, p^t)}{\max_{i,j} \text{JS}(p^i, p^j)}$$

**Proof:** Let  $(i, j) = \text{argmax} \text{JS}(p^i, p^j)$ . Note that since we allow unrestricted centers,

$$\text{MaxCost}_{\text{KL}}(p^i, p^j) \leq \text{MaxCost}_{\text{KL}}(p^1, \dots, p^t; c) ,$$

and let  $q$  minimize  $\max\{\text{KL}(p^i, q), \text{KL}(p^j, q)\}$ . But

$$\begin{aligned} 2 \max\{\text{KL}(p^i, q), \text{KL}(p^j, q)\} &\geq \text{KL}(p^i, q) + \text{KL}(p^j, q) \\ &\geq \min_q \text{KL}(p^i, q) + \text{KL}(p^j, q) \\ &= \text{JS}(p^i, p^j) . \end{aligned}$$

from which the Lemma follows. ■

We now are in a position to complete the proof of Theorem 17.

**Proof:** From Lemmas 21 and 22, and the fact that for any  $c$ ,

$$\text{MaxCost}_{\text{KL}}(p^1, \dots, p^t; c) \geq \text{MaxCost}_{\text{KL}}(p^1, \dots, p^t)$$

(by definition), we know that if we  $\alpha$ -approximate the problem of minimizing the maximum JS diameter of a cluster, we get a  $\min(O(\alpha \log d), O(\alpha \log n))$  approximation for  $k$ -center KL clustering. In the rest of the proof we show that we may assume  $\alpha = 4$ .

We use a variant of an algorithm by Gonzalez [16] that is applicable to divergences that satisfy a relaxed triangle inequality. Recall that JS satisfies,

$$\text{JS}(p, q) + \text{JS}(q, r) \geq \text{JS}(p, r)/2 .$$

for all  $p, q, r$ . The algorithm assumes knowledge of the optimum JS diameter (note that there are at most  $n^2$  possible values and thus we can check them all); let this value be  $D$ . Initially, let all  $p^j$  be “unmarked.” The algorithm proceeds by picking an arbitrary unmarked distribution  $p^i$ , marking all  $p^j$  such that  $\text{JS}(p^j, p^i) \leq D$  and repeating until all distributions are marked. Define the each cluster as the set of distributions marked in the same iteration and call  $p^i$  the “center” of this cluster. This results in a clustering such that the maximum diameter is at most  $2(D + D) = 4D$ . We need to show that the process does not determine more than  $k$  centers. Suppose we pick  $k + 1$  centers. Note that each of these centers are strictly greater than  $(D + D)/2 = D$  apart and hence no two may be in the same cluster for the optimum clustering. This is a contradiction. ■

## 5 Hardness Results

In this final section of our paper, we prove hardness of approximation results for  $\text{MMC}_{\text{KL}}$  and  $\text{MTC}_{\text{KL}}$ , i.e., we show lower bounds of the approximation factors possible in polynomial time on the assumption that  $P \neq NP$ . We consider two variants of these problems. For all the algorithms we presented in the previous sections, we insisted that the centers  $c^1, \dots, c^k$  lay in  $\Delta$  but other than this the centers were *unrestricted*. In some of the previous work on approximation algorithms for clustering a variant is considered in which it is required that  $c^1, \dots, c^k$  are chosen from among the input distributions  $\{p^1, \dots, p^n\}$ . We call this the *restricted center* version.

When a metric is used rather than KL, the restricted and unrestricted versions of the problems are closely related: it can be shown that the restriction can at most double the clustering cost. However, for KL we show that, while we have presented approximation algorithms for the unrestricted case, no approximation to any multiplicative factor is possible in the restricted case.

### 5.1 Unrestricted Centers

In this section, we prove an approximation hardness result for  $\text{MMC}_{\text{KL}}$ . Our result is based on demonstrating that near the center of the simplex KL behaves similarly to  $\ell_2^2$ . We then use a result by Feder and Greene [13] that showed an approximation hardness of 1.822 for  $k$ -center in the plane where distance are measured as  $\ell_2$ . (Hence, this gives a  $1.822^2 < 3.320$  approximation hardness result for  $\ell_2^2$ .)

Recall the definition,

$$A(r) = \{p \in \Delta : p_j = 1/d \pm r \text{ for all } j \in [d]\} . \quad (3)$$

**Lemma 23** For  $p, q \in A(\epsilon/d)$ ,

$$\text{KL}(p, q) = (1 \pm 5\epsilon)d\ell_2^2(p, q) .$$

**Proof:** We apply Taylor’s Theorem to the terms of the KL divergence:

$$\begin{aligned} \text{KL}(p, q) &= \sum_{i \in [d]} p_i \log \frac{p_i}{q_i} - p_i + q_i \\ &= \sum_{i \in [d]} -p_i \log \left( 1 - \frac{p_i - q_i}{p_i} \right) - p_i + q_i \\ &= \sum_{i \in [d]} \frac{(p_i - q_i)^2}{p_i} + \eta_i^3 p_i \end{aligned}$$

for some  $\eta_i$  with  $|\eta_i| \leq |p_i - q_i|/p_i$ . Note that

$$|\eta_i|^3 p_i \leq \frac{(p_i - q_i)^2}{p_i} \cdot \frac{|p_i - q_i|}{p_i} \leq 3\epsilon \frac{(p_i - q_i)^2}{p_i}$$

and therefore

$$\text{KL}(p, q) = (1 \pm 3\epsilon) \sum_{i \in [d]} \frac{(p_i - q_i)^2}{p_i} \leq (1 \pm 5\epsilon)d\ell_2^2(p, q) .$$

Using the above lemma, the next theorem shows that if the distributions to be clustered are near the center of the simplex, then we can use an approximation algorithm for  $\text{MTC}_{\text{KL}}$  or  $\text{MMC}_{\text{KL}}$  to get a good approximation for  $\text{MTC}_{\ell_2^2}$  or  $\text{MMC}_{\ell_2^2}$  respectively.

**Theorem 24** Let  $\tau \in (1, 10)$  and let

$$p^1, \dots, p^n \in A(\epsilon^2/(50^2 d^3)) .$$

Then, a  $\tau$ -approximation for  $\text{MTC}_{\text{KL}}$  yields a  $(\tau + 5\epsilon)$ -approximation for  $\text{MTC}_{\ell_2^2}$ . Similarly, a  $\tau$ -approximation for  $\text{MMC}_{\text{KL}}$  yields a  $(\tau + 5\epsilon)$ -approximation for  $\text{MMC}_{\ell_2^2}$ .

**Proof:** We first consider  $\text{MTC}_{\text{KL}}$ . Suppose we want to solve  $\text{MTC}_{\ell_2^2}$  on the input  $p^1, \dots, p^n \in \Delta$  and let

$$\{\tilde{c}^1, \dots, \tilde{c}^k\}$$

be a  $\tau$ -approximation for  $\text{MTC}_{\text{KL}}$ . Without loss of generality, we may assume that  $\tilde{c}^1, \dots, \tilde{c}^k$  are in the convex hull of  $p^1, \dots, p^n$  since if  $\tilde{c}^j$  is the closest center to  $\{p^i\}_{i \in I}$  then the objective function only decreases if we let  $\tilde{c}^j = \text{cent}(p^i : i \in I)$ .

Denote the convex hull of  $p^1, \dots, p^n$  by  $H$  and note that  $q \in H$  implies that  $q \in A(\epsilon^2/(50^2 d^3)) \subset A(\epsilon/(5d))$ . Hence, by appealing to Lemmas 7 and 23, we deduce,

$$\begin{aligned} &\sum_{j \in [n]} \min_{i \in [k]} \ell_2^2(p^j, \tilde{c}^i) \\ &= \frac{1}{d(1 \pm \epsilon)^2} \sum_{j \in [n]} \min_{i \in [k]} \text{KL}(p^j, \tilde{c}^i) \\ &\leq \frac{\tau}{d(1 \pm \epsilon)^2} \min_{c^1, \dots, c^k \in H} \sum_{j \in [n]} \min_{i \in [k]} \text{KL}(p^j, c^i) \\ &\leq \frac{\tau}{(1 \pm \epsilon)^4} \min_{c^1, \dots, c^k \in H} \sum_{j \in [n]} \min_{i \in [k]} \ell_2^2(p^j, c^i) \\ &= \frac{\tau}{(1 \pm \epsilon)^4} \min_{c^1, \dots, c^k \in \Delta} \sum_{j \in [n]} \min_{i \in [k]} \ell_2^2(p^j, c^i) . \end{aligned}$$

where the last line follows because the optimum centers for  $\text{MTC}_{\ell_2}$  lie in  $\text{convex}(P)$ .

We now consider  $\text{MMC}_{\text{KL}}$  and suppose  $\{\tilde{c}^1, \dots, \tilde{c}^k\}$  is a  $\tau$ -approximation for  $\text{MMC}_{\text{KL}}$ . By appealing to Lemma 7, we may assume

$$\tilde{c}^1, \dots, \tilde{c}^k \in A(10\sqrt{\epsilon^2/(50^2 d^2)}) = A(\epsilon/(5d)).$$

Hence,

$$\begin{aligned} & \max_{j \in [n]} \min_{i \in [k]} \ell_2^2(p^j, \tilde{c}^i) \\ &= \frac{1}{d(1 \pm \epsilon)^2} \max_{j \in [n]} \min_{i \in [k]} \text{KL}(p^j, \tilde{c}^i) \\ &\leq \frac{\tau}{d(1 \pm \epsilon)^2} \min_{c^1, \dots, c^k \in A(\frac{\epsilon}{5d})} \left( \max_{j \in [n]} \min_{i \in [k]} \text{KL}(p^j, c^i) \right) \\ &\leq \frac{\tau}{(1 \pm \epsilon)^4} \min_{c^1, \dots, c^k \in A(\frac{\epsilon}{5d})} \left( \max_{j \in [n]} \min_{i \in [k]} \ell_2^2(p^j, c^i) \right) \\ &= \frac{\tau}{(1 \pm \epsilon)^4} \min_{c^1, \dots, c^k \in \Delta} \left( \max_{j \in [n]} \min_{i \in [k]} \ell_2^2(p^j, c^i) \right). \end{aligned}$$

where the last line follows because the optimum centers for  $\text{MMC}_{\ell_2}$  lie in  $A(\epsilon/(5d))$ . This can be shown using ideas contained in Lemma 7:

$$\ell_2^2(p^i, p^1) \leq d \max_{j \in [d]} (p_j^i - p_j^1)^2 \leq 4\epsilon^4/(50^4 d^5)$$

while for  $q \notin A(\epsilon/(5d))$ ,

$$\ell_2^2(p^i, q) \geq (\epsilon^2/(5d)^2 - \epsilon^2/(50^2 d^3))^2 \geq 4\epsilon^4/(50^4 d^5). \quad \blacksquare$$

To show a hardness result for unrestricted centers it is therefore sufficient to show a hardness result for  $\text{MMC}_{\ell_2}$  when the points to be clustered lie near the middle of the probability simplex. We do this by taking Feder and Greene [13] result the showed the hardness of  $\text{MMC}_{\ell_2}$  in the plane and demonstrating that the plane can be mapped into the middle of the probability simplex in a manner that preserves approximation factors. This will give the following theorem.

**Theorem 25** *For any  $\alpha < 3.320$ , unless  $P = NP$ , no polynomial-time,  $\alpha$ -approximation algorithm exists for  $\text{MMC}_{\text{KL}}$ .*

**$k$ -means on the middle of  $\Delta$ :** Given an instance  $I$  of  $\text{MMC}_{\ell_2}$  on a bounded domain  $A$  of the  $x_1 - x_2$  plane, we show how to produce an instance  $I'$  of  $\text{MMC}_{\ell_2}$  on the three-dimensional simplex, such that, there is an approximation preserving bijection between the solutions to  $I$  and the solutions to  $I'$ .

To show this, we first assume without loss of generality that  $A \subseteq [0, 1/4] \times [0, 1/4]$ . We can assume this because translating and scaling scales down the distance between every pair of points by the same number. For any  $x \in A$ , we define a map  $\phi(x)$  as follows:

$$\phi(x) = Ux + [1/3, 1/3, 1/3]^T$$

where  $U$  is the matrix:

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{pmatrix}$$

**Lemma 26** *If  $x \in A$ ,  $\phi(x)$  lies on the simplex.*

**Proof:** We first show that if  $x$  lies on the  $x_1 - x_2$  plane, then  $Ux$  lies on the plane  $x_1 + x_2 + x_3 = 0$ . Let

$$y_1 = (1/\sqrt{2}, -1/\sqrt{2}, 0) \text{ and } y_2 = (0, 1/\sqrt{2}, -1/\sqrt{2}).$$

As  $y_1 = Ux_1$  and  $y_2 = Ux_2$ , if  $x = \alpha_1 x_1 + \alpha_2 x_2$ , then,  $Ux = \alpha_1 y_1 + \alpha_2 y_2$ . Since

$$y_1 \cdot (1, 1, 1) = y_2 \cdot (1, 1, 1) = 0,$$

$Ux \cdot (1, 1, 1) = 0$  as well, which means that  $Ux$  lies on the plane  $x_1 + x_2 + x_3 = 0$ .

Since  $Ux$  lies on the plane  $x_1 + x_2 + x_3 = 0$ , we deduce that  $\phi(x)$  lies on the plane  $x_1 + x_2 + x_3 = 1$ . Since  $x \in [0, 1/4] \times [0, 1/4]$ , for any  $i \in \{1, 2, 3\}$ ,

$$(Ux)_i \geq -\frac{1}{4} \times \frac{1}{\sqrt{2}} \geq -\frac{1}{3}.$$

Therefore, for any  $i$ ,  $(\phi(x))_i \geq 0$ . Again, as  $x \in [0, 1/4] \times [0, 1/4]$ , for any  $i \in \{1, 2, 3\}$ ,

$$(Ux)_i \leq \frac{1}{4} \times \frac{1}{\sqrt{2}} \leq \frac{2}{3}.$$

Therefore,  $(\phi(x))_i \leq 1$  for each  $i$ , from which the lemma follows.  $\blacksquare$

To map an instance  $I$  of  $\text{MMC}_{\ell_2}$  on the  $x_1 - x_2$  plane to the probability simplex, we simply apply the map  $\phi$  on each point of  $I$ . This produces another instance  $I'$  of the problem on the simplex, which has the following property.

**Lemma 27** *There is an approximation-preserving bijection between the solutions of  $I$  and the solutions of  $I'$ .*

**Proof:** We observe that as  $U$  is a unitary matrix, the map  $\phi$  is a bijection. Moreover, since  $\phi$  consists of a translation and a rotation, it preserves the distance between every pair of points. Therefore, applying  $\phi$  on a solution of  $I$  produces a solution of  $I'$  of the same cost. The mapping  $\phi$  is thus approximation preserving.  $\blacksquare$

Finally, by mapping each point  $x \in I'$  to  $\epsilon x + (1-\epsilon)u$  where  $u = [1/3, 1/3, 1/3]^T$  we generate a set of points that lie arbitrarily close to the center of  $\Delta$  (setting  $\epsilon$  as small as necessary.) Again, this transformation can be seen to be approximation preserving.

## 5.2 Restricted Centers

In this section, we consider the restricted version of  $\text{MMC}_{\text{KL}}$  and  $\text{MTC}_{\text{KL}}$  where we insist that the cluster centers  $c^1, \dots, c^n \in \{p^1, \dots, p^n\}$ . Our result is based on relating the problem to the SET-COVER problem and appealing to a result of Feige [14].

**Theorem 28** For any  $\alpha \geq 1$ , unless  $P = NP$ , no polynomial-time,  $\alpha$ -approximation algorithm exists for either  $\text{MTC}_{\text{KL}}$  or  $\text{MMC}_{\text{KL}}$ .

**Proof:** Consider a reduction from the problem SET-COVER: Consider  $S_1, \dots, S_{n-d-1} \in [d-1]$  and  $k \leq d$ . It was shown by Feige [14] that it is NP-hard to determine if there exists  $\mathcal{S} = \{S_{i_1}, \dots, S_{i_{k-1}}\}$  such that  $\bigcup S_{i_j} = [d-1]$ .

We first consider  $\text{MMC}_{\text{KL}}$ . Let  $c_1, c_2 > 1$  such that

$$(d-1)e^{-c_1} < 1 \text{ and } (d-1)e^{-c_2} < e^{-c_1} .$$

Let  $q^i$  be the probability distribution with mass  $e^{-c_1}$  on each element in  $S_i$ , and the remaining mass on  $\{d\}$ . Let  $p^i = e_i$  (i.e., the  $i$ -th vector of the standard basis) for  $i \in [d-1]$ . Let  $r$  be the probability distribution with  $e^{-c_2}$  mass on each element in  $[d-1]$  and the remaining mass on  $\{d\}$ .

Note that  $\text{KL}(p^i, q^j) = c_1$ ,  $\text{KL}(p^i, r) = c_2$ , and

$$\begin{aligned} \text{KL}(q^j, r) &= (1 - |S_j|e^{-c_1}) \ln \frac{1 - |S_j|e^{-c_1}}{1 - (d-1)e^{-c_2}} \\ &\quad + |S_j|e^{-c_1} \ln(e^{-c_1}/e^{-c_2}) \\ &\leq |S_j|e^{-c_1}(c_2 - c_1) \leq de^{-c_1}c_2 \end{aligned}$$

Hence, if there exists  $\mathcal{S}$ , the clustering with centers  $p^{i_1}, \dots, p^{i_{k-1}}, r$  costs at most  $\max\{c_1, de^{-c_1}c_2\}$  whereas otherwise the cost is

$$\max\{c_1, c_2, de^{-c_1}c_2\} \geq c_2 .$$

Hence the ratio difference is at least

$$\frac{c_2}{\max\{c_1, de^{-c_1}c_2\}}$$

which we can make arbitrarily large.

This also implies that no approximation is possible for  $\text{MTC}_{\text{KL}}$  because any  $\alpha'$ -approximate solution for  $\text{MTC}_{\text{KL}}$  is also a  $\alpha n$ -approximation solution for  $\text{MMC}_{\text{KL}}$  for  $k$ -median when centers must be original points. ■

**Bi-criteria Approximation:** We briefly mention an approximation algorithm for the related approximation problem of finding the minimum number  $k'$  of centers  $c^1, c^2, \dots, c^{k'} \in \{p^1, \dots, p^n\}$  such that for all  $i \in [n]$ ,

$$\min_{j \in [k']} \text{KL}(p^i, c^j) \leq r$$

for some given  $r$ . This can be approximated up to a factor of  $O(\log n)$  using a well-known approximation algorithm for SET-COVER. Specifically, for each  $p^i$  we define a set  $S_i = \{j \in [n] : \text{KL}(p^j, p^i) \leq r\}$ . Then, our problem becomes picking the smallest number of sets  $S_{i_1}, S_{i_2}, \dots$  such that  $\bigcup_{j \geq 1} S_{i_j} = [n]$ . An  $O(\log n)$ -approximation algorithm exists for this problem.

**Acknowledgements:** We would like to thank Sanjoy Dasgupta for helpful discussions.

## References

- [1] M. R. Ackerman, J. Blomer, and C. Sohler. Clustering for metric and non-metric distance measures. In *ACM-SIAM Symposium on Discrete Algorithms*, 2008.
- [2] V. Arya, N. Garg, R. Khandekar, K. Munagala, and V. Pandit. Local search heuristic for  $k$ -median and facility location problems. In *ACM Symposium on Theory of Computing*, pages 21–29, 2001.
- [3] M. Badoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *ACM Symposium on Theory of Computing*, pages 250–257, 2002.
- [4] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, AU, 1998. ACM Press, New York, US.
- [5] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.
- [6] K. Chen. On  $k$ -median clustering in high dimensions. In *Symposium on Discrete Algorithms*, 2006.
- [7] J. Chuzhoy, S. Guha, E. Halperin, S. Khanna, G. Kortsarz, R. Krauthgamer, and J. Naor. Asymmetric  $k$ -center is  $\log^* n$ -hard to approximate. *J. ACM*, 52(4):538–551, 2005.
- [8] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA, 1991.
- [9] K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. In *NIPS*, pages 321–328, 2006.
- [10] I. Csiszár. Why least squares and maximum entropy? an axiomatic approach to inference for linear inverse problems. *Ann. Statist.*, 19:2032–2056, 1991.
- [11] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *JMLR*, 3:1265–1287, 2003.
- [12] D. M. Endres and J. E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003.
- [13] T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *ACM Symposium on Theory of Computing*, 1988.
- [14] U. Feige. A threshold of  $\ln$  for approximating set cover. *J. ACM*, 45(4):634–652, 1998.
- [15] D. Feldman, M. Monemizadeh, and C. Sohler. A ptas for  $k$ -means clustering based on weak coresets. In *Symposium on Computational Geometry*, pages 11–18, 2007.
- [16] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985.
- [17] S. Guha, P. Indyk, and A. McGregor. Sketching

- information divergences. *Submitted to Journal of Machine Learning*, 2007.
- [18] S. Guha, A. McGregor, and S. Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 733–742, 2006.
- [19] S. Har-Peled and S. Mazumdar. Coresets for k-means and k-median clustering and their applications. In *ACM Symposium on Theory of Computing*, 2004.
- [20] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004.
- [21] S. G. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the euclidean kappa-median problem. In *European Symposium on Algorithms*, pages 378–389, 1999.
- [22] A. Kumar, Y. Sabharwal, and S. Sen. A simple linear time  $(1 + \epsilon)$ -approximation algorithm for k-means clustering in any dimensions. In *IEEE Symposium on Foundations of Computer Science*, pages 454–462. IEEE Computer Society, 2004.
- [23] R. Panigrahy and S. Vishwanathan. An  $o(\log^* n)$  approximation algorithm for the asymmetric  $p$ -center problem. *J. Algorithms*, 27(2):259–268, 1998.
- [24] F. C. N. Pereira, N. Tishby, and L. Lee. Distributional clustering of english words. In *ACL*, pages 183–190, 1993.
- [25] N. Slonim, R. Somerville, N. Tishby, and O. Lahav. Objective classification of galaxy spectra using the information bottleneck method. *Monthly Notes of the Royal Astronomical Society*, 323:270–284, 2001.
- [26] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *NIPS*, pages 617–623, 1999.
- [27] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.
- [28] F. Topsøe. Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory*, 46(4):1602–1609, 2000.

---

# An Information Theoretic Framework for Multi-view Learning

---

Karthik Sridharan and Sham M. Kakade  
Toyota Technological Institute at Chicago  
{karthik, sham}@tti-c.org

## Abstract

In the multi-view learning paradigm, the input variable is partitioned into two different views  $X_1$  and  $X_2$  and there is a target variable  $Y$  of interest. The underlying assumption is that either view *alone* is sufficient to predict the target  $Y$  accurately. This provides a natural semi-supervised learning setting in which unlabeled data can be used to eliminate hypothesis from either view, whose predictions tend to disagree with predictions based on the other view.

This work explicitly formalizes an information theoretic, multi-view assumption and studies the multi-view paradigm in the PAC style semi-supervised framework of Balcan and Blum [2006]. Underlying the PAC style framework is that an *incompatibility function* is assumed to be known — roughly speaking, this incompatibility function is a means to score how good a function is based on the unlabeled data alone. Here, we show how to derive incompatibility functions for certain loss functions of interest, so that minimizing this incompatibility over unlabeled data helps reduce expected loss on future test cases. In particular, we show how the class of empirically successful co-regularization algorithms fall into our framework and provide performance bounds (using the results in Rosenberg and Bartlett [2007], Farquhar et al. [2005]).

We also provide a normative justification for canonical correlation analysis (CCA) as a dimensionality reduction technique. In particular, we show (for strictly convex loss functions of the form  $\ell(w \cdot x, y)$ ) that we can first use CCA as dimensionality reduction technique and (if the multi-view assumption is satisfied) this projection does not throw away much predictive information about the target  $Y$  — the benefit being that subsequent learning with a labeled set need only work in this lower dimensional space.

## 1 Introduction

The “multi-view” approach to learning has been receiving increasing attention as a paradigm for semi-supervised learning. The implicit assumption is that either view *alone* has sufficient information about the target  $Y$ . The basic intuition as to why this assumption is helpful is that the complexity of the learning problem could be reduced by eliminating hypothesis from each view that tend not to agree with each other, which, crucially, can be done using unlabeled data.

There are many natural applications for which this assumption is applicable. For example, consider a setting where it is easy to obtain pictures of objects from different camera angles and say our supervised task is one of object recognition. Intuitively, we can think of unlabeled data as providing examples of viewpoint invariance. One can even consider multi-modal views, e.g. identity recognition where the task might be to identify a person with one view being a video stream and the other an audio stream — each of these views would be sufficient to determine the identity. In NLP, an example would be a paired document corpus, consisting of a document and its translation into another language, and the supervised task could be predicting some high level property of the document. The motivating example in Blum and Mitchell [1998] is a web-page classification task, where one view was the text in the page and the other was the hyper-link structure.

This work explicitly formalizes a general information theoretic multi-view assumption. Based on this assumption, we seek to understand the reduction in label complexity from using unlabeled data. There are two natural classes of algorithms in the literature which can be considered multi-view algorithms and algorithms based on CCA. For the former, we analyze the co-regularization algorithms of Sindhwani et al. [2005], Brefeld et al. [2006] (and the related SVM-2K algorithm of Farquhar et al. [2005]) in a generalization of the PAC style semi-supervised framework of Balcan and Blum [2006]. Technically, this PAC model is for the 0/1 loss, but we generalize the framework to arbitrary loss functions. For the latter class of algorithms, we generalize the CCA results in Kakade and Foster [2007] to show how CCA can be used for dimensionality reduction, when dealing with convex loss functions (under linear prediction). In the Discussion, we present a practical answer to the open problem presented in Balcan and Blum [2007] (presented at COLT 2007) using

co-regularization algorithms, under the theory of surrogate loss functions [Bartlett et al., 2006], and we also discuss the connection to the Information Bottleneck method of Tishby et al. [1999].

In the remainder of the Introduction, we present our setting and main information theoretic assumption, and then summarize our contributions and related work.

### 1.1 A Multi-View Assumption

In the (multi-view) semi-supervised setting, we assume that we have  $n$  labeled examples  $S = \{(x_1^i, x_2^i, y^i)\}_{i=1}^n$  and  $m$  unlabeled examples  $U = \{(x_1^i, x_2^i)\}_{i=n+1}^{n+m}$ , where  $y_i \in \mathcal{Y}$  and  $x_v^i \in \mathcal{X}_v$  for  $v \in \{1, 2\}$ , which are both sampled in an i.i.d. manner from some unknown underlying joint distribution (typically  $m \gg n$ ). In particular, the joint underlying distribution is over  $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}$ . As usual, the goal is to predict  $Y$ , as measured with respect to some known loss function.

Information theory provides the natural language to state an assumption for multi-view learning. In particular, the conditional mutual information  $I(A : B|C)$  (between random outcomes  $A$  and  $B$  conditioned on  $C$ ) measures how much information is shared between  $A$  and  $B$  conditioned on already knowing  $C$ , which can be viewed as how much knowing  $A$  reduces our uncertainty of  $B$ , conditioned on already knowing  $C$ . We now state our first main assumption.

**Assumption 1 (Multi-View Assumption)** *There exists an  $\epsilon_{\text{info}} > 0$  such that*

$$I(Y : X_2|X_1) \leq \epsilon_{\text{info}}$$

and

$$I(Y : X_1|X_2) \leq \epsilon_{\text{info}}$$

Let us try to get an intuitive feel for this assumption. The assumption states that (on average) if we already knew  $X_1$  then there is little more information that we could gain about  $Y$  from learning  $X_2$  (and vice-versa) — this small potential gain is quantified by  $\epsilon_{\text{info}}$ . Hence, we can think of this assumption as stating that both  $X_1$  and  $X_2$  are (approximately) redundant with regards to their information about  $Y$ .

Let us examine how the compatibility assumption made in the co-training case [Blum and Mitchell, 1998], where  $Y \in \{0, 1\}$ , is related to this assumption. Here, it was assumed that a perfect prediction of  $Y$  is possible using the knowledge of either view alone. This implies the above conditions are satisfied with  $\epsilon_{\text{info}} = 0$ , since conditioned on either view, the target  $Y$  is already known (so there is no possible reduction in uncertainty with knowledge from the remaining view).

However, note that under this assumption, neither view need accurately predict the target, just that they both carry (roughly) the same information about the target. Hence, the assumption is well suited to situations with noise. In fact, even if  $\epsilon_{\text{info}} = 0$ , there need not exist perfect predictions of the target — though for this case we would expect that the optimal predictions should perfectly agree (as they carry the same information about  $Y$ ), a point which we return to.

The work in Blum and Mitchell [1998] also introduced a further conditional independence assumption, which states

that  $X_1$  and  $X_2$  are independent conditioned on the knowledge of  $Y$ . The work of Dasgupta et al. [2001], Abney [2004] shows how unreasonably strong this extra assumption is, with regards to classification. In our work, we make no further assumptions on the underlying data distribution.

### 1.2 Co-Regularization

There is a recent class of algorithms which control model complexity in the two view setting by *co-regularizing* [Sindhwani et al., 2005, Brefeld et al., 2006]. A related algorithm is the two view SVM-2K algorithm of Farquhar et al. [2005]. These class of algorithms all have demonstrated empirical successes. The question we seek to understand is how unlabeled data improves the performance of these algorithms.

These co-regularization algorithms add an additional regularizer which penalizes using functions from either view which tend to disagree. The (kernelized) algorithm of Sindhwani et al. [2005], Brefeld et al. [2006] is to find two predictors  $f_1$  and  $f_2$  (where  $f_1 : \mathcal{X}_1 \rightarrow \mathcal{Y}$  and  $f_2 : \mathcal{X}_2 \rightarrow \mathcal{Y}$ ) which minimize the following co-regularized loss:

$$\frac{1}{2}(\widehat{E}_S[\ell(f_1(x_1), y)] + \widehat{E}_S[\ell(f_2(x_2), y)]) + \lambda \|f_1\|_K^2 + \lambda \|f_2\|_K^2 + \lambda_{co} \widehat{E}_U(f_1(x_1) - f_2(x_2))^2 \quad (1)$$

where  $\|\cdot\|_K$  is a pre-specified norm over functions;  $\widehat{E}_S$  and  $\widehat{E}_U$  are empirical averages with respect to the labeled and unlabeled sets  $S$  and  $U$ , respectively; and  $\ell$  is some convex loss (such as the hinge loss or squared loss). The last term is the co-regularizer. Note that if  $\lambda_{co} = 0$  then this problem just reduces to solving two independent (regularized) problems. The SVM-2K algorithm of Farquhar et al. [2005] is similar — it essentially imposes an agreement constraint into the SVM objective function, based on the  $L_1$  norm (which allows for an efficient implementation).

Rosenberg and Bartlett [2007] provide generalization bounds for co-regularization (using a co-regularizer that is the square loss) in terms of Rademacher complexities. Farquhar et al. [2005] also provide generalization bounds (again using Rademacher complexities) for the SVM-2K algorithm. These bounds characterize how much the complexity class of the hypothesis space decreases with the co-regularization. We can view these bounds as characterizing how much the variance of the algorithm decreases. In particular, as  $\lambda_{co}$  increases, this has the effect of decreasing the variance (as a harder constraint is being imposed). While these are valid generalization bounds (which compare the empirical expectation of a predictor to the true expectation), they do not address the bias issue of how performance could decrease as  $\lambda_{co}$  is increased too much. In particular, as  $\lambda_{co}$  is increased, the algorithm is not as free to use certain hypothesis (which we can think of as the bias). Roughly speaking, these previous multi-view results quantify how model complexity is reduced, but they do not specify *why* this is reasonable to do. Hence, to understand how unlabeled data could improve performance, we must characterize how much the co-regularization effects this bias-variance trade-off.

We address these issues under the recent PAC framework for semi-supervised learning of Balcan and Blum [2006] — though we generalize the setting for arbitrary loss functions (Balcan and Blum [2006] only considered the 0/1 loss).

Their framework assumes an *incompatibility* function — a function which scores how good hypothesis are just based on the underlying data distribution. They provide a general framework for characterizing how such an incompatibility function can reduce the need for labeled samples. Intuitively, one can view the co-regularizer as an incompatibility function, as it is scoring hypothesis based on unlabeled data — if a pair of hypothesis disagree strongly under the co-regularizer it is unlikely that they would be good predictors.

One of our main contributions for analyzing these co-regularization algorithms is that we show how the incompatibility function is really a derived property of the loss function — the incompatibility function needs to satisfy a rather mild inverse Lipschitz condition. Under relatively general conditions, incompatibility functions can be derived for many loss functions of interest — we provide examples for the (regularized) hinge loss, the square loss, for the 0/1 loss, and for strictly convex losses. Interestingly (and rather subtly), our incompatibility function for the 0/1 loss makes use of Tsybakov’s noise condition.

We then explicitly use the Rademacher bounds in Rosenberg and Bartlett [2007], Farquhar et al. [2005] to provide performance bounds under the multi-view assumption. These bounds characterize the bias-variance trade-off. We explicitly quantify how to set the co-regularization parameter  $\lambda_{co}$  in terms of  $\epsilon_{\text{info}}$ , showing that an appropriate setting of  $\lambda_{co}$  is  $O(1/\sqrt{\epsilon_{\text{info}}})$ . In particular, this shows it is appropriate for  $\lambda_{co} \rightarrow \infty$  as  $\epsilon_{\text{info}} \rightarrow 0$ , i.e. when the information theoretic assumption is as sharp as possible, we are permitted to co-regularize as hard as possible (without introducing any bias). For this case, the co-regularization algorithms obtain their maximal reduction in variance.

### 1.3 Dimensionality Reduction

While PCA is the time-honoured and simplest dimensionality reduction technique, there are few normative reasons as to why this technique is appropriate. The typical justification is that the top  $k$  principal directions are those which best reconstruct the data, in a mean squared sense. One common criticism of this oft used justification is that a rescaling of the data could change the outcome of PCA.

*Canonical Correlation Analysis* (CCA) [Hotelling, 1935] — like PCA but for the two view setting — also serves as a rather general and widely used dimensionality reduction technique. Roughly speaking, it uses the cross-correlation matrix between the two views to find the canonical directions — those directions which are most correlated (in a normalized sense) between the views. As a dimensionality reduction procedure, one can take the top  $k$  CCA directions which, roughly speaking, preserves the most correlated coordinates. However, unlike PCA, CCA is invariant to linear transformations of the data. (Under the linear transformation  $x_1 \rightarrow Lx_1$  and  $x_2 \rightarrow L'x_2$ , the result of CCA does not change. This is because CCA works in terms of normalized correlation coefficients.) We define CCA more precisely in Section 3.

In certain special cases, there are normative justifications for CCA as a dimensionality reduction technique. When  $x_1$  and  $x_2$  are jointly distributed as a Gaussian, the

Gaussian Information Bottleneck method [Chechik et al., 2005] shows that CCA provides an appropriate compression scheme (under the Information Bottleneck criterion [Tishby et al., 1999]). In a semi-supervised multi-view setting, Kakade and Foster [2007] show that CCA provides the natural dimensionality reduction technique by which one can project  $x$  onto a lower dimensional space (using CCA) and yet still retain predictive information about  $y$ . However, this work was rather specific to the square loss and used a multi-view assumption tailored to the square loss.

This work provides a normative justification of CCA in a rather broad sense — we generalize the work of Kakade and Foster [2007]. We consider a setting where we have a convex loss function of the form  $\ell(w \cdot x, y)$ , where either the loss function is strictly convex (e.g. log loss, square loss) or we use a strictly convex regularizer (e.g. hinge loss with  $L_2$  regularization). We show that, under the multi-view assumption above, if we perform CCA and project the data onto to the top  $k$  canonical directions (where  $k$  is determined by the canonical eigenspectrum), then this projection loses little predictive information about  $Y$ . Hence, our subsequent supervised learning problem is simpler as we can work with a lower dimensional space (with the knowledge that we have not thrown away useful predictive information in working with this lower dimensional space). We state this precisely in Section 3.

## 2 Co-Regularization and Compatibility

We now consider the PAC style semi-supervised framework introduced in Balcan and Blum [2006] and generalize the framework to general loss functions. We work with a prediction space  $\hat{\mathcal{Y}}$  that need not be equal to  $\mathcal{Y}$ . The goal is to learn a pair of predictors  $(f_1, f_2)$ , where  $f_1 : \mathcal{X}_1 \rightarrow \hat{\mathcal{Y}}$  and  $f_2 : \mathcal{X}_2 \rightarrow \hat{\mathcal{Y}}$ , based on the labeled and unlabeled data such that the expected loss of any one of these predictors is small. We work with loss functions (bounded in  $[0, 1]$ ) of the form  $\ell(f; (x_1, x_2, y))$  (usually the loss functions are of the more restricted form  $\ell(f(x), y)$  though in some cases, e.g. Example 4, this more general form is appropriate). Denote by  $L(f_1)$  the expected loss of  $f_1$ , i.e.  $L(f_1) = E\ell(f_1; (x_1, y))$ , and  $L(f_2)$  is similarly defined. Let  $\mathcal{F}_1$  and  $\mathcal{F}_2$  denote the hypothesis classes of interest, consisting of functions from  $\mathcal{X}_1$  (and, respectively,  $\mathcal{X}_2$ ) to the prediction space  $\hat{\mathcal{Y}}$ . Let a Bayes optimal predictor with respect to loss  $L$  based on input  $X_1, X_2$  be denoted by  $y^*(X_1, X_2)$ . So  $y^* \in \text{argmin}_f L(f)$ , where the argmin is over all functions. Similarly, let  $y_v^*$  for  $v \in \{1, 2\}$  be Bayes optimal predictors with respect to loss function  $L$  based on input  $X_v$ .

### 2.1 Compatible Function Classes

As discussed in the Introduction, to leverage our information theoretic assumption, we would like to say that a near optimal predictor using information from one view tends to agree with a near optimal predictor from another view. If this were the case, then the intuitive basis for an algorithm would be to find predictors from either view which tend to agree. However, quantifying this statement depends on the details of the loss function and the prediction space, since we need to specify a relationship between a measure of “closeness”

of the loss function and a measure of agreement between hypothesis. We do this in the following assumption, which can be considered an *inverse* Lipschitz condition, which bounds how close two functions are in terms of how close their loss is.

**Assumption 2 (Inverse Lipschitz Condition)** *There exists a symmetric function  $d : \hat{\mathcal{Y}} \times \hat{\mathcal{Y}} \rightarrow \mathbb{R}^+$  and a monotonically increasing non-negative function  $\Phi$  on the reals (with  $\Phi(0) = 0$ ) such that for all  $f$ ,*

$$E[d(f(x), y^*(x))] \leq \Phi(L(f) - L(y^*))$$

where the expectation is with respect to  $x = (x_1, x_2)$ , and  $y^*$  is some Bayes optimal predictor with respect to loss  $L$ . Furthermore, for  $v \in 1, 2$  and all  $f_v$ ,

$$E[d(f_v(x), y_v^*(x))] \leq \Phi(L(f_v) - L(y_v^*))$$

where  $y_v^*$  is a Bayes optimal predictor using only knowledge of  $x_v$ .

While we this assumption seems natural enough, we should note that there some subtleties. For example, if we are dealing with binary prediction and the 0/1 loss function (the binary classification loss), consider the case where the target function is complete noise. Here, all predictors are Bayes optimal and have the maximal error rate of 0.5. Hence, predictors can be far from agreeing yet they are all optimal. In general, for the 0/1 loss, the higher the noise, the less near-optimal predictors need to agree. In the next Subsection, we consider this case in more detail (in Example 2), and we also consider other commonly used loss functions.

While it is natural to assume that  $d$  satisfies the triangle inequality, there are some natural choices of  $d$  which do not satisfy this. In particular, in some cases we would like to use  $d(y, y') = (y - y')^2$ , which does not satisfy the triangle inequality. Hence, we only assume a relaxed version of the triangle inequality.

**Assumption 3 (Relaxed Triangle Inequality)** *For the function  $d$ , there exists a  $c_d \geq 1$  such that*

$$\forall \hat{y}_1, \hat{y}_2, \hat{y}_3 \in \hat{\mathcal{Y}}, \quad d(\hat{y}_1, \hat{y}_2) \leq c_d(d(\hat{y}_1, \hat{y}_3) + d(\hat{y}_3, \hat{y}_2))$$

We now introduce the incompatibility framework of Balcan and Blum [2006] for the multi-view setting. Here, we have a function  $\chi : \hat{\mathcal{Y}} \times \hat{\mathcal{Y}} \rightarrow \mathbb{R}^+$ , which we think of as scoring how incompatible two functions are. In particular, in this framework, they desire to use functions which are highly compatible. To formalize this, define the compatible function class with respect to incompatibility function  $\chi$  and some  $t \geq 0$  as those pairs of functions which are compatible to the tune of  $t$ , more precisely:

$$\mathcal{C}^X(t) = \{(f_1, f_2) : f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2 \text{ and } E[\chi(f_1, f_2)] \leq t\}$$

where we are slightly abusing notation by referring to  $\chi(f, f')$  as meaning  $\chi(f(x_1, x_2), f'(x_1, x_2))$ , which we do throughout.

In order to characterize how good this compatibility class is, in terms of our multi-view assumption, we need to also define the Bayes regret:

$$\epsilon_{\text{bayes}} = \max\{L(f_1^*) - L(y_1^*), L(f_2^*) - L(y_2^*)\}$$

where  $f_v^* \in \mathcal{F}_v$  is the optimal predictor for view  $v$  within the hypothesis class  $\mathcal{F}_v$ .

Our first result shows that for a particular choice of  $t$ , the incompatibility class contains a good pair of hypothesis.

**Theorem 1 (Bias)** *If Assumptions 1, 2, and 3 are satisfied, then given a loss function  $\ell$  bounded by 1 and if we set the incompatibility function to be  $d$ , i.e.  $\chi = d$ , then for  $t = 2c_d^2(\Phi(\sqrt{\epsilon_{\text{info}}}) + \Phi(\epsilon_{\text{bayes}}))$ , we have:*

$$\inf_{(f_1, f_2) \in \mathcal{C}^X(t)} \frac{L(f_1) + L(f_2)}{2} \leq L(y^*) + \epsilon_{\text{bayes}} + \sqrt{\epsilon_{\text{info}}}$$

(The proof is provided in the Appendix).

Of course, for convex loss functions we have  $L(\frac{f_1+f_2}{2}) \leq \frac{L(f_1)+L(f_2)}{2}$ .

The need for stating the bound in terms of the Bayes regret  $\epsilon_{\text{bayes}}$  is due to our information theoretic Assumption 1 not explicitly referring to any hypothesis classes  $\mathcal{F}_1$  and  $\mathcal{F}_2$ . The square root dependence on  $\epsilon_{\text{info}}$  is a result of using Pinsker's equality in the proof, which relates the  $L_1$  distance to the KL-distance (see Cover and Thomas [1991]).

Note that in Balcan and Blum [2006] they did *not* explicitly characterize the quality of the incompatibility class — they assumed that  $\chi$  was known and that a setting of  $t$  was known such that  $\mathcal{C}^X(t)$  contained a 'good' predictor. Here, we derive our incompatibility function and we specify a value  $t$ . Intuitively, this lemma characterizes the bias — the reduction in performance — by using  $\mathcal{C}^X(t)$  instead of the full hypothesis classes  $\mathcal{F}_1$  and  $\mathcal{F}_2$ , in terms of the error  $\epsilon_{\text{info}}$ .

We now provide examples of pairs  $\chi$  and  $\Phi$  for commonly used loss functions, showing that our multi-view framework is quite general.

## 2.2 Examples of Loss/Incompatibility Pairs

**Example 1 (Squared Loss)** *Let  $\mathcal{Y}, \hat{\mathcal{Y}} = \mathbb{R}$ . Consider the loss function  $\ell(\hat{y}, y) = (y - \hat{y})^2$ . Here, we can choose the incompatibility function  $\chi(\hat{y}_1, \hat{y}_2) = d(\hat{y}_1, \hat{y}_2) = (\hat{y}_1 - \hat{y}_2)^2$  and  $\Phi(x) = x$ . To see that this satisfies all the requisite assumptions, note that since  $(a - b)^2 \leq 2(a^2 + b^2)$ , we have that  $\chi$  satisfies the relaxed triangle inequality with  $c_d = 2$ . Also, since that  $y_v^* = E[Y|X_v]$  and  $y^* = E[Y|X_1, X_2]$ , we have:*

$$\begin{aligned} E(f_v - y_v^*)^2 &= E(f_v - y)^2 - E(y_v^* - y)^2, \\ E(f - y^*)^2 &= E(f - y)^2 - E(y^* - y)^2 \end{aligned}$$

so our inverse Lipschitz condition is satisfied with equality.

**Example 2 (Zero-one Loss)** *Here, we have  $\mathcal{Y}, \hat{\mathcal{Y}} = \{1, -1\}$  with  $\ell(\hat{y}, y) = \mathbb{1}_{\{y \neq \hat{y}\}}$ . As discussed in the previous Subsection, there is no natural choice of  $d$  and  $\Phi$  for this loss function, without further restrictions on the noise. Hence, let us assume that Tsybakov's noise condition [Tsybakov, 2004] holds for each view independently and for both views together for some noise exponent  $\alpha \in (0, 1]$ , which we define below. Now we can choose the incompatibility function  $\chi(\hat{y}_1, \hat{y}_2) = \mathbb{1}_{\{\hat{y}_1 \neq \hat{y}_2\}}$  with  $\Phi(x) = cx^\alpha$  where  $c > 0$  (defined below). Here,  $\chi$  is in fact a metric and hence satisfies the triangle inequality.*

To see that the choice of  $\Phi$  is appropriate, first note that by definition of Tsybakov's noise condition, for all  $f_1 : \mathcal{X}_1 \rightarrow \hat{\mathcal{Y}}$ ,  $f_2 : \mathcal{X}_2 \rightarrow \hat{\mathcal{Y}}$  and  $f : \mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \hat{\mathcal{Y}}$  there exists  $c > 0$  such that for  $v \in \{1, 2\}$

$$\Pr(f(X_v)(\eta_v(X_v) - \frac{1}{2}) \leq 0) \leq c(L(f_v) - L(y_v^*))^\alpha$$

and

$$\Pr(f(X_1, X_2)(\eta(X) - \frac{1}{2}) \leq 0) \leq c(L(f) - L(y^*))^\alpha$$

where  $\eta_v$  and  $\eta$  stand for  $P(Y = 1|X_v)$  and  $P(Y = 1|X_1, X_2)$  respectively. Now since  $\text{sign}(\eta(X) - \frac{1}{2})$  is the Bayes optimal predictor,  $\mathbb{1}_{\{f(X)(\eta(X) - \frac{1}{2}) \leq 0\}} = \mathbb{1}_{\{f(X) \neq y^*(X)\}} = \chi(f, y^*)$  and thus, under Tsybakov's noise condition, Assumption 2 is satisfied.

**Example 3 (Strictly Convex Losses)** Consider a loss function  $\ell(\hat{y}, y)$  where, for each  $y$ ,  $\ell(\cdot, y)$  is strictly convex with respect to pseudo-metric  $d$  with modulus of convexity  $\delta$  (defined below). Let the prediction space  $\hat{\mathcal{Y}}$  and output space  $\mathcal{Y}$  be bounded a subset of  $\mathbb{R}$ . Here,  $\chi(\hat{y}_1, \hat{y}_2) = \delta(d(\hat{y}_1, \hat{y}_2))$  satisfies Assumption 2 with  $\Phi(x) = \frac{x}{2}$  (provided the modulus of convexity function  $\delta(\epsilon) \leq \epsilon^p$  for some  $p > 0$ ). In this case it is easy to check that  $c_d = 1$  if  $p < 1$  and  $c_d = 2^{p-1}$  otherwise.

To see this, we first define modulus of convexity of the loss function  $\ell$  with respect to pseudometric  $d$  (in its first parameter). We say that for a given  $y$ ,  $\ell(\cdot, y)$  has modulus of convexity  $\delta$  if,

$$\delta_y(\epsilon) = \inf\left\{\frac{\ell(\hat{y}, y) + \ell(\hat{y}', y)}{2} - \ell\left(\frac{\hat{y} + \hat{y}'}{2}, y\right) : d(\hat{y}, \hat{y}') \geq \epsilon\right\}$$

where the inf is over  $\hat{y}, \hat{y}' \in \hat{\mathcal{Y}}$ . We actually want to work with a uniform bound on this function and so we define  $\delta$  to be any function satisfying,

$$\delta(\epsilon) \leq \inf_{y \in \mathcal{Y}} \delta_y(\epsilon)$$

Now note that

$$\frac{L(f_v) + L(y_v^*)}{2} - L\left(\frac{f_v + y_v^*}{2}\right) \geq E\delta(d(f_v, y_v^*))$$

and

$$\frac{L(f) + L(y^*)}{2} - L\left(\frac{f + y^*}{2}\right) \geq E\delta(d(f, y^*))$$

Since  $L(\frac{f_v + y_v^*}{2}) \geq L(y_v^*)$  and  $L(\frac{f + y^*}{2}) \geq L(y^*)$  we have that,

$$E[\chi(f_v, y_v^*)] = E\delta(d(f_v, y_v^*)) \leq \frac{L(f_v) - L(y_v^*)}{2}$$

and

$$E[\chi(f, y^*)] = E\delta(d(f, y^*)) \leq \frac{L(f) - L(y^*)}{2}$$

which shows our choice of  $\chi$  and  $\Phi$  is appropriate.

**Remark 1** It is worth noting that whenever Assumption 2 is satisfied with  $\chi(\hat{y}_1, \hat{y}_2) = g(d(\hat{y}_1, \hat{y}_2))$  where  $d$  is some pseudo-metric and  $g$  is an invertible convex function then Assumption 2 is also with  $\chi' = d$  as the incompatibility function and  $\Phi_{\chi'} = g^{-1}(\Phi)$ . This is a simple consequence of Jensen's inequality.

**Example 4 ( $L_2$  Regularized Losses)** Say we have some loss function  $\ell$  that is convex and  $\hat{\mathcal{Y}} = \mathbb{R}$ . Now consider the regularized loss functional for a certain RKHS function class  $\mathcal{F}$ ,

$$\ell_\lambda(f; x, y) := \ell(f(x), y) + \lambda\|f\|_K^2 \quad (2)$$

Taking  $\chi(\hat{y}_1, \hat{y}_2) = (\hat{y}_1 - \hat{y}_2)^2$  we can show that Assumption 2 is satisfied for the regularized loss with  $\Phi(x) = \frac{(K+\lambda)^2}{2\lambda}x$ , where  $K := \sup_{x \in \mathcal{X}} \sqrt{K(x, x)}$  (note that here we overload the notation  $K$ , but it is clear from context).

To see this, define for  $f, f' \in \mathcal{F}$  the metric

$$d_{\lambda, x}(f, f') = |f(x) - f'(x)| + \lambda\|f - f'\|_K$$

One can show that  $E[\ell_\lambda(f)]$  is strictly convex with respect to  $d_{\lambda, x}$  (Steinwart and Scovel [2006], Lemma 6.4) with modulus of convexity  $\delta(\epsilon) = \frac{\lambda\epsilon^2}{(K+\lambda)^2}$ . From this we see that

$$\begin{aligned} & \frac{E[\ell_\lambda(f; x, y)] - E[\ell_\lambda(f^*; x, y)]}{2} \\ & \geq E\left[\frac{\ell_\lambda(f; x, y) + \ell_\lambda(f^*; x, y)}{2} - \ell_\lambda\left(\frac{f + f^*}{2}; x, y\right)\right] \\ & \geq E\delta(d'_{\lambda, x}(f, f^*)) \\ & \geq E\delta(|f(x) - f^*(x)| + \lambda\|f - f^*\|) \\ & \geq E\delta(|f(x) - f^*(x)|) \\ & \geq \frac{\lambda}{(K + \lambda)^2} E(f(x) - f^*(x))^2 \end{aligned}$$

Thus we see that for the regularized loss functional  $\ell_\lambda$  the squared incompatibility satisfies Assumption 2, with our choice of  $\Phi(x) = \frac{(K+\lambda)^2}{2\lambda}x$ .

### 2.3 Convergence Bounds

We now characterize the sample complexity of an algorithm which uses a labeled and unlabeled data set, sampled from the underlying distribution. Our framework again parallels that of Balcan and Blum [2006] — broadened to include more general loss functions.

The basic algorithm we consider is identical to that in Balcan and Blum [2006]. Given an unlabeled data set  $U$ , we define the empirical compatibility class as:

$$\widehat{\mathcal{C}}^x(t) = \{(f_1, f_2) : f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2 \text{ and } \widehat{E}_U[\chi(f_1, f_2)] \leq t\}$$

where the empirical expectation is:

$$\widehat{E}_U[\chi(f_1, f_2)] = \frac{1}{m} \sum_{(x_1, x_2) \in U} \chi(f_1(x_1), f_2(x_2)).$$

The algorithm simply minimizes the average loss of predictions over labeled data subject to the constraint of choosing

hypothesis from  $\widehat{\mathcal{C}}^\chi(t)$ . More precisely, for a given  $t$ , the algorithm simply chooses the best pair in this class:

$$(\widehat{f}_1, \widehat{f}_2) = \operatorname{argmin}_{f_1, f_2 \in \widehat{\mathcal{C}}^\chi(t)} \widehat{E}_S[\ell(f_1(x_1), y) + \ell(f_2(x_2), y)] \quad (3)$$

The co-regularization algorithm can be viewed as a dual version of this algorithm, which we consider in the following Subsection.

As we are dealing with abstract hypothesis classes, as in Balcan and Blum [2006], we make an assumption about the learning complexity with respect to these abstract hypothesis class — we give examples shortly. This assumption is stated in terms of both  $S$  and  $U$ , which allows us to use data-dependent sample complexity bounds (such as the Rademacher bounds), which is important in the next Subsection (for the analysis of the co-regularization algorithms and SVM-2K).

**Assumption 4 (Sample Complexity)** For the hypothesis classes  $\mathcal{F}_1$  and  $\mathcal{F}_2$ ,

**Unlabeled:** With probability greater than  $1 - \delta$  over the i.i.d. sampling of unlabeled data set  $U$  we have that  $\forall (f_1, f_2) \in \mathcal{F}_1 \times \mathcal{F}_2$

$$\widehat{E}[\chi(f_1, f_2)] \leq E[\chi(f_1, f_2)] + G_\chi(\mathcal{F}_1 \times \mathcal{F}_2, U, \delta)$$

where  $G_\chi$  is some notion of the generalization of the function class.

**Labelled Case:** For any given unlabeled data set  $U$ , with probability greater than  $1 - \delta$  over i.i.d sampling of labeled data set  $S$  we have that for all pairs  $(f_1, f_2) \in \widehat{\mathcal{C}}^\chi(t)$ ,

$$|L(f_1) + L(f_2) - (\widehat{L}(f_1) + \widehat{L}(f_2))| \leq G_\ell(\widehat{\mathcal{C}}^\chi(t), S, \delta)$$

where  $G_\ell$  is some notion of the generalization of the function class.

We now provide some standard sample complexity bounds.

**Remark 2 (Examples of  $G_\chi$  and  $G_\ell$ )** Assumption 4 is satisfied in the following standard examples.

**Finite Hypothesis Class:** If the hypothesis classes are finite, then using Chernoff and union bounds we have

$$G_\chi(\mathcal{H}, U, \delta) = O\left(\sqrt{\frac{\log(|\mathcal{H}|) + \log(\frac{1}{\delta})}{m}}\right)$$

and  $G_\chi = G_\ell$ .

**Finite VC Class:** If the hypotheses map to  $[0, 1]$  and the VC dimension is finite, then

$$G_\chi(\mathcal{H}, U, \delta) = O\left(\sqrt{\frac{VCdim(\mathcal{H}) + \log(\frac{1}{\delta})}{m}}\right)$$

and  $G_\chi = G_\ell$ .

**Rademacher Bounds :** For bounded loss and incompatibility functions, Rademacher bounds give us:

$$G_\chi(\mathcal{H}, U, \delta) = O\left(\widehat{R}_m(\mathcal{H}) + 3\sqrt{\frac{\ln(2/\delta)}{2m}}\right)$$

and  $G_\chi = G_\ell$ . Here,  $\widehat{R}_n(\mathcal{H}) = \frac{1}{n} E_S \sup_{f \in \mathcal{H}} \sum_{i=1}^n \sigma_i f(x_i)$  where  $\sigma_i$  are Rademacher variables.

We are now ready to state our main result on the complexity of our multi-view algorithm.

**Theorem 2** Assume that the function  $\ell$  is bounded by 1, the incompatibility function  $\chi = d$  and that Assumptions 1, 2, 3 and 4 hold. Set

$$t = 2c_d^2(\Phi(\sqrt{\epsilon_{\text{info}}}) + \Phi(\epsilon_{\text{bayes}})) + G_\chi(\mathcal{F}_1 \times \mathcal{F}_2, U, \delta)$$

and let the pair  $(\widehat{f}_1, \widehat{f}_2)$  be the output of the algorithm (as defined by Equation 3) with this setting of  $t$ . Then with probability greater than  $1 - \delta$  over an i.i.d sample of both the labeled dataset  $S$  and unlabeled dataset  $U$ , we have

$$\frac{L(\widehat{f}_1) + L(\widehat{f}_2)}{2} \leq L(y^*) + G_\ell(\widehat{\mathcal{C}}^\chi(t), S, \delta/3) + \epsilon_{\text{bayes}} + \sqrt{\epsilon_{\text{info}}}$$

(The proof is provided in the Appendix).

This statement is analogous to the main complexity statements in the semi-supervised PAC framework of Balcan and Blum [2006]. In particular, the unlabeled complexity  $G_\chi$  only alters the setting of  $t$ , just as in Balcan and Blum [2006]. The labeled complexity term,  $G_\ell$ , appears as a penalization to the bound, again as in the semi-supervised PAC framework.

The main difference is that we now specify the value of  $t$  to be used and compare ourselves to the Bayes optimal. Note that in Balcan and Blum [2006], there is no explicit characterization as to how much bias is introduced by using  $\mathcal{C}^\chi(t)$  as opposed to using the unconstrained hypothesis space. The information theoretic assumption is what allows us to make this explicit characterization. The term  $\sqrt{\epsilon_{\text{info}}}$  is the bias introduced by using the constrained hypothesis space rather than the unconstrained hypothesis space. The benefit is that we could substantially reduce the variance. In particular, this variance reduction is reflected by that the labeled complexity term,  $G_\ell$ , only depends on the restricted hypothesis space,  $\widehat{\mathcal{C}}^\chi(t)$ , rather than the full hypothesis space — the former of which could have significantly less complexity.

We now show specific algorithms and analyses fit into this framework.

## 2.4 Algorithms

We now provide bounds for co-regularization algorithms and the SVM-2K algorithm of Farquhar et al. [2005]. For  $v \in \{1, 2\}$  let  $\mathcal{F}_v$  be some RKHS with respect to norm  $\|\cdot\|_K$ . Define  $\ell_\lambda$  as in Example 4, i.e.

$$\ell_\lambda(f; x, y) := \ell(f(x), y) + \lambda \|f\|_K^2 \quad (4)$$

where  $\ell(f(x), y)$  is convex. Define

$$L_\lambda(f) := E\ell_\lambda(f; (x_1, x_2, y)) .$$

Also let

$$f^* = \operatorname{argmin}_f E[L_\lambda(f)]$$

where the  $\operatorname{argmin}$  is over all functions (so  $f_*$  is the Bayes optimal predictor). By the Representer Theorem,  $f^*$  lives in the RKHS. This implies that  $\epsilon_{\text{baves}} = 0$ .

Throughout this section we overload notation by using  $K := \sup_{x \in \mathcal{X}} \sqrt{K(x, x)}$  (when it is clear from context).

### Co-Regularization (with squared incompatibility)

The original co-regularization algorithm introduced in Sindhwani et al. [2005] and also the co-regularized least squares regression Brefeld et al. [2006] both minimize the objective in Equation 1. Recall that for the regularized convex loss functions in Example 4, we already showed that  $\chi(f_1(x_1), f_2(x_2)) = (f_1(x_1) - f_2(x_2))^2$  satisfies Assumption 2. Therefore we see that Theorem 2 justifies these co-regularization algorithms under the information theoretic Assumption 1.

Rosenberg and Bartlett [2007] provide an estimate for the Rademacher complexity of kernel class for co-regularization in a transductive type setting (i.e. conditioned on the unlabeled data). The bound given is exactly of the form needed in Assumption 4. The subtlety in using these complexity bounds is that the co-regularization algorithms are a dual formulation of our Algorithm (see Equation 3), the latter of which imposes a hard agreement constraint. Hence, to provide a bound we need find an appropriate setting of the parameter  $\lambda_{co}$ . The following theorem does this.

**Corollary 3** *Assume we are working in the transductive setting (where  $U$  is known and the underlying data distribution is uniform over  $U$ ). Let  $C_{lip}$  be the Lipschitz constant for the loss. Let  $K_{S \times S}^v$ ,  $K_{S \times U}^v$  and  $K_{U \times U}^v$  stand for the kernel matrix between labeled examples, between labeled and unlabeled examples, and unlabeled and unlabeled samples for view  $v \in \{1, 2\}$  respectively.*

*Given  $\lambda > 0$ , if we set  $\lambda_{co} = \frac{\lambda}{4(K+\lambda)^2 \sqrt{\epsilon_{\text{info}}}}$  then for the pair of functions  $(\hat{f}_1, \hat{f}_2) \in \mathcal{F}_1 \times \mathcal{F}_2$  returned by the co-regularization algorithm (Equation 1), with probability at least  $1 - \delta$  over labeled samples,*

$$L_\lambda\left(\frac{\hat{f}_1 + \hat{f}_2}{2}\right) \leq L_\lambda(f^*) + \frac{1}{\sqrt{n}} \left( 2 + 3\sqrt{\frac{\ln(\frac{2}{\delta})}{2}} \right) + 2C_{Lip} \hat{R}_n\left(\hat{\mathcal{C}}^x\left(\frac{1}{\lambda_{co}}\right)\right) + \sqrt{\epsilon_{\text{info}}}$$

Where,

$$\hat{R}_n\left(\hat{\mathcal{C}}^x\left(\frac{1}{\lambda_{co}}\right)\right) \leq \frac{R}{n}$$

$$R^2 = \lambda^{-1} \operatorname{tr}(K_{S \times S}^1) + \lambda^{-1} \operatorname{tr}(K_{S \times S}^2) - \frac{\lambda}{4(K+\lambda)^2 \sqrt{\epsilon_{\text{info}}}} \operatorname{tr}(J^T (I + \lambda M)^{-1} J)$$

$$J = \lambda^{-1} K_{U \times S}^1 - \lambda^{-1} K_{U \times S}^2, \quad M = \lambda^{-1} K_{U \times U}^1 - \lambda^{-1} K_{U \times U}^2$$

(The proof is provided in the Appendix).

An important difference between our bounds and that in Rosenberg and Bartlett [2007] is that the above bound

compares to the Bayes optimal predictor  $f^*$ , while Rosenberg and Bartlett [2007] only compare to the best function in  $\hat{\mathcal{C}}^x(t)$  (without any normative justification for how to set the parameter  $t$ ). Our comparison to  $f^*$  leads to the additional penalty of  $\sqrt{\epsilon_{\text{info}}}$  (and we specify a value of  $\lambda_{co}$  in the bound).

Note that the appropriate setting of  $\lambda_{co}$  is  $O(1/\sqrt{\epsilon_{\text{info}}})$ . In particular, this shows it is appropriate for  $\lambda_{co} \rightarrow \infty$  as  $\epsilon_{\text{info}} \rightarrow 0$ , i.e. when the information theoretic assumption is as sharp as possible, we are permitted to co-regularize as hard as possible (without introducing any bias). For this case, the co-regularization algorithms obtain their maximal reduction in variance.

To convert the above corollary to an inductive bound (where  $U$  is a random sample) we need to establish an unlabeled complexity statement of the kind in Assumption 4. Note that if the prediction space is bounded then it can be shown using covering number arguments (Zhang [2002]) that  $G_\chi(\mathcal{F}_1 \times \mathcal{F}_2, U, \delta)$  will be  $c\sqrt{\frac{\log(1/\delta)}{m}}$  where  $c$  is some constant (which depends of  $\lambda_{co}$  and  $K$ ). Hence by setting  $t = 2c_d^2(\Phi(\epsilon_{\text{baves}}) + \Phi(\sqrt{\epsilon_{\text{info}}})) + c\sqrt{\frac{\log(1/\delta)}{m}}$  we can get the inductive statement required.

### Two View SVM

The SVM-2K approach proposed by Farquhar et al. [2005] can be formulated as the following optimization problem:

$$\operatorname{argmin}_{(f_1, f_2) \in \mathcal{F}_1 \times \mathcal{F}_2} \frac{1}{2} (\hat{E}_S[\ell(f_1(x_1), y)] + \hat{E}_S[\ell(f_2(x_2), y)]) + \lambda \|f_1\|_K^2 + \lambda \|f_2\|_K^2 + \lambda_{co} \hat{E}_U[|f_1(x_1) - f_2(x_2)|] \quad (5)$$

where  $\ell$  is the hinge loss. Technically, the formulation in Farquhar et al. [2005] uses slack variables (more in line with the usual SVM formulation), but the above formulation is identical.<sup>1</sup>

SVM-2K can be viewed as using the incompatibility function  $\chi(\hat{y}_1, \hat{y}_2) = |\hat{y}_1 - \hat{y}_2|$ . Recall that for regularized convex loss functions in Example 4, we already showed that  $(f_1(x_1) - f_2(x_2))^2$  satisfies Assumption 2. Hence using Remark 1 we see that this incompatibility function for SVM-2K also satisfies Assumption 3 and 2 with  $c_d = 1$  and  $\phi(x) = \sqrt{\frac{(K+\lambda)^2}{2\lambda}} x$ . Hence, we get the following Corollary.

**Corollary 4** *Assume we are working in the transductive setting (where  $U$  is known and the underlying data distribution is uniform over  $U$ ). Given  $\lambda > 0$ , if we set and  $\lambda_{co} = \frac{\lambda}{2(K+\lambda)^2 \sqrt{\epsilon_{\text{info}}}}$  then with probability at least  $1 - \delta$  over labeled samples, for the pair of functions  $(\hat{f}_1, \hat{f}_2) \in \mathcal{F}_1 \times \mathcal{F}_2$  returned by SVM-2K algorithm (Equation 5),*

$$L_\lambda\left(\frac{\hat{f}_1 + \hat{f}_2}{2}\right) \leq L_\lambda(f^*) + 2\hat{R}_n\left(\hat{\mathcal{C}}^x\left(\frac{1}{\lambda_{co}}\right)\right) + 3\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} + \sqrt{\epsilon_{\text{info}}}$$

where  $\hat{R}_n\left(\hat{\mathcal{C}}^x\left(\frac{1}{\lambda_{co}}\right)\right)$  is the data-dependent Rademacher complexity.

<sup>1</sup>Technically, the SVM-2K algorithm has a parameter  $\epsilon$  which allows a little more disagreement, but the algorithm we specify is equivalent to the SVM-2K algorithm with  $\epsilon = 0$ .

In particular, Farquhar et al. [2005] show how to upper bound  $\widehat{R}_n(\widehat{C}^\chi(t))$  as a solution to a particular optimization problem. The proof is essentially identical to the previous Corollary, and is not provided.

Again, the main extension in our work is that we compare the algorithm’s performance to the loss of the Bayes optimal predictor  $f^*$ , while Farquhar et al. [2005] only compares to the best function in  $\widehat{C}^\chi(t)$ . Our comparison to  $f^*$  leads to the additional penalty of  $\sqrt{\epsilon_{\text{info}}}$  (and we specify a value of  $t$  in the bound).

The appropriate setting of  $\lambda_{\text{co}}$  is  $O(1/\sqrt{\epsilon_{\text{info}}})$  which again shows that smaller  $\epsilon_{\text{info}}$  gets, the harder we can co-regularize.

### 3 Dimensionality Reduction and CCA

Consider a setting where  $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$  is a real vector space (of finite or countably infinite dimension). Here, we work with linear predictors of the form  $w^T x$  and convex losses of form  $\ell(w^T x, y)$  that satisfy Assumptions 2 and 3 with respect to the squared incompatibility function. For example, most strictly convex loss functions can be used with the squared incompatibility function, including the square loss, log loss, exponential loss, and  $L_2$  regularized losses. Let  $L(w) = E[\ell(w^T x, y)]$ . For simplicity, we work in the transductive setting — in particular, we only assume knowledge of the second order statistics of the underlying data distribution (i.e. we know the covariance matrix of  $\mathcal{X}$ ).

Assume that the loss function is twice differentiable and that the second derivative of the loss function is bounded from above by some constant  $C$ , that is

$$\forall z \frac{d^2 \ell(z, y)}{dz^2} \leq C \quad (6)$$

Note that this assumption is satisfied for common strictly convex losses.

Define canonical correlation analysis (CCA) as follows:

**Definition 5** *The bases  $B_1, B_2$  for  $\mathcal{X}_1$  and  $\mathcal{X}_2$  is the canonical basis for the two views if for  $(x_1, x_2)$  in this basis the following holds:*

1. *Orthogonality Conditions: For  $v \in \{1, 2\}$*

$$E[(x_v)_i (x_v)_j] = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

2. *Correlation Conditions:*

$$E[(x_1)_i (x_2)_j] = \begin{cases} \gamma_i & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

where  $\gamma_i$  is the  $i^{\text{th}}$  correlation coefficient. We assume without loss of generality that  $1 \geq \gamma_1 \geq \gamma_2 \geq \dots \geq 0$ .

Now we present the main algorithm, which uses CCA as a dimensionality reduction technique. Consider some threshold,  $0 < \gamma_{\text{thresh}} < 1$ . Let  $i_{\text{thresh}}$  be the smallest  $i$  such that

$$\gamma_i < \gamma_{\text{thresh}}$$

First, project  $x_v$  to the subspace spanned by the first  $1, \dots, i_{\text{thresh}}$  canonical coordinates. Denote this projection

by  $\Pi_{\text{cca}}(x_v)$ . Let  $\beta_{\text{proj}}^{(v)}$  be the optimal linear predictor for view  $v$  using only the projected  $\Pi_{\text{cca}}(x_v)$  as input.

We now show that the loss of performance due to this projection is small if  $\epsilon_{\text{info}}$  is small.

**Theorem 6** *Assume that Equation 6 holds, that Assumption 1 is satisfied, and that Assumptions 2 and 3 hold with respect to the squared incompatibility function. Then*

$$L(\beta_{\text{proj}}^{(v)}) - L(y_v^*) \leq \frac{4C (\Phi(\sqrt{\epsilon_{\text{info}}}) + \Phi(\epsilon_{\text{bayes}}))}{1 - \gamma_{\text{thresh}}} + \epsilon_{\text{bayes}}$$

where  $C$  satisfies Equation 6.

(The proof is provided in the Appendix).

In particular, if the cutoff,  $\gamma_{\text{thresh}}$ , is  $\frac{1}{2}$ , then makes the  $\frac{1}{1 - \gamma_{\text{thresh}}}$  factor in the bound into 2.

Let us consider the implications for learning with a random labeled data set  $S$  using  $\Pi_{\text{cca}}(x_v)$ . Here, the a learning algorithm only needs to work with the coordinates which have sufficiently large  $\gamma_i$ . Hence, the supervised learning problem is simpler as we can work with a lower dimensional space. This Theorem is analogous to the dimensionality reduction statements in Kakade and Foster [2007] — though there the statements were restricted to the square loss (and a multi-view assumption based on the square loss).

## 4 Discussion

### An Open Problem from Balcan and Blum [2007]

This problem (presented at COLT 2007) is where we have the 0/1 loss, and it is assumed that classifiers from either view can perfectly predict the data (so the best classifiers agree completely on the unlabeled data). Furthermore, they assume that the classifiers are linearly separable. The question posed is can an efficient algorithm be found? A more general and practically relevant question is this case but with noise, which of course makes the problem harder. Here, the optimal predictors (from either view) may not agree perfectly on the unlabeled data. However, under Example 2, we know that choosing  $d$  to be the 0/1 loss is a suitable discrepancy function (with  $\Phi$  being defined in terms of the Tsybakov noise exponent).

In practice, even in the single view case, one is rarely able to directly minimize the 0/1 loss. Instead, what one actually does is minimize a surrogate loss function, such as the hinge loss, logistic loss, or exponential loss. Furthermore, through the work of Bartlett et al. [2006], we have an understanding of how minimizing these surrogate losses relate to the 0/1 loss.

In our framework, we are able to choose a discrepancy functions tailored to our loss (as long as the discrepancy satisfies Assumption 2). Hence, if we are using a surrogate loss (for the 0/1 loss) then we should choose a incompatibility function that satisfies Assumption 2 with respect to this surrogate loss. We view both the co-regulation algorithms and the SVM-2K algorithm as the solution to this problem, under the theory of surrogate losses (where both these algorithms are using the surrogate hinge loss).

## 4.1 Relations to the Information Bottleneck

We end with a note on the connection to the Information Bottleneck method. In this method, the goal is to compress  $X_1$  to  $Z$  such that  $Z$  has maximum information about  $X_2$  — in particular,  $Z$  is a compression of  $X_1$  that retains all the information that  $X_1$  has about  $X_2$ , that is,

$$Z = \underset{A}{\operatorname{argmin}} I(A : X_1)$$

s.t.  $I(A : X_2) = I(X_1 : X_2)$

where the argmin is over compression functions  $A$  of  $X_1$ .

In the multi-view setting, if we find such a  $Z$  (with respect to  $X_1$  and  $X_2$ ), it can be shown that

$$I(Z : Y) \geq I(X_1 : Y) - \epsilon_{\text{info}}$$

This shows that  $Z$  loses little predictive information about  $Y$ . In this sense, the Information Bottleneck is not throwing much relevant information with regards to  $Y$  and can be used as a semi-supervised algorithm.

In fact, using Lemma 7, one can show that for any loss bounded by 1, the Bayes optimal predictor which uses only knowledge of  $Z$  has a regret of at most  $\sqrt{\epsilon_{\text{info}}}$  with respect to the Bayes optimal predictor  $y^*$ . An interesting direction to pursue is to learn with  $Z$  as inputs to our learning algorithm rather than  $X_v$ , since  $Z$  has lower entropy. Two issues to consider are: 1) the mapping  $Z$  has an abstract range (so one needs to take care in how to learn a function from  $Z \rightarrow Y$ ) and 2) it is not clear how to implement the Information Bottleneck without knowledge of the underlying distribution.

## Acknowledgements

We thank Gilles Blanchard for a number of helpful suggestions.

## References

- Steven Abney. Understanding the yarowsky algorithm. *Comput. Linguist.*, 30(3):365–395, 2004. ISSN 0891-2017.
- Maria-Florina Balcan and Avrim Blum. A pac-style model for learning from labeled and unlabeled data. In *Semi-Supervised Learning*, pages 111–126. MIT Press, 2006.
- Maria-Florina Balcan and Avrim Blum. Open problems in efficient semi-supervised pac learning. In *Conference on Computational Learning Theory (COLT)*, 2007.
- Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. In *Journal of the American Statistical Association*, volume 101, No. 473, pages 138–156, 2006.
- Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT' 98: Proceedings of the eleventh annual conference on computational learning theory*, pages 92–100, New York, NY, USA, 1998. ACM Press. ISBN 1-58113-057-0.
- Ulf Brefeld, Thomas Gartner, Tobias Scheffer, and Stefan Wrobel. Efficient co-regularised least squares regression. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 137–144, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-383-2.
- Gal Chechik, Amir Globerson, Naftali Tishby, and Yair Weiss. Information bottleneck for gaussian variables. *J. Mach. Learn. Res.*, 6:165–188, 2005. ISSN 1533-7928.
- Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, 1991.
- Sanjoy Dasgupta, Michael L. Littman, and David A. McAllester. Pac generalization bounds for co-training. In *NIPS*, pages 375–382, 2001.
- Jason D. R. Farquhar, David R. Hardoon, Hongying Meng, John Shawe-Taylor, and Sndor Szedmk. Two view learning: Svm-2k, theory and practice. In *NIPS*, 2005.
- H. Hotelling. The most predictable criterion. *Journal of Educational Psychology*, 26:139–142, 1935.
- Sham M. Kakade and Dean P. Foster. Multi-view regression via canonical correlation analysis. In Nader H. Bshouty and Claudio Gentile, editors, *COLT*, volume 4539 of *Lecture Notes in Computer Science*, pages 82–96. Springer, 2007.
- David Rosenberg and Peter L. Bartlett. The rademacher complexity of co-regularized kernel classes. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.
- V. Sindhwani, P. Niyogi, and M. Belkin. A Co-Regularization Approach to Semi-supervised Learning with Multiple Views. In *Workshop on Learning with Multiple Views, Proceedings of International Conference on Machine Learning*, 2005.
- Ingo Steinwart and C. Scovel. Fast rates for support vector machines using gaussian kernels. In Los Alamos National Laboratory Technical Report LA-UR-04-8796, editor, *Annals of Statistics*, 2006.
- N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.
- A. Tsybakov. Optimal aggregation of classifiers in statistical learning. In *Annals of Statistics*, volume 32 No. 1, 2004.
- Tong Zhang. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2:527–550, 2002.

## A Proofs

First we state two Lemmas that will be used in proving the theorem.

**Lemma 7** For  $v \in \{1, 2\}$ , if the loss function  $\ell$  is bounded by 1 then we have that

$$|L(y^*) - L(y_v^*)| \leq \sqrt{\epsilon_{\text{info}}} \quad \text{and} \quad |L(y_1^*) - L(y_2^*)| \leq 2\sqrt{\epsilon_{\text{info}}}$$

**Proof:** Consider some function  $g : \mathcal{X} \rightarrow [0, 1]$  and some two probability measures  $P$  and  $Q$ . We have that

$$\begin{aligned} \left| \int g(x)dQ - \int g(x)dP \right| &= \left| \int (1 - \beta)g(x)dQ \right| \\ &\leq \int |1 - \beta|dQ \\ &\leq \sqrt{DK(Q\|P)} \end{aligned} \quad (7)$$

where  $\beta = \frac{dP}{dQ}$  and the last step is because the  $L_1$  variational distance is bounded by square root of the KL divergence (Pinsker's Inequality). Now using this we get that for a fixed  $x_1, x_2$  we have that

$$\begin{aligned} |E_{Y|X_1=x_1} \ell(y^*(x_1, x_2), y) - E_{Y|X=(x_1, x_2)} \ell(y^*(x_1, x_2), y)| \\ \leq \sqrt{DK(P_{Y|X=(x_1, x_2)}\|P_{Y|X_1=x_1})} \end{aligned}$$

Taking expectation with respect to  $X = (X_1, X_2)$  and using Jensen's inequality twice (once on the left for convex function  $|x|$  and once on the right for concave function  $\sqrt{x}$ ) we get that

$$\begin{aligned} |E_X E_{Y|X_1=x_1} \ell(y^*(x_1, x_2), y) - L(y^*)| \\ \leq \sqrt{E_X DK(P_{Y|X=(x_1, x_2)}\|P_{Y|X_1=x_1})} \end{aligned}$$

Now note that since

$$L(y_1^*) \leq E_X E_{Y|X_1=x_1} \ell(y^*(x_1, x_2), y)$$

and  $L(y_1^*) \geq L(y^*)$ , we get

$$\begin{aligned} |L(y_1^*) - L(y^*)| \\ \leq \sqrt{E_X DK(P_{Y|X=(x_1, x_2)}\|P_{Y|X_1=x_1})} \end{aligned}$$

Also,

$$E_X DK(P_{Y|X=(x_1, x_2)}\|P_{Y|X_1=x_1}) = I_{Y: X_2 | X_1}$$

and so we have that

$$|L(y_1^*) - L(y^*)| \leq \sqrt{\epsilon_{\text{info}}}$$

similarly we have

$$|L(y_2^*) - L(y^*)| \leq \sqrt{\epsilon_{\text{info}}}$$

Also the above two inequalities together imply that

$$|L(y_1^*) - L(y_2^*)| \leq 2\sqrt{\epsilon_{\text{info}}}$$

**Lemma 8** For any  $f_1, f_2$  assume

$$L(f_1) - L(y_1^*) \leq \epsilon', \quad L(f_2) - L(y_1^*) \leq \epsilon'$$

then given Assumptions 1, 2 and 3 and that the loss function is bounded by  $B$ , we have that

$$E[\chi(f_1, f_2)] \leq 2c_d^2(\Phi(\epsilon') + \Phi(\sqrt{\epsilon_{\text{info}}}))$$

**Proof:** First note that by Assumptions 2 and 3 we have that for  $f_1$  and  $f_2$  there exists  $y_1^*$  and  $y_2^*$  such that

$$\begin{aligned} E[\chi(f_1, y_1^*)] &\leq \Phi(L(f_1) - L(y_1^*)) \quad \text{and} \\ E[\chi(f_2, y_2^*)] &\leq \Phi(L(f_2) - L(y_2^*)) \end{aligned}$$

and since  $\Phi$  is monotonically increasing we have that

$$\begin{aligned} E[\chi(f_1, y_1^*)] &\leq \Phi(\epsilon') \quad \text{and} \\ E[\chi(f_2, y_2^*)] &\leq \Phi(\epsilon') \end{aligned}$$

Again by Assumptions 2 and 3 we have that for some specific  $y^*$ ,

$$E[\chi(y_1^*, y^*)] \leq \Phi(L(y_1^*) - L(y^*)) \leq \Phi(\sqrt{\epsilon_{\text{info}}})$$

and

$$E[\chi(y_2^*, y^*)] \leq \Phi(L(y_2^*) - L(y^*)) \leq \Phi(\sqrt{\epsilon_{\text{info}}})$$

Since  $\chi$  satisfies the relaxed triangle inequality Assumption 3, we get that

$$E[\chi(y_2^*, y_1^*)] \leq c_d \Phi(\sqrt{\epsilon_{\text{info}}})$$

Again using relaxed triangle inequality Assumption 3, we get the required result that

$$\begin{aligned} E[\chi(f_1, f_2)] &\leq c_d^2(E[\chi(f_1, y_1^*)] + E[\chi(y_1^*, y_2^*)] + E[\chi(f_2, y_2^*)]) \\ &\leq 2c_d^2(\Phi(\epsilon') + \Phi(\sqrt{\epsilon_{\text{info}}})) \end{aligned}$$

**Proof:**[of Theorem 1]

Using Lemma 8 we see that

$$E[\chi(f_1^*, f_2^*)] \leq 2c_d^2(\Phi(\epsilon_{\text{bayes}}) + \Phi(\sqrt{\epsilon_{\text{info}}}))$$

Therefore setting  $t = 2c_d^2(\Phi(\epsilon_{\text{bayes}}) + \Phi(\sqrt{\epsilon_{\text{info}}}))$  we find that  $(f_1^*, f_2^*) \in \mathcal{C}^X(t)$  and thus,

$$(f_1^*, f_2^*) = \underset{(f_1, f_2) \in \mathcal{C}^X(t)}{\operatorname{argmin}} \frac{L(f_1) + L(f_2)}{2}$$

Now by definition of  $\epsilon_{\text{bayes}}$  we have that

$$\min_{f_v \in \mathcal{F}_v} L(f_v) - L(y_v^*) \leq \epsilon_{\text{bayes}}$$

Therefore,

$$\min_{(f_1, f_2) \in \mathcal{C}^X(t)} \frac{L(f_1) + L(f_2)}{2} \leq \frac{L(y_1^*) + L(y_2^*)}{2} + \epsilon_{\text{bayes}} \quad (8)$$

Now by Lemma 7 we see that for each  $v \in \{1, 2\}$ ,  $L(y_v^*) - L(y^*) \leq \sqrt{\epsilon_{\text{info}}}$ . Hence using this in Equation (8) we conclude that

$$\min_{(f_1, f_2) \in \mathcal{C}^X(t)} \frac{L(f_1) + L(f_2)}{2} \leq L(y^*) + \epsilon_{\text{bayes}} + \sqrt{\epsilon_{\text{info}}}$$

**Proof:**[of Theorem 2] Let  $(f_1^*_{\widehat{\mathcal{C}}_O}, f_2^*_{\widehat{\mathcal{C}}_O}) \in \widehat{\mathcal{C}}^X(t)$  be the minimizer of  $L(f_1) + L(f_2)$  in the class  $\widehat{\mathcal{C}}^X(t)$ . Using statement Assumption 4 (labeled) we have that with probability at least  $1 - \delta$  over the sample  $S$ ,

$$\begin{aligned} \widehat{L}(f_1^*_{\widehat{\mathcal{C}}_O}) + \widehat{L}(f_2^*_{\widehat{\mathcal{C}}_O}) - L(f_1^*_{\widehat{\mathcal{C}}_O}) - L(f_2^*_{\widehat{\mathcal{C}}_O}) \\ \leq G_\ell(\widehat{\mathcal{C}}^X(t), S, \delta) \end{aligned}$$

Also for any  $(f_1, f_2) \in \widehat{\mathcal{C}}^x(t)$  we have that with probability at least  $1 - \delta$  over the sample  $S$ ,

$$L(f_1) + L(f_2) - \widehat{L}(f_1) - \widehat{L}(f_2) \leq G_\ell(\widehat{\mathcal{C}}^x(t), S, \delta)$$

Hence combining the two, for the pair  $(\widehat{f}_1, \widehat{f}_2) \in \widehat{\mathcal{C}}^x(t)$  that minimizes  $\widehat{L}(f_1) + \widehat{L}(f_2)$  we have that with probability at least  $1 - 2\delta$  over the sample  $S$ ,

$$\begin{aligned} L(\widehat{f}_1) + L(\widehat{f}_2) - L(f_1^* \widehat{c} \circ t) - L(f_2^* \widehat{c} \circ t) \\ \leq 2G_\ell(\widehat{\mathcal{C}}^x(t), S, \delta) \end{aligned}$$

Now Let  $t' = 2c_d^2(\Phi(\sqrt{\epsilon_{\text{info}}}) + \Phi(\epsilon_{\text{baves}}))$  then we see that if  $(f_1, f_2) \in \mathcal{C}^x(t')$  then,

$$E[\chi(f_1, f_2)] \leq t'$$

However applying Assumption 4 (unlabeled) we find that with probability greater than  $1 - \delta$  over the unlabeled dataset  $U$  we have that

$$\widehat{E}\chi(f_1, f_2) \leq E[\chi(f_1, f_2)] + G_\chi(\mathcal{F}_1 \times \mathcal{F}_2, U, \delta)$$

Thus we can conclude that with probability greater than  $1 - \delta$  over the i.i.d. unlabeled sample we have that  $(f_1, f_2) \in \widehat{\mathcal{C}}^x(t)$ . Now using the above we see that with probability  $1 - \delta$  over unlabeled data

$$\min_{(f_1, f_2) \in \widehat{\mathcal{C}}^x(t)} L(f_1) + L(f_2) = \min_{(f_1, f_2) \in \mathcal{C}^x(t')} L(f_1) + L(f_2)$$

Hence using the result of Theorem 1 we can conclude that with probability  $1 - 3\delta$  over both labeled and unlabeled data we have that

$$\begin{aligned} L(\widehat{f}_1) + L(\widehat{f}_2) \leq 2L(y^*) + 2G_\ell(\widehat{\mathcal{C}}^x(t), S, \delta) \\ + 2\epsilon_{\text{baves}} + 2\sqrt{\epsilon_{\text{info}}} \end{aligned}$$

■

**Proof:**[Proof of Corollary 3] First note that we can write  $f_1 \in \mathcal{F}_1$  as  $(f_1, 0) \in \mathcal{F}_1 \times \mathcal{F}_2$  and similarly we can define any  $f_2 \in \mathcal{F}_2$  as  $(0, f_2) \in \mathcal{F}_1 \times \mathcal{F}_2$  so that we can consider only the joint RKHS defined by sum of  $f_1$  and  $f_2$ . From Example 4 we first of all have that for the regularized loss Assumption 2 is satisfied by the squared incompatibility (i.e..  $\chi(\widehat{y}_1, \widehat{y}_2) = (\widehat{y}_1 - \widehat{y}_2)^2$ ) function with  $\Phi(x) = \frac{(K+\lambda)^2}{2\lambda}x$ . Also note that in this case  $\epsilon_{\text{baves}} = 0$  since  $f^*$  is in the RKHS (in fact for the regularized loss to even be applicable the function needs to live in the RKHS). Hence if we restrict ourselves to the class  $\mathcal{C}^x(t)$  where  $t = \frac{8(\lambda+K)^2\sqrt{\epsilon_{\text{info}}}}{\lambda}$  then using Theorem 2, we see that we can get a low regularized regret with respect to  $f^*$ . Now without loss of generality assume that for the given loss  $\ell$  we have that  $\ell(0, y) = 1$ . Then using this in Equation 1 we see that,

$$\lambda_{co} \widehat{E}_U[f_1(x_1) - f_2(x_2)]^2 \leq 1$$

and so using  $\lambda_{co} = \frac{1}{t}$  we see that for any function pairs  $(f_1, f_2)$  returned by the algorithm  $\widehat{E}_U[\chi(f_1, f_2)] \leq t$ . However since we are in the transductive setting  $\widehat{E}_U[\chi(f_1, f_2)] = E[\chi(f_1, f_2)]$ . Now we use the result from Rosenberg and Bartlett [2007] to establish a statement

of the form Assumption 4 (labeled).

To this end define,

$$\begin{aligned} \mathcal{H}(t) = \{(f_1, f_2) : \lambda\|f_1\|^2 + \lambda\|f_2\|^2 \\ + \lambda_{co}\widehat{E}_U(f_1(x_1) - f_2(x_2))^2 \leq 1\} \end{aligned}$$

Notice that the solution of the co-regularization algorithm is contained in this class. Further as in Rosenberg and Bartlett [2007] define  $\mathcal{J}(t) = \{x \rightarrow \frac{f_1(x_1) + f_2(x_2)}{2} : (f_1, f_2) \in \mathcal{H}\}$ . Now we can directly use Theorem 2 of their paper (assuming  $\ell$  is bounded by 1) to get that with probability at least  $1 - \delta$  over labeled samples, for all  $(f_1, f_2) \in \widehat{\mathcal{C}}^x(t)$

$$\begin{aligned} L(f_1) + L(f_2) \leq \widehat{L}(f_1) + \widehat{L}(f_2) \\ + 2C_{Lip}\widehat{R}_n(\mathcal{J}(t)) + \frac{1}{\sqrt{n}}(2 + 3\sqrt{\frac{\ln(2/\delta)}{2}}) \end{aligned} \quad (9)$$

Where by Theorem 3 of Rosenberg and Bartlett [2007] we find that

$$\widehat{R}_n(\mathcal{J}(t)) \leq \frac{R}{n}$$

where

$$\begin{aligned} R^2 = \lambda^{-1}tr(K_{S \times S}^1) + \lambda^{-1}tr(K_{S \times S}^2) \\ - \frac{\lambda}{(\lambda + K)^2 t} tr(J^T(I + \lambda M)^{-1}J) \end{aligned}$$

and

$$J = \lambda^{-1}K_{U \times S}^1 - \lambda^{-1}K_{U \times S}^2 \quad M = \lambda^{-1}K_{U \times U}^1 - \lambda^{-1}K_{U \times U}^2$$

Now this establishes the Assumption 4, labeled statement we were aiming for.

Now putting the regularization term on both sides of the inequality in Equation 9 we get that

$$\begin{aligned} E[\ell_\lambda(f_1, x_1, y) + \ell_\lambda(f_2, x_2, y)] \leq \\ \widehat{E}[\ell_\lambda(f_1, x_1, y) + \ell_\lambda(f_2, x_2, y)] \\ + 4C_{Lip}\widehat{R}_n(\mathcal{J}(t)) + \frac{1}{\sqrt{n}}(2 + 3\sqrt{\frac{\ln(2/\delta)}{2}}) \end{aligned}$$

Now this is essentially the labeled statement in Assumption 4 and since we are in the transductive case we do not need the unlabeled part of the assumption. Hence using Theorem 2 we see that with probability at least  $1 - \delta$  over labeled samples for the pair  $\widehat{f}_1, \widehat{f}_2^A$  returned by co-regularization algorithm,

$$\begin{aligned} E\left[\frac{\ell(\widehat{f}_1 x_1, y) + \ell(\widehat{f}_2, x_2, y)}{2}\right] \leq E[\ell_\lambda(f^*, x_1, x_2, y)] \\ + 2C_{Lip}\widehat{R}_n(\mathcal{J}(t)) + \frac{1}{\sqrt{n}}(2 + 3\sqrt{\frac{\ln(2/\delta)}{2}}) + \sqrt{\epsilon_{\text{info}}} \end{aligned}$$

Now using Jensen's Inequality we see that the regularized loss of the average predictor is bounded by average of regularized loss of the predictors and hence the result. ■

**Proof:**[of Theorem 6] Without loss of generality we assume we are in the CCA basis. For each  $v \in \{1, 2\}$  let  $\beta^{(v)}$  be the

minimizer with respect to  $\beta$  of  $E[\ell(\beta^T x_v, y)]$ . From the result of Lemma 8 using the squared incompatibility function ( $c_d = 2$  in this case) we have that

$$\begin{aligned} 8\Phi(\sqrt{\epsilon_{\text{info}}}) + 8\Phi(\epsilon_{\text{bayes}}) &\geq E[(x_1^T \beta^{(1)} - x_2^T \beta^{(2)})^2] \\ &= \sum_i [(\beta_i^{(1)})^2 + (\beta_i^{(2)})^2 - 2\gamma_i \beta_i^{(1)} \beta_i^{(2)}] \\ &\geq \sum_i [(1 - \gamma_i)(\beta_i^{(1)})^2 + (1 - \gamma_i)(\beta_i^{(2)})^2] \end{aligned}$$

(the last step is due to the identity  $2ab \leq a^2 + b^2$ ). Hence we conclude that

$$\sum_i (1 - \gamma_i)(\beta_i^{(v)})^2 \leq 8\Phi(\sqrt{\epsilon_{\text{info}}}) + 8\Phi(\epsilon_{\text{bayes}}) \quad (10)$$

Let  $\beta_P^{(v)}$  be the projection of  $\beta^{(v)}$  on to the first  $i_{\text{thresh}}$  coordinates. Consider a twice differentiable loss function. By Taylor's theorem (second order) we have that there exists some  $\tilde{\beta}$  such that

$$\begin{aligned} \ell(x_v^T \beta_P^{(v)}, y) &= \ell(x_v^T \beta^{(v)}, y) + (\beta_P^{(v)} - \beta^{(v)})^T \nabla \ell(\beta^{(v)}) \\ &\quad + \frac{1}{2} (\beta_P^{(v)} - \beta^{(v)})^T \nabla^2 \ell(\tilde{\beta}^T x_v, y) (\beta_P^{(v)} - \beta^{(v)}) \end{aligned}$$

Taking expectation and noting that since  $\beta^{(v)}$  is the minimizer of the expected loss we find that

$$\begin{aligned} L(\beta_P^{(v)}) - L(\beta^{(v)}) &= \\ &= \frac{1}{2} (\beta^{(v)} - \beta_P^{(v)})^T E[\nabla^2 \ell(\tilde{\beta}^T x_v, y)] (\beta^{(v)} - \beta_P^{(v)}) \end{aligned}$$

Let  $\beta_{\text{res}}^{(v)} = \beta^{(v)} - \beta_P^{(v)}$ . Note that since  $(\beta_P^{(v)})_i = (\beta^{(v)})_i$  for all  $i$ 's corresponding to correlation values greater than the threshold we see that  $\beta_{\text{res}}^{(v)}$  is zero in the first  $i_{\text{thresh}}$  coordinates and is equal to  $\beta^{(v)}$  on the rest. Now note that for a loss function that is twice differentiable and a function of  $\tilde{\beta}^T x_v$  we have that by chain rule

$$\nabla^2 \ell(\beta \cdot x_v, y) = \frac{d^2 \ell(\tilde{\beta}^T x_v, y)}{d(\tilde{\beta}^T x_v)^2} x_v x_v^T$$

Now using the assumption that the second derivative of the loss function is bounded by some  $C$  we then see that

$$L(\beta_P^{(v)}) - L(\beta^{(v)}) \leq \frac{C}{2} (\beta_{\text{res}}^{(v)})^T E[x_v x_v^T] (\beta_{\text{res}}^{(v)})$$

Note that since we are in the CCA basis we have that  $E[(x_v)_i (x_v)_j] = 0$  when  $i \neq j$  and is 1 otherwise. Now note that for all  $i > i_{\text{thresh}}$  we have that  $1 - \gamma_i > 1 - \gamma_{\text{thresh}}$  and so,

$$\begin{aligned} L(\beta_P^{(v)}) - L(\beta^{(v)}) &\leq \frac{C}{2} \|\beta_{\text{res}}^{(v)}\|^2 \\ &= \frac{C}{2} \sum_{i > i_{\text{thresh}}} (\beta_i^{(v)})^2 \\ &\leq \frac{C}{2} \sum_{i > i_{\text{thresh}}} \frac{1 - \gamma_i}{1 - \gamma_{\text{thresh}}} (\beta_i^{(v)})^2 \\ &\leq \frac{C}{2(1 - \gamma_{\text{thresh}})} \sum_{i > i_{\text{thresh}}} (1 - \gamma_i) (\beta_i^{(v)})^2 \\ &\leq \frac{C}{2(1 - \gamma_{\text{thresh}})} \sum_i (1 - \gamma_i) (\beta_i^{(v)})^2 \end{aligned}$$

Hence using Equation 10 we can conclude that

$$L(\beta_P^{(v)}) - L(\beta^{(v)}) \leq \frac{4C (\Phi(\sqrt{\epsilon_{\text{info}}}) + \Phi(\epsilon_{\text{bayes}}))}{(1 - \gamma_{\text{thresh}})}$$

Now since  $L(\beta_P^{(v)}) \geq L(\beta_{\text{proj}}^{(v)})$  we conclude that

$$L(\beta_{\text{proj}}^{(v)}) - L(\beta^{(v)}) \leq \frac{4C (\Phi(\sqrt{\epsilon_{\text{info}}}) + \Phi(\epsilon_{\text{bayes}}))}{(1 - \gamma_{\text{thresh}})}$$

Finally since  $L(\beta^{(v)}) - L(y_v^*) \leq \epsilon_{\text{bayes}}$  we have the required result. ■

---

# Optimal Strategies and Minimax Lower Bounds for Online Convex Games

---

Jacob Abernethy\*

UC Berkeley

jake@cs.berkeley.edu

Alexander Rakhlin\*

UC Berkeley

rakhlin@cs.berkeley.edu

Peter L. Bartlett†

UC Berkeley

bartlett@cs.berkeley.edu

Ambuj Tewari

TTI Chicago

ambuj@cs.berkeley.edu

## Abstract

A number of learning problems can be cast as an Online Convex Game: on each round, a learner makes a prediction  $x$  from a convex set, the environment plays a loss function  $f$ , and the learner's long-term goal is to minimize regret. Algorithms have been proposed by Zinkevich, when  $f$  is assumed to be convex, and Hazan et al., when  $f$  is assumed to be strongly convex, that have provably low regret. We consider these two settings and analyze such games from a minimax perspective, proving minimax strategies and lower bounds in each case. These results prove that the existing algorithms are essentially optimal.

## 1 Introduction

The decision maker's greatest fear is *regret*: knowing, with the benefit of hindsight, that a better alternative existed. Yet, given only hindsight and not the gift of foresight, imperfect decisions can not be avoided. It is thus the decision maker's ultimate goal to suffer as little regret as possible.

In the present paper, we consider the notion of “regret minimization” for a particular class of decision problems. Assume we are given a set  $X$  and some set of functions  $\mathcal{F}$  on  $X$ . On each round  $t = 1, \dots, T$ , we must choose some  $\mathbf{x}_t$  from a set  $X$ . After we have made this choice, the *environment* chooses a function  $f_t \in \mathcal{F}$ . We incur a cost (loss)  $f_t(\mathbf{x}_t)$ , and the game proceeds to the next round. Of course, had we the fortune of perfect foresight and had access to the sum  $f_1 + \dots + f_T$ , we would know the optimal choice  $\mathbf{x}^* = \arg \min_{\mathbf{x}} \sum_{t=1}^T f_t(\mathbf{x})$ . Instead, at time  $t$ , we will have only seen  $f_1, \dots, f_{t-1}$ , and we must make the decision  $\mathbf{x}_t$  with only historical knowledge. Thus, a natural long-term goal is to minimize the *regret*, which here we define as

$$\sum_{t=1}^T f_t(\mathbf{x}_t) - \inf_{\mathbf{x} \in X} \sum_{t=1}^T f_t(\mathbf{x}).$$

A special case of this setting is when the decision space  $X$  is a convex set and  $\mathcal{F}$  is some set of convex functions on  $X$ . In

the literature, this framework has been referred to as Online Convex Optimization (OCO), since our goal is to minimize a global function, i.e.  $f_1 + f_2 + \dots + f_T$ , while this objective is revealed to us but one function at a time. Online Convex Optimization has attracted much interest in recent years [4, 9, 6, 1], as it provides a general analysis for a number of standard online learning problems including, among others, online classification and regression, prediction with expert advice, the portfolio selection problem, and online density estimation.

While instances of OCO have been studied over the past two decades, the general problem was first analyzed by Zinkevich [9], who showed that a very simple and natural algorithm, online gradient descent, elicits a bound on the regret that is on the order of  $\sqrt{T}$ . Online gradient descent can be described simply by the update  $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f_t(\mathbf{x}_t)$ , where  $\eta$  is some parameter of the algorithm. This regret bound only required that  $f_t$  be smooth, convex, and with bounded derivative.

A regret bound of order  $O(\sqrt{T})$  is not surprising: a number of online learning problems give rise to similar bounds. More recently, however, Hazan et al. [4] showed that when  $\mathcal{F}$  consists of *curved* functions, i.e.  $f_t$  is strongly convex, then we get a bound of the form  $O(\log T)$ . It is quite surprising that curvature gives such a great advantage to the player. Curved loss functions, such as square loss or logarithmic loss, are very natural in a number of settings.

Finding algorithms that can guarantee low regret is, however, only half of the story; indeed, it is natural to ask “can we obtain even lower regret?” or “do better algorithms exist?” The goal of the present paper is to address these questions, in some detail, for several classes of such online optimization problems. We answer both in the negative: the algorithms of Zinkevich and Hazan et al. are tight even up to their multiplicative constants.

This is achieved by a game-theoretic analysis: if we pose the above online optimization problem as a game between a Player who chooses  $\mathbf{x}_t$  and an Adversary who chooses  $f_t$ , we may consider the regret achieved when each player is playing optimally. This is typically referred to as the *value*  $V_T$  of the game. In general, computing the value of zero-sum games is difficult, as we may have to consider exponentially many, or even uncountably many, strategies of the Player and the Adversary. Ultimately we will show that this value, as well as the optimal strategies of both the player and the adversary,

---

\*Division of Computer Science

†Division of Computer Science, Department of Statistics

can be computed *exactly and efficiently* for certain classes of online optimization games.

The central results of this paper are as follows:

- When the adversary plays *linear* loss functions, we use a known randomized argument to lower bound the value  $V_T$ . We include this mainly for completeness.
- We show that indeed this same linear game can be solved *exactly* for the case when the input space  $X$  is a ball, and we provide the optimal strategies for the player and adversary.
- We perform a similar analysis for the *quadratic game*, that is where the adversary must play quadratic functions. We describe the adversary's strategy, and we prove that the well-known Follow the Leader strategy is optimal for the player.
- We show that the above results apply to a much wider class of games, where the adversary can play either convex or strongly convex functions, suggesting that indeed the linear and quadratic games are the "hard cases".

## 2 Online Convex Games

The general optimization game we consider is as follows. We have two agents, a player and an adversary, and the game proceeds for  $T$  rounds with  $T$  known in advance to both agents. The player's choices will come from some convex set  $X \subset \mathbb{R}^n$ , and the adversary will choose functions from the class  $\mathcal{F}$ . For the remainder of the paper,  $n$  denotes the dimension of the space  $X$ . To consider the game in full generality, we assume that the adversary's "allowed" functions may change on each round, and thus we imagine there is a sequence of allowed sets  $L_1, L_2, \dots, L_T \subset \mathcal{F}$ .

### Online Convex Game

$\mathcal{G}(X, \{L_t\})$ :

- 1: **for**  $t = 1$  to  $T$  **do**
- 2:   Player chooses (predicts)  $\mathbf{x}_t \in X$ .
- 3:   Adversary chooses a function  $f_t \in L_t$ .
- 4: **end for**
- 5: Player suffers regret

$$R_T = \sum_{t=1}^T f_t(\mathbf{x}_t) - \inf_{\mathbf{x} \in X} \sum_{t=1}^T f_t(\mathbf{x}).$$

From this general game, we obtain each of the examples above with appropriate choice of  $X$ ,  $\mathcal{F}$  and the sets  $\{L_t\}$ . We define a number of particular games in the definitions below.

It is useful to prove regret bounds within this model as they apply to any problem that can be cast as an Online Convex Game. The known general upper bounds are as follows:

- **Zinkevich [9]:** If  $L_1 = \dots = L_T = \mathcal{F}$  consist of continuous twice differentiable functions  $f$ , where  $\|\nabla f\| \leq G$  and  $\nabla^2 f \succeq \mathbf{0}$ , then<sup>1</sup>

$$R_T \leq \frac{1}{2} DG\sqrt{T}.$$

<sup>1</sup>This bound can be obtained by a slight modification of the analysis in [9].

where  $D := \max_{\mathbf{x}, \mathbf{y} \in X} \|\mathbf{x} - \mathbf{y}\|$  and  $G$  is some positive constant.

- **Hazan et al. [4]:** If  $L_1 = \dots = L_T = \mathcal{F}$  consist of continuous twice differentiable functions  $f$ , where  $\|\nabla f\| \leq G$  and  $\nabla^2 f \succeq \sigma I$ , then

$$R_T \leq \frac{1}{2} \frac{G^2}{\sigma} \log T,$$

where  $G$  and  $\sigma$  are positive constants.

- **Bartlett et al. [1]:** If  $L_t$  consists of continuous twice differentiable functions  $f$ , where  $\|\nabla f\| \leq G_t$  and  $\nabla^2 f \succeq \sigma_t I$ , then

$$R_T \leq \frac{1}{2} \sum_{t=1}^T \frac{G_t^2}{\sum_{s=1}^t \sigma_s},$$

where  $G_t$  and  $\sigma_t$  are positive constants. Moreover, the algorithm does not need to know  $G_t, \sigma_t$  on round  $t$ .

All three of these games posit an upper bound on  $\|\nabla f\|$  which is required to make the game nontrivial (and is natural in most circumstances). However, the first requires only that the second derivative be nonnegative, while the second and third game has a strict positive lower bound on the eigenvalues of the Hessian  $\nabla^2 f$ . Note that the bound of Bartlett et al recovers the logarithmic regret of Hazan et al whenever  $G_t$  and  $\sigma_t$  do not vary with time.

In the present paper, we analyze each of these games with the goal of obtaining the exact minimax value of the game, defined as:

$$V_T(\mathcal{G}(X, \{L_t\})) =$$

$$\inf_{\mathbf{x}_1 \in X} \sup_{f_1 \in L_1} \dots \inf_{\mathbf{x}_T \in X} \sup_{f_T \in L_T} \left( \sum_{t=1}^T f_t(\mathbf{x}_t) - \inf_{\mathbf{x} \in X} \sum_{t=1}^T f_t(\mathbf{x}) \right).$$

The quantity  $V_T(\mathcal{G})$  tells us the worst case regret of an *optimal* strategy in this game.

First, in the spirit of [1], we consider  $V_T$  for the games where constants  $G$  and  $\sigma$ , which respectively bound the first and second derivatives of  $f_t$ , can change throughout the game. That is, the Adversary is given two sequences before the game begins,  $\langle G_1, \dots, G_T \rangle$  and  $\langle \sigma_1, \dots, \sigma_T \rangle$ . We also require only that the gradient of  $f_t$  is bounded *at the point*  $\mathbf{x}_t$ , i.e.  $\|\nabla f_t(\mathbf{x}_t)\| \leq G_t$ , as opposed to the global constraint  $\|\nabla f_t(\mathbf{x})\| \leq G_t$  for all  $\mathbf{x} \in X$ . We may impose both of the above constraints by carefully choosing the sets  $L_t \subseteq \mathcal{F}$ , and we note that these sets will depend on the choices  $\mathbf{x}_t$  made by the Player.

We first define the Linear and Quadratic Games, which are the central objects of this paper.

**Definition 1** *The Linear Game  $\mathcal{G}_{lin}(X, \langle G_t \rangle)$  is the game  $\mathcal{G}(X, \{L_t\})$  where*

$$L_t = \{f : f(\mathbf{x}) = v^\top(\mathbf{x} - \mathbf{x}_t) + c, v \in \mathbb{R}^n, c \in \mathbb{R}; \|v\| \leq G_t\}.$$

**Definition 2** *The Quadratic Game  $\mathcal{G}_{quad}(X, \langle G_t \rangle, \langle \sigma_t \rangle)$  is the game  $\mathcal{G}(X, \{L_t\})$  where*

$$L_t = \{f : f(\mathbf{x}) = v^\top(\mathbf{x} - \mathbf{x}_t) + \frac{\sigma_t}{2} \|\mathbf{x} - \mathbf{x}_t\|^2 + c, v \in \mathbb{R}^n, c \in \mathbb{R}; \|v\| \leq G_t\}.$$

The functions in these definitions are parametrized through  $\mathbf{x}_t$  to simplify proofs of the last section. In Section 4, however, we will just consider the standard parametrization  $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ .

We also introduce more general games: the Convex Game and the Strongly Convex Game. While being defined with respect to a much richer class of loss functions, we show that these games are indeed no harder than the Linear and the Quadratic Games defined above.

**Definition 3** *The Convex Game  $\mathcal{G}_{conv}(X, \langle G_t \rangle)$  is the game  $\mathcal{G}(X, \{L_t\})$  where*

$$L_t = \{f : \|\nabla f(\mathbf{x}_t)\| \leq G_t, \nabla^2 f \succeq 0\}.$$

**Definition 4** *The Strongly Convex Game  $\mathcal{G}_{st-conv}(X, \langle G_t \rangle, \langle \sigma_t \rangle)$  is the game  $\mathcal{G}(X, \{L_t\})$  where*

$$L_t = \{f : \|\nabla f(\mathbf{x}_t)\| \leq G_t, \nabla^2 f - \sigma_t I \succeq 0\}.$$

We write  $\mathcal{G}(G)$  instead of  $\mathcal{G}(\langle G_t \rangle)$  when all values  $G_t = G$  for some fixed  $G$ . This holds similarly for  $\mathcal{G}(\sigma)$  instead of  $\mathcal{G}(\langle \sigma_t \rangle)$ . Furthermore, we suppose that  $\sigma_1 > 0$  throughout the paper.

### 3 Previous Work

Several lower bounds for various online settings are available in the literature. Here we review a number of such results relevant to the present paper and highlight our primary contributions.

The first result that we mention is the lower bound of Vovk in the online linear regression setting [8]. It is shown that there exists a randomized strategy of the Adversary such that the expected regret is at least  $[(n - \epsilon)G^2 \ln T - C_\epsilon]$  for any  $\epsilon > 0$  and  $C_\epsilon$  a constant. One crucial difference between this particular setting and ours is that the loss functions of the form  $(y_t - \mathbf{x}_t \cdot \mathbf{w}_t)^2$  used in linear regression are curved in only one direction and linear in all other, thus this setting does not quite fit into any of the games we analyze. The lower bound of Vovk scales roughly as  $n \log T$ , which is quite interesting given that  $n$  does not enter into the lower bound of the Strongly Convex Game we analyze.

The lower bound for the log-loss functions of Ordentlich and Cover [5] in the setting of Universal Portfolios is also logarithmic in  $T$  and linear in  $n$ . Log-loss functions are parametrized as  $f_t(\mathbf{x}) = -\log(\mathbf{w} \cdot \mathbf{x})$  for  $\mathbf{x}$  in the simplex, and these fit more generally within the class of “exp-concave” functions. Upper bounds on the class of log-loss functions were originally presented by Cover [3] whereas Hazan et al. [4] present an efficient method for competing against the more general exp-concave functions. The log-loss lower bound of [5] is quite elegant yet, contrary to the minimax results we present, the optimal play is not efficiently computable.

The work of Takimoto and Warmuth [7] is most closely related to our results for the Quadratic Game. The authors consider functions  $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$  corresponding to the log-likelihood of the datapoint  $\mathbf{y}$  for a unit-variance Gaussian with mean  $\mathbf{x}$ . The lower bound of  $\frac{1}{2}D^2(\ln T - \ln \ln T + O(\ln \ln T / \ln T))$  is obtained, where  $D$  is the bound on the norm of adversary’s choices  $\mathbf{y}$ . Furthermore, they exhibit the

minimax strategy which, in the end, corresponds to a biased maximum-likelihood solution. We emphasize that these results differ from ours in several ways. First, we enforce a constraint on the size of the gradient of  $f_t$  whereas [7] constrain the location of the point  $\mathbf{y}$  when  $f_t(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$ . With our slightly weaker constraint, we can achieve a regret bound of the order  $\log T$  instead of the  $\log T - \log \log T$  of Takimoto and Warmuth. Interestingly, the authors describe the “ $-\log \log T$ ” term of their lower bound as “surprising” because many known games “were shown to have  $O(\log T)$  upper bounds”. They conjecture that the apparent slack is due to the learner being unaware of the time horizon  $T$ . In the present paper, we resolve this issue by noting that our slightly weaker assumption erases the additional term; it is thus the limit on the adversary, and not knowledge of the horizon, that gives rise to the slack. Furthermore, the minimax strategy of Takimoto and Warmuth, a biased maximum likelihood estimate on each round, is also an artifact of their assumption on the boundedness of adversary’s choices. With our weaker assumption, the minimax strategy is *exactly* maximum likelihood (generally called “Follow The Leader”).

All previous work mentioned above deals with “curved” functions. We now discuss known lower bounds for the Linear Game. It is well-known that in the expert setting, it is impossible to do better than  $O(\sqrt{T})$ . The lower bound in Cesa-Bianchi and Lugosi [2], Theorem 3.7, proves an asymptotic bound: in the limit of  $T \rightarrow \infty$ , the value of the game behaves as  $\sqrt{(\ln N)T/2}$ , where  $N$  is the number of experts. We provide a similar randomized argument, which has been sketched in the literature (e.g. Hazan et al [4]), but our additional minimax analysis indeed gives the tightest bound possible for any  $T$ .

Finally, we provide reductions between Quadratic and Strongly Convex as well as Linear and Convex Games. While apparent that the Adversary does better by playing linear approximations instead of convex functions, it requires a careful analysis to show that this holds for the minimax setting.

### 4 The Linear Game

In this section we begin by providing a relatively standard proof of the  $O(\sqrt{T})$  lower bound on regret when competing against linear loss functions. The more interesting result is our *minimax* analysis which is given in Section 4.2.

#### 4.1 The Randomized Lower Bound

Lower bounds for games with linear loss functions have appeared in the literature though often not in detail. The rough idea is to imagine a randomized Adversary and to compute the Player’s expected regret. This generally produces an  $O(\sqrt{T})$  lower bound yet it is not fully satisfying since the analysis is not tight. In the following section we provide a much improved analysis with minimax strategies for both the Player and Adversary.

**Theorem 5** *Suppose  $X = [-D/(2\sqrt{n}), D/(2\sqrt{n})]^n$ , so that the diameter of  $X$  is  $D$ . Then*

$$V_T(\mathcal{G}_{lin}(X, \langle G_t \rangle)) \geq \frac{D}{2\sqrt{2}} \sqrt{\sum_{t=1}^T G_t^2}$$

**Proof:** Define the scaled cube

$$\mathcal{C}_t = \{-G_t/\sqrt{n}, G_t/\sqrt{n}\}^n.$$

Suppose the Adversary chooses functions from

$$\hat{L}_t = \{f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} : \mathbf{w} \in \mathcal{C}_t\}.$$

Note that  $\|\nabla f\| = \|\mathbf{w}_t\| = G_t$  for any  $f \in \hat{L}_t$ .

Since we are restricting the Adversary to play linear functions with restricted  $\mathbf{w}$ ,

$$\begin{aligned} V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle)) &\geq V_T(\mathcal{G}(X, \hat{L}_1, \dots, \hat{L}_T)) \\ &= \inf_{\mathbf{x}_1 \in X} \sup_{f_1 \in \hat{L}_1} \dots \inf_{\mathbf{x}_T \in X} \sup_{f_T \in \hat{L}_T} \left[ \sum_{t=1}^T f_t(\mathbf{x}_t) - \inf_{\mathbf{x} \in X} \sum_{t=1}^T f_t(\mathbf{x}) \right] \\ &= \inf_{\mathbf{x}_1 \in X} \sup_{\mathbf{w}_1 \in \mathcal{C}_1} \dots \inf_{\mathbf{x}_T \in X} \sup_{\mathbf{w}_T \in \mathcal{C}_T} \left[ \sum_{t=1}^T \mathbf{w}_t \cdot \mathbf{x}_t - \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^T \mathbf{w}_t \right] \\ &\geq \inf_{\mathbf{x}_1 \in X} \mathbb{E}_{\mathbf{w}_1} \dots \inf_{\mathbf{x}_T \in X} \mathbb{E}_{\mathbf{w}_T} \left[ \sum_{t=1}^T \mathbf{w}_t \cdot \mathbf{x}_t - \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^T \mathbf{w}_t \right], \end{aligned}$$

where  $\mathbb{E}_{\mathbf{w}_t}$  denotes expectation with respect to any distribution over the set  $\mathcal{C}_t$ . In particular, it holds for the uniform distribution, i.e. when the coordinates of  $\mathbf{w}_t$  are  $\pm G_t/\sqrt{n}$  with probability 1/2. Since in this case  $\mathbb{E}_{\mathbf{w}_T} \mathbf{w}_T \cdot \mathbf{x}_T = 0$  for any  $\mathbf{x}_T$ , we obtain

$$\begin{aligned} V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle)) &\geq \inf_{\mathbf{x}_1 \in X} \mathbb{E}_{\mathbf{w}_1} \dots \inf_{\mathbf{x}_{T-1} \in X} \mathbb{E}_{\mathbf{w}_{T-1}} \inf_{\mathbf{x}_T \in X} \\ &\quad \mathbb{E}_{\mathbf{w}_T} \left[ \sum_{t=1}^T \mathbf{w}_t \cdot \mathbf{x}_t - \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^T \mathbf{w}_t \right] \\ &= \inf_{\mathbf{x}_1 \in X} \mathbb{E}_{\mathbf{w}_1} \dots \inf_{\mathbf{x}_{T-1} \in X} \mathbb{E}_{\mathbf{w}_{T-1}} \inf_{\mathbf{x}_T \in X} \\ &\quad \left[ \sum_{t=1}^{T-1} \mathbf{w}_t \cdot \mathbf{x}_t - \mathbb{E}_{\mathbf{w}_T} \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^T \mathbf{w}_t \right] \\ &= \inf_{\mathbf{x}_1 \in X} \mathbb{E}_{\mathbf{w}_1} \dots \inf_{\mathbf{x}_{T-1} \in X} \\ &\quad \mathbb{E}_{\mathbf{w}_{T-1}} \left[ \sum_{t=1}^{T-1} \mathbf{w}_t \cdot \mathbf{x}_t - \mathbb{E}_{\mathbf{w}_T} \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^T \mathbf{w}_t \right], \end{aligned}$$

where the last equality holds because the expression no longer depends on  $\mathbf{x}_T$ . Repeating the process, we obtain

$$\begin{aligned} V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle)) &\geq -\mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_T} \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^T \mathbf{w}_t \\ &= -\mathbb{E}_{\{\epsilon_{i,t}\}} \min_{\mathbf{x} \in \left\{ -\frac{D}{2\sqrt{n}}, \frac{D}{2\sqrt{n}} \right\}^n} \left( \mathbf{x} \cdot \sum_{t=1}^T \mathbf{w}_t \right), \end{aligned}$$

where  $\mathbf{w}_t(i) = \epsilon_{i,t} G_t/\sqrt{n}$ , with i.i.d. Rademacher variables  $\epsilon_{i,t} = \pm 1$  with probability 1/2. The last equality is due to the fact that a linear function is minimized at the vertices of the cube. In fact, the dot product is minimized by matching the sign of  $\mathbf{x}(i)$  with that of the  $i$ th coordinate of  $\sum_{t=1}^T \mathbf{w}_t$ .

Hence,

$$\begin{aligned} V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle)) &\geq -\mathbb{E}_{\{\epsilon_{i,t}\}} \sum_{i=1}^n -\frac{D}{2\sqrt{n}} \left| \sum_{t=1}^T \epsilon_{i,t} \frac{G_t}{\sqrt{n}} \right| \\ &= \frac{D}{2} \mathbb{E}_{\{\epsilon_{i,t}\}} \left| \sum_{t=1}^T \epsilon_{i,t} G_t \right| \geq \frac{D}{2\sqrt{2}} \sqrt{\sum_{t=1}^T G_t^2}, \end{aligned}$$

where the last inequality follows from the Khinchine's inequality [2].  $\blacksquare$

## 4.2 The Minimax Analysis

While in the previous section we found a particular lower bound on  $V_T(\mathcal{G}_{\text{lin}})$ , here we present a complete minimax analysis for the case when  $X$  is a ball in  $\mathbb{R}^n$  (of dimension  $n$  at least 3). We are indeed able to compute exactly the value

$$V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle))$$

and we provide the simple minimax strategies for both the Player and the Adversary. The unit ball, while a special case, is a very natural choice for  $X$  as it is the *largest* convex set of diameter 2.

For the remainder of this section, let  $f_t(\mathbf{x}) := \mathbf{w}_t \cdot \mathbf{x}$  where  $\mathbf{w}_t \in \mathbb{R}^n$  with  $\|\mathbf{w}_t\| \leq G_t$ . Also, we define  $\mathbf{W}_t = \sum_{s=1}^t \mathbf{w}_s$ , the cumulative functions chosen by the Adversary.

**Theorem 6** *Let  $X = \{\mathbf{x} : \|\mathbf{x}\|_2 \leq D/2\}$  and suppose the Adversary chooses functions from*

$$L_t = \{f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} : \|\mathbf{w}\|_2 \leq G_t\}.$$

*Then the value of the game*

$$V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle)) = \frac{D}{2} \sqrt{\sum_{t=1}^T G_t^2}.$$

*Furthermore, the optimal strategy for the player is to choose*

$$\mathbf{x}_{t+1} = \left( \frac{-D}{2\sqrt{\|\mathbf{W}_t\|^2 + \sum_{s=t+1}^T G_s}} \right) \mathbf{W}_t.$$

To prove the theorem, we will need a series of short lemmas.

**Lemma 7** *When  $X$  is the unit ball  $B = \{\mathbf{x} : \|\mathbf{x}\| \leq 1\}$ , the value  $V_T$  can be written as*

$$\inf_{\mathbf{x}_1 \in B} \sup_{\mathbf{w}_1 \in L_1} \dots \inf_{\mathbf{x}_T \in B} \sup_{\mathbf{w}_T \in L_T} \left[ \sum_{t=1}^T \mathbf{w}_t \cdot \mathbf{x}_t + \|\mathbf{W}_T\| \right] \quad (1)$$

*In addition, if we choose a larger radius  $D$ , the value of the game will scale linearly with this radius and thus it is enough to assume  $X = B$ .*

**Proof:** The last term in the regret

$$\inf_{\mathbf{x} \in B} \sum_t f_t(\mathbf{x}) = \inf_{\mathbf{x} \in B} \mathbf{W}_T \cdot \mathbf{x} = -\|\mathbf{W}_T\|$$

since the infimum is obtained when  $\mathbf{x} = \frac{\mathbf{W}_T}{\|\mathbf{W}_T\|}$ . This implies equation (1). The fact that the bound scales linearly with  $D/2$  follows from the fact that both the norm  $\|\mathbf{W}_T\|$  will scale with  $D/2$  as well as the terms  $\mathbf{w}_t \cdot \mathbf{x}_t$ . ■

For the remainder of this section, we simply assume that  $X = B$ , the unit ball with diameter  $D = 2$ .

**Lemma 8** *Regardless of the Player's choices, the Adversary can always obtain regret at least*

$$\sqrt{\sum_{t=1}^T G_t^2} \quad (2)$$

whenever the dimension  $n$  is at least 3.

**Proof:** Consider the following adversarial strategy and assume  $X = B$ . On round  $t$ , after the Player has chosen  $\mathbf{x}_t$ , the adversary chooses  $\mathbf{w}_t$  such that  $\|\mathbf{w}_t\| = G_t$ ,  $\mathbf{w}_t \cdot \mathbf{x}_t = 0$  and  $\mathbf{w}_t \cdot \mathbf{W}_{t-1} = 0$ . Finding a vector of length  $G_t$  that is perpendicular to two arbitrary vectors can always be done when the dimension is at least 3. With this strategy, it is guaranteed that  $\sum_t \mathbf{w}_t \cdot \mathbf{x}_t = 0$  and we claim also that

$$\|\mathbf{W}_T\| = \sqrt{\sum_{t=1}^T G_t^2}.$$

This follows from a simple induction. Assuming  $\|\mathbf{W}_{t-1}\| = \sqrt{\sum_{s=1}^{t-1} G_s^2}$ , then

$$\|\mathbf{W}_t\| = \|\mathbf{W}_{t-1} + \mathbf{w}_t\| = \sqrt{\|\mathbf{W}_{t-1}\|^2 + \|\mathbf{w}_t\|^2},$$

implying the desired conclusion. ■

The result of the last lemma is quite surprising: the adversary need only play some vector with length  $G_t$  which is perpendicular to both  $\mathbf{x}_t$  and  $\mathbf{W}_{t-1}$ . Indeed, this lower bound has a very different flavor from the randomized argument of the previous section. To obtain a full minimax result, all that remains is to show that the Adversary can *do no better!*

**Lemma 9** *Let  $\mathbf{w}_0 = \mathbf{0}$ . If the player always plays the point*

$$\mathbf{x}_t = \frac{-\mathbf{W}_{t-1}}{\sqrt{\|\mathbf{W}_{t-1}\|^2 + \sum_{s=t}^T G_s^2}} \quad (3)$$

then

$$\sup_{\mathbf{w}_1} \sup_{\mathbf{w}_2} \dots \sup_{\mathbf{w}_T} \left[ \sum_{t=1}^T \mathbf{w}_t \cdot \mathbf{x}_t + \|\mathbf{W}_T\| \right] \leq \sqrt{\sum_{t=1}^T G_t^2}$$

i.e., the regret can be no greater than the value in (2).

**Proof:** As before,  $\mathbf{W}_t = \sum_{s=1}^t \mathbf{w}_s$ . Define  $\Gamma_t^2 = \sum_{s=t}^T G_s^2$ , the forward sum, with  $\Gamma_{T+1} = 0$ . Define

$$\Phi_t(\mathbf{w}_1, \dots, \mathbf{w}_{t-1}) = \sum_{s=1}^{t-1} \mathbf{x}_s \cdot \mathbf{w}_s + \sqrt{\|\mathbf{W}_{t-1}\|^2 + \Gamma_t^2}$$

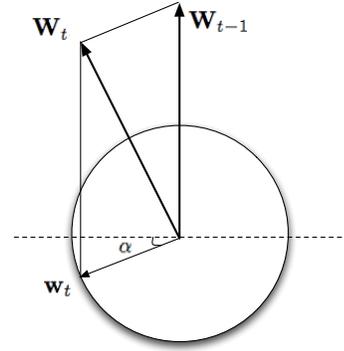


Figure 1: Illustration for the proof of the minimax strategy for the ball. We suppose that  $\mathbf{x}_t$  is aligned with  $\mathbf{W}_{t-1}$  and depict the plane spanned by  $\mathbf{W}_{t-1}$  and  $\mathbf{w}_t$ . We assume that  $\mathbf{w}_t$  has angle  $\alpha$  with the line perpendicular to  $\mathbf{W}_{t-1}$  and show that  $\alpha = 0$  is optimal.

where  $\mathbf{x}_t$  is as defined in (3) and  $\Phi_1$  is  $\sqrt{\sum_{t=1}^T G_t^2}$ . Let

$$V_t(\mathbf{w}_1, \dots, \mathbf{w}_{t-1}) = \sup_{\mathbf{w}_t} \dots \sup_{\mathbf{w}_T} \left[ \sum_{t=1}^T \mathbf{w}_t \cdot \mathbf{x}_t + \|\mathbf{W}_T\| \right]$$

be the optimum payoff to the adversary given that he plays  $\mathbf{w}_1, \dots, \mathbf{w}_{t-1}$  in the beginning and then plays optimally. The player plays according to (3) throughout. Note that the value of the game is  $V_1$ .

We prove by backward induction that, for all  $t \in \{1, \dots, T\}$ ,

$$V_t(\mathbf{w}_1, \dots, \mathbf{w}_{t-1}) \leq \Phi_t(\mathbf{w}_1, \dots, \mathbf{w}_{t-1})$$

The base case,  $t = T + 1$  is obvious. Now assume it holds for  $t + 1$  and we will prove it for  $t$ . We have

$$\begin{aligned} & V_t(\mathbf{w}_1, \dots, \mathbf{w}_{t-1}) \\ &= \sup_{\mathbf{w}_t} V_{t+1}(\mathbf{w}_1, \dots, \mathbf{w}_t) \\ (\text{induc.}) &\leq \sup_{\mathbf{w}_t} \Phi_{t+1}(\mathbf{w}_1, \dots, \mathbf{w}_t) \\ &= \sum_{s=1}^{t-1} \mathbf{x}_s \cdot \mathbf{w}_s + \\ (*) &\sup_{\mathbf{w}_t} \left[ \mathbf{x}_t \cdot \mathbf{w}_t + \sqrt{\|\mathbf{W}_{t-1} + \mathbf{w}_t\|^2 + \Gamma_{t+1}^2} \right] \end{aligned}$$

Let us consider the final supremum term above. If we can show that it is no more than

$$\sqrt{\|\mathbf{W}_{t-1}\|^2 + \Gamma_t^2} \quad (4)$$

then we will have proved  $V_t \leq \Phi_t$  thus completing the induction. This is the objective of the remainder of this proof.

We begin by noting two important facts about the expression (\*). First, the supremum is taken over a convex function of  $\mathbf{w}_t$  and thus the maximum occurs at the boundary, i.e. where  $\|\mathbf{w}_t\| = G_t$  exactly. This is easily checked by computing the Hessian with respect to  $\mathbf{w}_t$ . Second, since  $\mathbf{x}_t$  is chosen parallel to  $\mathbf{W}_{t-1}$ , the only two vectors of interest are

$\mathbf{w}_t$  and  $\mathbf{W}_{t-1}$ . Without loss of generality, we can assume that  $\mathbf{W}_{t-1}$  is the 2-dim vector  $\langle F, 0 \rangle$ , where  $F = \|\mathbf{W}_{t-1}\|$ , and that  $\mathbf{w}_t = \langle -G_t \sin \alpha, G_t \cos \alpha \rangle$  for any  $\alpha$ . Plugging in the choice of  $\mathbf{x}_t$  in (3), we may now rewrite (\*) as

$$\sup_{\alpha} \underbrace{\frac{FG_t \sin \alpha}{\sqrt{F^2 + G_t^2 + \Gamma_{t+1}^2}} + \sqrt{F^2 + G_t^2 + \Gamma_{t+1}^2} - 2FG_t \sin \alpha}_{\phi(\alpha)}$$

We illustrate this problem in Figure 1. Bounding the above expression requires some care, and thus we prove it in Lemma 16 found in the appendix. The result of Lemma 16 gives us that, indeed,

$$\phi(\alpha) \leq \sqrt{F^2 + G_t^2 + \Gamma_{t+1}^2} = \sqrt{\|\mathbf{W}_{t-1}\|^2 + \Gamma_t^2}.$$

Since (\*) is exactly  $\sup_{\alpha} \phi(\alpha)$ , which is no greater than

$$\sqrt{F^2 + \Gamma_t^2},$$

we are done.  $\blacksquare$

We observe that the minimax strategy for the ball is exactly the Online Gradient Descent strategy<sup>2</sup> of Zinkevich [9]. The value of the game for the ball is exactly the upper bound for the proof of Online Gradient Descent if the initial point is the center of the ball. The lower bound of the randomized argument in the previous section differs from the upper bound for Online Gradient Descent by  $\sqrt{2}$ .

## 5 The Quadratic Game

As in the last section, we now give a minimax analysis of the game  $\mathcal{G}_{\text{quad}}$ . Ultimately we will be able to compute the exact value of  $V_T(\mathcal{G}_{\text{quad}}(X, \langle G_t \rangle, \langle \sigma_t \rangle))$  and provide the optimal strategy of both the Player and the Adversary. What is perhaps most interesting is that the optimal Player strategy is the well-known Follow The Leader approach. This general strategy can be defined simply as

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{x} \in X} \sum_{s=1}^t f_s(\mathbf{x});$$

that is, we choose the best  $\mathbf{x}$  “in hindsight”. As has been pointed out by several authors, this strategy can incur  $\Omega(T)$  regret when the loss functions are linear. It is thus quite surprising that this strategy is optimal when instead we are competing against quadratic loss functions.

For this section, define  $F_t(\mathbf{x}) := \sum_{s=1}^t f_s(\mathbf{x})$  and  $\mathbf{x}_t^* := \arg \min_{\mathbf{x}} F_t(\mathbf{x})$ . Define  $\sigma_{1:t} = \sum_{s=1}^t \sigma_s$ . We assume from the outset that  $\sigma_1 > 0$ . We also set  $\sigma_{1:0} = 0$ .

### 5.1 A Necessary Restriction

Recall that the upper bound in Hazan et al. [4] is

$$R_T \leq \frac{1}{2} \frac{G^2}{\sigma} \log T$$

<sup>2</sup>This does require some work to show, and more information will be available in the full version of this paper.

and note that this expression has no dependence on the size of  $X$ . We would thus ideally like to consider the case when  $X = \mathbb{R}^n$ , for this would seem to be the “hardest” case for the Player. The unbounded assumption is problematic, however, not because the game is too difficult for the Player, but the game is *too difficult for the Adversary!*. This ought to come as quite a surprise, but arises from the particular restrictions we place on the Adversary.

**Proposition 5.1** *For  $G, \sigma > 0$ , if  $\max_{\mathbf{x}, \mathbf{y} \in X} \|\mathbf{x} - \mathbf{y}\| = D > 4G/\sigma$ , there is an  $\alpha > 0$  such that  $V_T(\mathcal{G}_{\text{quad}}(X, G, \sigma)) \leq -\alpha T$ .*

**Proof:** Fix  $\mathbf{x}_o, \mathbf{x}_e \in X$  with  $\|\mathbf{x}_o - \mathbf{x}_e\| > 4G/\sigma$ . Consider a player that plays  $\mathbf{x}_{2k-1} = \mathbf{x}_o, \mathbf{x}_{2k} = \mathbf{x}_e$ . Then for any  $\mathbf{x} \in X$ ,

$$f_{2k-1}(\mathbf{x}) \geq f_{2k-1}(\mathbf{x}_o) - G\|\mathbf{x} - \mathbf{x}_o\| + \frac{\sigma}{2}\|\mathbf{x} - \mathbf{x}_o\|^2,$$

And similarly for  $f_{2k}$  and  $\mathbf{x}_e$ . Summing over  $t$  (assuming that  $T$  is even) shows that  $V_t(\mathcal{G}_{\text{quad}}(X, G, \sigma))$  is no more than

$$\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) \leq \frac{T}{2} \left( G\|\mathbf{x} - \mathbf{x}_o\| - \frac{\sigma}{2}\|\mathbf{x} - \mathbf{x}_o\|^2 + G\|\mathbf{x} - \mathbf{x}_e\| - \frac{\sigma}{2}\|\mathbf{x} - \mathbf{x}_e\|^2 \right).$$

But by the triangle inequality, any  $\mathbf{x} \in X$  has  $\|\mathbf{x} - \mathbf{x}_o\| + \|\mathbf{x} - \mathbf{x}_e\| \geq D$ . Subject to this constraint, plus the constraints  $0 \leq \|\mathbf{x} - \mathbf{x}_o\| \leq D, 0 \leq \|\mathbf{x} - \mathbf{x}_e\| \leq D$  shows that  $V_t(\mathcal{G}_{\text{quad}}(X, G, \sigma)) \leq T(GD - \sigma D^2/4)/2 \leq -\alpha T$  for some  $\alpha > 0$ , since  $D > 4G/\sigma$ .  $\blacksquare$

As we don’t generally expect regret to be negative, this example suggests that the Quadratic Game is uninteresting without further constraints on the Player. While an explicit bound on the size of  $X$  is a possibility, it is easier for the analysis to place a slightly weaker restriction on the Player.

**Assumption 5.1** *Let  $\mathbf{x}_{t-1}^*$  be the minimizer of  $F_{t-1}(\mathbf{x})$ . We assume that the Player must choose  $\mathbf{x}_t$  such that*

$$\sigma_t \|\mathbf{x}_t - \mathbf{x}_{t-1}^*\| < 2G_t.$$

This restriction is necessary for non-negative regret. Indeed, it can be shown that if we increase the size of the above ball by only  $\epsilon$ , the method of Proposition 5.1 above shows that the regret will be negative for large enough  $T$ .

### 5.2 Minimax Analysis

With the above restriction in place, we now simply write the game as  $\mathcal{G}'_{\text{quad}}(\langle G_t \rangle, \langle \sigma_t \rangle)$ , omitting the input  $X$ . We now proceed to compute the value of this game exactly.

**Theorem 10** *Under Assumption 5.1, the value of the game*

$$V_T(\mathcal{G}'_{\text{quad}}(\langle G_t \rangle, \langle \sigma_t \rangle)) = \sum_{t=1}^T \frac{G_t^2}{2\sigma_{1:t}}.$$

With uniform  $G_t$  and  $\sigma_t$ , we obtain the harmonic series, giving us our logarithmic regret bound. We note that this is *exactly* the upper bound proven in [1, 4], even with the constant.

**Corollary 11** For the uniform parameters of the game,

$$\frac{G}{2\sigma} \log(T+1) \leq V_T(\mathcal{G}'_{quad}(G, \sigma)) \leq \frac{G}{2\sigma}(1 + \log T).$$

The main argument in the proof of Theorem 10 boils down to reducing the multiple round game to a single round game. The following lemma gives the value of this single round game. Since the proof is somewhat technical, we postpone it to the Appendix.

**Lemma 12** For arbitrary  $G_t, \sigma_t, \sigma_{1:t-1} > 0$ ,

$$\begin{aligned} \inf_{\Delta: \|\Delta\| \leq \frac{2G_t}{\sigma_t}} \sup_{\delta} \left( G_t \|\Delta - \delta\| - \frac{1}{2} \sigma_t \|\Delta - \delta\|^2 - \frac{1}{2} \sigma_{1:t-1} \|\delta\|^2 \right) \\ = \frac{G_t^2}{2\sigma_{1:t}} = \frac{G_t^2}{2\sigma_{1:t}}, \end{aligned}$$

and indeed the optimal strategy pair is  $\Delta = \mathbf{0}$  and  $\delta$  any vector for which  $\|\delta\| = \frac{G_t}{\sigma_{1:t}}$ .

We now show how to “unwind” the recursive inf sup definition of  $V_T(\mathcal{G}'_{quad}(\langle G_t \rangle, \langle \sigma_t \rangle))$ , where the final term we chop off is the object we described in the above lemma.

**Proof:** [Proof of Theorem 10] Let  $\mathbf{x}_{t-1}^*$  be the minimizer of  $F_{t-1}(\mathbf{x})$  and  $\mathbf{z} \in X$  be arbitrary. Note that  $F_t$  is  $\sigma_{1:t}$ -quadratic, so

$$\begin{aligned} F_t(\mathbf{z}) &= F_{t-1}(\mathbf{z}) + f_t(\mathbf{z}) \\ &= F_{t-1}(\mathbf{x}_{t-1}^* + (\mathbf{z} - \mathbf{x}_{t-1}^*)) + f_t(\mathbf{z}) \\ &= F_{t-1}(\mathbf{x}_{t-1}^*) + \nabla F_{t-1}(\mathbf{x}_{t-1}^*)(\mathbf{z} - \mathbf{x}_{t-1}^*) \\ &\quad + \frac{1}{2} \sigma_{1:t-1} \|\mathbf{z} - \mathbf{x}_{t-1}^*\|^2 + f_t(\mathbf{z}) \\ &= F_{t-1}(\mathbf{x}_{t-1}^*) + \frac{1}{2} \sigma_{1:t-1} \|\mathbf{z} - \mathbf{x}_{t-1}^*\|^2 + f_t(\mathbf{z}), \end{aligned}$$

where the last equality holds by the definition of  $\mathbf{x}_{t-1}^*$ . Hence,

$$\begin{aligned} \sum_{s=1}^t f_s(\mathbf{x}_s) - F_t(\mathbf{z}) &= \left( \sum_{s=1}^{t-1} f_s(\mathbf{x}_s) - F_{t-1}(\mathbf{x}_{t-1}^*) \right) \\ &\quad + \left( f_t(\mathbf{x}_t) - f_t(\mathbf{z}) - \frac{1}{2} \sigma_{1:t-1} \|\mathbf{z} - \mathbf{x}_{t-1}^*\|^2 \right). \end{aligned}$$

Expanding  $f_t$  around  $\mathbf{x}_t$ ,

$$f_t(\mathbf{x}_t) - f_t(\mathbf{z}) = -\nabla f_t(\mathbf{x}_t)(\mathbf{z} - \mathbf{x}_t) - \frac{1}{2} \sigma_t \|\mathbf{z} - \mathbf{x}_t\|^2.$$

Substituting,

$$\begin{aligned} \sum_{s=1}^t f_s(\mathbf{x}_s) - F_t(\mathbf{z}) &= \left( \sum_{s=1}^{t-1} f_s(\mathbf{x}_s) - F_{t-1}(\mathbf{x}_{t-1}^*) \right) \\ &\quad + \left( \nabla f_t(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{z}) - \frac{1}{2} \sigma_t \|\mathbf{z} - \mathbf{x}_t\|^2 - \frac{1}{2} \sigma_{1:t-1} \|\mathbf{z} - \mathbf{x}_{t-1}^*\|^2 \right). \end{aligned}$$

Then

$$\begin{aligned} V_t &:= \inf_{\mathbf{x}_1} \sup_{f_1} \dots \inf_{\mathbf{x}_t} \sup_{f_t} \left( \sum_{s=1}^t f_s(\mathbf{x}_s) - \inf_{\mathbf{z}} F_t(\mathbf{z}) \right) \\ &= \inf_{\mathbf{x}_1} \sup_{f_1} \dots \inf_{\mathbf{x}_t} \sup_{f_t, \mathbf{z}} \left( \sum_{s=1}^t f_s(\mathbf{x}_s) - F_t(\mathbf{z}) \right) \\ &= \inf_{\mathbf{x}_1} \sup_{f_1} \dots \inf_{\mathbf{x}_{t-1}} \sup_{f_{t-1}} \left[ \left( \sum_{s=1}^{t-1} f_s(\mathbf{x}_s) - F_{t-1}(\mathbf{x}_{t-1}^*) \right) \right. \\ &\quad \left. + \inf_{\mathbf{x}_t} \sup_{f_t, \mathbf{z}} \left( \nabla f_t(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{z}) - \frac{1}{2} \sigma_t \|\mathbf{z} - \mathbf{x}_t\|^2 \right. \right. \\ &\quad \left. \left. - \frac{1}{2} \sigma_{1:t-1} \|\mathbf{z} - \mathbf{x}_{t-1}^*\|^2 \right) \right]. \end{aligned}$$

However, we can simplify the final inf sup as follows. We note that the quantity  $\nabla f_t(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{z})$  is maximized when  $\nabla f_t(\mathbf{x}_t) = G_t \frac{\mathbf{x}_t - \mathbf{z}}{\|\mathbf{x}_t - \mathbf{z}\|}$ . Second, we can instead use the variables  $\Delta = \mathbf{x}_t - \mathbf{x}_{t-1}^*$  and  $\delta = \mathbf{z} - \mathbf{x}_{t-1}^*$  in the optimization. Recall from Assumption 5.1 that  $\|\mathbf{x}_t - \mathbf{x}_{t-1}^*\| = \|\Delta\| \leq \frac{2G_t}{\sigma_t}$ . Then,

$$\begin{aligned} V_t &= \inf_{\mathbf{x}_1} \sup_{f_1} \dots \inf_{\mathbf{x}_{t-1}} \sup_{f_{t-1}} \left[ \left( \sum_{s=1}^{t-1} f_s(\mathbf{x}_s) - F_{t-1}(\mathbf{x}_{t-1}^*) \right) \right. \\ &\quad \left. + \inf_{\Delta: \|\Delta\| \leq \frac{2G_t}{\sigma_t}} \sup_{\delta} \left( G_t \|\Delta - \delta\| \right. \right. \\ &\quad \left. \left. - \frac{1}{2} \sigma_t \|\Delta - \delta\|^2 - \frac{1}{2} \sigma_{1:t-1} \|\delta\|^2 \right) \right] \\ &= \inf_{\mathbf{x}_1} \sup_{f_1} \dots \inf_{\mathbf{x}_{t-1}} \\ &\quad \sup_{f_{t-1}} \left[ \left( \sum_{s=1}^{t-1} f_s(\mathbf{x}_s) - F_{t-1}(\mathbf{x}_{t-1}^*) \right) + \frac{G_t^2}{2\sigma_{1:t}} \right] \\ &= V_{t-1} + \frac{G_t^2}{2\sigma_{1:t}}, \end{aligned}$$

where the second equality is obtained by applying Lemma 12. Unwinding the recursion proves the theorem. ■

**Corollary 13** The optimal Player strategy is to set  $\mathbf{x}_t = \mathbf{x}_{t-1}^*$  on each round.

**Proof:** In analyzing the game, we found that the optimal choice of  $\Delta = \mathbf{x}_t - \mathbf{x}_{t-1}^*$  was shown to be  $\mathbf{0}$  in Lemma 12. ■

## 6 General Games

While the minimax results shown above are certainly interesting, we have only shown them to hold for the rather restricted games  $\mathcal{G}_{\text{lin}}$  and  $\mathcal{G}_{\text{quad}}$ . For these particular cases, the class of functions that the Adversary may choose from is quite small: both the set of linear functions and the set quadratic functions can be parameterized by  $O(n)$  variables. It would of course be more satisfying if our minimax analysis held for more richer loss function spaces.

Indeed, we prove in this section that both of our minimax results hold much more generally. In particular, we prove that even if the Adversary were able to choose *any* convex function on round  $t$ , with derivative bounded by  $G_t$ , then he can do no better than if he only had access to linear functions. On a similar note, if the Adversary is given the weak restriction that his functions be  $\sigma_t$ -strongly convex on round  $t$ , then he can do no better than if he could only choose  $\sigma_t$ -quadratic functions.

**Theorem 14** For fixed  $X$ ,  $\langle G_t \rangle$ , and  $\langle \sigma_t \rangle$ , the values of the Quadratic Game and the Strongly Convex Game are equal<sup>3</sup>:

$$V_T(\mathcal{G}_{st\text{-conv}}(X, \langle G_t \rangle, \langle \sigma_t \rangle)) = V_T(\mathcal{G}_{quad}(X, \langle G_t \rangle, \langle \sigma_t \rangle)).$$

For a fixed  $X$  and  $\langle G_t \rangle$ , the values of the Convex Game and the Linear Game are equal:

$$V_T(\mathcal{G}_{conv}(X, \langle G_t \rangle)) = V_T(\mathcal{G}_{lin}(X, \langle G_t \rangle)).$$

We need the following lemma whose proof is postponed to the appendix. Define the regret function

$$R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_T, f_T) = \sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in X} \sum_{t=1}^T f_t(\mathbf{x}).$$

**Lemma 15** Consider a sequence of sets  $\{N_s\}_{s=1}^T$  and  $M \subseteq N_t$  for some  $t$ . Suppose that for all  $f_t \in N_t$  and  $\mathbf{x}_t \in X$  there exists  $f_t^* \in M$  such that for all

$$\begin{aligned} & (\mathbf{x}_1, f_1, \dots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_{t+1}, f_{t+1}, \dots, \mathbf{x}_T, f_T), \\ & R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_t, f_t, \dots, \mathbf{x}_T, f_T) \\ & \leq R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_t, f_t^*, \dots, \mathbf{x}_T, f_T). \end{aligned}$$

Then

$$\begin{aligned} & \inf_{\mathbf{x}_1} \sup_{f_1 \in N_1} \dots \inf_{\mathbf{x}_t} \sup_{f_t \in N_t} \dots \inf_{\mathbf{x}_T} \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_T, f_T) \\ & = \inf_{\mathbf{x}_1} \sup_{f_1 \in N_1} \dots \inf_{\mathbf{x}_t} \sup_{f_t \in M} \dots \inf_{\mathbf{x}_T} \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_T, f_T). \end{aligned}$$

**Proof:**[Proof of Theorem 14] Given the sequences  $\langle G_t \rangle, \langle \sigma_t \rangle$ , let  $L_t(\mathbf{x}_t)$  be defined as for the Strongly Convex Game (Definition 3) and  $L_t^*(\mathbf{x}_t)$  be defined as for the Quadratic Game (Definition 2). Observe that  $L_t^* \subseteq L_t$  for any  $t$ . Moreover, for any  $f_t \in L_t$  and  $\mathbf{x}_t \in X$ , define  $f_t^*(\mathbf{x}) = f_t(\mathbf{x}_t) + \nabla f_t(\mathbf{x}_t)^\top (\mathbf{x} - \mathbf{x}_t) + \frac{1}{2} \sigma_t \|\mathbf{x} - \mathbf{x}_t\|^2$ . By definition,  $f_t(\mathbf{x}_t) = f_t^*(\mathbf{x}_t)$  and  $\nabla f_t(\mathbf{x}_t) = \nabla f_t^*(\mathbf{x}_t)$ . Hence,  $f_t^* \in L_t^*$ . Furthermore,  $f_t(\mathbf{x}) \geq f_t^*(\mathbf{x})$  for any  $\mathbf{x} \in X$ , and  $\mathbf{x}^*$  in particular. Hence, for all  $(\mathbf{x}_1, f_1, \dots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_{t+1}, f_{t+1}, \dots, \mathbf{x}_T, f_T)$ ,

$$\begin{aligned} & R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_t, f_t, \dots, \mathbf{x}_T, f_T) \\ & \leq R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_t, f_t^*, \dots, \mathbf{x}_T, f_T). \end{aligned}$$

The statement of the first part of the theorem follows by Lemma 15, applied for every  $t \in \{1, \dots, T\}$ . The second part is proved by analogous reasoning. ■

<sup>3</sup>We note that the computation of  $V_T$  for the Quadratic Game required a particular restriction on the player, Assumption 5.1, where here we only consider a fixed domain  $X$ .

## Acknowledgments

We gratefully acknowledge the support of DARPA under grant FA8750-05-2-0249 and NSF under grant DMS-0707060.

## Appendix

**Proof:**[Proof of Lemma 12] We write

$$P_t(\Delta, \delta) := G_t \|\Delta - \delta\| - \frac{1}{2} \sigma_t \|\Delta - \delta\|^2 - \frac{1}{2} \sigma_{1:t-1} \|\delta\|^2$$

and

$$Q_t(\Delta) := \sup_{\delta} P_t(\Delta, \delta),$$

then our goal is to obtain  $\inf_{\Delta: \|\Delta\| \leq \frac{2G_t}{\sigma_t}} Q_t(\Delta)$ . We now proceed to show that the choice  $\Delta = \mathbf{0}$  is optimal. For this choice,

$$Q_t(\mathbf{0}) = \sup_{\delta} G_t \|\delta\| - \frac{1}{2} \sigma_{1:t} \|\delta\|^2 = \frac{G_t^2}{2\sigma_{1:t}}.$$

Here the optimal choice of  $\delta$  is any vector such that  $\|\delta\| = \frac{G_t}{\sigma_{1:t}}$ .

Now let us consider the case that  $\Delta \neq \mathbf{0}$ . First, suppose  $\Delta \neq \delta$ . Note that the optimum  $\sup_{\delta} P_t(\Delta, \delta)$  will be obtained when the gradient with respect to  $\delta$  is zero, i.e.

$$-G_t \frac{\Delta - \delta}{\|\Delta - \delta\|} - \sigma_t(\delta - \Delta) - \sigma_{1:t-1}\delta = \mathbf{0}$$

implying that  $\delta$  is a linear scaling of  $\Delta$ , i.e.  $\delta = c\Delta$ . The second case,  $\Delta = \delta$ , also implies that  $\delta$  is a linear scaling of  $\Delta$ . Substituting this optimal form of  $\delta$ ,

$$\begin{aligned} Q_t(\Delta) &= \sup_{c \in \mathbb{R}} [G_t |1 - c| \cdot \|\Delta\| \\ & \quad - \frac{1}{2} \sigma_t (1 - c)^2 \|\Delta\|^2 - \frac{1}{2} \sigma_{1:t-1} c^2 \|\Delta\|^2]. \end{aligned}$$

We now claim that the supremum over  $c \in \mathbb{R}$  occurs at some  $c^* \leq 1$  for any choice of  $\Delta$ . Assume by contradiction that  $c^* > 1$  for some  $\Delta$ . Then  $\tilde{c} = -c^* + 2$  achieves at least the same value as  $c^*$  since  $|1 - c^*| = |1 - \tilde{c}|$  while  $(c^*)^2 > (\tilde{c})^2$ , making the last term larger, which is a contradiction. Hence,  $c \leq 1$  and, collecting the terms,

$$\begin{aligned} Q_t(\Delta) &= \sup_{c \leq 1} \left[ \left( G_t \|\Delta\| - \frac{1}{2} \sigma_t \|\Delta\|^2 \right) \right. \\ & \quad \left. + c \cdot \left( \sigma_t \|\Delta\|^2 - G_t \|\Delta\| \right) - c^2 \cdot \left( \frac{1}{2} \sigma_{1:t} \|\Delta\|^2 \right) \right]. \end{aligned}$$

Since we now assume  $\|\Delta\| \neq \mathbf{0}$ , we see that the supremum is achieved for  $c^* = \frac{\sigma_t \|\Delta\|^2 - G_t \|\Delta\|}{\sigma_{1:t} \|\Delta\|^2} = \frac{\sigma_t \|\Delta\| - G_t}{\sigma_{1:t} \|\Delta\|} \leq 1$  and

$$\begin{aligned} Q_t(\Delta) &= \frac{(\sigma_t \|\Delta\|^2 - G_t \|\Delta\|)^2}{2\sigma_{1:t} \|\Delta\|^2} + (G_t \|\Delta\| - \frac{1}{2} \sigma_t \|\Delta\|^2) \\ &= \frac{\sigma_t^2 \|\Delta\|^2 - \sigma_t \|\Delta\| G_t + G_t^2}{2\sigma_{1:t}} \\ & \quad + (G_t \|\Delta\| - \frac{1}{2} \sigma_t \|\Delta\|^2) \\ &= \frac{\sigma_t}{\sigma_{1:t}} \left( \frac{1}{2} \sigma_t \|\Delta\|^2 - \|\Delta\| G_t \right) \\ & \quad + (G_t \|\Delta\| - \frac{1}{2} \sigma_t \|\Delta\|^2) + \frac{G_t^2}{2\sigma_{1:t}} \\ &= \frac{\sigma_{1:t-1}}{\sigma_{1:t}} \left( G_t - \frac{1}{2} \sigma_t \|\Delta\| \right) \|\Delta\| + \frac{G_t^2}{2\sigma_{1:t}} > \frac{G_t^2}{2\sigma_{1:t}}, \end{aligned}$$

where the last inequality holds by because  $\|\Delta\| \leq \frac{2G_t}{\sigma_t}$ . Hence, the value  $Q_t(\Delta)$  is strictly larger than  $G_t^2/(2\sigma_{1:t})$  whenever  $\|\Delta\| > 0$  and is equal to this value if  $\Delta = \mathbf{0}$ . Hence, the optimal choice for the Player is to choose  $\Delta = \mathbf{0}$ . ■

**Proof:**[Proof of Lemma 15] Fix  $f_t \in L_t$  and  $\mathbf{x}_t \in X$ . Let  $f_t^* \in M$  be as in the statement of the lemma. Define

$$h_1(\mathbf{x}_1, f_1, \dots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_{t+1}, f_{t+1}, \dots, \mathbf{x}_T, f_T) \\ := R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_t, f_t, \dots, \mathbf{x}_T, f_T)$$

$$h_2(\mathbf{x}_1, f_1, \dots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_{t+1}, f_{t+1}, \dots, \mathbf{x}_T, f_T) \\ := R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_t, f_t^*, \dots, \mathbf{x}_T, f_T).$$

By assumption,  $h_1 \leq h_2$ . Hence, we can inf/sup over the variables  $\mathbf{x}_{t+1}, f_{t+1}, \dots, \mathbf{x}_T, f_T$ , obtaining

$$\inf_{\mathbf{x}_{t+1}} \sup_{f_{t+1} \in N_{t+1}} \dots \inf_{\mathbf{x}_T} \\ \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_t, f_t, \dots, \mathbf{x}_T, f_T) \\ \leq \inf_{\mathbf{x}_{t+1}} \sup_{f_{t+1} \in N_{t+1}} \dots \inf_{\mathbf{x}_T} \\ \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_t, f_t^*, \dots, \mathbf{x}_T, f_T)$$

for any  $(\mathbf{x}_1, f_1, \dots, \mathbf{x}_{t-1}, f_{t-1})$ . Hence, since  $f_t^* \in M$

$$\sup_{f_t \in N_t} \inf_{\mathbf{x}_{t+1}} \sup_{f_{t+1} \in N_{t+1}} \dots \inf_{\mathbf{x}_T} \\ \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_t, f_t, \dots, \mathbf{x}_T, f_T) \\ \leq \sup_{f_t \in M} \inf_{\mathbf{x}_{t+1}} \sup_{f_{t+1} \in N_{t+1}} \dots \inf_{\mathbf{x}_T} \\ \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \dots, \mathbf{x}_t, f_t, \dots, \mathbf{x}_T, f_T)$$

for all  $(\mathbf{x}_1, f_1, \dots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_t)$ . Since  $M \subseteq N_t$ , the above is in fact an equality. Since the two functions of the variables  $(\mathbf{x}_1, f_1, \dots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_t)$  are equal, taking inf's and sup's over these variables we obtain the statement of the lemma. ■

**Lemma 16** *The expression*

$$\frac{FG \sin \alpha}{\sqrt{F^2 + G^2 + K^2}} + \sqrt{F^2 + G^2 + K^2 - 2FG \sin \alpha}$$

is no more than  $\sqrt{F^2 + G^2 + K^2}$  for constants  $F, G, K > 0$  and any  $\alpha$ .

**Proof:** We are interested in proving that the supremum of

$$\phi(\alpha) = \frac{FG \sin \alpha}{\sqrt{F^2 + G^2 + K^2}} + \sqrt{F^2 + G^2 + K^2 - 2FG \sin \alpha}$$

over  $[-\pi/2, \pi/2]$  is attained at  $\alpha = 0$ . Setting the derivative of  $\Phi(\alpha)$  to zero,

$$\frac{FG \cos \alpha}{\sqrt{F^2 + G^2 + K^2}} - \frac{FG \cos \alpha}{\sqrt{F^2 + G^2 + K^2 - 2FG \sin \alpha}} = 0$$

which implies that either  $\cos \alpha = 0$  or  $\sin \alpha = 0$ , i.e.  $\alpha \in \{-\pi/2, 0, \pi/2\}$ . Taking the second derivative, we get

$$\phi''(\alpha) = -\frac{FG \sin \alpha}{\sqrt{F^2 + G^2 + K^2}} \\ - \left( -\frac{FG \sin \alpha}{\sqrt{F^2 + G^2 + K^2 - 2FG \sin \alpha}} \right. \\ \left. + \frac{(FG \cos \alpha)(FG \cos \alpha)}{(F^2 + G^2 + K^2 - 2FG \sin \alpha)^{3/2}} \right).$$

Thus,  $\phi''(0) < 0$ . We conclude that the optimum is attained at  $\alpha = 0$  and therefore

$$\phi(\alpha) \leq \sqrt{F^2 + G^2 + K^2}$$

■

## References

- [1] Peter Bartlett, Elad Hazan, and Alexander Rakhlin. Adaptive online gradient descent. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.
- [2] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [3] T.M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, January 1991.
- [4] Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *COLT*, pages 499–513, 2006.
- [5] Erik Ordentlich and Thomas M. Cover. The cost of achieving the best portfolio in hindsight. *Math. Oper. Res.*, 23(4):960–982, 1998.
- [6] Shai Shalev-Shwartz and Yoram Singer. Convex repeated games and Fenchel duality. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.
- [7] Eiji Takimoto and Manfred K. Warmuth. The mini-max strategy for gaussian density estimation. pp. In *COLT '00: Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 100–106, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [8] V. Vovk. Competitive on-line linear regression. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 364–370, Cambridge, MA, USA, 1998. MIT Press.
- [9] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.



---

# Regret Bounds for Sleeping Experts and Bandits

---

**Robert D. Kleinberg\***  
Department of Computer Science  
Cornell University  
Ithaca, NY 14853  
rdk@cs.cornell.edu

**Alexandru Niculescu-Mizil†**  
Department of Computer Science  
Cornell University  
Ithaca, NY 14853  
alexnm@cs.cornell.edu

**Yogeshwer Sharma‡**  
Department of Computer Science  
Cornell University  
Ithaca, NY 14853  
yogi@cs.cornell.edu

## Abstract

We study on-line decision problems where the set of actions that are available to the decision algorithm vary over time. With a few notable exceptions, such problems remained largely unaddressed in the literature, despite their applicability to a large number of practical problems. Departing from previous work on this “Sleeping Experts” problem, we compare algorithms against the payoff obtained by the *best ordering* of the actions, which is a natural benchmark for this type of problem. We study both the full-information (best expert) and partial-information (multi-armed bandit) settings and consider both stochastic and adaptive adversaries. For all settings we give algorithms achieving (almost) information-theoretically optimal regret bounds (up to a constant or a sub-logarithmic factor) with respect to the best-ordering benchmark.

## 1 Introduction

In on-line decision problems, or sequential prediction problems, an algorithm must choose, in each of the  $T$  consecutive rounds, one of the  $n$  possible actions. In each round, each action receives a real valued positive payoff in  $[0, 1]$ , initially unknown to the algorithm. At the end of each round the algorithm is revealed some information about the payoffs of the actions in that round. The goal of the algorithm is to maximize the total payoff, i.e. the sum of the payoffs of the chosen actions in each round. The standard on-line decision settings are the *best expert* setting (or the full-information setting) in which, at the end of the round, the payoffs of *all*  $n$  strategies are revealed to the algorithm, and the *multi-armed bandit* setting (or the partial-information setting) in which only the payoff of the chosen strategy is revealed. Customarily, in the best expert setting the strategies are called *experts* and in the multi-armed bandit setting the strategies are called *bandits* or *arms*. We use *actions* to generically refer to both

types of strategies, when we do not refer particularly to either.

The performance of the algorithm is typically measured in terms of *regret*. The regret is the difference between the expected payoff of the algorithm and the payoff of a single fixed strategy for selecting actions. The usual single fixed strategy to compare against is the one which always selects the expert or bandit that has the highest total payoff over the  $T$  rounds (in hindsight).

The usual assumption in online learning problems is that all actions are available at all times. In many applications, however, this assumption is not appropriate. In network routing problems, for example, some of the routes are unavailable at some point in time due to router or link crashes. Or, in electronic commerce problems, items are out of stock, sellers are not available (due to maintenance or simply going out of business), and buyers do not buy all the time. Even in the setting that originally motivated the multi-armed bandit problems, a gambler playing slot machines, some of the slot machines might be occupied by other players at any given time.

In this paper we relax the assumption that all actions are available at all times, and allow the set of available actions to vary from one round to the next, a model known as “predictors that specialize” or “sleeping experts” in prior work. The first foundational question that needs to be addressed is how to define regret when the set of available actions may vary over time. Defining regret with respect to the best action in hindsight is no longer appropriate since that action might sometimes be unavailable. A useful thought experiment for guiding our intuition is the following: if each action had a fixed payoff distribution that was *known* to the decision-maker, what would be the best way to choose among the available actions? The answer is obvious: one should order all of the actions according to their expected payoff, then choose among the available actions by selecting the one which ranks highest in this ordering. Guided by the outcome of this thought experiment, we define our base to be the best ordering of actions in hindsight (see Section 2 for a formal definition) and contend that this is a natural and intuitive way to define regret in our setting. This contention is also supported by the informal observation that order-based decision rules seem to resemble the way people make choices in situations with a varying set of actions, e.g. choosing which brand of beer to buy at a store.

We prove lower and upper bounds on the regret with re-

---

\*Supported by NSF grants CCF-0643934 and CCF-0729102.

†Supported by NSF grants 0347318, 0412930, 0427914, and 0612031.

‡Supported by NSF grant CCF-0514628.

spect to the best ordering for both the best expert setting and the multi-armed bandit settings. We first explore the case of stochastic adversary, where the payoffs received by expert (bandit)  $i$  at each time step are independent samples from an unknown but fixed distribution  $P_i(\cdot)$  supported on  $[0, 1]$  with mean  $\mu_i$ . Assuming that  $\mu_1 > \mu_2 > \dots > \mu_n$  (and the algorithm, of course, does not know the identities of these actions) we show that the regret of any learning algorithm will necessarily be at least  $\Omega\left(\sum_{i=1}^{n-1} \frac{1}{\mu_i - \mu_{i+1}}\right)$  in the best expert setting, and  $\Omega\left(\log(T) \sum_{i=1}^{n-1} \frac{1}{\mu_i - \mu_{i+1}}\right)$  in the multi-armed bandit setting if the game is played for  $T$  rounds (for  $T$  sufficiently large). We also present efficient learning algorithms for both settings. For the multi-armed bandit setting, our algorithm, called AUER, is an adaptation of the UCB1 algorithm in Auer et al [ACBF02], which comes within a constant factor of the lower bound mentioned above. For the expert setting, a very simple algorithm, called “follow-the-awake-leader”, which is a variant of “follow-the-leader” [Han57, KV05], comes within a constant factor of the lower bound above. While our algorithms are adaptations of existing techniques, the proofs of the upper and lower bounds hinge on some technical innovations. For the lower bound, we must modify the classic asymptotic lower bound proof of Lai and Robbins [LR85] to obtain a bound which holds at all sufficiently large finite times. We also prove a novel lemma (Lemma 3) that allows us to relate a regret upper bound arising from application of UCB1 to a sum of lower bounds for two-armed bandit problems.

Next we explore the fully adversarial case where we make no assumptions on how the payoffs for each action are generated. We show that the regret of any learning algorithm must be at least  $\Omega\left(\sqrt{Tn \log(n)}\right)$  for the best expert setting and  $\Omega\left(\sqrt{Tn^2}\right)$  for the multi-armed bandit setting. We also present algorithms whose regret is within a constant factor of the lower bound for the best expert setting, and within  $\mathcal{O}\left(\sqrt{\log(n)}\right)$  of the lower bound for the multi-armed bandit setting. It is worth noting that the gap of  $\mathcal{O}\left(\sqrt{\log(n)}\right)$  also exists in the all-awake bandit problem.

The fully adversarial case, however, proves to be harder, and neither algorithm is computationally efficient. To appreciate the hardness of the fully adversarial case, one can prove<sup>1</sup> that, unless  $P = NP$ , any low regret algorithm that learns internally a consistent ordering over experts can not be computationally efficient. Note that this does not mean that there can be no computationally efficient, low regret algorithms for the fully adversarial case. There might exist learning algorithms that are able to achieve low regret without actually learning a consistent ordering over experts. Finding such algorithms, if they do indeed exist, remains an open problem.

## 1.1 Related work

**Sequential prediction problems.** The best-expert and multi-armed bandit problems correspond to special cases of our model in which every action is always available. These prob-

lems have been widely studied, and we draw on this literature to design algorithms and prove lower bounds for the generalizations considered here. The adversarial expert paradigm was introduced by Littlestone and Warmuth [LW94], and Vovk [Vov90]. Cesa-Bianchi et al [CBFH<sup>+</sup>97] further developed this paradigm in work which gave optimal regret bounds of  $\sqrt{T(\ln n)}$  and Vovk [Vov98] characterized the achievable regret bounds in these settings.

The multi-armed bandit model was introduced by Robbins [Rob]. Lai and Robbins [LR85] gave asymptotically optimal strategies for the stochastic version of bandit problem—in which there is a distribution of rewards on each arm and the rewards in each time step are drawn according to this distribution. Auer, Cesa-Bianchi, Fischer [ACBF02] introduced the algorithm UCB1 and showed that the optimal regret bounds of  $\mathcal{O}(\log T)$  can be achieved uniformly over time for the stochastic bandit problem. (In this bound, the big-O hides a constant depending on the means and differences of means of payoffs.) For the adversarial version of the multi-armed bandit problem, Auer, Cesa-Bianchi, Freund, and Schapire [ACBFS02] proposed the algorithm Exp3 which achieves the regret bound of  $\mathcal{O}(\sqrt{Tn \log n})$ , leaving a  $\sqrt{\log n}$  factor gap from the lower bound of  $\Omega(\sqrt{nT})$ . It is worth noting that the lower bound holds even for an oblivious adversary, one which chooses a sequence of payoff functions independently of the algorithm’s choices.

**Prediction with sleeping experts.** Freund, Schapire, Singer, and Warmuth [FSSW97] and Blum and Mansour [BM05] have considered sleeping experts problems before, analyzing algorithms in a framework different from the one we adopt here. In the model of Freund et al., as in our model, a set of awake experts is specified in each time period. The goal of the algorithm is to choose one expert in each time period so as to minimize regret against the best “mixture” of experts (which constitutes their benchmark). A mixture  $\mathbf{u}$  is a probability distribution  $(u_1, u_2, \dots, u_n)$  over  $n$  experts which in time period  $t$  selects an expert according to the restriction of  $\mathbf{u}$  to the set of awake experts.

We consider a natural evaluation criterion, namely the best ordering of experts. In the special case when all experts are always awake, both evaluation criteria degenerate to picking the best expert. Our “best ordering” criterion can be regarded as a degenerate case of the “best mixture” criterion of Freund et al. as follows. For the ordering  $\sigma$ , we assign probabilities  $\frac{1}{Z}(1, \epsilon, \epsilon^2, \dots, \epsilon^{n-1})$  to the sequence of experts  $(\sigma(1), \sigma(2), \dots, \sigma(n))$  where  $Z = \frac{1-\epsilon^n}{1-\epsilon}$  is the normalization factor and  $\epsilon > 0$  is an arbitrarily small positive constant. The only problem is that the bounds that we get from [FSSW97] in this degenerate case are very weak. As  $\epsilon \rightarrow 0$ , their bound reduces to comparing the algorithm’s performance to the ordering  $\sigma$ ’s performance only for time periods when  $\sigma(1)$  expert is awake, and ignoring the time periods when  $\sigma(1)$  is not awake. Therefore, a natural reduction of our problem to the problem considered by Freund et al. defeats the purpose of giving equal importance to all time periods.

Blum and Mansour [BM05] consider a generalization of the sleeping expert problem, where one has a set of *time selection functions* and the algorithm aims to have low regret

<sup>1</sup>It is a simple reduction from feedback arc set problem, which is omitted from this extended abstract.

with respect to every expert, according to every time selection function. It is possible to solve our regret-minimization problem (with respect to the best ordering of experts) by reducing to the regret-minimization problem solved by Blum and Mansour, but this leads to an algorithm which is neither computationally efficient nor information-theoretically optimal. We now sketch the details of this reduction. One can define a time selection function for each (ordering, expert) pair  $(\sigma, i)$ , according to  $I_{\sigma,i}(t) = 1$  if  $i \preceq_{\sigma} j$  for all  $j \in A_t$  (that is,  $\sigma$  chooses  $i$  in time period  $t$  if  $I_{\sigma,i}(t) = 1$ ). The regret can now be bounded, using Blum and Mansour’s analysis, as

$$\begin{aligned} & \sum_{i=1}^n \mathcal{O} \left( \sqrt{T_i \log(n \cdot n! \cdot n)} + \log(n! \cdot n^2) \right) \\ &= \mathcal{O} \left( \sqrt{Tn^2 \log n} + n \log n \right). \end{aligned}$$

This algorithm takes exponential time (due to the exponential number of time selection functions) and gives a regret bound of  $\mathcal{O}(\sqrt{Tn^2 \log n})$  against the best ordering, a bound which we improve in Section 4 using a different algorithm which also takes exponential time but is information-theoretically optimal. (Of course, Blum and Mansour were designing their algorithm for a different objective, not trying to get low regret with respect to best ordering. Our improved bound for regret with respect to the best ordering does not imply an improved bound for experts learning with time selection functions.)

A recent paper by Langford and Zhang [LZ07] presents an algorithm called the *Epoch-Greedy algorithm* for bandit problems with side information. This is a generalization of the multi-armed bandit problem in which the algorithm is supplied with a piece of *side information* in each time period before deciding which action to play. Given a hypothesis class  $\mathcal{H}$  of functions mapping side information to actions, the Epoch-Greedy algorithm achieves low regret against a sequence of actions generated by applying a single function  $h \in \mathcal{H}$  to map the side information in every time period to an action. (The function  $h$  is chosen so that the resulting sequence has the largest possible total payoff.) The stochastic case of our problem is reducible to theirs, by treating the set of available actions,  $A_t$ , as a piece of side information and considering the hypothesis class  $\mathcal{H}$  consisting of functions  $h_{\sigma}$ , for each total ordering  $\sigma$  of the set of actions, such that  $h_{\sigma}(A)$  selects the element of  $A$  which appears first in the ordering  $\sigma$ . The regret bound in [LZ07] is expressed implicitly in terms of the expected regret of an empirical reward maximization estimator, which makes it difficult to compare this bound with ours. Instead of pursuing this reduction from our problem to the contextual bandit problem in [LZ07], Section 3.1.1 presents a very simple bandit algorithm for the stochastic setting with an explicit regret bound that is provably information-theoretically optimal.

## 2 Terminology and Conventions

We assume that there is a fixed pool of actions,  $\{1, 2, \dots, n\}$ , with  $n$  known. We will sometimes refer to an action by *expert* in the best expert setting and by *arm* or *bandit* in the multi-armed bandit setting. At each time step  $t \in \{1, 2, \dots, T\}$ ,

an adversary chooses a subset  $A_t \subseteq \{1, 2, \dots, n\}$  of the actions to be available. The algorithm can only choose among available actions, and only available actions receive rewards. The reward received by an available action  $i$  at time  $t$  is  $r_i(t) \in [0, 1]$ .

We will consider two models for assigning rewards to actions: a stochastic model and an adversarial model. (In contrast, the choice of the set of awake experts is always adversarial.) In the stochastic model the reward for arm  $i$  at time  $t$ ,  $r_i(t)$ , is drawn independently from a fixed unknown distribution  $P_i(\cdot)$  with mean  $\mu_i$ . In the adversarial model we make no stochastic assumptions on how the rewards are assigned to actions. Instead, we assume that the rewards are selected by an adversary. The adversary is potentially but not necessarily randomized.

Let  $\sigma$  be an ordering (permutation) of the  $n$  actions, and  $A$  a subset of the actions. We denote by  $\sigma(A)$  the action in  $A$  that is highest ranked in  $\sigma$ . The reward of an ordering is the reward obtained by selecting at each time step the highest ranked action available.

$$R_{\sigma,T} = \sum_{t=1}^T r_{\sigma(A_t)}(t) \quad (1)$$

Let  $R_T = \max_{\sigma} R_{\sigma,T}$  ( $\max_{\sigma} \mathbb{E}[R_{\sigma,T}]$  in the stochastic rewards model) be the reward obtained by the best ordering. We define the regret of an algorithm with respect to the best ordering as the expected difference between the reward obtained by the best ordering and the total reward of the algorithm’s chosen actions  $x(1), x(2), \dots, x(t)$ :

$$REG_T = \mathbb{E} \left[ R_T - \sum_{t=1}^T r_{x(t)}(t) \right] \quad (2)$$

where the expectation is taken over the algorithm’s random choices and the randomness of the reward assignment in the stochastic reward model.

## 3 Stochastic Model of Rewards

We first explore the stochastic rewards model, where the reward for action  $i$  at each time step is drawn independently from a fixed unknown distribution  $P_i(\cdot)$  with mean  $\mu_i$ . For simplicity of presentation, throughout this section we assume that  $\mu_1 > \mu_2 > \dots > \mu_n$ . That is the lower numbered actions are better than the higher numbered actions. Let  $\Delta_{i,j} = \mu_i - \mu_j$  for all  $i < j$  be the expected increase in the reward of expert  $i$  over expert  $j$ .

We present optimal (up to a constant factor) algorithms for both the best expert and the multi-armed bandit setting. Both algorithms are natural extensions of algorithms for the all-awake problem to the sleeping-experts problem. The analysis of the algorithms, however, is not a straightforward extension of the analysis for the all-awake problem and new proof techniques are required.

### 3.1 Best expert setting

In this section we study algorithms for the best expert setting with stochastic rewards. We prove matching (up to a constant factor) information-theoretic upper and lower bounds on the regret of such algorithms.

### 3.1.1 Upper bound (algorithm: FTAL)

To get an upper bound on regret we adapt the “follow the leader” algorithm [Han57, KV05] to the sleeping experts setting: at each time step the algorithm chooses the awake expert that has the highest average payoff, where the average is taken over the time steps when the expert was awake. If an expert is awake for the first time, then the algorithm chooses it. (If there are more than one such experts, then the algorithm chooses one of them arbitrarily.) The pseudocode for the algorithm is shown in Algorithm 1. The algorithm is called **Follow The Awake Leader** (FTAL for short).

```

1 Initialize  $z_i = 0$  and  $n_i = 0$  for all  $i \in [n]$ .
2 for  $t = 1$  to  $T$  do
3   if  $\exists j \in A_t$  s.t.  $n_j = 0$  then
4     Play expert  $x(t) = j$ 
5   else
6     Play expert  $x(t) = \arg \max_{i \in A_t} \left( \frac{z_i}{n_i} \right)$ 
7   end
8   Observe payoff  $r_i(t)$  for all  $i \in A_t$ 
9    $z_i \leftarrow z_i + r_i(t)$  for all  $i \in A_t$ 
10   $n_i \leftarrow n_i + 1$  for all  $i \in A_t$ 
11 end

```

**Algorithm 1:** Follow-the-awake-leader (FTAL) algorithm for sleeping experts problem with stochastic adversary.

**Theorem 1** *The FTAL algorithm has a regret of at most*

$$\sum_{j=1}^{n-1} \frac{32}{\Delta_{j,j+1}}$$

with respect to the best ordering.

The theorem follows immediately from the following pair of lemmas. The second of these lemmas will also be used in Section 3.2.

**Lemma 2** *The FTAL algorithm has a regret of at most*

$$\sum_{j=2}^n \sum_{i=1}^{j-1} \frac{8}{\Delta_{i,j}^2} (\Delta_{i,i+1} + \Delta_{j-1,j})$$

with respect to the best ordering.

**Proof:** Let  $n_{i,t}$  be the number of times expert  $i$  has been awake until time  $t$ . Let  $\hat{\mu}_{i,t}$  be expert  $i$ 's average payoff until time  $t$ . The Azuma-Hoeffding Inequality [Azu67, Hoe63] says that

$$\begin{aligned} & \mathbb{P}[n_{j,t} \hat{\mu}_{j,t} > n_{j,t} \mu_j + n_{j,t} \Delta_{i,j}/2] \\ & \leq e^{-\frac{n_{j,t}^2 \Delta_{i,j}^2}{8 \cdot n_{j,t}}} = e^{-\frac{\Delta_{i,j}^2 n_{j,t}}{8}}, \end{aligned}$$

and

$$\begin{aligned} & \mathbb{P}[n_{i,t} \hat{\mu}_{i,t} < n_{i,t} \mu_i - n_{i,t} \Delta_{i,j}/2] \\ & \leq e^{-\frac{n_{i,t}^2 \Delta_{i,j}^2}{8 \cdot n_{i,t}}} = e^{-\frac{\Delta_{i,j}^2 n_{i,t}}{8}}. \end{aligned}$$

Let us say that the FTAL algorithm suffers an  $(i, j)$ -anomaly of type 1 at time  $t$  if  $x_t = j$  and  $\hat{\mu}_{j,t} - \mu_j > \Delta_{i,j}/2$ . Let us say that FTAL suffers an  $(i, j)$ -anomaly of type 2 at time  $t$  if  $i_t^* = i$  and  $\mu_i - \hat{\mu}_{i,t} > \Delta_{i,j}/2$ . Note that when FTAL picks a strategy  $x_t = j \neq i = i_t^*$ , it suffers an  $(i, j)$ -anomaly of type 1 or 2, or possibly both. We will denote the event of an  $(i, j)$ -anomaly of type 1 (resp. type 2) at time  $t$  by  $\mathcal{E}_{i,j}^{(1)}(t)$  (resp.  $\mathcal{E}_{i,j}^{(2)}(t)$ ), and we will use  $M_{i,j}^{(1)}$ , resp.  $M_{i,j}^{(2)}$ , to denote the total number of  $(i, j)$ -anomalies of types 1 and 2, respectively. We can bound the expected value of  $M_{i,j}^{(1)}$  by

$$\mathbb{E}[M_{i,j}^{(1)}] \leq \sum_{t=1}^{\infty} e^{-\frac{\Delta_{i,j}^2 n_{j,t}}{8}} \mathbf{1}\{j \in A_t\} \quad (3)$$

$$\begin{aligned} & \leq \sum_{n=1}^{\infty} e^{-\frac{\Delta_{i,j}^2 n}{8}} \quad (4) \\ & = \frac{1}{e^{\Delta_{i,j}^2/8} - 1} \leq \frac{8}{\Delta_{i,j}^2}, \end{aligned}$$

where line (4) is justified by observing that distinct nonzero terms in (3) have distinct values of  $n_{j,t}$ . The expectation of  $M_{i,j}^{(2)}$  is also bounded by  $8/\Delta_{i,j}^2$ , via an analogous argument.

Recall that  $A_t$  denotes the set of awake experts at time  $t$ ,  $x_t \in A_t$  denotes the algorithm's choice at time  $t$ , and  $r_i(t)$  denotes the payoff of expert  $i$  at time  $t$  (which is distributed according to  $P_i(\cdot)$ ). Let  $i_t^* \in A_t$  denote the optimal expert at time  $t$  (i.e., the lowest-numbered element of  $A_t$ ). Let us bound the regret of the FTAL algorithm now.

$$\begin{aligned} & \mathbb{E} \left[ \sum_{t=1}^T (r_{i_t^*}(t) - r_{x_t}(t)) \right] \\ & = \mathbb{E} \left[ \sum_{t=1}^T \Delta_{i_t^*, x_t} \right] \\ & = \mathbb{E} \left[ \sum_{t=1}^T \mathbf{1} \left\{ \mathcal{E}_{i_t^*, x_t}^{(1)}(t) \vee \mathcal{E}_{i_t^*, x_t}^{(2)}(t) \right\} \Delta_{i_t^*, x_t} \right] \\ & \leq \mathbb{E} \left[ \sum_{t=1}^T \mathbf{1} \left\{ \mathcal{E}_{i_t^*, x_t}^{(1)}(t) \right\} \Delta_{i_t^*, x_t} \right] \\ & \quad + \mathbb{E} \left[ \sum_{t=1}^T \mathbf{1} \left\{ \mathcal{E}_{i_t^*, x_t}^{(2)}(t) \right\} \Delta_{i_t^*, x_t} \right] \end{aligned}$$

With the convention that  $\Delta_{i,j} = 0$  for  $j \leq i$ , the first term can be bounded by:

$$\begin{aligned} & \mathbb{E} \left[ \sum_{t=1}^T \mathbf{1} \left\{ \mathcal{E}_{i_t^*, x_t}^{(1)}(t) \right\} \Delta_{i_t^*, x_t} \right] \\ & = \mathbb{E} \left[ \sum_{t=1}^T \sum_{j=2}^n \mathbf{1} \left\{ \mathcal{E}_{i_t^*, j}^{(1)}(t) \right\} \Delta_{i_t^*, j} \right] \end{aligned}$$

(Since the event  $\mathcal{E}_{i_t^*, j}^{(1)}(t)$  occurs only for  $j = x_t$ .)

$$= \mathbb{E} \left[ \sum_{t=1}^T \sum_{j=2}^n \mathbf{1} \left\{ \mathcal{E}_{i_t^*, j}^{(1)}(t) \right\} \sum_{i=i_t^*}^{j-1} (\Delta_{i,j} - \Delta_{i+1,j}) \right] \quad (5)$$

$$\leq \mathbb{E} \left[ \sum_{t=1}^T \sum_{j=2}^n \sum_{i=i_t^*}^{j-1} \mathbf{1} \left\{ \mathcal{E}_{i,j}^{(1)}(t) \right\} \Delta_{i,i+1} \right]$$

(Since  $\mathbf{1} \left\{ \mathcal{E}_{i_1,j}^{(1)}(t) \right\} \leq \mathbf{1} \left\{ \mathcal{E}_{i_2,j}^{(1)}(t) \right\}$  for all  $i_1 \leq i_2 < j$ .)

$$\begin{aligned} &\leq \mathbb{E} \left[ \sum_{j=2}^n \sum_{i=1}^{j-1} \Delta_{i,i+1} \sum_{t=1}^T \mathbf{1} \left\{ \mathcal{E}_{i,j}^{(1)}(t) \right\} \right] \\ &= \sum_{j=2}^n \sum_{i=1}^{j-1} \Delta_{i,i+1} \mathbb{E}[M_{i,j}^{(1)}] \\ &\leq \sum_{1 \leq i < j \leq n} \frac{8}{\Delta_{i,j}^2} \Delta_{i,i+1}. \end{aligned}$$

Similarly, the second term can be bounded by

$$\begin{aligned} &\mathbb{E} \left[ \sum_{t=1}^T \mathbf{1} \left\{ \mathcal{E}_{i_t^*, x_t}^{(2)}(t) \right\} \Delta_{i_t^*, x_t} \right] \\ &= \mathbb{E} \left[ \sum_{t=1}^T \sum_{i=1}^{n-1} \mathbf{1} \left\{ \mathcal{E}_{i, x_t}^{(2)}(t) \right\} \Delta_{i, x_t} \right] \end{aligned}$$

(Since event  $\mathcal{E}_{i, x_t}^{(2)}(t)$  occurs only for  $i = i_t^*$ .)

$$\begin{aligned} &= \mathbb{E} \left[ \sum_{t=1}^T \sum_{i=1}^{n-1} \mathbf{1} \left\{ \mathcal{E}_{i, x_t}^{(2)}(t) \right\} \sum_{j=i+1}^{x_t} (\Delta_{i,j} - \Delta_{i,j-1}) \right] \quad (6) \\ &\leq \mathbb{E} \left[ \sum_{t=1}^T \sum_{i=1}^{n-1} \sum_{j=i+1}^{x_t} \mathbf{1} \left\{ \mathcal{E}_{i,j}^{(2)}(t) \right\} \Delta_{j-1,j} \right] \end{aligned}$$

(Since  $\mathbf{1} \left\{ \mathcal{E}_{i,j_1}^{(2)}(t) \right\} \geq \mathbf{1} \left\{ \mathcal{E}_{i,j_2}^{(2)}(t) \right\}$  for all  $i < j_1 \leq j_2$ .)

$$\begin{aligned} &\leq \mathbb{E} \left[ \sum_{i=1}^{n-1} \sum_{j=i+1}^n \Delta_{j-1,j} \sum_{t=1}^T \mathbf{1} \left\{ \mathcal{E}_{i,j}^{(2)}(t) \right\} \right] \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n \Delta_{j-1,j} \mathbb{E}[M_{i,j}^{(2)}] \\ &\leq \sum_{1 \leq i < j \leq n} \frac{8}{\Delta_{i,j}^2} \Delta_{j-1,j} \end{aligned}$$

Adding the two bounds gives the statement of the lemma. ■

**Lemma 3** For  $\Delta_{i,j} = \mu_i - \mu_j$  defined as above

$$\sum_{1 \leq i < j \leq n} \Delta_{i,j}^{-2} \Delta_{i,i+1} \leq 2 \sum_{j=2}^n \Delta_{j-1,j}^{-1}$$

and

$$\sum_{1 \leq i < j \leq n} \Delta_{i,j}^{-2} \Delta_{j-1,j} \leq 2 \sum_{j=2}^n \Delta_{j-1,j}^{-1}.$$

**Proof:** It suffices to prove the first of the two inequalities stated in the lemma; the second follows from the first by replacing each  $\mu_i$  with  $1 - \mu_i$ , which has the effect of replacing  $\Delta_{i,j}$  with  $\Delta_{n+1-j, n+1-i}$ .

For a fixed  $i \in [n]$ , we write  $\sum_{j:j>i} \Delta_{i,j}^{-2}$  as follows.

$$\begin{aligned} \sum_{j:j>i} \Delta_{i,j}^{-2} &= \sum_{j=2}^n \mathbf{1} \{j > i\} \Delta_{i,j}^{-2} \quad (7) \\ &= \int_{x=0}^{\infty} \# \{j : j > i, \Delta_{i,j}^{-2} \geq x\} dx \\ &= \int_{x=0}^{\infty} \# \{j > i, \Delta_{i,j} \leq x^{-1/2}\} dx \\ &= -2 \int_{y=0}^{\infty} \# \{j > i, \Delta_{i,j} \leq y\} y^{-3} dy \end{aligned}$$

(Changing the variable of integration  $x^{-1/2} = y$ )

$$= 2 \int_{y=0}^{\infty} \# \{j > i, \Delta_{i,j} \leq y\} y^{-3} dy. \quad (8)$$

Let us make the following definition, which will be used in the proof below.

**Definition 4** For an expert  $j$  and  $y \geq 0$ , let  $i_y(j)$  be the minimum numbered expert  $i \leq j$  such that  $\Delta_{i,j}$  is no more than  $y$ . That is

$$i_y(j) := \arg \min \{i : i \leq j, \Delta_{i,j} \leq y\}.$$

Now we can write the following chain of inequalities. (Note that the best (highest payoff) expert is indexed as 1, and lowest payoff is indexed  $n$ .)

$$\begin{aligned} &\sum_{j=2}^n \sum_{i=1}^{j-1} \Delta_{i,j}^{-2} \Delta_{i,i+1} \quad (9) \\ &= \sum_{i=1}^{n-1} \Delta_{i,i+1} \sum_{j:j>i} \Delta_{i,j}^{-2} \\ &= 2 \sum_{i=1}^{n-1} \Delta_{i,i+1} \left( \int_{y=0}^{\infty} \# \{j : j > i, \Delta_{i,j} \leq y\} y^{-3} dy \right) \end{aligned}$$

(From (8).)

$$= 2 \int_{y=0}^{\infty} y^{-3} \left( \sum_{i=1}^{n-1} \Delta_{i,i+1} \cdot \# \{j > i, \Delta_{i,j} \leq y\} \right) dy$$

(Changing the order of integration and summation.)

$$= 2 \int_{y=0}^{\infty} y^{-3} \left( \sum_{i=1}^{n-1} \Delta_{i,i+1} \sum_{j=i+1}^n \mathbf{1} \{j > i, \Delta_{i,j} \leq y\} \right) dy$$

(Expanding  $\# \{\cdot\}$  into sum of  $\mathbf{1} \{\cdot\}$ .)

$$= 2 \int_{y=0}^{\infty} y^{-3} \left( \sum_{j=2}^n \sum_{i=1}^{j-1} \Delta_{i,i+1} \mathbf{1} \{j > i, \Delta_{i,j} \leq y\} \right) dy$$

(Changing the order of summation.) Recall from Definition 4 that for any  $j$  and  $y \geq 0$ ,  $i_y(j)$  is the least indexed expert  $i$  such that  $\Delta_{i,j}$  is still less than  $y$ . We get the following.

$$\begin{aligned} &= 2 \int_{y=0}^{\infty} y^{-3} \left( \sum_{j=2}^n \sum_{i=i_y(j)}^{j-1} \Delta_{i,i+1} \right) dy \\ &= 2 \int_{y=0}^{\infty} y^{-3} \left( \sum_{j=2}^n (\mu_{i_y(j)} - \mu_j) \right) dy \\ &= 2 \sum_{j=2}^n \int_{y=0}^{\infty} y^{-3} (\mu_{i_y(j)} - \mu_j) dy \end{aligned}$$

(Changing the order of summation and integration.)

$$= 2 \sum_{j=2}^n \int_{y=\Delta_{j-1,j}}^{\infty} y^{-3} (\mu_{i_y(j)} - \mu_j) dy \quad (10)$$

(This is because for values of  $y$  less than  $\Delta_{j-1,j}$ ,  $i_y(j) = j$  and integrand is equal to zero.)

$$\leq 2 \sum_{j=2}^n \int_{y=\Delta_{j-1,j}}^{\infty} y^{-3} \cdot y dy$$

(Since  $\mu_{i_y(j)} - \mu_j \leq y$ .)

$$\begin{aligned} &= 2 \sum_{j=2}^n \int_{y=\Delta_{j-1,j}}^{\infty} y^{-2} dy \\ &= 2 \sum_{j=2}^n \Delta_{j-1,j}^{-1} \end{aligned} \quad (11)$$

This concludes the proof of the lemma.  $\blacksquare$

**Remarks for small  $\Delta_{i,i+1}$**  Note that the upper bound stated in Theorem 1 become very large when  $\Delta_{i,i+1}$  is very small for some  $i$ . Indeed, when mean payoffs of all experts are equal,  $\Delta_{i,i+1} = 0$  for all  $i$  and upper bound becomes trivial, while the algorithm does well (picking any expert is as good as any other). We suggest a slight modification of the proof to take care of such case.

Let  $\epsilon > 0$  be fixed (the original theorem corresponds to the case  $\epsilon = 0$ ). Recall the definition of  $i_\epsilon(j)$  from Definition 4. We also define the inverse,  $j_\epsilon(i)$  as the maximum numbered expert  $j$  such that  $\Delta_{i,j}$  is no more than  $\epsilon$ , i.e.,  $j_\epsilon(i) = \arg \max \{j : j \geq i, \Delta_{i,j} \leq \epsilon\}$ . Note that the three conditions: (1)  $i < i_\epsilon(j)$ , (2)  $j > j_\epsilon(i)$ , and (3)  $\Delta_{i,j} > \epsilon$  are equivalent. The idea in this new analysis is to “identify” experts that have means within  $\epsilon$  of each other. (We cannot just make equivalence classes based on this, since the relation of “being within  $\epsilon$  of each other” is not an equivalence relation.)

Lemma 2 can be modified to prove that the regret of the algorithm is bounded by

$$2\epsilon T + \sum_{\substack{1 \leq i < j \leq n, \\ \Delta_{i,j} > \epsilon}} \frac{8}{\Delta_{i,j}^2} (\Delta_{i,i+1} + \Delta_{j-1,j}).$$

This can be seen by rewriting Equation (5) as

$$\begin{aligned} &\mathbb{E} \left[ \sum_{t=1}^T \sum_{j=2}^n \mathbf{1} \left\{ \mathcal{E}_{i_t^*,j}^{(1)}(t) \right\} \sum_{i=i_t^*}^{i_\epsilon(j)-1} \Delta_{i,i+1} \right] \\ &+ \mathbb{E} \left[ \sum_{t=1}^T \sum_{j=2}^n \mathbf{1} \left\{ \mathcal{E}_{i_t^*,j}^{(1)}(t) \right\} \sum_{i=i_\epsilon(j)}^{j-1} \Delta_{i,i+1} \right] \end{aligned}$$

and noting that the second term is at most

$$\mathbb{E} \left[ \sum_{t=1}^T \sum_{j=2}^n \mathbf{1} \left\{ \mathcal{E}_{i_t^*,j}^{(1)}(t) \right\} \epsilon \right] = \mathbb{E} \left[ \epsilon \sum_{t=1}^T \mathbf{1} \right] = \epsilon T,$$

since only one of the events  $\mathcal{E}_{i_t^*,j}^{(1)}(t)$  (corresponding to  $j = x_t$ ) can occur for each  $t$ . Equation (6) can be similarly modified by splitting the summation  $j = i + 1 \dots x_t$  to  $j = i + 1 \dots j_\epsilon(i)$  and  $j = j_\epsilon(i) + 1 \dots x_t$ .

Similarly, Lemma 3 can be modified as follows. In equation (7), instead of rewriting  $\sum_{j:j>i} \Delta_{i,j}^{-2}$ , we rewrite

$$\sum_{j:j>i, i < i_\epsilon(j)} \Delta_{i,j}^{-2}$$

to get

$$2 \int_{y=0}^{\infty} \# \{j > i, \epsilon < \Delta_{i,j} \leq y\} y^{-3} dy,$$

in Equation (8).

Equation (9) can be rewritten as

$$\sum_{j=1}^n \sum_{i=1}^{i_\epsilon(j)-1} \Delta_{i,j}^{-2} \Delta_{i,i+1}.$$

The rest of the analysis goes through as it is written, except that the limits of integration in Equation (10) now become  $y = \max\{\epsilon, \Delta_{j-1,j}\} \dots \infty$  instead of  $y = \Delta_{j-1,j} \dots \infty$ , resulting in the final expression of

$$2 \sum_{j=2}^n (\max\{\epsilon, \Delta_{j-1,j}\})^{-1},$$

in Equation (11).

Therefore, the denominators of regret expression in Theorem 1 can be made at least  $\epsilon$ , if we are willing to pay  $2\epsilon T$  upfront in terms of regret.

### 3.1.2 Lower bound

In this section, assuming that the means  $\mu_i$  are bounded away from 0 and 1, we prove that in terms of the regret, the FTAL algorithm presented in the section above is optimal (up to constant factors). This is done by showing the following lower bound on the regret guarantee of any algorithm.

**Lemma 5** Assume that the means  $\mu_i$  are bounded away from 0 and 1. Any algorithm for the stochastic version of the best expert problem must have regret at least

$$\Omega \left( \sum_{i=1}^{n-1} \frac{1}{\Delta_{i,i+1}} \right),$$

as  $T$  becomes large enough.

To prove this lemma, we first prove its special case for the case of two experts.

**Lemma 6** *Suppose we are given two numbers  $\mu_1 > \mu_2$ , both lying in an interval  $[a, b]$  such that  $0 < a < b < 1$ , and suppose we are given any online algorithm  $\phi$  for the best expert problem with two experts. Then there is an input instance in the stochastic rewards model, with two experts  $L$  and  $R$  whose payoff distributions are Bernoulli random variables with means  $\mu_1$  and  $\mu_2$  or vice-versa, such that for large enough  $T$ , the regret of algorithm  $\phi$  is*

$$\Omega(\delta^{-1}),$$

where  $\delta = \mu_1 - \mu_2$  and the constants inside the  $\Omega(\cdot)$  may depend on  $a, b$ .

**Proof:** Let us define some joint distributions:  $p$  is the distribution in which both experts have average payoff  $\mu_1$ ,  $q_L$  is the distribution in which they have payoffs  $(\mu_1, \mu_2)$  (left is better), and  $q_R$  is the distribution in which they have payoffs  $(\mu_2, \mu_1)$  (right expert is better).

Let us define the following events:  $E_t^L$  is true if  $\phi$  picks  $L$  at time  $t$ , and similarly  $E_t^R$ .

We denote by  $p^t(\cdot)$  the joint distribution for first  $t$  time steps, where the distribution of rewards in each time period is  $p(\cdot)$ . Similarly for  $q^t(\cdot)$ . We have  $p^t[E_t^L] + p^t[E_t^R] = 1$ . Therefore, for every  $t$ , there exists  $M \in \{L, R\}$  such that  $p^t[E_t^M] \geq 1/2$ . Similarly, there exists  $M \in \{L, R\}$  such that

$$\# \left\{ t : 1 \leq t \leq T, \quad p^t[E_t^M] \geq \frac{1}{2} \right\} \geq \frac{T}{2}.$$

Take  $T_0 = \frac{c}{\delta^2}$  for a small enough constant  $c$ . We will prove the claim below for  $T = T_0$ ; for larger values of  $T$ , the claim follows easily from this.

Without loss of generality, assume that  $M = L$ . Now assume the algorithm faces the input distribution  $q_R$ , and define  $q = q_R$ . Using  $\text{KL}(\cdot; \cdot)$  to denote the KL-divergence of two distributions, we have

$$\begin{aligned} \text{KL}(p^t; q^t) &\leq \text{KL}(p^T; q^T) = T \cdot \text{KL}(p; q) \\ &= c\delta^{-2} \cdot \text{KL}(\mu_1; \mu_2) \leq c\delta^{-2} \cdot \mathcal{O}(\delta^2) \leq \frac{1}{50}, \end{aligned}$$

for a small enough value of  $c$  which depends on  $a$  and  $b$  because the constant inside the  $\mathcal{O}(\cdot)$  in the line above depends on  $a$  and  $b$ .

Karp and Kleinberg [KK07] prove the following lemma. If there is an event  $E$  with  $p(E) \geq 1/3$  and  $q(E) < 1/3$ , then

$$\text{KL}(p; q) \geq \frac{1}{3} \ln \left( \frac{1}{3q(E)} \right) - \frac{1}{e}. \quad (12)$$

We have that for at least  $T/2$  values of  $t$ ,  $p^t(E_t^L) \geq 1/3$  (it is actually at least  $1/2$ ). In such time steps, we either have  $q^t(E_t^L) \geq 1/3$  or the lemma applies, yielding

$$\frac{1}{50} \geq \text{KL}(p^t; q^t) \geq \frac{1}{3} \ln \left( \frac{1}{q^t(E_t^L)} \right) - \frac{1}{e}.$$

This gives

$$q^t(E_t^L) \geq \frac{1}{10}.$$

Therefore, the regret of the algorithm in time period  $t$  is at least

$$\mu_1 - \left( \frac{9}{10}\mu_1 + \frac{1}{10}\mu_2 \right) \geq \frac{1}{10}\delta.$$

Since  $T = \Omega(\delta^{-2})$ , we have that the regret is at least

$$\frac{1}{10}\delta \cdot \Omega(\delta^{-2}) = \Omega(\delta^{-1}).$$

This finishes the proof of the lower bound for two experts. ■

**Proof of Lemma 5:** Let us group experts in pairs of 2 as  $(2i - 1, 2i)$  for  $i = 1, 2, \dots, \lfloor n/2 \rfloor$ . Apply the two-expert lower bound from Lemma 6 by creating a series of time steps when  $A_t = \{2i - 1, 2i\}$  for each  $i$ . (We need a sufficiently large time horizon — namely  $T \geq \sum_{i=1}^{\lfloor n/2 \rfloor} c\Delta_{2i-1, 2i}^{-2}$  — in order to apply the lower bound to all  $\lfloor n/2 \rfloor$  two-expert instances.) The total regret suffered by any algorithm is the sum of regret suffered in the independent  $\lfloor n/2 \rfloor$  instances defined above. Using the lower bound from Lemma 6, we get that the regret suffered by any algorithm is at least

$$\sum_{i=1}^{\lfloor n/2 \rfloor} \Omega \left( \frac{1}{\Delta_{2i-1, 2i}} \right).$$

Similarly, if we group the experts in pairs according to  $(2i, 2i+1)$  for  $i = 1, 2, \dots, \lfloor n/2 \rfloor$ , then we get a lower bound of

$$\sum_{i=1}^{\lfloor n/2 \rfloor} \Omega \left( \frac{1}{\Delta_{2i, 2i+1}} \right).$$

Since both of these are lower bounds, so is their average, which is

$$\frac{1}{2} \sum_{i=1}^{n-1} \Omega \left( \frac{1}{\Delta_{i, i+1}} \right) = \Omega \left( \sum_{i=1}^{n-1} \Delta_{i, i+1}^{-1} \right).$$

This proves the lemma. ■

### 3.2 Multi-armed bandit setting

We now turn our attention to the multi-armed bandit setting against a stochastic adversary. We first present a variant of UCB1 algorithm [ACBF02], and then present a matching lower bound based on idea from Lai and Robbins [LR85], which is a constant factor away from the UCB1-like upper bound.

#### 3.2.1 Upper bound (algorithm: AUER)

Here the optimal algorithm is again a natural extension of the UCB1 algorithm [ACBF02] to the sleeping-bandits case. In a nutshell, the algorithm keeps track of the running average of payoffs received from each arm, and also a confidence interval of width  $2\sqrt{\frac{8 \ln t}{n_{j,t}}}$  around arm  $j$ , where  $t$  is the current time interval and  $n_{j,t}$  is the number of times  $j$ 's payoff has been observed (number of times arm  $j$  has been played). At

time  $t$ , if an arm becomes available for the first time then the algorithm chooses it. Otherwise the algorithm optimistically picks the arm with highest “upper estimated reward” (or “upper confidence bound” in UCB1 terminology) among the available arms. That is, it picks the arm  $j \in A_t$  with maximum  $\hat{\mu}_{j,t} + \sqrt{\frac{8 \ln t}{n_{j,t}}}$  where  $\hat{\mu}_{j,t}$  is the mean of the observed rewards of arm  $j$  up to time  $t$ . The algorithm is shown in Figure 2. The algorithm is called **Awake Upper Estimated Reward (AUER)**.

```

1 Initialize  $z_i = 0$  and  $n_i = 0$  for all  $i \in [n]$ .
2 for  $t = 1$  to  $T$  do
3   if  $\exists j \in A_t$  s.t.  $n_j = 0$  then
4     Play arm  $x(t) = j$ 
5   else
6     Play arm
7      $x(t) = \arg \max_{i \in A_t} \left( \frac{z_i}{n_i} + \sqrt{\frac{8 \log t}{n_i}} \right)$ 
8   end
9   Observe payoff  $r_{x(t)}(t)$  for arm  $x(t)$ 
10   $z_{x(t)} \leftarrow z_{x(t)} + r_{x(t)}(t)$ 
11   $n_{x(t)} \leftarrow n_{x(t)} + 1$ 
12 end

```

**Algorithm 2:** The AUER algorithm for sleeping bandit problem with stochastic adversary.

We first need to state a claim about the confidence intervals that we are using.

**Lemma 7** *With the definition of  $n_{i,t}$  and  $\mu_i$  and  $\hat{\mu}_i$ , the following holds for all  $1 \leq i \leq n$  and  $1 \leq t \leq T$ :*

$$\mathbb{P} \left[ \mu_i \notin \left[ \hat{\mu}_{i,t} - \sqrt{\frac{8 \ln t}{n_{i,t}}}, \hat{\mu}_{i,t} + \sqrt{\frac{8 \ln t}{n_{i,t}}} \right] \right] \leq \frac{1}{t^4}.$$

**Proof:** The proof is an application of Chernoff-Hoeffding bounds, and follows from [ACBF02, pp. 242–243]. ■

**Theorem 8** *The regret of the AUER algorithm is at most*

$$(64 \ln T) \cdot \sum_{j=1}^{n-1} \frac{1}{\Delta_{j,j+1}}.$$

up to time  $T$ .

The theorem follows immediately from the following lemma and Lemma 3.

**Lemma 9** *The AUER algorithm has a regret of at most*

$$(32 \ln T) \cdot \sum_{j=2}^n \sum_{i=1}^{j-1} \left( \frac{1}{\Delta_{i,j}^2} \right) \Delta_{i,i+1}$$

**Proof:** We bound the regret of the algorithm arm by arm. Let us consider an arm  $2 \leq j \leq n$ . Let us count the number of times  $j$  was played, where some arm in  $1, 2, \dots, i$  could have been played (in these iterations, the regret accumulated

is at least  $\Delta_{i,j}$  and at most  $\Delta_{1,j}$ ). Call this  $N_{i,j}$  for  $i < j$ . We claim that  $N_{i,j} \leq \frac{32 \ln T}{\Delta_{i,j}^2}$  with probability  $1 - \frac{2}{t^4}$ .

Let us define  $Q_{i,j} = \frac{32 \ln T}{\Delta_{i,j}^2}$ . We want to claim that after playing  $j$  for  $Q_{i,j}$  number of times, we will not make the mistake of choosing  $j$  instead of something from the set  $\{1, 2, \dots, i\}$ ; that is, if some arm in  $[i]$  is awake as well as  $j$  is awake, then some awake arm in  $[i]$  will be chosen, and not the arm  $j$  (with probability at least  $1 - \frac{2}{t^4}$ ).

Let us bound the probability of choosing  $j$  when  $A_t \cap [i] \neq \emptyset$  after  $j$  has been played  $Q_{i,j}$  number of times.

$$\begin{aligned} & \sum_{t=Q_{i,j}+1}^T \sum_{k=Q_{i,j}+1}^T \mathbb{P} \left[ (x_t = j) \wedge (j \text{ is played } k\text{-th time}) \right. \\ & \qquad \qquad \qquad \left. \wedge (A_t \cap [i] \neq \emptyset) \right] \\ & \leq \sum_{t=Q_{i,j}+1}^T \sum_{k=Q_{i,j}+1}^T \mathbb{P} \left[ (n_{j,t} = k) \right. \\ & \qquad \qquad \qquad \left. \wedge \left( \hat{\mu}_{j,t} + \sqrt{\frac{8 \ln t}{k}} \geq \hat{\mu}_{h_t,t} + \sqrt{\frac{8 \ln t}{n_{h_t,t}}} \right) \right], \end{aligned}$$

where  $h_t$  is the index  $g$  in  $A_t \cap [i]$  which maximizes  $\hat{\mu}_{g,t} + \sqrt{(8 \ln t)/n_{g,t}}$ , i.e.  $h = \arg \max_{g \in A_t} \hat{\mu}_{g,t} + \sqrt{(8 \ln t)/n_{g,t}}$

$$\begin{aligned} & = \sum_{t=Q_{i,j}+1}^T \sum_{k=Q_{i,j}+1}^T \mathcal{O} \left( \frac{1}{t^4} \right) + \mathbb{P} [\mu_j + \Delta_{i,j} \geq \mu_{h_t}] \\ & = \mathcal{O}(1). \end{aligned}$$

Here, the first  $\frac{1}{t^4}$  term comes from the probability that  $j$ 's confidence interval might be wrong, or  $h_t$ 's confidence interval might be wrong (it follows from Lemma 7). Since  $k > \frac{32 \ln t}{\Delta_{i,j}^2}$ ,  $j$ 's confidence interval is at most  $\Delta_{i,j}/2$  wide.

Therefore, with probability  $1 - \frac{2}{t^4}$ , we have  $\hat{\mu}_{j,t} + \sqrt{\frac{8 \ln t}{k}} \leq \mu_j + \Delta_{i,j}$  and  $\hat{\mu}_{h_t,t} + \sqrt{\frac{8 \ln t}{n_{h_t,t}}} \geq \mu_{h_t}$ . Also, the probability  $\mathbb{P}[\mu_j + \Delta_{i,j} \geq \mu_{h_t}] = 0$  since we know that  $\mu_j + \Delta_{i,j} \leq \mu_{h_t}$  as  $h_t \in [i]$ . Therefore, we can mess up only constant number of times between  $[i]$  and  $j$  after  $j$  has been played  $Q_{i,j}$  number of times. We get that

$$\mathbb{E}[N_{i,j}] \leq Q_{i,j} + \mathcal{O}(1).$$

Now, it is easy to bound the total regret of the algorithm, which is

$$\begin{aligned} & \mathbb{E} \left[ \sum_{j=2}^n \sum_{i=1}^{j-1} (N_{i,j} - N_{i-1,j}) \Delta_{i,j} \right] \tag{13} \\ & = \sum_{j=2}^n \sum_{i=1}^{j-1} N_{i,j} (\Delta_{i,j} - \Delta_{i+1,j}), \end{aligned}$$

which follows by regrouping of terms and the convention that  $N_{0,j} = 0$  and  $\Delta_{j,j} = 0$  for all  $j$ . Taking the expectation of this gives the regret bound of

$$(32 \ln T) \cdot \sum_{j=2}^n \sum_{i=1}^{j-1} \left( \frac{1}{\Delta_{i,j}^2} \right) (\Delta_{i,j} - \Delta_{i+1,j}).$$

This gives the statement of the lemma.  $\blacksquare$

**Remarks for small  $\Delta_{i,i+1}$**  As noted in the case of expert setting, the upper bound above become trivial if some  $\Delta_{i,i+1}$  are small. In such case, the proof can be modified by changing equation (13) as follows.

$$\begin{aligned}
& \sum_{j=2}^n \sum_{i=1}^{j-1} (N_{i,j} - N_{i-1,j}) \Delta_{i,j} \\
&= \sum_{j=2}^n \sum_{i=1}^{i_\epsilon(j)} (N_{i,j} - N_{i-1,j}) \Delta_{i,j} \\
&\quad + \sum_{j=2}^n \sum_{i=i_\epsilon(j)+1}^{j-1} (N_{i,j} - N_{i-1,j}) \Delta_{i,j} \\
&\leq \sum_{j=2}^n \sum_{i=1}^{i_\epsilon(j)-1} N_{i,j} \Delta_{i,i+1} + \sum_{j=2}^n N_{i_\epsilon(j),j} \Delta_{i_\epsilon(j),j} \\
&\quad + \sum_{j=2}^n \sum_{i=i_\epsilon(j)+1}^{j-1} (N_{i,j} - N_{i-1,j}) \epsilon \\
&\leq \sum_{j=2}^n \sum_{i=1}^{i_\epsilon(j)-1} N_{i,j} \Delta_{i,i+1} + \epsilon \sum_{j=2}^n N_{i_\epsilon(j),j} \\
&\quad + \epsilon \sum_{j=2}^n (N_{j-1,j} - N_{i_\epsilon(j),j}) \\
&\leq \sum_{1 \leq i < j \leq n, \Delta_{i,j} > \epsilon} N_{i,j} \Delta_{i,i+1} + \epsilon T,
\end{aligned}$$

where the last step follows from  $\sum_{j=2}^n N_{j-1,j} \leq T$ .

Taking the expectation, and using the modification of Lemma 3 suggested in Section 3.1.1 gives us an upper bound of

$$\epsilon T + (64 \ln T) \sum_{i=1}^{n-1} (\max\{\epsilon, \Delta_{i,i+1}\})^{-1},$$

for any  $\epsilon \geq 0$ .

### 3.2.2 Lower bound

In this section, we prove that the AUER algorithm presented is information theoretically optimal up to constant factors when the means of arms  $\mu_i$ 's are bounded away from 0 and 1. We do this by presenting a lower bound of

$$\Omega \left( \ln T \cdot \sum_{i=1}^{n-1} \Delta_{i,i+1}^{-1} \right)$$

for this problem. This is done by closely following the lower bound of Lai and Robbins [LR85] for two armed bandit problems. The difference is that Lai and Robbins prove their lower bound only in the case when  $T$  approaches  $\infty$ , but we want to get bounds that hold for finite  $T$ . Our main result is stated in the following lemma.

**Lemma 10** Suppose there are  $n$  arms and  $n$  Bernoulli distributions  $P_i$  with means  $\mu_i$ , with each  $\mu_i \in [\alpha, \beta]$  for some

$0 < \alpha < \beta < 1$ . Let  $\phi$  be an algorithm for picking among  $n$  arms which, up to time  $t$ , plays a suboptimal bandit at most  $o(t^a)$  number of times for every  $a > 0$ . Then, there is an input instance with  $n$  arms endowed with some permutation of above mentioned  $n$  distributions, such that the regret of  $\phi$  has to be at least

$$\Omega \left( \sum_{i=1}^{n-1} \frac{(\log t)(\mu_i - \mu_{i+1})}{\text{KL}(\mu_{i+1}; \mu_i)} \right),$$

for  $t \geq n^2$ .

We first prove the result for two arms. For this, in the following, we extend the Lai and Robbins result so that it holds (with somewhat worse constants) for finite  $T$ , rather than only in the limit  $T \rightarrow \infty$ .

**Lemma 11** Let there be two arms and two distributions  $P_1(\cdot)$  and  $P_2(\cdot)$  with means  $\mu_1$  and  $\mu_2$  with  $\mu_i \in [\alpha, \beta]$  for  $i = 1, 2$  and  $0 < \alpha < \beta < 1$ . Let  $\phi$  be any algorithm for choosing the arms which never picks the worse arm (for any values of  $\mu_1$  and  $\mu_2$  in  $[\alpha, \beta]$ ) more than  $o(T^a)$  times (for any value of  $a > 0$ ).

Then there exists an instance for  $\phi$  with two arms endowed with two distributions above (in some order) such that the regret of the algorithm if presented with this instance is at least

$$\Omega \left( \frac{(\log t)(\mu_1 - \mu_2)}{\text{KL}(\mu_2; \mu_1)} \right),$$

where the constant inside the big-omega is at least  $1/2$ .

**Proof:** Since we are proving a lower bound, we just focus on Bernoulli distributions, and prove that if we have two bandits, with Bernoulli payoffs with means  $\mu_1$  and  $\mu_2$  such that  $\alpha \leq \mu_2 < \mu_1 \leq \beta$ , then we can get the above mentioned lower bound.

Let us fix a  $\delta < 1/10$ . From the assumption that  $\mu_1$  and  $\mu_2$  are bounded away from 0 and 1, there exists a Bernoulli distribution with mean  $\lambda > \mu_1$  with

$$|\text{KL}(\mu_2; \lambda) - \text{KL}(\mu_2; \mu_1)| \leq \delta \cdot \text{KL}(\mu_2; \mu_1),$$

because of the continuity of KL divergence in its second argument.

This claim provides us with a Bernoulli distribution with mean  $\lambda$  and

$$\text{KL}(\mu_2; \lambda) \leq (1 + \delta) \text{KL}(\mu_2; \mu_1). \quad (14)$$

From now on, until the end of the proof, we work with the following two distributions on  $t$ -step histories:  $p$  is the distribution induced by Bernoulli arms with means  $(\mu_1, \mu_2)$ , and  $q$  is the distribution induced by Bernoulli arms with means  $(\mu_1, \lambda)$ . From the assumption of the lemma, we have

$$\mathbb{E}_q[t - n_{2,t}] \leq o(t^a), \quad \text{for all } a > 0.$$

We choose any  $a < \delta$ . By an application of Markov's inequality, we get that

$$\begin{aligned}
& \mathbb{P}_q[n_{2,t} < (1 - \delta)(\log t) / \text{KL}(\mu_2; \lambda)] \\
& \leq \frac{\mathbb{E}_q[t - n_{2,t}]}{t - (1 - \delta)(\log t) / \text{KL}(\mu_2; \lambda)} \leq o(t^{a-1}). \quad (15)
\end{aligned}$$

Let  $\mathcal{E}$  denote the event that  $n_{2,t} < (1-\delta) \log t / \text{KL}(\mu_2; \lambda)$ . If  $\mathbb{P}_p(\mathcal{E}) < 1/3$ , then

$$\begin{aligned} \mathbb{E}_p[n_{2,t}] &\geq \mathbb{P}_p(\bar{\mathcal{E}}) \cdot (1-\delta) \log t / \text{KL}(\mu_2, \lambda) \\ &\geq \frac{2}{3} \cdot (1-\delta) \log t / \text{KL}(\mu_2, \lambda) \\ &\geq \frac{2}{3} \left( \frac{1-\delta}{1+\delta} \frac{\log t}{\text{KL}(\mu_2; \mu_1)} \right), \end{aligned}$$

which implies the stated lower bound for  $\delta = 1/10$ .

Henceforth, we will assume  $\mathbb{P}_p(\mathcal{E}) \geq 1/3$ . We have  $\mathbb{P}_q(\mathcal{E}) < 1/3$  using (15). Now we can apply the lemma from [KK07] stated in (12), we have

$$\begin{aligned} \text{KL}(p; q) &\geq \frac{1}{3} \ln \left( \frac{1}{3 \mathcal{O}(t^{a-1})} \right) - \frac{1}{e} \\ &= (1-a) \ln t - \mathcal{O}(1). \end{aligned} \quad (16)$$

The chain rule for KL divergence [CT99, Theorem 2.5.3] implies

$$\text{KL}(p; q) = \mathbb{E}_p[n_{2,t}] \cdot \text{KL}(\mu_2; \lambda) \quad (17)$$

Combining (16) with (17), we get

$$\begin{aligned} \mathbb{E}_{\mu_1, \mu_2}[n_{2,t}] &\geq \frac{(1-a) \ln t - \mathcal{O}(1)}{\text{KL}(\mu_2; \lambda)} \\ &\geq \frac{1-a}{1+\delta} \frac{\ln t}{\text{KL}(\mu_2; \mu_1)} - \mathcal{O}(1). \end{aligned} \quad (18)$$

Using  $a < \delta < 1/10$ , the regret bound follows.  $\blacksquare$

We now extend the result from 2 to  $n$  bandits.

**Proof of Lemma 10:** A naive way to extend the lower bound is to divide the time line between  $n/2$  blocks of length  $2T/n$  each and use  $n/2$  separate two-armed bandit lower bounded as done in the proof of Lemma 5.

We can pair the arms in pairs of  $(2i-1, 2i)$  for  $i = 1, 2, \dots, \lfloor n/2 \rfloor$ . We present the algorithm with two arms  $2i-1$  and  $2i$  in the  $i$ -th block of time. The lower bound then is

$$\begin{aligned} \log \left( \frac{T}{n} \right) &\left( \frac{\mu_1 - \mu_2}{\text{KL}(\mu_2; \mu_1)} + \dots + \frac{\mu_{2\lfloor n/2 \rfloor - 1} - \mu_{2\lfloor n/2 \rfloor}}{\text{KL}(\mu_{2\lfloor n/2 \rfloor}; \mu_{2\lfloor n/2 \rfloor - 1})} \right) \\ &= \Omega \left( (\log T) \cdot \left( \sum_{i=1}^{\lfloor n/2 \rfloor} \Delta_{2i, 2i-1}^{-1} \right) \right), \end{aligned}$$

if we take  $T > n^2$ . Using the fact that  $\mu_i \in [\alpha, \beta]$ , we have  $\text{KL}(\mu_i; \mu_j) = \mathcal{O}(\Delta_{i,j}^{-2})$  which justifies the derivation of the second line above.

We get a similar lower bound by presenting the algorithm with  $(2i, 2i+1)$ , which gives us a lower bound of

$$\Omega \left( (\log T) \cdot \left( \sum_{i=1}^{\lfloor n/2 \rfloor} \Delta_{2i, 2i+1}^{-1} \right) \right).$$

Taking their averages gives the required lower bound, proving the lemma.  $\blacksquare$

## 4 Adversarial Model of Rewards

We now turn our attention to the case where no distributional assumptions are made on the generation of rewards. In this section we prove information theoretic lower bounds on the regret of any online learning algorithm for both the best expert and the multi-armed bandit settings. We also present online algorithms whose regret is within a constant factor of the lower bound for the expert setting and within a sublogarithmic factor of the lower bound for the bandit setting. Unlike in the stochastic rewards setting, however, these algorithms are not computationally efficient. It is an open problem if there exists an efficient algorithm whose regret grows as polynomial in  $n$ .

### 4.1 Best expert

**Theorem 12** *For every online algorithm ALG and every time horizon  $T$ , there is an adversary such that the algorithm's regret with respect to the best ordering, at time  $T$ , is*

$$\Omega(\sqrt{Tn \log(n)}).$$

**Proof:** We construct a randomized oblivious adversary (i.e., a distribution on input sequences) such that the regret of any algorithm ALG is at least  $\Omega(\sqrt{Tn \log(n)})$ . The adversary partitions the timeline  $\{1, 2, \dots, T\}$  into a series of *two-expert games*, i.e. intervals of consecutive rounds during which only two experts are awake and all the rest are asleep. In total there will be  $Q(n) = \Theta(n \log n)$  two-expert games, where  $Q(n)$  is a function to be specified later in (20). For  $i = 1, 2, \dots, Q(n)$ , the set of awake experts throughout the  $i$ -th two-experts game is a pair  $A^{(i)} = \{x_i, y_i\}$ , determined by the adversary based on the (random) outcomes of previous two-experts games. The precise rule for determining the elements of  $A^{(i)}$  will be explained later in the proof.

Each two-experts game runs for  $T_0 = T/Q(n)$  rounds, and the payoff functions for the rounds are independent, random bijections from  $A^{(i)}$  to  $\{0, 1\}$ . Letting  $g^{(i)}(x_i)$ ,  $g^{(i)}(y_i)$  denote the payoffs of  $x_i$  and  $y_i$ , respectively, during the two-experts game, it follows from Khintchine's inequality [KHi23] that

$$\mathbb{E} \left( \left| g^{(i)}(x_i) - g^{(i)}(y_i) \right| \right) = \Omega \left( \sqrt{T_0} \right). \quad (19)$$

The expected payoff for any algorithm can be at most  $\frac{T_0}{2}$ , so for each two-experts game the regret of any algorithm is at least  $\Omega(\sqrt{T_0})$ . For each two-experts game we define the *winner*  $W_i$  to be the element of  $\{x_i, y_i\}$  with the higher payoff in the two-experts game; we will adopt the convention that  $W_i = x_i$  in case of a tie. The *loser*  $L_i$  is the element of  $\{x_i, y_i\}$  which is not the winner.

The adversary recursively constructs a sequence of  $Q(n)$  two-experts games and an ordering of the experts such that the winner of every two-experts game precedes the loser in this ordering. (We call such an ordering *consistent* with the sequence of games.) In describing the construction, we assume for convenience that  $n$  is a power of 2. If  $n = 2$  then we set  $Q(2) = 1$  and we have a single two-experts game and an ordering in which the winner precedes the loser. If  $n > 2$  then we recursively construct a sequence of games and an ordering consistent with those games, as follows:

1. We construct  $Q(n/2)$  games among the experts in the set  $\{1, 2, \dots, n/2\}$  and an ordering  $\prec_1$  consistent with those games.
2. We construct  $Q(n/2)$  games among the experts in the set  $\{(n/2) + 1, \dots, n\}$  and an ordering  $\prec_2$  consistent with those games.
3. Let  $k = 2Q(n/2)$ . For  $i = 1, 2, \dots, n/2$ , we define  $x_{k+i}$  and  $y_{k+i}$  to be the  $i$ -th elements in the orderings  $\prec_1, \prec_2$ , respectively. The  $(k+i)$ -th two-experts game uses the set  $A^{(k+i)} = \{x_{k+i}, y_{k+i}\}$ .
4. The ordering of the experts puts the winner of the game between  $x_{k+i}$  and  $y_{k+i}$  before the loser, for every  $i = 1, 2, \dots, n/2$ , and it puts both elements of  $A^{(k+i)}$  before both elements of  $A^{(k+i+1)}$ .

By construction, it is clear that the ordering of experts is consistent with the games, and that the number of games satisfies the recurrence

$$Q(n) = 2Q(n/2) + n/2, \quad (20)$$

whose solution is  $Q(n) = \Theta(n \log n)$ .

The best ordering of experts achieves a payoff at least as high as that achieved by the constructed ordering which is consistent with the games. By (19), the expected payoff of that ordering is  $T/2 + Q(n) \cdot \Omega(\sqrt{T_0})$ . The expected payoff of ALG in each round  $t$  is  $1/2$ , because the outcome of that round is independent of the outcomes of all prior rounds. Hence the expected payoff of ALG is only  $T/2$ , and its regret is

$$\begin{aligned} Q(n) \cdot \Omega(\sqrt{T_0}) &= \Omega(n \log n \sqrt{T/(n \log n)}) \\ &= \Omega(\sqrt{T n \log n}). \end{aligned}$$

This proves the theorem.  $\blacksquare$

It is interesting to note that the adversary that achieves this lower bound is not adaptive in either choosing the payoffs or choosing the awake experts at each time step. It only needs to be able to carefully coordinate which experts are awake based on the payoffs at previous time steps.

Even more interesting, this lower bound is tight, so an adaptive adversary is not more powerful than an oblivious one. There is a learning algorithm that achieves a regret of  $O(\sqrt{T n \log(n)})$ , albeit not computationally efficient. To achieve this regret we transform the sleeping experts problem to a problem with  $n!$  experts that are always awake. In the new problem, we have one expert for each ordering of the original  $n$  experts. At each round, each of the  $n!$  experts makes the same prediction as the highest ranked expert in its corresponding ordering, and receives the payoff of that expert.

**Theorem 13** *An algorithm that makes predictions using Hedge on the transformed problem achieves  $O(\sqrt{T n \log(n)})$  regret with respect to the best ordering.*

**Proof:** Every expert in the transformed problem receives the payoff of its corresponding ordering in the original problem. Since Hedge achieves regret  $O(\sqrt{T \log(n!)})$  with respect to the best expert in the transformed problem, the same regret is achieved by the algorithm in the original problem.  $\blacksquare$

## 4.2 Multi-armed bandit setting

**Theorem 14** *For every online algorithm ALG and every time horizon  $T$ , there is an adversary such that the algorithm's regret with respect to the best ordering, at time  $T$ , is  $\Omega(n\sqrt{T})$ .*

**Proof:** To prove the lower bound we will rely on the lower bound proof for the multi-armed bandit in the usual setting when all the experts are awake [ACBFS02]. In the usual bandit setting with a time horizon of  $T_0$ , any algorithm will have at least  $\Omega(\sqrt{T_0 n})$  regret with respect to the best expert. To ensure this regret, the input sequence is generated by sampling  $T_0$  times independently from a distribution in which every bandit but one receives a payoff of 1 with probability  $\frac{1}{2}$  and 0 otherwise. The remaining bandit, which is chosen at random, incurs a payoff of 1 with probability  $\frac{1}{2} + \epsilon$  for an appropriate choice of  $\epsilon$ .

To obtain the lower bound for the sleeping bandits setting we set up a sequence of  $n$  multi-armed bandit games as described above. Each game will run for  $T_0 = \frac{T}{n}$  rounds. The bandit that received the highest payoff during the game will become asleep and unavailable in the rest of the games.

In game  $i$ , any algorithm will have a regret of at least  $\Omega\left(\sqrt{\frac{T}{n}(n-i)}\right)$  with respect to the best bandit in that game. In consequence, the total regret of any learning algorithm with respect to the best ordering is:

$$\begin{aligned} \sum_{i=1}^{n-1} \sqrt{\frac{T}{n}(n-i)} &= \sqrt{\frac{T}{n}} \sum_{j=1}^{n-1} j^{1/2} \\ &\geq \sqrt{\frac{T}{n}} \int_{x=0}^{n-1} x^{1/2} dx = \sqrt{\frac{T}{n}} \frac{2}{3} \left((n-1)^{3/2}\right) \\ &= \Omega\left(n\sqrt{T}\right). \end{aligned}$$

The theorem follows.  $\blacksquare$

To get an upper bound on regret, we will use the Exp4 algorithm [ACBFS02]. Since Exp4 requires an oblivious adversary, in the following, we assume that the adversary is oblivious (as opposed to adaptive). Exp4 chooses an action by combining the advice of a set of “experts.” At each round, each expert provides advice in the form of a probability distribution over actions. In particular the advice can be a point distribution concentrated on a single action. (It is required that at least one of the experts is the *uniform expert* whose advice is always the uniform distribution over actions.) To use Exp4 for the sleeping experts setting, in addition to the uniform expert we have an expert for each ordering over actions. At each round, the advice of that expert is a point distribution concentrated on the highest ranked action in the corresponding ordering.

Since the uniform expert may advise us to pick actions which are not awake, we assume for convenience that the problem is modified as follows. Instead of being restricted to choose an action in the set  $A_t$  at time  $t$ , the algorithm is allowed to choose any action at all, with the proviso that the payoff of an action in the complement of  $A_t$  is defined to be 0. Note that any algorithm for this modified problem can easily be transformed into an algorithm for the original

problem: every time the algorithm chooses an action in the complement of  $A_t$  we instead play an arbitrary action in  $A_t$ . Such a transformation can only increase the algorithm's payoff, i.e. decrease the regret. Hence, to prove the regret bound asserted in Theorem 15 below, it suffices to prove the same bound for the modified problem.

**Theorem 15** *Against an oblivious adversary, the Exp4 algorithm as described above achieves a regret of  $O(n\sqrt{T\log(n)})$  with respect to the best ordering.*

**Proof:** We have  $n$  actions and  $1 + n!$  experts, so the regret of Exp4 with respect to the payoff of the best expert is  $O(\sqrt{Tn\log(n! + 1)})$  [ACBFS02]. Since the payoff of each expert is exactly the payoff of its corresponding ordering we obtain the statement of the theorem. ■

The upper bound and lower bound differ by a factor of  $O(\sqrt{\log(n)})$ . The same gap exists in the usual multi-armed bandit setting where all actions are available at all times, hence closing the logarithmic gap between the lower and upper bounds in Theorems 14 and 15 is likely to be as difficult as closing the corresponding gap for the nonstochastic multi-armed bandit problem itself.

## 5 Conclusions

We have analyzed algorithms for full-information and partial-information prediction problems in the “sleeping experts” setting, using a novel benchmark which compares the algorithm's payoff against the best payoff obtainable by selecting available actions using a fixed total ordering of the actions. We have presented algorithms whose regret is information-theoretically optimal in both the stochastic and adversarial cases. In the stochastic case, our algorithms are simple and computationally efficient. In the adversarial case, the most important open question is whether there is a computationally efficient algorithm which matches (or nearly matches) the regret bounds achieved by the exponential-time algorithms presented here.

## References

- [ACBF02] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.
- [ACBFS02] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002.
- [Azu67] K. Azuma. Weighted sums of certain dependent random variables. *Tohoku Math. J.*, 19:357–367, 1967.
- [BM05] Avrim Blum and Yishay Mansour. From external to internal regret. In *COLT*, pages 621–636, 2005.
- [CBFH<sup>+</sup>97] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997.
- [CT99] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. J. Wiley, 1999.
- [FSSW97] Yoav Freund, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. Using and combining predictors that specialize. In *STOC*, pages 334–343, 1997.
- [Han57] J. Hannan. Approximation to Bayes risk in repeated plays. volume 3, pages 97–139, 1957. in: M. Dresher, A. Tucker, P. Wolfe (Eds.), *Contributions to the Theory of Games*, Princeton University Press.
- [Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. American Stat. Assoc.*, 58:13–30, 1963.
- [Khi23] Aleksandr Khintchine. Über dyadische Brüche. *Math Z.*, 18:109–116, 1923.
- [KK07] Richard M. Karp and Robert Kleinberg. Noisy binary search and its applications. In *SODA*, pages 881–890, 2007.
- [KV05] Adam Tauman Kalai and Santosh Vempala. Efficient algorithms for on-line optimization. *J. Computer and System Sciences*, 71(3):291–307, 2005.
- [LR85] T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocations rules. *Adv. in Appl. Math.*, 6:4–22, 1985.
- [LW94] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994. An extended abstract appeared in *IEEE Symposium on Foundations of Computer Science*, 1989, pp. 256–261.
- [LZ07] John Langford and Tong Zhang. The epoch-greedy algorithm for multiarmed bandits with side information. In *NIPS*, 2007.
- [Rob] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535.
- [Vov90] V. G. Vovk. Aggregating strategies. In *COLT*, pages 371–386, 1990.
- [Vov98] V. G. Vovk. A game of prediction with expert advice. *J. Comput. Syst. Sci.*, 56(2):153–173, 1998. An extended abstract appeared in *COLT* 1995, pp. 51–60.

---

# Optimal Strategies from Random Walks

---

**Jacob Abernethy\***  
Division of Computer Science  
UC Berkeley  
jake@cs.berkeley.edu

**Manfred K. Warmuth†**  
Department of Computer Science  
UC Santa Cruz  
manfred@cse.ucsc.edu

**Joel Yellin**  
Division of Physical and  
Biological Sciences  
UC Santa Cruz  
yellin@soe.ucsc.edu

## Abstract

We analyze a sequential game between a Gambler and a Casino. The Gambler allocates bets from a limited budget over a fixed menu of gambling events that are offered at equal time intervals, and the Casino chooses a binary loss outcome for each of the events. We derive the optimal min-max strategies for both participants. We then prove that the minimum cumulative loss of the Gambler, assuming optimal play by the Casino, is exactly a well-known combinatorial quantity: the expected number of draws needed to complete a multiple set of “cards” in the generalized Coupon Collector’s Problem. We show that this quantity and the optimal strategy of the Gambler can be efficiently estimated from a simple random walk.

## 1 Introduction

This paper analyzes the problem of sequential prediction and decision making from the perspective of a two player game. The game is played by a learner, called here the Gambler, who makes a sequence of betting decisions. The Gambler’s opponent is the Casino in which he plays.

### Gambler vs. Casino:

1. On each day, the Gambler arrives at the Casino with \$1. The Casino presents  $n$  events and each event is played once per day. The Gambler chooses a distribution vector  $\mathbf{w} \in [0, 1]^n$ , where  $\sum w_i = 1$ , and bets the portion  $w_i$  of his \$1 budget on event  $i$ .
2. On each day the Casino determines the outcome of each event with the objective of winning as much money from the Gambler as possible. In particular, after observing the distribution of the Gambler’s bets the Casino decides between a *loss* or a *no loss* for all daily events. These choices are summarized by a *loss vector*  $\ell \in \{1, 0\}^n$  where  $\ell_i = 1$  implies that on event  $i$ , the Gambler lost. (For simplicity, we assume the only relevant quantities are losses. By shifting our baseline we can model *wins* as *non-losses*).

3. At the end of each day, the Gambler leaves the Casino having lost  $\mathbf{w} \cdot \ell = \sum_i w_i \ell_i$  and the cumulative loss of the gambler is updated as  $L \leftarrow L + \mathbf{w} \cdot \ell$ . The Gambler also monitors the cumulative performance of each event with a state vector  $\mathbf{s} \in \mathbb{N}^n$ , where  $s_i$  is the current total loss of event  $i$ . After incurring loss  $\ell$  at the current day, the state vector is updated to  $\mathbf{s} \leftarrow \mathbf{s} + \ell$ .
4. The Gambler stops playing as soon as he observes that each even has suffered more than  $k$  losses, where  $k$  is some fixed positive integer known to both. The Casino is aware of this decision and behaves accordingly.

Gambling against a casino may seem an unlikely starting point for a model of sequential decision making – we generally consider the typical environment for learning to be *stochastic* rather than *adversarial*. Yet are these two environments necessarily incompatible? Among the objectives of this paper is to address questions such as: “What will be the Gambler’s worst-case cumulative loss?”; and “What is the optimal betting strategy?” These questions, while clearly game-theoretic, are ultimately answered here by considering a *randomized* Casino rather than an adversarial one. From this perspective, randomness may indeed be the Gambler’s worst adversary.

Early work on sequential decision making focused on the problem of predicting a binary outcome given advice from a set of  $n$  experts. In that setting, the goal of the predictor is to combine the predictions of the experts to make his own prediction, with the objective of performing well, in hindsight, compared to the best expert. The performance of both the learner and the experts is measured by a loss function that compares predictions to outcomes. One of the early algorithms, the Weighted Majority algorithm [LW94], utilizes a distribution corresponding to the degree of *trust* in each expert.

It was observed by Freund and Schapire [FS97] that the analysis of the Weighted Majority algorithm can be applied to the so-called *hedge setting*. Rather than predict a binary outcome, the learner now plays some distribution over the experts on every round, a loss value is assigned to each expert independently, and the learner suffers the expected loss according to his chosen distribution. In this case, the learner bears the exact burden of the Gambler - that of “hedging” his bets so as to minimize his cumulative loss. To emphasize that the Gambler/Casino game is useful for settings other than prediction, we use the term “event” rather than “expert”.

---

\*Supported by DARPA grant FA8750-05-2-0249 and NSF grant DMS-0707060.

†Supported by NSF grant CCR 9821087.

A central theme of much of the sequential decision making literature is the use of so-called “exponential weights” to determine the learner’s distribution on each round. Use of the exponential weighting scheme in the case of the Casino game results in the following strategy for the Gambler: At a state  $\mathbf{s}$ , bet

$$w_i = \frac{\beta^{s_i}}{\sum_j \beta^{s_j}} \quad \text{on event } i, \quad (1)$$

where the factor  $\beta$  lies in  $[0, 1)$ .

From the analysis of the Weighted Majority algorithm it follows that the cumulative loss of the Gambler using the above strategy is bounded by

$$\frac{\ln n + k \ln \frac{1}{\beta}}{1 - \beta}.$$

Under the assumption that the loss of the best event is at most  $k$ , the factor  $\beta$  can be tuned [FS97] so that the above bound becomes

$$k + \sqrt{2k \ln n} + \ln n.$$

The exponential weights framework, as well as other on-line learning techniques, can be motivated using the method of relative entropy regularization [KW99]. While the resulting algorithms are elegant and in some cases can be shown to be asymptotically optimal [CBFHW96], they do not optimally solve the underlying game. Some improvements have been made using, for example, binomial weights that lead to slightly better but still non-optimal solutions [CBFHW96] in a setting where the experts must produce a prediction. While it is formally easy to define the optimal algorithms using minimax expressions, it has generally been assumed that actually computing an efficient solution is quite challenging [CBFH<sup>+</sup>97]. More recently, however, a minimax result [ALW07] was obtained for the specific game of prediction with absolute loss. The resulting algorithm, Binning, is efficient and optimal in a slightly relaxed setting.

In this paper we show that the minimax solution to the Gambler/Casino prediction game, which is identical to the underlying game of the hedge setting with binary losses, can be obtained efficiently. In addition, the game can be fully analyzed using a simple Markov process: a random walk on an  $n$ -dimensional lattice. The value of the game, that is the cumulative loss of an optimal Gambler, can be interpreted as the expected length of such a random walk. The Gambler’s optimal play, the portion of his budget he should bet on a given event, can similarly be interpreted as manifesting an assessment of the probability of a specific random outcome of this walk.

The game’s stopping criterion, that is when all events have lost at least  $k + 1$  times, may seem unusual at first yet fits quite naturally within the experts framework. Indeed, online learning bounds are often tuned with an explicit a priori knowledge of the cumulative loss of the best expert, which here would be  $k^1$ . While perhaps not realistic in prac-

<sup>1</sup>Strictly speaking, in the expert setting it is assumed that at least one expert has not crossed the  $k$ -mistake threshold, while here we stop the Gambler/Casino game when the loss of the *last* expert/event goes beyond this threshold. It is easy to show that this slight modification, made for convenience, increases the worst-case loss of the Gambler by exactly 1.

tice,  $k$  can be estimated and various techniques such as successive doubling can be used to obtain near-optimal bounds [CBFH<sup>+</sup>97].

The paper is structured as follows. In Section 2 we give a minimax definition of the optimal value of the game considered here. In Section 2.1 we modify the game by restricting the adversary’s choices to unit loss vectors. In Section 3, we then turn our attention to a specific Markov process with a number of relevant properties. We apply this randomized approach to the Casino game in Section 4, where we prove our main results. In Section 5 we give recurrences and exact formulas, based on sums over multinomials, for the value of the game and for the optimal probabilities. We set out an efficient method to compute both the optimal strategy of the Gambler and the value of the game. In Section 6 we compare the optimal regret bound to previous results, and in Section 7 we draw a connection between our game and a well studied version of the coupon collector problem. We also briefly summarize what is known about the asymptotics of this problem. We conclude with a discussion of our results and list open problems (Section 8).

## 2 The Value of the Game

Assume that in each event the Gambler has already suffered some losses specified by state vector  $\mathbf{s}$ . Define  $V(\mathbf{s})$  to be the total money lost by an optimal Gambler playing against an optimal Casino *starting from the state  $\mathbf{s}$* . That is,  $V(\mathbf{s})$  is the amount of money that the optimal Gambler will lose (against an optimal Casino) from now until the end of the game. Roughly speaking, the value of the game is computed as:

$$V(\mathbf{s}) \stackrel{?}{=} \min_{\text{dist. } \mathbf{w}} \max_{\ell \in \{0,1\}^n} \mathbf{w} \cdot \ell + V(\mathbf{s} + \ell)$$

The Gambler chooses  $\mathbf{w}$  to minimize the loss while the Casino chooses  $\ell$  to maximize the loss, where the loss is computed as the loss  $\mathbf{w} \cdot \ell$  on this round plus the worst case loss  $V(\mathbf{s} + \ell)$  on future rounds. However, we have to be careful, as this recursive definition doesn’t address the following issues:

- When is the game over? What is the base case of  $V(\cdot)$ ?
- Is this recursion bounded?
- Do we need to record the losses  $s_i$  that go above  $k$ ?

We address these issues beginning with some simplifications and notational conventions. First, we assume that the state vector  $\mathbf{s}$  lies within the set  $\mathcal{S} = \{0, 1, \dots, k+1\}^n$ . Note that it is not necessary to record the losses of events that have already crossed the  $k$  threshold. We call such events *dead*. Since the losses of dead events are not restricted, having loss  $k + 3$  is the same as loss  $k + 100$ . We therefore “round” all states  $\mathbf{s}$  into the state space  $\{0, 1, \dots, k+1\}^n$  using the notation  $\dagger$  which we define below. We use the notation  $\lambda(\mathbf{s})$  to record the set of live events; the statement  $i \notin \lambda(\mathbf{s})$  is exactly the statement  $s_i = k + 1$ .

Second, as the game is defined recursively, we must guarantee that this recursion terminates. If the Casino repeatedly chose  $\ell = \langle 0, \dots, 0 \rangle$ , for example, the game would make no progress. The same problem occurs if the Casino causes

losses on only dead events. We must therefore place additional restrictions so that the dead state is reached eventually. The simplest way to ensure this is to forbid the Casino from inflicting loss on only dead events. Yet this is not sufficient: with this restriction alone the Gambler would have a guaranteed non-losing strategy by betting solely on dead events. We thus assume that neither can the Casino can inflict losses on dead events nor can the Gambler bet money on them (keeping in mind that all such bets are in any case non-optimal). We must enforce this explicitly in order to have a well-defined game.

We use two notational conventions to describe the above restrictions. First, we write  $\mathbf{w} \sim \lambda(\mathbf{s})$  to describe the set  $\{\mathbf{w} \in \Delta_n \mid w_i = 0 \forall i \notin \lambda(\mathbf{s})\}$  where  $\Delta_n$  is the  $n$ -simplex. We also abuse notation slightly and write  $\ell \subset \lambda(\mathbf{s})$  to mean that  $\ell \in \{0, 1\}^n$  and  $\ell_i = 0$  for all  $i \notin \lambda(\mathbf{s})$ .

We now define the value of the game precisely.

**Definition 1** Define the value  $V(\mathbf{s})$  of the game as follows.

- At the dead state,  $V(\mathbf{d}) := 0$ .
- For any other  $\mathbf{s} \in \mathcal{S}$ , we define  $V(\mathbf{s})$  recursively as

$$V(\mathbf{s}) := \min_{\mathbf{w} \sim \lambda(\mathbf{s})} \max_{0 \neq \ell \subset \lambda(\mathbf{s})} \mathbf{w} \cdot \ell + V(\mathbf{s} + \ell). \quad (2)$$

In our notation, we commonly make use of several special states. The state where the game begins is the “initial” state,  $\mathbf{s} = \mathbf{0}$ . Once all events have lost more than  $k$  times the game is over and we refer to this as the *dead* state  $\mathbf{d}$ . It will also be useful to consider *one-live* states  $\mathbf{o}_i$ , where all events except  $i$  are dead, and the remaining event has exactly  $k$  losses. By the game definition, it is easy to check that  $V(\mathbf{o}_i) = 1$ , since the Gambler must bet all of his money on this event, and the Casino must inflict a corresponding loss, charging the Gambler \$1 and ending the game.

Below, we include a list of notations for reference:

**Notation:**

$\mathcal{S} := \{0, \dots, k + 1\}^n$	(the state space)
$\mathbf{0} := \langle 0, 0, \dots, 0 \rangle = \langle 0^n \rangle$	(the initial state)
$\mathbf{d} := \langle (k + 1)^n \rangle$	(the dead state)
$\mathbf{o}_i := \mathbf{d} - \mathbf{e}_i$	( $i$ th one-live state)
$\lambda(\mathbf{s}) := \{i \in [n] : s_i \leq k\}$	(set of live events)
$\mathbf{s} \dot{+} \ell := \langle \min(s_i + \ell_i, k + 1) \rangle$	(“rounded” addition)
$ \mathbf{s}  := \sum s_i$	(elementwise sum)
$\Delta_n := \{\mathbf{w} \in \mathbf{R}_+^n :  \mathbf{w}  = 1\}$	(the $n$ -simplex)

### 2.1 The Modified Game

We also consider a modified game that we make easier for the Gambler. In this new game, we restrict the Casino to inflict loss on exactly *one* event in each round, i.e.  $\ell$  must be a basis vector  $\mathbf{e}_1, \dots, \mathbf{e}_n$ . So for  $\ell = \mathbf{e}_i$  we have  $\mathbf{w} \cdot \ell = w_i$ . We can then precisely define the value  $\widehat{V}(\cdot)$  of the modified game:

**Definition 2** Define  $\widehat{V}(\mathbf{d}) := V(\mathbf{d}) = 0$ . Otherwise

$$\widehat{V}(\mathbf{s}) := \min_{\mathbf{w} \sim \lambda(\mathbf{s})} \max_{i \in \lambda(\mathbf{s})} w_i + \widehat{V}(\mathbf{s} + \mathbf{e}_i). \quad (3)$$

One of the central results of this paper is that the above game, while seemingly more restricted, is ultimately just as difficult for the Gambler as the original game. It is easy to show that  $V(\mathbf{s}) \geq \widehat{V}(\mathbf{s})$ , since the Casino has strictly more choices in the original game. We go further and prove as our main result in Theorem 12 that

$$V(\mathbf{s}) = \widehat{V}(\mathbf{s}).$$

Thus both games have the same worst-case outcome.

Both the analysis of the modified game, as well as the proof of the above result, requires a different formulation of the Casino’s actions.

## 3 A Randomized Casino

In Section 2 we presented a game-theoretic analysis of a well-known sequential prediction problem characterized as a game between a Gambler and a Casino. In the present section, we consider a different framework, in which the Casino uses random events. We will show that introducing a randomized strategy of the Casino enables us to specify the optimal strategy of the Gambler.

### 3.1 A Random Walk on the State Graph

Let us now imagine that our Casino does not fix outcomes deterministically, but instead chooses the outcome of each event using the following random process. Assume we are at state  $\mathbf{s}$  and that, on each day, an event  $i$  is chosen uniformly at random from  $\{1, \dots, n\}$  and a loss is assigned to event  $i$ . In other words, the loss vector  $\ell$  is a uniformly sampled unit vector  $\mathbf{e}_i$ , and after the loss the new state is  $\mathbf{s} + \mathbf{e}_i$ . This process continues until we reach the dead state  $\mathbf{d}$ .

We can model this behavior as a Markov process on the state space as follows. Consider any sequence of indices  $I_1, I_2, \dots \in [n]$ , and let  $S_t := \sum_{m=1}^t \mathbf{e}_{I_m}$ , where  $S_0 := \mathbf{0}$ . Assuming that we start at state  $\mathbf{s}$ , this induces a sequence of states

$$\mathbf{s} = \mathbf{s} \dot{+} S_0 \rightarrow \mathbf{s} \dot{+} S_1 \rightarrow \mathbf{s} \dot{+} S_2 \rightarrow \dots \rightarrow \mathbf{s} \dot{+} S_t.$$

Notice that this process has “self-loops”; i.e. it is quite possible that  $\mathbf{s} \dot{+} S_t = \mathbf{s} \dot{+} S_{t+1}$ . This occurs when  $(\mathbf{s} \dot{+} S_t)_{I_{t+1}}$  is already at  $k + 1$ .

If we imagine the state space  $\mathcal{S}$  as an  $n$ -dimensional lattice, which we will call the *state lattice*, then the Markov process above can be interpreted as a random walk on this lattice. The walker starts at the initial state  $\mathbf{0}$ , and on every time interval a positively directed single step is taken along an axis drawn uniformly at random. If the walker has already reached the  $k + 1$  boundary in this dimension, he remains in place. The walk stops once the dead state  $\mathbf{d}$  is reached. We will show that the value  $V$  is  $1/n$  times the expected total number of random draws that achieves this position. Thus  $V$  is the expected walk/path length from  $\mathbf{s}$  to  $\mathbf{d}$ .

### 3.2 Survival Probabilities

We now define a *survival probability* at a state  $\mathbf{s}$ . We will show in the next section that such probabilities are the basis for the Gambler’s optimal strategy.

**Definition 3** Assume we are at state  $\mathbf{s}$ , and let the random state  $\mathbf{s} \dot{+} S_t$  be the result of the above random walk after  $t$

steps. Define the  $i$ th survival probability  $\widehat{p}_i(\mathbf{s})$  to be the probability that

$$\exists t : \mathbf{s} \dot{+} S_t = \mathbf{o}_i.$$

Equivalently,

$$\widehat{p}_i(\mathbf{s}) = \Pr(\lambda(\mathbf{s} \dot{+} S_t) = \{i\} \text{ for some } t).$$

We call these survival probabilities since  $\widehat{p}_i(\mathbf{s})$  is the probability that, if the losses were assigned randomly to the events in sequence, the  $i$ th event would be the last non-dead event.

**Lemma 4** For any  $\mathbf{s} \neq \mathbf{d}$ , the vector

$$\widehat{\mathbf{p}}(\mathbf{s}) := \langle \widehat{p}_i(\mathbf{s}) \rangle_{i=1}^n$$

defines a distribution on  $\{1, \dots, n\}$ .

**Proof:** The quantity  $\sum_i \widehat{p}_i(\mathbf{s})$  is the probability that eventually there is exactly one live event. This probability is exactly 1, given that the current state is not the dead state  $\mathbf{d}$ . ■

We list some examples of survival probabilities:

- When  $\mathbf{s} = \mathbf{0}$  (or any other symmetric state), we have

$$\widehat{p}_i(\mathbf{s}) = \frac{1}{n}, \forall i$$

because there is a uniform chance of survival.

- When  $i$  is a dead event, i.e.  $s_i = k + 1$ , then

$$\widehat{p}_i(\mathbf{s}) = 0$$

because no dead event can be the last remaining live event.

- If there is only one remaining live event, i.e.  $\lambda(\mathbf{s}) = \{i\}$ , then

$$\widehat{p}_i(\mathbf{s}) = 1.$$

Computing  $\widehat{p}_i(\mathbf{s})$  for more general  $\mathbf{s}$  requires a recursion, and we leave this discussion for Section 5.

### 3.3 Expected Path Lengths

Another important quantity we consider is the *length of a random path*, i.e. the number of steps in the random walk on the state lattice required until the dead state  $\mathbf{d}$  is reached.

**Definition 5** For a sequence  $S_0, S_1, \dots$ , let

$$T(\mathbf{s}) := \min\{t \geq 0 : \mathbf{s} \dot{+} S_t = \mathbf{d}\}.$$

That is,  $T(\mathbf{s})$  is the length of the random path starting at  $\mathbf{s}$  and just entering  $\mathbf{d}$ . Furthermore, let

$$\tau(\mathbf{s}) := \mathbb{E} T(\mathbf{s})$$

be the expected path length.

We note that paths may be infinitely long due to self-loops, yet such paths occur with probability 0. A key fact is that the expected path length  $\tau(\mathbf{s})$  can be rewritten using indicator variables:

$$T(\mathbf{s}) = \sum_{t=0}^{\infty} \mathbf{1}[\mathbf{s} \dot{+} S_t \neq \mathbf{d}], \quad (4)$$

i.e.  $T(\mathbf{s})$  is the number of initial segments (including the empty segment) of a random path starting at  $\mathbf{s}$  that has not reached the dead state  $\mathbf{d}$ .

We now prove a relationship between expected path length  $\tau(\mathbf{s})$  and survival probabilities  $\widehat{p}_i(\mathbf{s})$ :

**Lemma 6** For any state  $\mathbf{s}$  and event  $i$ ,

$$\widehat{p}_i(\mathbf{s}) = \frac{1}{n}(\tau(\mathbf{s}) - \tau(\mathbf{s} \dot{+} \mathbf{e}_i)).$$

**Proof:** When  $i \notin \lambda(\mathbf{s})$ , then  $\mathbf{s} = \mathbf{s} \dot{+} \mathbf{e}_i$  and it is trivially true that

$$\widehat{p}_i(\mathbf{s}) = 0 = \frac{1}{n}(\tau(\mathbf{s}) - \tau(\mathbf{s} \dot{+} \mathbf{e}_i)).$$

The interesting case is when  $i \in \lambda(\mathbf{s})$ . Indeed, Using (4), we have

$$\begin{aligned} \tau(\mathbf{s}) - \tau(\mathbf{s} \dot{+} \mathbf{e}_i) &= \mathbb{E} T(\mathbf{s}) - \mathbb{E} T(\mathbf{s} \dot{+} \mathbf{e}_i) \\ &= \mathbb{E} \left[ \sum_{t=0}^{\infty} \mathbf{1}[\mathbf{s} \dot{+} S_t \neq \mathbf{d}] - \mathbf{1}[(\mathbf{s} \dot{+} \mathbf{e}_i) \dot{+} S_t \neq \mathbf{d}] \right]. \end{aligned}$$

Since the dead state  $\mathbf{d}$  is an absorbing state we have that for any path  $S$ , if  $\mathbf{s} \dot{+} S = \mathbf{d}$ , then  $\mathbf{s} \dot{+} \mathbf{e}_i \dot{+} S = \mathbf{d}$  as well. Equivalently, if  $(\mathbf{s} \dot{+} \mathbf{e}_i) \dot{+} S \neq \mathbf{d}$ , then  $\mathbf{s} \dot{+} S \neq \mathbf{d}$ . Thus in the difference between the expectations, we only need be concerned with sequences  $S_t$  that are accounted for in the first expectation but not in the second. Therefore the above difference becomes

$$= \mathbb{E} \left[ \sum_{t=0}^{\infty} \mathbf{1}[(\mathbf{s} \dot{+} S_t \neq \mathbf{d}) \wedge ((\mathbf{s} \dot{+} \mathbf{e}_i) \dot{+} S_t) = \mathbf{d}] \right].$$

We claim that any sequence  $S_t$  that satisfies the conjunction must have the property that  $(S_t)_i = k - s_i$ . This is true because  $(\mathbf{s} \dot{+} \mathbf{e}_i) \dot{+} S_t = \mathbf{d}$  and therefore  $(S_t)_i \geq k + 1 - s_i$ . Also  $(S_t)_j \geq k + 1 - s_j$ , for  $j \neq i$ . This implies that  $\mathbf{s} \dot{+} S_t = \mathbf{o}_i$  and the above difference becomes

$$\mathbb{E} \left[ \sum_{t=0}^{\infty} \mathbf{1}[\mathbf{s} \dot{+} S_t = \mathbf{o}_i] \right].$$

The last term is exactly  $\widehat{p}_i(\mathbf{s})$ , the probability that  $\mathbf{s} \dot{+} S_t$  eventually arrives at  $\mathbf{o}_i$ , times the expected number of iterations spent in state  $\mathbf{o}_i$  before arriving at  $\mathbf{d}$ . To leave  $\mathbf{o}_i$ , the random walk must make a step in the  $i$ th direction, and thus the expected “waiting time” at  $\mathbf{o}_i$  can be computed as

$$\sum_{q=1}^{\infty} q \underbrace{\left(1 - \frac{1}{n}\right)^{q-1}}_{\text{prob. of } q-1 \text{ loops}} \underbrace{\frac{1}{n}}_{\text{prob. of leaving}} = n.$$

The last lemma implies an important fact about the state lattice. Interpret the state lattice as a directed graph with directed edges at all pairs  $(\mathbf{s}, \mathbf{s} \dot{+} \mathbf{e}_i)$  for each  $i \in \lambda(\mathbf{s})$ . Also associate the edge  $(\mathbf{s}, \mathbf{s} \dot{+} \mathbf{e}_i)$  with the survival probability  $\widehat{p}_i(\mathbf{s})$ . Consider starting at state  $\mathbf{s}$  and walking through this directed graph:

$$\mathbf{s} \rightarrow \mathbf{s} \dot{+} \mathbf{e}_{i_1} \rightarrow \mathbf{s} \dot{+} \mathbf{e}_{i_1} \dot{+} \mathbf{e}_{i_2} \rightarrow \dots$$

**Corollary 7** Consider any two states  $\mathbf{s}, \mathbf{s}'$ . For any path from  $\mathbf{s}$  to  $\mathbf{s}'$  through the directed state graph, the sum of all edge weights  $\widehat{p}_i(\cdot)$  along this path is independent of the choice of path.

**Proof:** Assume the path  $\mathbf{s} = \mathbf{s}^1, \mathbf{s}^2, \dots, \mathbf{s}^T, \mathbf{s}^{T+1} = \mathbf{s}'$  defined by a sequence of moves is  $i_1, i_2, \dots, i_T$ , where  $\mathbf{s}^{t+1} = \mathbf{s}^t + \mathbf{e}_{i_t}$ . By Lemma 6 the total weight sum is

$$\sum_{t=1}^T \hat{p}_{i_t}(\mathbf{s}^t) = \sum_{t=1}^T \frac{1}{n} (\tau(\mathbf{s}^t) - \tau(\mathbf{s}^{t+1})) = \frac{1}{n} (\tau(\mathbf{s}) - \tau(\mathbf{s}')),$$

which is independent of the choice of path.  $\blacksquare$

Note that in the definition of the directed state graph above and in the corollary we ignore loops, which occur when  $\mathbf{s} = \mathbf{s} + \mathbf{e}_i$  (or equivalently  $i \notin \lambda(\mathbf{s})$ ). Such loops out of state  $\mathbf{s}$  are immaterial because they correspond to dead events, and  $i \notin \lambda(\mathbf{s})$  iff  $\hat{p}_i(\mathbf{s}) = 0$ .

## 4 The Optimal Strategy

We now have the all the tools to express  $\hat{V}(\mathbf{s})$  in terms of the expected path length  $\tau(\mathbf{s})$ , prove that  $V(\mathbf{s}) = \hat{V}(\mathbf{s})$ , and show that the optimal betting strategy for the gambler is  $\hat{p}(\mathbf{s})$ .

We prove two major theorems in this section. We provide the mathematically precise argument for each but, as formality often obscures the true intuition, we also provide an “English Version” so that the reader sees a rough sketch. Our mathematical proofs require induction on the state space  $\mathcal{S}$ , so we need a “measure of progress” for state vectors  $\mathbf{s}$ . For any  $\mathbf{s} \in \mathcal{S}$ , define  $m(\mathbf{s}) := n(k+1) - |\mathbf{s}|$ , the number of steps required before reaching the dead state. Clearly  $m(\mathbf{s}) = 0$  if and only if  $\mathbf{s} = \mathbf{d}$ .

**Theorem 8** For all states  $\mathbf{s}$ ,

$$\hat{V}(\mathbf{s}) = \frac{1}{n} \tau(\mathbf{s}).$$

**Proof:** (*English Version*) Assume that the Gambler always plays according to the distribution vector  $\hat{p}(\mathbf{s})$ . Then we may think of the Casino’s choices as a walk around the state graph and, as we discussed at the end of Section 3, a collection of the “weights”  $\hat{p}_i(\cdot)$  along the way, ending at  $\mathbf{d}$ . But as we proved in Corollary 7 for the weights  $\hat{p}(\cdot)$ , it doesn’t matter what path is taken: the Casino will always receive  $\frac{1}{n} (\tau(\mathbf{s}) - \tau(\mathbf{d})) = \frac{1}{n} \tau(\mathbf{s})$  on any path from  $\mathbf{s}$  that just ended in  $\mathbf{d}$ .

If the Gambler ever chooses a distribution  $\mathbf{w}$  different from  $\hat{p}(\mathbf{s})$  at some state  $\mathbf{s}$ , then the Casino can simply let  $\ell = \mathbf{e}_j$  for any  $j$  for which  $w_j > \hat{p}_j(\mathbf{s})$ , and on this round the casino will force loss *greater* than  $\hat{p}_j(\mathbf{s})$ . This means that on some path starting from  $\mathbf{s}$ , the Casino will accrue total weight/loss larger than  $\frac{1}{n} \tau(\mathbf{s})$ , and therefore that the distribution  $\mathbf{w}$  at  $\mathbf{s}$  was non-optimal for the Gambler. We conclude that for the Gambler  $\hat{p}(\cdot)$  is the only optimal assignment of distributions to states.  $\blacksquare$

**Proof:** (*Formal Version*) We induct on  $m(\mathbf{s})$ . First we check the base case  $\mathbf{s} = \mathbf{d}$ . In this case, the expected path length is exactly 0 since we have already reached the dead state. Thus  $\frac{\tau(\mathbf{s})}{n} = 0 = \hat{V}(\mathbf{d})$  as desired.

Now assume that  $m(\mathbf{s}) > 0$ . Then

$$\begin{aligned} \hat{V}(\mathbf{s}) &= \min_{\mathbf{w} \sim \lambda(\mathbf{s})} \max_{i \in \lambda(\mathbf{s})} w_i + \hat{V}(\mathbf{s} + \mathbf{e}_i) \\ (\text{induc.}) &= \min_{\mathbf{w} \sim \lambda(\mathbf{s})} \max_{i \in \lambda(\mathbf{s})} w_i + \frac{1}{n} \tau(\mathbf{s} + \mathbf{e}_i) \\ &\leq \max_{i \in \lambda(\mathbf{s})} \hat{p}_i(\mathbf{s}) + \frac{1}{n} \tau(\mathbf{s} + \mathbf{e}_i) \\ (\text{Lem. 6}) &= \max_i \frac{1}{n} (\tau(\mathbf{s}) - \tau(\mathbf{s} + \mathbf{e}_i)) + \frac{1}{n} \tau(\mathbf{s} + \mathbf{e}_i) \\ &= \frac{1}{n} \tau(\mathbf{s}). \end{aligned}$$

We prove  $\hat{V}(\mathbf{s}) \geq \frac{1}{n} \tau(\mathbf{s})$  by a similar induction. Assume that the Gambler chooses the optimal distribution  $\mathbf{w}^*$  which may indeed be different from  $\hat{p}(\mathbf{s})$ . For any  $i \notin \lambda(\mathbf{s})$ ,  $\hat{p}_i(\mathbf{s})$  is defined as zero. For the optimal strategy  $w_i^* = 0$  as well because otherwise the Casino can incur unbounded loss by playing  $\mathbf{e}_i$  repeatedly. Since  $\mathbf{w}^*$  and  $\hat{p}(\mathbf{s})$  are different distributions on the live events  $\lambda(\mathbf{s})$ , there must exist some  $j \in \lambda(\mathbf{s})$  for which  $w_j^* > \hat{p}_j(\mathbf{s})$ . We now have

$$\begin{aligned} \hat{V}(\mathbf{s}) &= \max_{i \in \lambda(\mathbf{s})} w_i^* + \hat{V}(\mathbf{s} + \mathbf{e}_i) \\ (\text{induc.}) &= \max_{i \in \lambda(\mathbf{s})} w_i^* + \frac{1}{n} \tau(\mathbf{s} + \mathbf{e}_i) \\ &\geq w_j^* + \frac{1}{n} \tau(\mathbf{s} + \mathbf{e}_i) \\ &> \hat{p}_j(\mathbf{s}) + \frac{1}{n} \tau(\mathbf{s} + \mathbf{e}_i) \\ (\text{Lem.6}) &= \frac{1}{n} (\tau(\mathbf{s}) - \tau(\mathbf{s} + \mathbf{e}_i)) + \frac{1}{n} \tau(\mathbf{s} + \mathbf{e}_i) \\ &= \frac{1}{n} \tau(\mathbf{s}). \end{aligned}$$

**Corollary 9** For any  $\mathbf{s} \neq \mathbf{d}$ ,  $\hat{p}(\mathbf{s})$  is the unique optimal probability vector for the learner for the game related to  $\hat{V}$ .  $\blacksquare$

**Proof:** See end of last proof.  $\blacksquare$

**Corollary 10** For all  $\mathbf{s}$  and all  $i \in [n]$ ,

$$\hat{p}_i(\mathbf{s}) = \hat{V}(\mathbf{s}) - \hat{V}(\mathbf{s} + \mathbf{e}_i)$$

**Proof:** This follows from the previous theorem and Lemma 6.  $\blacksquare$

We need one more lemma before we can prove our main result.

**Lemma 11** For any state  $\mathbf{s}$  and distinct events  $i, j \in \lambda(\mathbf{s})$ , we have

$$\hat{p}_i(\mathbf{s}) < \hat{p}_i(\mathbf{s} + \mathbf{e}_j).$$

This fact is intuitive: if losses are randomly assigned then the probability that the  $i$ th event will survive last *strictly increases* when another event suffers a loss. We prove this precisely below.

**Proof:** To show that  $\widehat{p}_i(\mathbf{s}) \leq \widehat{p}_i(\mathbf{s} + \mathbf{e}_j)$  is straightforward. Any sequence  $S_0, S_1, S_2, \dots$  that brings  $\mathbf{s}$  to the one-live state  $\mathbf{o}_i$  also brings  $\mathbf{s} + \mathbf{e}_j$  to  $\mathbf{o}_i$ . Indeed, if  $\mathbf{s} \dot{+} S_t = \mathbf{o}_i$  for some  $t$  then certainly  $(\mathbf{s} + \mathbf{e}_j) \dot{+} S_t = \mathbf{o}_i$  as well.

To show that this inequality is strict, we need only find one random sequence for which  $\mathbf{s} + \mathbf{e}_j$  is brought to  $\mathbf{o}_i$  but not  $\mathbf{s}$ . Take any sequence  $S_0, S_1, \dots$  such that  $\mathbf{s} \dot{+} S_t = \mathbf{d} - \mathbf{e}_i - \mathbf{e}_j$  (where the only events remaining are  $i$  and  $j$ ) and where  $S_{t+1} = S_t + \mathbf{e}_i$ . Then  $(\mathbf{s} + \mathbf{e}_j) \dot{+} S_t = \mathbf{o}_i$  but  $\mathbf{s} \dot{+} S_{t+1} = \mathbf{s} \dot{+} (S_t + \mathbf{e}_i) = \mathbf{o}_j$ . ■

**Theorem 12** For all states  $\mathbf{s}$ ,

$$V(\mathbf{s}) = \widehat{V}(\mathbf{s}) = \frac{1}{n} \tau(\mathbf{s}).$$

**Proof:** (English Version) Imagine a gambler who plays the distribution  $\widehat{p}(\mathbf{s})$  at every state  $\mathbf{s}$ . We already know that the Casino can use its modified game strategy and simply play unit vectors  $\boldsymbol{\ell} = \mathbf{e}_i$  on each round to force  $\frac{1}{n} \tau(\mathbf{s})$  loss. Yet since  $\boldsymbol{\ell}$  is unrestricted, can it obtain more? The answer is No: consider what happens if the Casino decides to choose  $\boldsymbol{\ell}$  larger than a unit vector, e.g. let  $\boldsymbol{\ell} = \mathbf{e}_i + \mathbf{e}_j$  for simplicity. Then on this round it obtains  $\widehat{p}_i(\mathbf{s}) + \widehat{p}_j(\mathbf{s})$ , but it can do better! We proved in Lemma 11 that survival probabilities strictly increase and therefore  $\widehat{p}_i(\mathbf{s}) < \widehat{p}_i(\mathbf{s} + \mathbf{e}_j)$ . Thus, a more patient Casino could choose  $\boldsymbol{\ell} = \mathbf{e}_j$  on this round, obtain  $\widehat{p}_j(\mathbf{s})$ , and then choose  $\boldsymbol{\ell} = \mathbf{e}_i$  on the next round to obtain  $\widehat{p}_i(\mathbf{s} + \mathbf{e}_j)$ . As  $\widehat{p}_j(\mathbf{s}) + \widehat{p}_i(\mathbf{s} + \mathbf{e}_j) > \widehat{p}_j(\mathbf{s}) + \widehat{p}_i(\mathbf{s})$ , the Casino only does worse by playing non-unit vectors. Indeed, this suggests that the Gambler has a strategy by which the Casino can inflict only as much loss as in the modified game, and thus the value  $V(\mathbf{s})$  is no different from  $\widehat{V}(\mathbf{s})$ . ■

**Proof:** (Formal Version) Certainly  $V(\mathbf{s}) \geq \widehat{V}(\mathbf{s})$ , since the Casino is given strictly fewer choices in the modified game. Thus we are left to show that  $V(\mathbf{s}) \leq \widehat{V}(\mathbf{s})$ . We proceed via induction on  $m(\mathbf{s})$ . By definition,  $V(\mathbf{s}) = \widehat{V}(\mathbf{s})$  for the case  $\mathbf{s} = \mathbf{d}$ . Now assume that, for all successive states  $\mathbf{s}'$  where  $m(\mathbf{s}') < m(\mathbf{s})$ ,  $V(\mathbf{s}') = \widehat{V}(\mathbf{s}')$ . We proceed by directly analyzing the recursive definition (2). Assume that the Gambler has chosen the (possibly non-optimal) distribution  $\mathbf{w} = \widehat{p}(\mathbf{s})$  to distribute his wealth on the live events  $\lambda(\mathbf{s})$ , and let  $\boldsymbol{\ell}^* \in \{0, 1\}^n$  be an optimal choice of the Casino (which can depend on the Gambler's choice). By definition (1) of  $V(\mathbf{s})$ , the chosen loss vector can't be  $\mathbf{0}$  and all events with loss one must be in  $\lambda(\mathbf{s})$ . More precisely,

$$\begin{aligned} V(\mathbf{s}) &= \min_{\mathbf{w} \sim \lambda(\mathbf{s})} \max_{\mathbf{0} \neq \boldsymbol{\ell} \subset \lambda(\mathbf{s})} \mathbf{w} \cdot \boldsymbol{\ell} + V(\mathbf{s} + \boldsymbol{\ell}) \\ (\text{ind.}) &= \min_{\mathbf{w} \sim \lambda(\mathbf{s})} \max_{\mathbf{0} \neq \boldsymbol{\ell} \subset \lambda(\mathbf{s})} \mathbf{w} \cdot \boldsymbol{\ell} + \widehat{V}(\mathbf{s} + \boldsymbol{\ell}) \\ &\leq \max_{\mathbf{0} \neq \boldsymbol{\ell} \subset \lambda(\mathbf{s})} \widehat{p}(\mathbf{s}) \cdot \boldsymbol{\ell} + \widehat{V}(\mathbf{s} + \boldsymbol{\ell}) \\ &= \widehat{p}(\mathbf{s}) \cdot \boldsymbol{\ell}^* + \widehat{V}(\mathbf{s} + \boldsymbol{\ell}^*) \end{aligned}$$

If  $\boldsymbol{\ell}^*$  is any unit vector  $\mathbf{e}_i$ , s.t.  $i \in \lambda(\mathbf{s})$ , then

$$\begin{aligned} V(\mathbf{s}) &\leq \widehat{p}_i(\mathbf{s}) \cdot \mathbf{e}_i + \widehat{V}(\mathbf{s} + \mathbf{e}_i) \\ &= \widehat{p}_i(\mathbf{s}) + \widehat{V}(\mathbf{s} + \mathbf{e}_i) = \widehat{V}(\mathbf{s}) \end{aligned}$$

and in this case,  $V(\mathbf{s}) = \widehat{V}(\mathbf{s})$  and we are done. We now prove by contradiction that  $\boldsymbol{\ell}^*$  can have no more than one non-zero coordinate. Assume indeed that  $|\boldsymbol{\ell}^*| > 1$ , i.e. it admits a decomposition  $\boldsymbol{\ell}^* = \mathbf{e}_i + \bar{\boldsymbol{\ell}}$  for some  $i$  and bit vector  $\bar{\boldsymbol{\ell}} \neq \mathbf{0}$  with  $\bar{\ell}_i = 0$ . Applying Lemma 11 repeatedly, we have that  $\widehat{p}_i(\mathbf{s}) < \widehat{p}_i(\mathbf{s} + \bar{\boldsymbol{\ell}})$  and therefore

$$\begin{aligned} &\widehat{p}(\mathbf{s}) \cdot \boldsymbol{\ell}^* + \widehat{V}(\mathbf{s} + \boldsymbol{\ell}^*) \\ &= \widehat{p}_i(\mathbf{s}) + \widehat{p}(\mathbf{s}) \cdot \bar{\boldsymbol{\ell}} + \widehat{V}(\mathbf{s} + \boldsymbol{\ell}^*) \\ (\text{Lem. 11}) &< \widehat{p}_i(\mathbf{s} + \bar{\boldsymbol{\ell}}) + \widehat{p}(\mathbf{s}) \cdot \bar{\boldsymbol{\ell}} + \widehat{V}(\mathbf{s} + \boldsymbol{\ell}^*) \\ (\text{Cor. 10}) &= \widehat{V}(\mathbf{s} + \bar{\boldsymbol{\ell}}) - \widehat{V}(\mathbf{s} + \boldsymbol{\ell}^*) + \widehat{p}(\mathbf{s}) \cdot \bar{\boldsymbol{\ell}} + \widehat{V}(\mathbf{s} + \boldsymbol{\ell}^*) \\ &= \widehat{p}(\mathbf{s}) \cdot \bar{\boldsymbol{\ell}} + \widehat{V}(\mathbf{s} + \bar{\boldsymbol{\ell}}). \end{aligned}$$

But the statement  $\widehat{p}(\mathbf{s}) \cdot \boldsymbol{\ell}^* + \widehat{V}(\mathbf{s} + \boldsymbol{\ell}^*) < \widehat{p}(\mathbf{s}) \cdot \bar{\boldsymbol{\ell}} + \widehat{V}(\mathbf{s} + \bar{\boldsymbol{\ell}})$  implies  $\boldsymbol{\ell}^*$  is a non-optimal choice for the Casino and this contradicts our assumption that  $\boldsymbol{\ell}^*$  was optimum. ■

**Corollary 13** For any  $\mathbf{s} \neq \mathbf{d}$ , if the learner plays with the optimum probability vector  $\widehat{p}(\mathbf{s})$ , then the only optimal responses of the adversary in the recurrence (2) for  $V$  is to choose a unit vector of a live event.

**Proof:** Proved at the end of the last theorem. ■

## 5 Recurrences, Combinatorics and Randomized Algorithms

The quantities  $V(\mathbf{s})$ ,  $\tau(\mathbf{s})$  and  $\widehat{p}_i(\mathbf{s})$  have a number of interesting properties that we lay out in this section.

### 5.1 Some Recurrences

The expected path length,  $\tau(\mathbf{s})$  satisfies a very natural recursion. When  $\mathbf{s} = \mathbf{d}$ , then the path length is deterministically 0 and therefore  $\tau(\mathbf{d}) = 0$ . Otherwise, we see that the expected path length is

$$\tau(\mathbf{s}) = 1 + \frac{\sum_{i=1}^n \tau(\mathbf{s} + \mathbf{e}_i)}{n}. \quad (5)$$

That is, the expected path length is 1, for the current step in the path, plus the expected path length of the next random state. Since the next state is chosen randomly from the set  $\{\mathbf{s} + \mathbf{e}_i : i = 1, \dots, n\}$ , the probability of any given state is  $\frac{1}{n}$ , hence the normalization factor.

Of course, our original quantity of interest is  $V(\mathbf{s})$ , and as we showed in Theorem 12  $V(\mathbf{s}) = \frac{1}{n} \tau(\mathbf{s})$ . This immediately gives us a recursion for  $V$ :

$$\begin{aligned} V(\mathbf{s}) &= \frac{1}{n} \left( 1 + \frac{1}{n} \sum_{i=1}^n \tau(\mathbf{s} + \mathbf{e}_i) \right) \\ &= \frac{1 + \sum_{i=1}^n V(\mathbf{s} + \mathbf{e}_i)}{n}. \end{aligned}$$

This recurrence, while true for the function  $V(\cdot)$ , is ambiguous because  $V(\mathbf{s})$  can occur on both sides of the equation. Indeed, whenever  $i \notin \lambda(\mathbf{s})$ ,  $V(\mathbf{s} + \mathbf{e}_i) = V(\mathbf{s})$ . However, we can rearrange all  $V(\mathbf{s})$  terms to obtain the following well-defined recursion:

$$V(\mathbf{s}) = \frac{1 + \sum_{i \in \lambda(\mathbf{s})} V(\mathbf{s} + \mathbf{e}_i)}{|\lambda(\mathbf{s})|}. \quad (6)$$

We can find a similar recurrence for  $\hat{p}_i(\cdot)$ . For the one-live states  $\mathbf{o}_i$  we have  $\hat{p}_j(\mathbf{o}_i) = 1$  if  $i = j$  and 0 otherwise. If  $|\lambda(\mathbf{s})| > 1$ , then

$$\hat{p}_i(\mathbf{s}) = \frac{\sum_{j=1}^n \hat{p}_i(\mathbf{s} + \mathbf{e}_j)}{n}.$$

As  $\hat{p}_i(\mathbf{s})$  is the probability of ending at state  $\mathbf{o}_i$  after executing the Markov chain, this formula is obtained by conditioning on one step of the Markov process. That is, the probability of ending at state  $\mathbf{o}_i$  is

$$\sum_j Pr(j \text{ chosen}) Pr(\text{random process takes } \mathbf{s} + \mathbf{e}_j \text{ to } \mathbf{o}_i).$$

This recurrence suffers from the same problem as did our initial recurrence for  $V(\cdot)$ :  $\hat{p}_i(\mathbf{s})$  can occur on both sides of the equality. We again solve this problem by rearranging terms and obtain

$$\hat{p}_i(\mathbf{s}) = \frac{\sum_{j \in \lambda(\mathbf{s})} \hat{p}_i(\mathbf{s} + \mathbf{e}_j)}{|\lambda(\mathbf{s})|}.$$

## 5.2 Combinatorial Sums

A further analysis gives us exact expressions for both  $\hat{p}_i(\mathbf{s})$  and  $V(\mathbf{s})$  in terms of infinite sums of multinomials.

**Proposition 5.1** For any state  $\mathbf{s} \in \mathcal{S}$ ,

$$\hat{p}_i(\mathbf{s}) = \sum_{\mathbf{r}: \mathbf{s} + \mathbf{r} = \mathbf{o}_i} \binom{|\mathbf{r}|}{r_1, r_2, \dots, r_n} \left(\frac{1}{n}\right)^{|\mathbf{r}|+1}.$$

**Proof:** By definition,  $\hat{p}_i(\mathbf{s})$  is the probability that  $\mathbf{s}$  reaches the one-live state  $\mathbf{o}_i$  eventually. To compute this probability, we consider at what point the Markov process *exits* the state  $\mathbf{o}_i$  and into  $\mathbf{d}$ . Recall the random variable  $S_t$  defined in Section 3. Take any  $\mathbf{r}$  for which  $\mathbf{s} + \mathbf{r} = \mathbf{o}_i$  and condition on  $S_t = \mathbf{r}$ . Then

$$\hat{p}_i(\mathbf{s}) = \sum_{\mathbf{r}: \mathbf{s} + \mathbf{r} = \mathbf{o}_i} Pr(S_t = \mathbf{r}) Pr(S_{t+1} = \mathbf{r} + \mathbf{e}_i | S_t = \mathbf{r})$$

The first probability is exactly  $\binom{|\mathbf{r}|}{r_1, r_2, \dots, r_n} n^{-|\mathbf{r}|}$  and the second probability is exactly  $1/n$ . ■

Since  $V(\mathbf{s})$  can be written as an expected path length, we can obtain a similar expression as a sum of multinomials for  $V(\mathbf{s})$ :

**Proposition 5.2**

$$V(\mathbf{s}) = \sum_{i=1}^n \sum_{\mathbf{r}: \mathbf{s} + \mathbf{r} = \mathbf{o}_i} (|\mathbf{r}| + 1) \binom{|\mathbf{r}|}{r_1, r_2, \dots, r_n} \left(\frac{1}{n}\right)^{|\mathbf{r}|+1}.$$

## 5.3 Randomized Approximations

Computing the exact value  $V(\mathbf{s})$  for large but non-asymptotic values of the state vector is difficult because we have no polynomial time algorithm. On the other hand, finding a randomized approximation to  $V(\mathbf{s})$  can be done very efficiently. Indeed, as we now have a representation of  $V(\mathbf{s})$  in terms of the length of a random walk, we can simply run the random walk  $S_1, S_2, \dots$  several times, note the length  $T(\mathbf{s})$ , and return the

mean. Such random approximations require that the distribution on  $T(\mathbf{s})$  has low-variance, yet this certainly holds in the case at hand. While the random walk requires at least  $n(k + 1)$  iterations to finish, a simple argument shows that with probability  $1 - \delta$  the random walk completes in less than  $nk \log(nk/\delta)$  rounds.

---

### Algorithm 1 Random Approximation to $V(\mathbf{s})$

---

```

Input: state  $\mathbf{s}$ 
 $t \leftarrow 0$ 
for  $i = 1, \dots, \text{NUMITER}$  do
   $\mathbf{z} \leftarrow \mathbf{s}$ 
  repeat
    Sample  $i \in \{1, \dots, n\}$  u.a.r.
     $\mathbf{z} \leftarrow \mathbf{z} + \mathbf{e}_i$ 
     $t \leftarrow t + 1$ 
  until  $\mathbf{z} = \mathbf{d}$ 
end for
Return  $\frac{t}{n \cdot \text{NUMITER}}$ .

```

---

If  $R(\mathbf{s})$  is the random variable returned by the above algorithm, then clearly  $\mathbb{E}R(\mathbf{s}) = V(\mathbf{s})$ . By increasing NUMITER, the variance of this estimate can be reduced quickly.

A randomized approximation for  $\hat{p}_i(\mathbf{s})$  can be obtained similarly. Again the above algorithm approximately com-

---

### Algorithm 2 Random Approximation to $\hat{p}_i(\mathbf{s})$

---

```

Input: state  $\mathbf{s} \neq \mathbf{d}$ 
 $\mathbf{p} \leftarrow \mathbf{0}$ 
for  $i = 1, \dots, \text{NUMITER}$  do
   $\mathbf{z} \leftarrow \mathbf{s}$ 
  repeat
    Sample  $i \in \{1, \dots, n\}$  u.a.r.
     $\mathbf{z} \leftarrow \mathbf{z} + \mathbf{e}_i$ 
  until  $\mathbf{z} = \mathbf{o}_j$  for some  $j$ 
   $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{e}_j$ 
end for
Return  $\frac{\mathbf{p}}{\text{NUMITER}}$ .

```

---

putes  $\hat{p}_i(\mathbf{s})$  in the following sense: If  $R(\mathbf{s})$  is the random variable returned by the above algorithm, then clearly  $\mathbb{E}R(\mathbf{s}) = \hat{p}_i(\mathbf{s})$ . Again increasing NUMITER, reduces the variance of the estimate.

## 5.4 A Simple Strategy in a Randomized Setting

In the particular case of betting against the Casino, it may be necessary for the Gambler to compute  $\hat{p}_i(\mathbf{s})$  in order to place his bets optimally. In an alternative setting, however, a randomized algorithm may be sufficient. Let us consider the case in which the Gambler chooses to bet according to the outcome of several coin tosses. Further assume that the Casino can observe his strategy but cannot see the outcome of the coin tosses or his final bets. In this scenario, the Gambler can even bet all of his money on a random event  $I \in \{1, \dots, n\}$  drawn according to some distribution as long

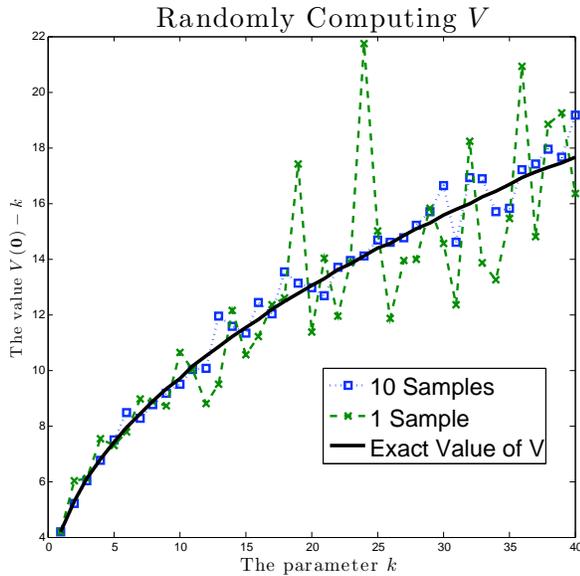


Figure 1: We illustrate the accuracy of the randomized approximation to  $V(\mathbf{0})$  stated in Algorithm 1. The plot compares the exact value of  $V(\mathbf{0})$  to that obtained by using either 1 or 10 samples of the random walk. Here  $n = 100$ .

as  $\mathbb{E}\mathbf{1}[I = i] = \hat{p}_i(\mathbf{s})$  for all  $i$ , and indeed his expected loss would be  $\hat{p}(\mathbf{s}) \cdot \ell$ .

For this scenario, randomly approximating  $\hat{p}$  is not necessary: *only one sample is needed!* To be precise, the Gambler can take the state  $\mathbf{s}$ , run the random walk until the state reaches  $\mathbf{o}_i$  for some  $i$ , and then bet his full dollar on event  $i$ . This bet will be correct in expectation, i.e. he will pick event  $i$  with probability  $\hat{p}_i(\mathbf{s})$ , and thus his expected loss will be exactly  $\hat{p} \cdot \ell$ . The key here is that sampling from the distribution  $\hat{p}(\mathbf{s})$  may be quite easy even when computing it exactly may take more time.

Note that the above method based on one sample is similar to the way the Randomized Weighted Majority algorithm approximates the Weighted Majority algorithm (more precisely the WMC algorithm of [LW94]). More precisely `NUMITER=1` of Algorithm 5.3 corresponds to WMR, and `NUMITER`  $\rightarrow \infty$  corresponds to WMC.

## 6 Comparison to Previous Bounds

As mentioned in the introduction, the bound obtainable based on exponential weights [FS97] is

$$k + \sqrt{2k \log n} + \log n \quad (7)$$

and can be shown to be asymptotically optimal [Vov98].<sup>2</sup> Having computed the minimax solution to the same game, we can compute the game-theoretically optimal bound of  $V(\mathbf{0})$  using Algorithm 1. For small values of  $n$  and  $k$ , these bounds do differ quite substantially. We present in Figure 2 a

<sup>2</sup>A slightly better but more complicated bound than (7) was given in [Vov98]. In the full paper we compare the optimal bound to this one as well.

comparison of the regret for  $n = 2, 10, 100$  and  $k = 1, \dots, 20$ .

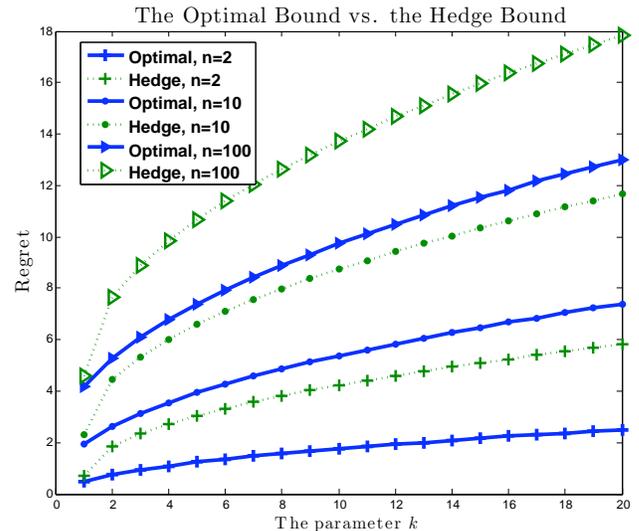


Figure 2: We compare the optimal regret bound we obtain from  $V(\mathbf{0})$  to that found in [FS97], which we refer to as the hedge bound. While asymptotically optimal, we observed that the hedge bound of  $k + \sqrt{2k \log n} + \log n$  is not tight for small values of  $n$  and  $k$ .

## 7 Connections to classic problems of probabilistic enumerative combinatorics.

Theorem 12 shows that an optimal strategy for the Casino requires unit vector plays. This leads to alternative interpretations of the game in terms of well studied random processes.

For example, one can easily confirm that our game also describes the random process underlying a generalized form of the Coupon Collector's Problem [?] in which the collector buys cereal boxes one by one in order to obtain  $K = k+1$  complete sets of  $n$  baseball cards, assuming one card is randomly placed within each cereal box. The value of our game,  $V(0,0)$ , is in fact the expected number of cereal boxes, per baseball card, needed to obtain the desired  $K$  complete sets.

Specifically, the probability generating function for the generalized Coupon Collector's Problem is [MW03]

$$G_{n,K}(z) = \frac{n}{(K-1)!} \int_0^\infty e^{-nt/z} t^{K-1} \left[ \sum_{j \geq k} \frac{t^j}{j!} \right]^{n-1} dt.$$

Taking the derivative at  $z = 1$  and dividing by  $n$ , we derive the expected number of steps to obtain  $K$  sets, which is also the value of our game, viz.

$$V(0^n) = \frac{n}{(K-1)!} \int_0^\infty t^K e^{-nt} \left[ \sum_{j \geq k} \frac{t^j}{j!} \right]^{n-1} dt. \quad (8)$$

Equation (8) gives us an elegant closed form for the two-card case ( $n = 2$ ):

$$V((0,0)) = K + \frac{K}{2^{2K}} \binom{2K}{K}$$

From (8) we also obtain the well known asymptotic expression for the value, for large  $n$  and fixed  $K$ ,

$$V(0^n) \rightarrow_{n \rightarrow \infty} \log n + (K - 1) \log \log n [1 + o(1)].$$

The same asymptotic form appears in the analysis of an evolving random graph. [ER60] The random walk on the state lattice provides yet another interpretation of the same dynamics.

For  $K \gg n \gg 1$ , the law of large numbers gives [NS60]

$$V((0^n)) = K + O(K^{1/2}).$$

## 8 Conclusion

We showed in Corollary 13 that against the optimal learning algorithm the optimal strategy of the adversary is to choose one of the unit loss vectors as his response. Curiously enough it can be show that this is also true of the Weighted Majority algorithm (1). That is, any trial in which  $q > 1$  experts incurred a unit of loss can be split into  $q$  trials in which a single expert has a unit of loss, and doing this always increases the loss of the algorithm for all update factor  $\beta \in [0, 1)$ . This observation about the Weighted Majority algorithm might actually lead to improved loss bounds for this algorithm, perhaps in the way the parameter  $\beta$  is tuned.

There remains also a deep question regarding the techniques introduced in this paper: how general is this method of computing the value of a game based on a random path? Can it handle slightly more involved problems? Examples we have considered include competing against  $m$ -sized sets of experts, discussed in [WK06], in which the loss of the algorithm is compared to the loss of the best  $m$ -subset. Another example is the problem of competing against permutations of  $n$  objects [HW07], where the loss of a permutation is linearly assigned. Our preliminary investigation suggests that similar techniques can be adapted to also handle such more complex problems. In the full paper we hope to delineate the scope of our new method of optimal algorithm design.

## References

- [ALW07] J. Abernethy, J. Langford, and M. K. Warmuth. Continuous experts and the Binning algorithm. In *Proceedings of the 19th Annual Conference on Learning Theory (COLT06)*, pages 544–558. Springer, June 2007.
- [CBFH<sup>+</sup>97] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997.
- [CBFHW96] N. Cesa-Bianchi, Y. Freund, D. P. Helmbold, and M. K. Warmuth. On-line prediction and conversion strategies. *Machine Learning*, 25:71–110, 1996.
- [ER60] P. Erdos and A. Renyi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5A:17–61, 1960.
- [FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to Boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997. Special Issue for EuroCOLT '95.
- [HW07] D. Helmbold and M. K. Warmuth. Learning permutations with exponential weights. In *Proceedings of the 20th Annual Conference on Learning Theory (COLT07)*. Springer, 2007.
- [KW99] Jyrki Kivinen and Manfred K. Warmuth. Averaging expert predictions. In *Computational Learning Theory, 4th European Conference, EuroCOLT '99, Nordkirchen, Germany, March 29-31, 1999, Proceedings*, volume 1572 of *Lecture Notes in Artificial Intelligence*, pages 153–167. Springer, 1999.
- [LW94] N. Littlestone and M. K. Warmuth. The Weighted Majority algorithm. *Inform. Comput.*, 108(2):212–261, 1994. Preliminary version in in FOCS 89.
- [MW03] A. Myers and H. S. Wilf. Some new aspects of the Coupon-Collector's problem. *SIAM J. Disc. Math.*, 17:1–17, 2003.
- [NS60] D. Newman and L. Shepp. The Double Dixie Cup problem. *Amer. Math Monthly.*, 67:541–574, 1960.
- [Vov98] V. Vovk. A game of prediction with expert advice. *J. of Comput. Syst. Sci.*, 56(2):153–173, 1998. Special Issue: Eighth Annual Conference on Computational Learning Theory.
- [WK06] M. K. Warmuth and D. Kuzmin. Randomized PCA algorithms with regret bounds that are logarithmic in the dimension. In *Advances in Neural Information Processing Systems 19 (NIPS 06)*. MIT Press, December 2006.



---

# On-line sequential bin packing

---

András György<sup>1</sup> and Gábor Lugosi<sup>2</sup> and György Ottucsák<sup>3</sup>

<sup>1</sup>Machine Learning Research Group, Computer and Automation Research Institute, Budapest, Hungary \*  
gya@szit.bme.hu

<sup>2</sup>ICREA and Department of Economics, Universitat Pompeu Fabra, Barcelona, Spain †  
gabor.lugosi@gmail.com

<sup>3</sup>GusGus AB, Budapest, Hungary  
ottucsak@gmail.com

## Abstract

We consider a sequential version of the classical bin packing problem in which items are received one by one. Before the size of the next item is revealed, the decision maker needs to decide whether the next item is packed in the currently open bin or the bin is closed and a new bin is opened. If the new item doesn't fit, it is lost. If a bin is closed, the remaining free space in the bin accounts for a loss. The goal of the decision maker is to minimize the loss accumulated over  $n$  periods. The main result of the paper is an algorithm that has a cumulative loss not much larger than any strategy that uses a fixed threshold at each step to decide whether a new bin is opened.

## 1 Introduction

In the classical *off-line* bin packing problem, an algorithm receives *items* (also called *pieces*)  $x_1, x_2, \dots, x_n \in (0, 1]$ . We have an infinite number of bins, each with capacity 1, and every item is to be assigned to a bin. Further, the sum of the sizes of the items assigned to any bin cannot exceed its capacity. A bin is empty if no item is assigned to it, otherwise, it is used. The goal of the algorithm is to minimize the number of used bins. This is one of the classical NP-hard problems and heuristic and approximation algorithms have been investigated thoroughly, see, e.g., Coffman, Garey, and Johnson [3].

Another well-studied version of the problem is the so-called *on-line* bin packing problem. Here items arrive one by one and each item  $x_t$  must be assigned to a bin (with free space at least  $x_t$ ) immediately, without any knowledge of the next pieces. In this setting the goal is the same as in the off-line problem, that is, the number of used bins is to be minimized, see, e.g., Seiden [8], He and Dósa [6].

In both the off-line and on-line problems the algorithm has access to the bins in arbitrary order. In this paper we

abandon this assumption and introduce a more restricted version that we call *sequential bin packing*. In this setting items arrive one by one (just like in the on-line problem) but in each round the algorithm has only two possible choices: assign the given item to the (only) open bin or to the “next” empty bin (in this case this will be the new open bin), and items cannot be assigned anymore to closed bins. An algorithm thus determines a sequence of binary decisions  $i_1, \dots, i_n$  where  $i_t = 0$  means that the next item is assigned to the open bin and  $i_t = 1$  means that a new bin is opened and the next item is assigned to that bin. Of course, if  $i_t = 0$ , then it may happen that the item  $x_t$  doesn't fit in the open bin. In that case the item is “lost.” If the decision is  $i_t = 1$  then the remaining empty space in the last closed bin is counted as a loss. The measure of performance we use is the total sum of all lost items and wasted empty space.

Just as in the original bin packing problem, we may distinguish off-line and on-line versions of the sequential bin packing problem. In the *off-line sequential* bin packing problem the entire sequence  $x_1, \dots, x_n$  is known to the algorithm at the outset. Note that unlike in the classical bin packing problem, the order of the items is relevant. This problem turns out to be computationally significantly easier than its non-sequential counterpart. In Section 3 we present a simple algorithm with running time of  $O(n^2)$  that minimizes the total loss in the off-line sequential bin packing problem.

Much more interesting is the on-line variant of the sequential bin packing problem. Here the items  $x_t$  are revealed one by one, *after* the corresponding decision  $i_t$  has been made. In other words, each decision has to be made without any knowledge on the size of the item. Formulated this way, the problem is reminiscent to an on-line *prediction problem*, see [2]. However, unlike in standard formulations of on-line prediction, here the loss the predictor suffers depends not only on the outcome  $x_t$  and decision  $i_t$  but also on the “state” defined by the fullness of the open bin.

Our goal is to extend the usual bin packing problems to situations in which one can handle only one bin at a time, and items must be processed immediately so they cannot wait for bin changes. To motivate the on-line sequential model, one may imagine a simple revenue management problem in which a decision maker has a unit storage capacity at his disposal. A certain product arrives in packages of different size and after each arrival, it has to be decided whether the stored packages are shipped or not. (Storing of the product is costly.) If the stored goods are shipped, the entire storage

---

\*The first author acknowledges support by the Hungarian Scientific Research Fund (OTKA F60787) and the Mobile Innovation Center of Hungary.

†The second author acknowledges support by the Spanish Ministry of Science and Technology grant MTM2006-05650 and by the PASCAL Network of Excellence under EC grant no. 506778.

capacity becomes available again. If they are not shipped one waits for the arrival of the next package. However, if the next package is too large to fit in the remaining open space, it is lost.

In another example of application, a sensor collects measurements that can be compressed to variable size (these are the items). The sensor communicates its measurements by sending frames of some fixed size (bins). Since it has limited memory, it cannot store more data than one frame. To save energy, the sensor must maximize its throughput (the proportion of useful data in each frame) and at the same time minimize data loss (this trade-off is reflected in the definition of the loss function).

Just like in on-line prediction, we compare the performance of an algorithm with the best in a pool of reference algorithms (experts). Arguably the most natural comparison class contains all algorithms that use a fixed threshold to decide whether a new bin is opened. In other words, reference predictors are parameterized by a real number  $p \in (0, 1]$ . An expert with parameter  $p$  simply decides to open a new bin whenever the remaining free space in the open bin is less than  $p$ . We call such an expert a *constant-threshold* strategy. The main result of this paper is the construction of a randomized algorithm for the sequential on-line bin packing problem that achieves a cumulative loss (measured as the sum of the total wasted capacity and lost items) that is less than the total loss of the best constant-threshold strategy (determined in hindsight) plus a quantity of the order of  $n^{2/3} \log^{1/3} n$ .

The principal difficulty of the problem lies in the fact that each action of the decision maker takes the problem in a new “state” (determined by the remaining empty space in the open bin) which has an effect on future losses. Moreover, the state of the algorithm is typically different from the state of the experts which makes comparison difficult. In related work, Merhav, Ordentlich, Seroussi, and Weinberger [7] considered a similar setup in which the loss function has a “memory,” that is, the loss of a predictor depends on the loss of past actions. Furthermore, Even-Dar, Kakade and Mansour [4] considered the MDP case where the adversarial reward function changes according to some fixed stochastic dynamics. However, there are several main additional difficulties in the present case. First, unlike in [7], but similarly to [4], the loss function has an unbounded memory as the state may depend on an arbitrarily long sequence of past predictions. Second, the state space is infinite (the  $[0, 1)$  interval) and the class of experts we compare to is also infinite, in contrast to both of the above papers. However, the special properties of the bin packing problem make it possible to design a prediction strategy with small regret.

Note that the MDP setting of [4] would be a too pessimistic approach to our problem, as in our case there is a strong connection between the rewards in different states, thus the absolute adversarial reward function results in an overestimated worst case. Also in the present case, state transitions are deterministically given by the outcome, the previous state, and the action of the decision maker, while in the setup of [4] transitions are stochastic and depend only on the state and the decision of the algorithm, but not on the reward (or on the underlying individual sequence generating the reward).

We also mention here the similar *on-line bin packing with rejection* problem where the algorithm has an opportunity to reject some items and the loss function is the sum of the number of the used bins and the “costs” of the rejected items<sup>1</sup> (see He and Dósa [6]). However, instead of the number of used bins, we use the sum of idle capacities (missed or free spaces) in the used bins to measure the loss.

The following example may help explain the difference between various versions of the problem.

**Example 1** Let the sequence of the items be  $\langle 0.4, 0.5, 0.2, 0.5, 0.5, 0.3, 0.5, 0.1 \rangle$ . Then the cumulative loss of the optimal off-line bin packing is 0 and it is 0.4 in the case of sequential off-line bin packing (see Figure 1). In the sequential case the third item (0.2) has been rejected.

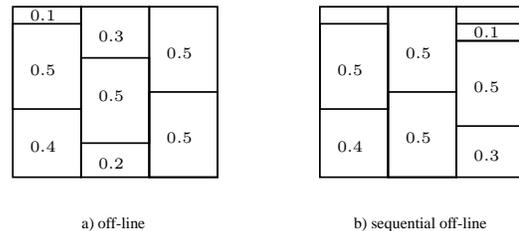


Figure 1: The difference between the optimal solutions for the off-line and sequential off-line problems.

The rest of the paper is organized as follows. In Section 2 the problem is defined formally. In Section 3 the complexity of the off-line sequential bin packing problem is analyzed. The main results of the paper are presented in Section 4.

## 2 Setup

We use a terminology borrowed from the theory of on-line prediction with expert advice. Thus, we call the sequential decisions of the on-line algorithm *predictions* and we use *forecaster* as a synonym for algorithm.

We denote by  $I_t \in \{0, 1\}$  the action of the forecaster at time  $t$  (that is, when  $t - 1$  items have been received). Action 0 means that the next item will be assigned to the open bin and action 1 represents the fact that a new bin is opened and the next item is assigned to the next empty bin. Note that we assume that we start with an open empty bin, thus for any reasonable algorithm,  $I_1 = 0$ , and we will restrict our attention to such algorithms. The sequence of decisions up to time  $t$  is denoted by  $\mathbf{I}_t \in \{0, 1\}^t$ .

Denote by  $\hat{s}_t \in [0, 1)$  the free space in the open (last) bin at time  $t \geq 1$ , that is, after having placed the items  $x_1, x_2, \dots, x_t$  according to the sequence  $\mathbf{I}_t$  of actions. This is the *state* of the forecaster. More precisely, the state of the forecaster is defined, recursively, as follows: As at the beginning we have an empty bin,  $\hat{s}_0 = 1$ . For  $t = 1, 2, \dots, n$ ,

- $\hat{s}_t = 1 - x_t$ , when the algorithm assigns the item to the next empty bin (i.e.,  $I_t = 1$ );

<sup>1</sup>In sequential bin packing we assume that the cost of the items coincides with their size. In this case the optimal solution of bin-packing with rejection is to reject all items.

- $\widehat{s}_t = \widehat{s}_{t-1}$ , when the assigned item does not fit in the open bin (i.e.,  $I_t = 0$  and  $\widehat{s}_{t-1} < x_t$ );
- $\widehat{s}_t = \widehat{s}_{t-1} - x_t$ , when the assigned item fits in the open bin (i.e.,  $I_t = 1$  and  $\widehat{s}_{t-1} \geq x_t$ ).

This may be written in a more compact form:

$$\begin{aligned}\widehat{s}_t &= \widehat{s}_t(I_t, x_t, \widehat{s}_{t-1}) \\ &= I_t(1 - x_t) + (1 - I_t)(\widehat{s}_{t-1} - \mathbb{I}_{\{\widehat{s}_{t-1} \geq x_t\}})\end{aligned}\quad (1)$$

where  $\mathbb{I}_{\{\cdot\}}$  denotes the indicator function of the event in brackets, that is, it equals 1 if the event is true and 0 otherwise. The loss suffered by the forecaster at round  $t$  is

$$\ell(I_t, x_t \mid \widehat{s}_{t-1}),$$

where the loss function  $\ell$  is defined by

$$\ell(0, x \mid s) = \begin{cases} 0, & \text{if } s \geq x; \\ x, & \text{otherwise} \end{cases}\quad (2)$$

and

$$\ell(1, x \mid s) = s. \quad (3)$$

The goal of the forecaster is to minimize its cumulative loss defined by

$$\widehat{L}_t = L_{\mathbf{I}_t, t} = \sum_{s=1}^t \ell(I_s, x_s \mid \widehat{s}_{s-1}).$$

In the off-line version of the problem, the entire sequence  $x_1, \dots, x_n$  is given and the solution is the optimal sequence  $\mathbf{I}_n^*$  of actions

$$\mathbf{I}_n^* = \arg \min_{\mathbf{I}_n \in \{0,1\}^n} L_{\mathbf{I}_n, n}.$$

In the on-line version of the problem the forecaster does not know the size of the next items, and the sequence of items can be completely arbitrary. We allow the forecaster to randomize its decisions, that is, at each time instance  $t$ ,  $I_t$  is allowed to depend on a random variable  $U_t$  where  $U_1, \dots, U_n$  are i.i.d. uniformly distributed random variables in  $[0, 1]$ .

Since we allow the forecaster to randomize, it is important to clarify that the entire sequence of items are determined *before* the forecaster starts making decisions, that is,  $x_1, \dots, x_n \in (0, 1]$  are fixed and cannot depend on the randomizing variables. (This is the so-called *oblivious adversary* model known in the theory of sequential prediction, see, e.g., [2].)

The performance of a sequential on-line algorithm is measured by its cumulative loss. It is natural to compare it to the cumulative loss of the off-line solution  $\mathbf{I}_n^*$ . However, it is easy to see that in general it is impossible to achieve an on-line performance that is comparable to the optimal solution. (This is in contrast with the non-sequential counterpart of the bin packing problem in which there exist on-line algorithms for which the number of used bins is within a constant factor of that of the optimal solution.)

So in order to measure the performance of a sequential on-line algorithm in a meaningful way, we adopt an approach extensively used in on-line prediction (the so-called “experts” framework). We define a set of reference forecasters, the so-called *experts*. The performance of the algorithm

SEQUENTIAL ON-LINE BIN PACKING PROBLEM  
WITH EXPERT ADVICE

**Parameters:** set  $\mathcal{E}$  of experts, state space  $\mathcal{S} = [0, 1)$ , action space  $\mathcal{A} = \{0, 1\}$ , nonnegative loss function  $\ell : (\mathcal{A} \times (0, 1] \mid \mathcal{S}) \rightarrow [0, 1)$ , number  $n$  of items.

**Initialization:**  $\widehat{s}_0 = 1$  and  $s_{E,0} = 1$  for all  $E \in \mathcal{E}$ .

For each round  $t = 1, \dots, n$ ,

- (a) each expert forms its action  $f_{E,t} \in \mathcal{A}$ ;
- (b) the forecaster observes the actions of the experts and forms its own decision  $I_t \in \mathcal{A}$ ;
- (c) the next item  $x_t \in (0, 1]$  is revealed;
- (d) the algorithm incurs loss  $\ell(I_t, x_t \mid \widehat{s}_{t-1})$  and each expert incurs loss  $\ell(f_{E,t}, x_t \mid s_{E,t-1})$ . The states of the experts and the algorithm are updated.

Figure 2: Sequential on-line bin packing problem with expert advice.

is evaluated relative to this set of experts, and the goal is to perform asymptotically as well as the best expert from the reference class.

Formally, let  $f_{E,t} \in \{0, 1\}$  be the decision of an expert  $E$  at round  $t$ , where  $E \in \mathcal{E}$  and  $\mathcal{E}$  is the set of the experts. This set may be finite or infinite, we consider both cases below. Similarly we denote the state of expert  $E$  with  $s_{E,t}$  after the  $t$ -th item has been revealed. Then the loss of expert  $E$  at round  $t$  is

$$\ell(f_{E,t}, x_t \mid s_{E,t-1})$$

and the cumulative loss of expert  $E$  is

$$L_{E,n} = \sum_{t=1}^n \ell(f_{E,t}, x_t \mid s_{E,t-1}).$$

The goal of the algorithm is to perform almost as well as the best expert from the reference class  $\mathcal{E}$ . Ideally, the normalized difference of the cumulative losses (the so-called *regret*) should vanish as  $n$  grows, that is, one wishes to achieve

$$\limsup_{n \rightarrow \infty} \frac{1}{n} (\widehat{L}_n - \inf_{E \in \mathcal{E}} L_{E,n}) \leq 0$$

with probability one, regardless of the sequence of items. This property is called *Hannan consistency*, see [5]. The model of sequential on-line bin packing with expert advice is given in Figure 2.

In Section 4 we design sequential on-line bin packing algorithms for two cases. In the first (and simpler) case we assume that the class  $\mathcal{E}$  of experts is finite. In the second case we consider the (infinite) class of experts defined by constant-threshold strategies. But before turning to the on-line problem, we show how the off-line problem can be solved by a simple quadratic-time algorithm.

### 3 Sequential off-line bin packing

As it is well known, most variants of the bin packing problem are NP-hard, including bin packing with rejection [6], and maximum resource bin packing [1]. In this section we show that the sequential bin packing problem is significantly easier. Indeed, we offer an algorithm to find the optimal sequential strategy with time complexity  $O(n^2)$  where  $n$  is the number of the items.

The key property is that after the  $t$ -th item has been received, the  $2^t$  possible sequences of decisions cannot lead to more than  $t$  different states.

**Lemma 1** *For any fixed sequence of items  $x_1, x_2, \dots, x_n$  and for every  $1 \leq t \leq n$ ,*

$$|\mathcal{S}_t| \leq t,$$

where

$$\mathcal{S}_t = \{s : s = s_{\mathbf{I}_t, t}, \mathbf{I}_t \in \{0, 1\}^t\}$$

and  $s_{\mathbf{I}_t, t}$  is the state reached after the sequence  $\mathbf{I}_t$  of decisions.

**Proof:** The proof goes by induction. Note that since  $I_1 = 0$ , we always have  $s_{\mathbf{I}_1, 1} = 1 - x_1$ , and therefore  $|\mathcal{S}_1| = 1$ . Now assume that  $|\mathcal{S}_{t-1}| \leq t - 1$ . At time  $t$ , the state of every sequence of decisions with  $I_t = 0$  belongs to the set  $\mathcal{S}'_t = \{s' : s' = s - \mathbb{I}_{\{s \geq x_t\}} x_t, s \in \mathcal{S}_{t-1}\}$  and the state of those with  $I_t = 1$  becomes  $1 - x_t$ . Therefore,

$$|\mathcal{S}_t| \leq |\mathcal{S}'_t| + 1 \leq |\mathcal{S}_{t-1}| + 1 \leq t$$

as desired.  $\blacksquare$

To describe a computationally efficient algorithm to compute  $\mathbf{I}_n^*$ , we set up a graph with the set of possible states as a vertex set (there are  $O(n^2)$  of them by Lemma 1) and we show that the shortest path on this graph yields the optimal solution of the sequential off-line bin packing problem.

To formalize the problem, consider a finite directed acyclic graph with a set of vertices  $V = \{v_1, \dots, v_{|V|}\}$  and a set of edges  $E = \{e_1, \dots, e_{|E|}\}$ . Each vertex  $v_k = v(s_k, t_k)$  of the graph is defined by a time index  $t_k$  and a state  $s_k \in \mathcal{S}_{t_k}$  and corresponds to state  $s_k$  reachable after  $t_k$  steps. To show the latter dependence, we will write  $v_k \in \mathcal{S}_{t_k}$ . Two vertices  $(v_i, v_j)$  are connected by an edge if and only if  $v_i \in \mathcal{S}_{t-1}$ ,  $v_j \in \mathcal{S}_t$  and state  $v_j$  is reachable from state  $v_i$ . That is, by choosing either action 0 or action 1 in state  $v_i$ , the new state becomes  $v_j$  after item  $x_t$  has been placed. Each edge has a label and a weight: the label corresponds to the action (zero or one) and the weight equals the loss, depending on the initial state, the action, and the size of item. Figure 3 shows the proposed graph. Moreover a sink vertex  $v_{|V|}$  is introduced that is connected with all vertices in  $\mathcal{S}_n$ . These edges have weight equal to the loss of the final states. The losses of these edges only depend on the initial state of the edges. More precisely, for  $(v_i, v_{|V|})$  the loss is  $1 - v_i$ , where  $v_i \in \mathcal{S}_n$ .

Notice that there is a one to one correspondence between paths from  $v_1$  to  $v_{|V|}$  and possible sequences of actions of length  $n$ . Furthermore, the total weight of each path (calculated as the sum of the weights on the edges of the path) is

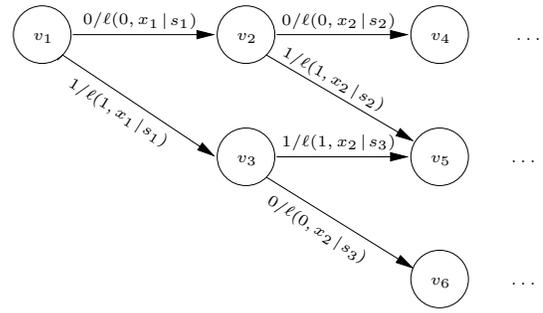


Figure 3: The graph corresponding to the off-line sequential bin packing problem.

equal to the loss of the corresponding sequence of actions. Thus, if we find a path with minimal total weight from  $v_1$  to  $v_{|V|}$ , we also find the optimal sequence of actions for the off-line bin packing problem. It is well known that this can be done in  $O(|V| + |E|)$  time. Now by Lemma 1,  $|V| \leq n(n+1)/2 + 1$ , where the additional vertex accounts for the sink. Moreover it is easy to see that  $|E| \leq n(n-1) + n = n^2$ . Hence the total time complexity of finding the off-line solution is  $O(n^2)$ .

### 4 Sequential on-line bin packing

In this section we study the sequential on-line bin packing problem with expert advice, as described in Section 2. We deal with two special cases. First we consider finite classes of experts (i.e., reference algorithms) without any assumption on the form or structure of the experts. We construct a randomized algorithm that, with large probability, achieves a cumulative loss not larger than that of the best expert plus  $O(n^{2/3} \ln^{1/3} N)$  where  $N = |\mathcal{E}|$  is the number of experts. Then we consider the class of all constant-threshold experts and show that a regret of the order  $O(n^{2/3} \ln^{1/3} n)$  may be achieved with high probability.

The following simple lemma is a key ingredient of the results of this section. It shows that in sequential on-line bin packing the cumulative loss is not sensitive to the initial states in the sense that the cumulative loss depends on the initial state in a minor way.

**Lemma 2** *Let  $i_1, \dots, i_m \in \{0, 1\}$  be a fixed sequence of decisions and let  $x_1, \dots, x_m \in (0, 1)$  be a sequence of items. Let  $s_0, s'_0 \in [0, 1)$  be two different initial states. Finally, let  $s_0, \dots, s_m$  and  $s'_0, \dots, s'_m$  denote the sequences of states generated by  $i_1, \dots, i_m$  and  $x_1, \dots, x_m$  starting from initial states  $s_0$  and  $s'_0$ , respectively. Then*

$$\left| \sum_{t=1}^m \ell(i_t, x_t | s'_{t-1}) - \sum_{t=1}^m \ell(i_t, x_t | s_{t-1}) \right| \leq s'_0 + s_0 \leq 2.$$

**Proof:** Let  $m'$  denote the smallest index for which  $i_{m'} = 1$ .

Note that  $s_{t-1} = s'_{t-1}$  for all  $t > m'$ . Therefore, we have

$$\begin{aligned} & \sum_{t=1}^m \ell(i_t, x_t | s'_{t-1}) - \sum_{t=1}^m \ell(i_t, x_t | s_{t-1}) \\ &= \sum_{t=1}^{m'} \ell(i_t, x_t | s'_{t-1}) - \sum_{t=1}^{m'} \ell(i_t, x_t | s_{t-1}) \\ &= \sum_{t=1}^{m'-1} \ell(0, x_t | s'_{t-1}) - \sum_{t=1}^{m'-1} \ell(0, x_t | s_{t-1}) \\ & \quad + \ell(1, x_{m'} | s'_{m'-1}) - \ell(1, x_{m'} | s_{m'-1}). \end{aligned}$$

Now using the definition of the loss (see (2) and (3)), we write

$$\begin{aligned} & \sum_{t=1}^m \ell(i_t, x_t | s'_{t-1}) - \sum_{t=1}^m \ell(i_t, x_t | s_{t-1}) \\ &= \sum_{t=1}^{m'-1} x_t (\mathbb{I}_{\{s'_{t-1} < x_t\}} - \mathbb{I}_{\{s_{t-1} < x_t\}}) \\ & \quad + s'_{m'-1} - s_{m'-1} \\ &\leq \sum_{t=1}^{m'-1} x_t (1 - \mathbb{I}_{\{s_{t-1} < x_t\}}) + s'_{m'-1} - s_{m'-1} \\ &\leq \sum_{t=1}^{m'-1} x_t (1 - \mathbb{I}_{\{s_{t-1} < x_t\}}) + s'_0 \\ &\leq s_0 + s'_0 \end{aligned}$$

where the next-to-last inequality holds because  $s'_{m'-1} \leq s'_0$  and  $s_{m'-1} \geq 0$ , and the last inequality follows from the fact that

$$\begin{aligned} 0 \leq s_{m'-1} &= s_{m'-2} - \mathbb{I}_{\{s_{m'-2} \geq x_{m'-1}\}} x_{m'-1} \\ &= s_{m'-3} - \mathbb{I}_{\{s_{m'-3} \geq x_{m'-2}\}} x_{m'-2} \\ & \quad - \mathbb{I}_{\{s_{m'-2} \geq x_{m'-1}\}} x_{m'-1} \\ &= s_0 - \sum_{t=1}^{m'-1} \mathbb{I}_{\{s_{t-1} \geq x_t\}} x_t. \end{aligned}$$

Similarly,

$$\begin{aligned} & \sum_{t=1}^m \ell(i_t, x_t | s_{t-1}) - \sum_{t=1}^m \ell(i_t, x_t | s'_{t-1}) \\ & \leq s'_0 + s_0 \end{aligned}$$

and the statement follows.  $\blacksquare$

The following example shows that upper bound of the lemma is tight.

**Example 2** Let  $x_1 = s_0$ ,  $s'_0 < s_0$ , and  $m' = 2$ . Then

$$\begin{aligned} & \sum_{t=1}^m \ell(i_t, x_t | s'_{t-1}) - \sum_{t=1}^m \ell(i_t, x_t | s_{t-1}) \\ &= \ell(0, x_1 | s'_0) + \ell(1, x_2 | s'_1) \\ & \quad - (\ell(0, x_1 | s_0) + \ell(1, x_2 | s_1)) \\ &= \ell(0, s_0 | s'_0) + \ell(1, x_2 | s'_0) \\ & \quad - (\ell(0, s_0 | s_0) + \ell(1, x_2 | 0)) \\ &= s_0 + s'_0 - (0 + 0). \end{aligned}$$

## 4.1 Finite sets of experts

First we consider the on-line sequential bin packing problem when the goal of the algorithm is to keep its cumulative loss close to the best in a finite set of experts. In other words, we assume that the class of experts is finite, say  $|\mathcal{E}| = N$ , but we do not assume any additional structure of the experts. The ideas presented here will be used below when we consider the infinite class of constant-threshold experts.

The proposed algorithm partitions the time period  $t = 1, \dots, n$  into segments of length  $m$  where  $m < n$  is a positive integer whose value will be specified later. This way we obtain  $n' = \lfloor n/m \rfloor$  segments of length  $m$ , and, if  $m$  does not divide  $n$ , an extra segment of length less than  $m$ . At the beginning of each segment, the algorithm selects an expert randomly, according to an exponentially weighted average distribution. During the entire segment, the algorithm follows the advice of the selected expert. By changing actions so rarely, the algorithm achieves a certain synchronization with the chosen expert, since the effect of the difference in the initial states is minor, according to Lemma 2. (A similar idea was used in [7] in a different context.) The algorithm is described in Figure 4. Recall that each expert  $E \in \mathcal{E}$  recommends an action  $f_{E,t} \in \{0, 1\}$  at every time instance  $t = 1, \dots, n$ . Since we have  $N$  experts, we may identify  $\mathcal{E}$  with the set  $\{1, \dots, N\}$ . Thus, experts will be indexed by the positive integers  $i \in \{1, \dots, N\}$ . At the beginning of each segment, the algorithm chooses expert  $i$  randomly, with probability  $p_{i,t}$ , where the distribution  $\mathbf{p}_t = (p_{1,t}, \dots, p_{N,t})$  is specified below. The random selection is made independently for each segment.

The following theorem establishes a performance bound of the algorithm. Recall that  $\widehat{L}_n$  denotes the cumulative loss of the algorithm while  $L_{i,n}$  is that of expert  $i$ .

**Theorem 3** Let  $n, N \geq 1$ ,  $\eta > 0$ ,  $1 \leq m \leq n$ , and  $\delta \in (0, 1)$ . For any sequence  $x_1, \dots, x_n \in (0, 1]$  of items, the cumulative loss  $\widehat{L}_n$  of the randomized strategy defined above satisfies, with probability at least  $1 - \delta$ ,

$$\begin{aligned} & \widehat{L}_n - \min_{i=1, \dots, N} L_{i,n} \\ & \leq \frac{m \ln N}{\eta} + \frac{n\eta}{8} + \sqrt{\frac{nm}{2} \ln \frac{1}{\delta}} + \frac{2n}{m} + 2m \end{aligned}$$

In particular, choosing  $m = (16n/\ln(N/\delta))^{1/3}$  and  $\eta = \sqrt{8m \ln N/n}$ , one has

$$\begin{aligned} & \widehat{L}_n - \min_{i=1, \dots, N} L_{i,n} \\ & \leq \frac{3}{\sqrt[3]{2}} n^{2/3} \ln^{1/3} \frac{N}{\delta} + 4 \left( \frac{2n}{\ln(N/\delta)} \right)^{1/3}. \end{aligned}$$

**Proof:** We introduce an auxiliary quantity, the so-called *hypothetical loss*, defined as the loss the algorithm would suffer if it had been in the same state as the selected expert. This hypothetical loss does not depend on previous decisions of the algorithm. More precisely, the true loss of the algorithm at time instance  $t$  is  $\ell(I_t, x_t | \widehat{s}_t)$  and its hypothetical loss is  $\ell(I_t, x_t | s_{J_t, t})$ . Introducing the notation

$$\ell_{i,t} = \ell(f_{i,t}, x_t | s_{i,t}),$$

SEQUENTIAL ON-LINE BIN PACKING ALGORITHM

**Parameters:** Real number  $\eta > 0$  and  $m \in \mathbb{N}^+$ .

**Initialization:**  $w_{i,0} = 1$  and  $s_{i,0} = 1$  for  $i = 1, \dots, N$ , and  $\widehat{s}_0 = 1$ .

For each round  $t = 1, \dots, n$ ,

- (a) If  $((t - 1) \bmod m) = 0$  then  
 – calculate the updated probability distribution

$$p_{i,t} = \frac{w_{i,t-1}}{\sum_{j=1}^N w_{j,t-1}}$$

for  $i = 1, \dots, N$ ;

- randomly select an expert  $J_t \in \{1, \dots, N\}$  according to the probability distribution  $\mathbf{p}_t = (p_{1,t}, \dots, p_{N,t})$ ;

otherwise, let  $J_t = J_{t-1}$ .

- (b) Follow the chosen expert:  $I_t = f_{J_t,t}$ .  
 (c) The size of next item  $x_t \in (0, 1]$  is revealed.  
 (d) The algorithm incurs loss

$$\ell(I_t, x_t \mid \widehat{s}_{t-1})$$

and each expert  $i$  incurs loss  $\ell(f_{i,t}, x_t \mid s_{i,t-1})$ . The states of the experts and the algorithm are changed.

- (e) Update the weights

$$w_{i,t} = w_{i,t-1} e^{-\eta \ell(f_{i,t}, x_t \mid s_{i,t-1})}$$

for all  $i \in \{1, \dots, N\}$ .

Figure 4: Sequential on-line bin packing algorithm.

the hypothetical loss of the algorithm is just

$$\ell(I_t, x_t \mid s_{J_t,t}) = \ell(f_{J_t,t}, x_t \mid s_{J_t,t}) = \ell_{J_t,t}.$$

Now it follows by a well-known result of randomized on-line prediction (see, e.g., [2, Corollary 4.2]) that the hypothetical loss of the sequential on-line bin packing algorithm satisfies, with probability at least  $1 - \delta$ ,

$$\begin{aligned} & \sum_{t=1}^n \ell_{J_t,t} - \min_{i=1, \dots, N} \sum_{t=1}^n \ell_{i,t} \\ & \leq m \left( \frac{\ln N}{\eta} + \frac{n'\eta}{8} + \sqrt{\frac{n'}{2} \ln \frac{1}{\delta}} \right) + m, \end{aligned} \quad (4)$$

where  $n' = \lfloor \frac{n}{m} \rfloor$  and the last  $m$  term comes from bounding the difference on the last, not necessarily complete segment.

Now we may decompose the regret as follows:

$$\begin{aligned} & \widehat{L}_n - \min_{i=1, \dots, N} L_{i,n} \\ & = \left( \widehat{L}_n - \sum_{t=1}^n \ell_{J_t,t} \right) \\ & \quad + \left( \sum_{t=1}^n \ell_{J_t,t} - \min_{i=1, \dots, N} L_{i,n} \right). \end{aligned}$$

The second term on the right-hand side is bounded using (4). To bound the first term, observe that by Lemma 2,

$$\begin{aligned} & \widehat{L}_n - \min_{i=1, \dots, N} L_{i,n} \\ & = \sum_{t=1}^n \ell(I_t, x_t \mid \widehat{s}_{t-1}) - \sum_{t=1}^n \ell(I_t, x_t \mid s_{J_t,t-1}) \\ & \leq m + \sum_{s=0}^{n'-1} \sum_{t=1}^m (\ell(I_{sm+t}, x_{sm+t} \mid \widehat{s}_{sm+t-1}) \\ & \quad - \ell(I_{sm+t}, x_{sm+t} \mid s_{J_{sm+t-1}, sm+t-1})) \\ & \leq m + 2n' \end{aligned}$$

where in the second inequality we bounded the difference on the last segment separately. ■

## 4.2 Constant-threshold experts

Now we are prepared to address the sequential on-line bin packing problem when the goal is to perform almost as well as the best in the class of all constant-threshold strategies. Recall that a constant-threshold strategy is parameterized by a number  $p \in (0, 1]$  and it opens a new bin if and only if the remaining empty space in the bin is less than  $p$ . More precisely, if the state of the algorithm defined by expert with parameter  $p$  is  $s_{p,t-1}$ , then at time  $t$  the expert's advice is  $\mathbb{I}_{\{s_{p,t-1} < p\}}$ . To simplify notation, we will refer to each expert with its parameter, and, similarly to the previous section,  $f_{p,t}$  and  $s_{p,t}$  will denote the decision of expert  $p$  at time  $t$ , and its state after the decision, respectively.

The difficulty in this setup is that there are uncountably many constant-threshold experts. In this section we provide a solution to this problem by reducing it to the case of finite expert classes. The main observation that enables this reduction is that on any sequence of  $n$  items, experts can exhibit only a finite number of different behaviors. In a sense, the “effective” number of experts is not too large and this fact may be exploited by the algorithm.

For  $t = 1, \dots, n$  we call two experts  $t$ -indistinguishable (with respect to the sequence of items  $x_1, \dots, x_t$ ) if their decision sequences are identical up to time  $t$ . This property defines a natural partitioning of the class of experts into maximal  $t$ -indistinguishable sets, where any two experts that belong to the same set are  $t$ -indistinguishable, and experts from different sets are not  $t$ -indistinguishable. Obviously, there are no more than  $2^t$  maximal  $t$ -indistinguishable sets. This bound, although finite, is still too large to be useful. However, it turns out that the number of maximal  $t$ -indistinguishable sets only grows quadratically with  $t$ .

The first step in proving this fact is the next lemma that shows that the maximal  $t$ -indistinguishable expert sets are intervals.

**Lemma 4** *Let  $1 \geq p > r > 0$  be such that expert  $p$  and expert  $r$  are  $t$ -indistinguishable. Then for any  $p > q > r$  expert  $q$  is  $t$ -indistinguishable from both experts  $p$  and  $r$ . Thus, the maximal  $t$ -indistinguishable expert sets form subintervals of  $(0, 1]$ .*

**Proof:** By the assumption of the lemma the decision sequences of experts  $p$  and  $r$  coincide, that is,

$$f_{p,u} = f_{r,u} \quad \text{and} \quad s_{p,u} = s_{r,u}$$

for all  $u = 1, 2, \dots, t$ . Let  $t_1, t_2, \dots$  denote the time instances when expert  $p$  (or expert  $r$ ) assigns the next item to the next empty bin (i.e.,  $f_{p,u} = 1$  for  $u = t_1, t_2, \dots$ ). If expert  $q$  also decides 1 at time  $t_k$  for some  $k$ , then it will decide 0 for  $t = t_k + 1, \dots, t_{k+1} - 1$  since so does expert  $p$  and  $p > q$ , and will decide 1 at time  $t_{k+1}$  as  $q > r$ . Thus the decision sequence of expert  $q$  coincides with that of expert  $p$  and  $r$  for time instances  $t_k + 1, \dots, t_{k+1}$  in this case. Since all experts start with the empty bin at time 0, the statement of the lemma follows by induction. ■

Based on the lemma we can identify the  $t$ -indistinguishable sets by their end points. Let  $\mathcal{Q}_t = \{q_{1,t}, \dots, q_{N_t,t}\}$  denote the set of the end points after receiving  $t$  items, where  $N_t = |\mathcal{Q}_t|$  is the number of maximal  $t$ -indistinguishable sets, and  $q_{0,t} = 0 < q_{1,t} < q_{2,t} < \dots < q_{N_t,t} = 1$ . Then the  $t$ -indistinguishable sets are  $(q_{k-1,t}, q_{k,t}]$  for  $k = 1, \dots, N_t$ . The next result shows that the number of maximal  $t$ -indistinguishable sets cannot grow too fast.

**Lemma 5** *The number of the maximal  $t$ -indistinguishable sets is at most quadratic in the number of the items  $t$ . More precisely,  $N_t \leq 1 + (t-1)t/2$  for any  $1 \leq t \leq n$ .*

**Proof:** The proof is by induction. First,  $N_1 = 1$  (and  $\mathcal{Q}_1 = \{1\}$ ) since the first decision of each expert is 1. Now assume that  $N_{t-1} \leq 1 + (t-2)(t-1)/2$  for some  $1 \leq t \leq n-1$ . When the next item  $x_t$  arrives, an expert  $p$  with state  $s$  decides 1 in the next step if and only if  $0 \leq s - x_t < p$ . Therefore, as each expert belonging to the same indistinguishable set has the same state, the  $k$ -th maximal  $(t-1)$ -indistinguishable interval with state  $s$  is split into two subintervals if and only if  $q_{k-1,t-1} < s - x_t \leq q_{k,t-1}$  (experts in this interval with parameters larger than  $s - x_t$  will form one subset, and the ones with parameter at most  $s - x_t$  will form the other one). As the number of possible states at time  $t-1$  is at most  $t-1$  by Lemma 1, it follows that at most  $t-1$  intervals can be split, and so  $N_t \leq N_{t-1} + t - 1 \leq 1 + (t-1)t/2$ , where the second inequality holds by the induction hypothesis. ■

This lemma makes it possible to apply our earlier algorithm for the case of finite expert classes. However, note that the number of “distinguishable” experts, that is, the number of the maximal indistinguishable sets, constantly grows with time, and each indistinguishable set contains a continuum number of experts. Therefore we need to redefine the algorithm carefully. This may be done by a two-level random

#### SEQUENTIAL ON-LINE BIN PACKING ALGORITHM WITH CONSTANT-THRESHOLD EXPERTS

**Parameters:**  $\eta > 0$  and  $m \in \mathbb{N}^+$ .

**Initialization:**  $w_{0,1} = 1$ ,  $N_1 = 1$ ,  $\mathcal{Q}_1 = \{1\}$ ,  $s_{1,0} = 1$  and  $\hat{s}_0 = 1$ .

For each round  $t = 1, \dots, n$ ,

- (a) If  $((t-1) \bmod m) = 0$  then
  - for  $i = 1, \dots, N_t$ , compute the probabilities
 
$$p_{i,t} = \frac{w_{i,t-1}}{\sum_{j=1}^{N_t} w_{j,t-1}};$$
  - randomly select an interval  $J_t \in \{1, \dots, N_t\}$  according to the probability distribution  $\mathbf{p}_t = (p_{1,t}, \dots, p_{N_t,t})$ ;
  - choose an expert  $p_t$  uniformly from the interval  $(q_{J_t-1,t}, q_{J_t,t}]$ ;
- otherwise, let  $p_t = p_{t-1}$ .
- (b) Follow the decision of expert  $p_t$ :  $I_t = f_{p_t,t}$ .
- (c)  $x_t \in (0, 1]$ , the size of the next item is revealed.
- (d) The algorithm incurs loss

$$\ell(I_t, x_t \mid \hat{s}_{t-1})$$

and each expert  $p \in (0, 1]$  incurs loss  $\ell(f_{p,t}, x_t \mid s_{p,t-1})$ , where  $p \in [0, 1]$ .

- (e) Compute the state  $\hat{s}_t$  of the algorithm by (2), and calculate the auxiliary weights and states of the expert sets for all  $i = 1, \dots, N_t$  by

$$\begin{aligned} \tilde{w}_{i,t} &= w_{i,t-1} e^{-\eta \ell(f_{i,t}, x_t \mid s_{i,t-1})} \\ \tilde{s}_{i,t} &= f_{i,t}(1 - x_t) \\ &\quad + (1 - f_{i,t})(s_{i,t} - \mathbb{I}_{\{s_{i,t} \geq x_t\}}). \end{aligned}$$

- (f) Update the end points of the intervals:

$$\mathcal{Q}_{t+1} = \mathcal{Q}_t \cup \bigcup_{i=1}^{N_t} \{\tilde{s}_{i,t} : q_{i-1,t} < \tilde{s}_{i,t} \leq q_{i,t}\}$$

and  $N_{t+1} = |\mathcal{Q}_{t+1}|$ .

- (g) Assign the new states and weights to the  $(t+1)$ -indistinguishable sets

$$s_{i,t+1} = \tilde{s}_{j,t} \quad \text{and} \quad w_{i,t+1} = \tilde{w}_{j,t}$$

for all  $i = 1, \dots, N_{t+1}$  and  $j = 1, \dots, N_t$  such that  $q_{j-1,t} < q_{i,t+1} \leq q_{j,t}$ .

Figure 5: Sequential on-line bin packing algorithm with constant-threshold experts.

choice of the experts: first we choose an indistinguishable expert set, then we pick one expert from this set randomly. The resulting algorithm is given in Figure 5.

Up to step (e) the algorithm is essentially the same as in the case of finitely many experts. The two-level random choice of the expert is performed in step (a). In step (f) we update the  $t$ -indistinguishable sets, and usually introduce new indistinguishable expert sets. Because of these new expert sets, the update of the weights  $w_{i,t}$  and the states  $s_{i,t}$  are performed in two steps, (e) and (g), where the actual update is made in step (e), and reordering of these quantities according to the new indistinguishable sets is performed in step (g) together with the introduction of the weights and states for the newly formed expert sets.

The performance and complexity of the algorithm is given in the next theorem.

**Theorem 6** *Let  $N = 1+n(n-1)/2$ ,  $m = (16n/\ln(n^2/\delta))^{1/3}$  and  $\eta = 4\sqrt{m \ln n/n}$  and  $\delta \in (0, 1)$ . Then the regret of the algorithm defined above is bounded, with probability at least  $1 - \delta$ , by*

$$\begin{aligned} \widehat{L}_n - \inf_{p \in (0,1)} L_{p,n} \\ \leq \frac{3}{\sqrt[3]{2}} n^{2/3} \ln^{1/3} \frac{n^2}{\delta} + 4 \left( \frac{2n}{\ln(n^2/\delta)} \right)^{1/3}. \end{aligned}$$

Moreover, the algorithm can be implemented with time complexity  $O(n^3)$  and space complexity  $O(n^2)$ .

**Proof:** It is easy to see that the two-level choice of the expert  $p_t$  ensures that the algorithm is the same as for the finite expert class with the experts defined by  $\mathcal{Q}_n$ . Thus, Theorem 3 can be used to bound the regret, where the number of experts is  $N_t$ . By Lemma 5, the latter is bounded by  $N < n^2$ , which finishes the proof of the first statement.

For the second part note that the algorithm has to store the states, the intervals, the weights and the probabilities, each on the order of  $O(n^2)$  based on Lemma 5. Concerning time complexity, the algorithm has to update the weights and states in each round (requiring  $O(n^2)$  computations per round), and has to compute the probabilities in every  $m$  step, which requires  $O(n^3/m)$  computations. Thus the time complexity of the algorithm is  $O(n^3)$ . ■

The next example reveals that the loss of the best expert can be arbitrarily far from that of the optimal sequential off-line packing.

**Example 3** *Let the sequence of items be*

$$\left( \underbrace{\varepsilon, 1-\varepsilon, \varepsilon, 1-\varepsilon, \dots, \varepsilon, 1-\varepsilon}_{2k}, \underbrace{\varepsilon, 1, 1, \dots, 1}_k \right),$$

where the number of items is  $n = 3k + 1$  and  $0 < \varepsilon < 1$ . An optimal sequential off-line packing is achieved if we drop anyone of the  $\varepsilon$  terms; then the total loss is  $\varepsilon$ . In contrast to this, the loss of the constant-threshold experts is  $1 - \varepsilon + k$  independently of the choice of the parameter  $p$ . Namely, if  $p \leq 1 - \varepsilon$  then the loss is 0 for the first  $2k$  items, but after the algorithm is stuck and suffers  $k + 1 - \varepsilon$  loss. If  $p > 1 - \varepsilon$ , then the loss is  $k$  for the first  $2k$  items and after that  $1 - \varepsilon$  for the rest of the sequence.

## 5 Conclusions

In this paper we provide an extension of the classical bin packing problems to an on-line sequential scenario. In this setting items are received one by one, and before the size of the next item is revealed, the decision maker needs to decide whether the next item is packed in the currently open bin or the bin is closed and a new bin is opened. If the new item doesn't fit, it is lost. If a bin is closed, the remaining free space in the bin accounts for a loss. The goal of the decision maker is to minimize the loss accumulated over  $n$  periods.

As the main result of the paper, we give an algorithm that has a cumulative loss not much larger than any finite set of reference algorithms, and, more importantly, another algorithm that has a cumulative loss not much larger than any strategy that uses a fixed threshold at each step to decide whether a new bin is opened. An interesting aspect of the problem is that the loss function has an (unbounded) memory. The presented solutions rely on the fact that one can "synchronize" the loss function in the sense that no matter in what state an algorithm is started, its loss may change only by a small additive constant. The second result is obtained by a covering of the uncountable set of constant-threshold experts such that the cardinality of the chosen finite set of experts grows only quadratically with the sequence length. The approach in the paper can easily be extended to any control problem where the loss function has such a synchronizable property.

## References

- [1] J. Boyar, L. Epstein, L.M. Favrholdt, J.S. Kohrt, K.S. Larsen, M.M. Pedersen, and S. Wøhlk. The maximum resource bin packing problem. *Theoretical Computer Science*, 362:127–139, 2006.
- [2] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [3] E.G. Coffman, M.R. Garey, and D.S. Johnson. *Approximation algorithms for bin packing: a survey*. In *Approximation algorithms for NP-hard problems*, pp. 46–93, PWS Publishing Co., Boston, MA, 1997.
- [4] E. Even-Dar, S.M. Kakade, and Y. Mansour. Experts in a Markov Decision Process. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pp. 401–408. MIT Press, Cambridge, MA, 2005.
- [5] J. Hannan. Approximation to Bayes risk in repeated plays. In M. Dresher, A. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pp. 97–139. Princeton University Press, 1957.
- [6] Y. He and Gy. Dósa. Bin packing and covering problems with rejection. *Lecture Notes in Computer Science 3595*, pp. 885–894, 2005.
- [7] N. Merhav, E. Ordentlich, G. Seroussi, and M. J. Weinberger. On sequential strategies for loss functions with memory. *IEEE Transactions on Information Theory*, 48:1947–1958, 2002.
- [8] S.S. Seiden. On the online bin packing problem. in *Proceedings of the 28th International Colloquium on Automata, Languages and Programming*, pp. 237 - 248, 2001.

---

# Time Varying Undirected Graphs

---

Shuheng Zhou, John Lafferty and Larry Wasserman\*

Carnegie Mellon University

{szhou, lafferty}@cs.cmu.edu, larry@stat.cmu.edu

## Abstract

Undirected graphs are often used to describe high dimensional distributions. Under sparsity conditions, the graph can be estimated using  $\ell_1$  penalization methods. However, current methods assume that the data are independent and identically distributed. If the distribution, and hence the graph, evolves over time then the data are not longer identically distributed. In this paper, we show how to estimate the sequence of graphs for non-identically distributed data, where the distribution evolves over time.

## 1 Introduction

Let  $Z = (Z_1, \dots, Z_p)^T$  be a random vector with distribution  $P$ . The distribution can be represented by an undirected graph  $G = (V, F)$ . The vertex set  $V$  has one vertex for each component of the vector  $Z$ . The edge set  $F$  consists of pairs  $(j, k)$  that are joined by an edge. If  $Z_j$  is independent of  $Z_k$  given the other variables, then  $(j, k)$  is not in  $F$ . When  $Z$  is Gaussian, missing edges correspond to zeroes in the inverse covariance matrix  $\Sigma^{-1}$ . Suppose we have independent, identically distributed data  $D = (Z^1, \dots, Z^t, \dots, Z^n)$  from  $P$ . When  $p$  is small, the graph may be estimated from  $D$  by testing which partial correlations are not significantly different from zero [DP04]. When  $p$  is large, estimating  $G$  is much more difficult. However, if the graph is sparse and the data are Gaussian, then several methods can successfully estimate  $G$ ; see [MB06, BGd08, FHT07, LF07, BL08, RBLZ07].

All these methods assume that the graphical structure is stable over time. But it is easy to imagine cases where such stability would fail. For example,  $Z^t$  could

represent a large vector of stock prices at time  $t$ . The conditional independence structure between stocks could easily change over time. Another example is gene expression levels. As a cell moves through its metabolic cycle, the conditional independence relations between proteins could change.

In this paper we develop a nonparametric method for estimating time varying graphical structure for multivariate Gaussian distributions using  $\ell_1$  regularization method. We show that, as long as the covariances change smoothly over time, we can estimate the covariance matrix well (in predictive risk) even when  $p$  is large. We make the following theoretical contributions: (i) nonparametric predictive risk consistency and rate of convergence of the covariance matrices, (ii) consistency and rate of convergence in Frobenius norm of the inverse covariance matrix, (iii) large deviation results for covariance matrices for non-identically distributed observations, and (iv) conditions that guarantee smoothness of the covariances. In addition, we provide simulation evidence that we can recover graphical structure. We believe these are the first such results on time varying undirected graphs.

## 2 The Model and Method

Let  $Z^t \sim N(0, \Sigma(t))$  be independent. It will be useful to index time as  $t = 0, 1/n, 2/n, \dots, 1$  and thus the data are  $D_n = (Z^t : t = 0, 1/n, \dots, 1)$ . Associated with each  $Z^t$  is its undirected graph  $G(t)$ . Under the assumption that the law  $\mathcal{L}(Z^t)$  of  $Z^t$  changes smoothly, we estimate the graph sequence  $G(1), G(2), \dots$ . The graph  $G(t)$  is determined by the zeroes of  $\Sigma(t)^{-1}$ . This method can be used to investigate a simple time series model of the form:  $W^0 \sim N(0, \Sigma(0))$ , and

$$W^t = W^{t-1} + Z^t, \text{ where } Z^t \sim N(0, \Sigma(t)).$$

Ultimately, we are interested in the general time series model where the  $Z^t$ 's are dependent and the graphs change over time. For simplicity, however, we assume independence but allow the graphs to change. Indeed, it is the changing graph, rather than the dependence, that is the biggest hurdle to deal with.

In the iid case, recent work [BGd08, FHT07] has considered  $\ell_1$ -penalized maximum likelihood estimators

---

\*This research was supported in part by NSF grant CCF-0625879. SZ thanks Alan Frieze and Giovanni Leoni for helpful discussions on sparsity and smoothness of functions. We thank J. Friedman, T. Hastie and R. Tibshirani for making GLASSO publicly available, and anonymous reviewers for their constructive comments.

over the entire set of positive definite matrices,

$$\widehat{\Sigma}_n = \arg \min_{\Sigma > 0} \{ \text{tr}(\Sigma^{-1} \widehat{S}_n) + \log |\Sigma| + \lambda |\Sigma^{-1}|_1 \} \quad (1)$$

where  $\widehat{S}_n$  is the sample covariance matrix. In the non-iid case our approach is to estimate  $\Sigma(t)$  at time  $t$  by

$$\widehat{\Sigma}_n(t) = \arg \min_{\Sigma > 0} \{ \text{tr}(\Sigma^{-1} \widehat{S}_n(t)) + \log |\Sigma| + \lambda |\Sigma^{-1}|_1 \}$$

$$\text{where } \widehat{S}_n(t) = \frac{\sum_s w_{st} Z_s Z_s^T}{\sum_s w_{st}} \quad (2)$$

is a weighted covariance matrix, with weights  $w_{st} = K\left(\frac{|s-t|}{h_n}\right)$  given by a symmetric nonnegative function kernel over time; in other words,  $\widehat{S}_n(t)$  is just the kernel estimator of the covariance at time  $t$ . An attraction of this approach is that it can use existing software for covariance estimation in the iid setting.

### 2.1 Notation

We use the following notation throughout the rest of the paper. For any matrix  $W = (w_{ij})$ , let  $|W|$  denote the determinant of  $W$ ,  $\text{tr}(W)$  the trace of  $W$ . Let  $\varphi_{\max}(W)$  and  $\varphi_{\min}(W)$  be the largest and smallest eigenvalues, respectively. We write  $W^\sim = \text{diag}(W)$  for a diagonal matrix with the same diagonal as  $W$ , and  $W^\diamond = W - W^\sim$ . The matrix Frobenius norm is given by  $\|W\|_F = \sqrt{\sum_i \sum_j w_{ij}^2}$ . The operator norm  $\|W\|_2$  is given by  $\varphi_{\max}(WW^T)$ . We write  $|\cdot|_1$  for the  $\ell_1$  norm of a matrix vectorized, i.e., for a matrix  $|W|_1 = \|\text{vec}W\|_1 = \sum_i \sum_j |w_{ij}|$ , and write  $\|W\|_0$  for the number of non-zero entries in the matrix. We use  $\Theta(t) = \Sigma^{-1}(t)$ .

### 3 Risk Consistency

In this section we define the loss and risk. Consider estimates  $\widehat{\Sigma}_n(t)$  and  $\widehat{G}_n(t) = (V, \widehat{F}_n)$ . The first risk function is

$$U(G(t), \widehat{G}_n(t)) = \mathbf{E}L(G(t), \widehat{G}_n(t)) \quad (3)$$

where  $L(G(t), \widehat{G}_n(t)) = |F(t) \Delta \widehat{F}_n(t)|$ , that is, the size of the symmetric difference between two edge sets. We say that  $\widehat{G}_n(t)$  is *sparsistent* if  $U(G(t), \widehat{G}_n(t)) \xrightarrow{P} 0$  as  $n \rightarrow \infty$ .

The second risk is defined as follows. Let  $Z \sim N(0, \Sigma_0)$  and let  $\Sigma$  be a positive definite matrix. Let

$$R(\Sigma) = \text{tr}(\Sigma^{-1} \Sigma_0) + \log |\Sigma|. \quad (4)$$

Note that, up to an additive constant,

$$R(\Sigma) = -2E_0(\log f_\Sigma(Z)),$$

where  $f_\Sigma$  is the density for  $N(0, \Sigma)$ . We say that  $\widehat{G}_n(t)$  is *persistent* [GR04] with respect to a class of positive definite matrices  $\mathcal{S}_n$  if  $R(\widehat{\Sigma}_n) - \min_{\Sigma \in \mathcal{S}_n} R(\Sigma) \xrightarrow{P} 0$ . In the iid case,  $\ell_1$  regularization yields a persistent estimator, as we now show.

The maximum likelihood estimate minimizes

$$\widehat{R}_n(\Sigma) = \text{tr}(\Sigma^{-1} \widehat{S}_n) + \log |\Sigma|,$$

where  $\widehat{S}_n$  is the sample covariance matrix. Minimizing  $\widehat{R}_n(\Sigma)$  without constraints gives  $\widehat{\Sigma}_n = \widehat{S}_n$ . We would like to minimize  $\widehat{R}_n(\Sigma)$  subject to  $\|\Sigma^{-1}\|_0 \leq L$ . This would give the “best” sparse graph  $G$ , but it is not a convex optimization problem. Hence we estimate  $\widehat{\Sigma}_n$  by solving a convex relaxation problem as written in (1) instead. Algorithms for carrying out this optimization are given by [BGd08, FHT07]. Given  $L_n, \forall n$ , let

$$\mathcal{S}_n = \{ \Sigma : \Sigma \succ 0, |\Sigma^{-1}|_1 \leq L_n \}. \quad (5)$$

We define the oracle estimator and write (1) as (7)

$$\Sigma^*(n) = \arg \min_{\Sigma \in \mathcal{S}_n} R(\Sigma), \quad (6)$$

$$\widehat{\Sigma}_n = \arg \min_{\Sigma \in \mathcal{S}_n} \widehat{R}_n(\Sigma). \quad (7)$$

Note that one can choose to only penalize off-diagonal elements of  $\Sigma^{-1}$  as in [RBLZ07], if desired. We have the following result, whose proof appears in Section 3.2.

**Theorem 1** Suppose that  $p_n \leq n^\xi$  for some  $\xi \geq 0$  and

$$L_n = o\left(\frac{n}{\log p_n}\right)^{1/2}$$

for (5). Then for the sequence of empirical estimators as defined in (7) and  $\Sigma^*(n), \forall n$  as in (6),

$$R(\widehat{\Sigma}_n) - R(\Sigma^*(n)) \xrightarrow{P} 0.$$

### 3.1 Risk Consistency for the Non-identical Case

In the non-iid case we estimate  $\Sigma(t)$  at time  $t \in [0, 1]$ . Given  $\Sigma(t)$ , let

$$\widehat{R}_n(\Sigma(t)) = \text{tr}(\Sigma(t)^{-1} \widehat{S}_n(t)) + \log |\Sigma(t)|.$$

For a given  $\ell_1$  bound  $L_n$ , we define  $\widehat{\Sigma}_n(t)$  as the minimizer of  $\widehat{R}_n(\Sigma)$  subject to  $\Sigma \in \mathcal{S}_n$ ,

$$\widehat{\Sigma}_n(t) = \arg \min_{\Sigma \in \mathcal{S}_n} \{ \text{tr}(\Sigma^{-1} \widehat{S}_n(t)) + \log |\Sigma| \} \quad (8)$$

where  $\widehat{S}_n(t)$  is given in (2), with  $K(\cdot)$  a symmetric non-negative function with compact support:

**A1** The kernel function  $K$  has a bounded support  $[-1, 1]$ .

**Lemma 2** Let  $\Sigma(t) = [\sigma_{jk}(t)]$ . Suppose the following conditions hold:

1. There exists  $C_0 > 0, C$  such that  $\max_{j,k} \sup_t |\sigma'_{jk}(t)| \leq C_0$  and  $\max_{j,k} \sup_t |\sigma''_{jk}(t)| \leq C$ .
2.  $p_n \leq n^\xi$  for some  $\xi \geq 0$ .
3.  $h_n \asymp n^{-1/3}$ .

Then  $\max_{j,k} |\widehat{S}_n(t, j, k) - \Sigma(t, j, k)| = O_P\left(\frac{\sqrt{\log n}}{n^{1/3}}\right)$  for all  $t > 0$ .

**Proof:** By the triangle inequality,

$$|\widehat{S}_n(t, j, k) - \Sigma(t, j, k)| \leq |\widehat{S}_n(t, j, k) - \mathbf{E}\widehat{S}_n(t, j, k)| + |\mathbf{E}\widehat{S}_n(t, j, k) - \Sigma(t, j, k)|.$$

In Lemma 14 we show that

$$\max_{j,k} \sup_t |\mathbf{E}\widehat{S}_n(t, j, k) - \Sigma(t, j, k)| = O(C_0 h_n).$$

In Lemma 15, we show that

$$\mathbf{P}\left(|\widehat{S}_n(t, j, k) - \mathbf{E}\widehat{S}_n(t, j, k)| > \epsilon\right) \leq \exp\{-c_1 h_n n \epsilon^2\}$$

for some  $c_1 > 0$ . Hence,

$$\mathbf{P}\left(\max_{j,k} |\widehat{S}_n(t, j, k) - \mathbf{E}\widehat{S}_n(t, j, k)| > \epsilon\right) \leq \exp\{-nh_n(C\epsilon^2 - 2\xi \log n/(nh_n))\} \quad \text{and} \quad (9)$$

$$\max_{j,k} |\widehat{S}_n(t, j, k) - \mathbf{E}\widehat{S}_n(t, j, k)| = O_P\left(\sqrt{\frac{\log n}{nh_n}}\right).$$

Hence the result holds for  $h_n \asymp n^{-1/3}$ .  $\square$

With the use of Lemma 2, the proof of the following follows the same lines as that of Theorem 1.

**Theorem 3** *Suppose all conditions in Lemma 2 and the following hold:*

$$L_n = o\left(n^{1/3}/\sqrt{\log n}\right). \quad (10)$$

Then,  $\forall t > 0$ , for the sequence of estimators as in (8),

$$R(\widehat{\Sigma}_n(t)) - R(\Sigma^*(t)) \xrightarrow{P} 0.$$

**Remark 4** *If a local linear smoother is substituted for a kernel smoother, the rate can be improved from  $n^{1/3}$  to  $n^{2/5}$  as the bias will be bounded as  $O(h^2)$  in (3.1).*

**Remark 5** *Suppose that  $\forall i, j$ , if  $\theta_{ij} \neq 0$ , we have  $\theta_{ij} = \Omega(1)$ . Then Condition (10) allows that  $|\Theta|_1 = L_n$ ; hence if  $p = n^\xi$  and  $\xi < 1/3$ , we have that  $\|\Theta\|_0 = \Omega(p)$ . Hence the family of graphs that we can guarantee persistency for, although sparse, is likely to include connected graphs, for example, when  $\Omega(p)$  edges were formed randomly among  $p$  nodes.*

The smoothness condition in Lemma 2 is expressed in terms of the elements of  $\Sigma(t) = [\sigma_{ij}(t)]$ . It might be more natural to impose smoothness on  $\Theta(t) = \Sigma(t)^{-1}$  instead. In fact, smoothness of  $\Theta_t$  implies smoothness of  $\Sigma_t$  as the next result shows. Let us first specify two assumptions. We use  $\sigma_i^2(x)$  as a shorthand for  $\sigma_{ii}(x)$ .

**Definition 6** *For a function  $u : [0, 1] \rightarrow \mathbf{R}$ , let  $\|u\|_\infty = \sup_{x \in [0, 1]} |u(x)|$ .*

**A2** *There exists some constant  $S_0 < \infty$  such that*

$$\max_{i=1, \dots, p} \sup_{t \in [0, 1]} |\sigma_i(t)| \leq S_0 < \infty, \quad \text{hence} \quad (11)$$

$$\max_{i=1, \dots, p} \|\sigma_i\|_\infty \leq S_0. \quad (12)$$

**A3** *Let  $\theta_{ij}(t), \forall i, j$ , be twice differentiable functions such that  $\theta'_{ij}(t) < \infty$  and  $\theta''_{ij}(t) < \infty, \forall t \in [0, 1]$ . In addition, there exist constants  $S_1, S_2 < \infty$  such that*

$$\sup_{t \in [0, 1]} \sum_{k=1}^p \sum_{\ell=1}^p \sum_{i=1}^p \sum_{j=1}^p |\theta'_{ki}(t) \theta'_{\ell j}(t)| \leq S_1 \quad (13)$$

$$\sup_{t \in [0, 1]} \sum_{k=1}^p \sum_{\ell=1}^p |\theta''_{k\ell}(t)| \leq S_2, \quad (14)$$

where the first inequality guarantees that  $\sup_{t \in [0, 1]} \sum_{k=1}^p \sum_{\ell=1}^p |\theta'_{k\ell}(t)| < \sqrt{S_1} < \infty$ .

**Lemma 7** *Denote the elements of  $\Theta(t) = \Sigma(t)^{-1}$  by  $\theta_{jk}(t)$ . Under A 2 and A 3, the smoothness condition in Lemma 2 holds.*

The proof is in Section 6. In Section 7, we show some preliminary results on achieving upper bounds on quantities that appear in Condition 1 of Lemma 2 through the sparsity level of the inverse covariance matrix, i.e.,  $\|\Theta_t\|_0, \forall t \in [0, 1]$ .

### 3.2 Proof of Theorem 1

Note that  $\forall n, \sup_{\Sigma \in \mathcal{S}_n} |R(\Sigma) - \widehat{R}_n(\Sigma)| \leq$

$$\sum_{j,k} |\Sigma_{jk}^{-1}| |\widehat{S}_n(j, k) - \Sigma_0(j, k)| \leq \delta_n |\Sigma^{-1}|_1,$$

where it follows from [RBLZ07] that

$$\delta_n = \max_{j,k} |\widehat{S}_n(j, k) - \Sigma_0(j, k)| = O_P(\sqrt{\log p/n}).$$

Hence, minimizing over  $\mathcal{S}_n$  with  $L_n = o\left(\frac{n}{\log p_n}\right)^{1/2}$ ,  $\sup_{\Sigma \in \mathcal{S}_n} |R(\Sigma) - \widehat{R}_n(\Sigma)| = o_P(1)$ . By the definitions of  $\Sigma^*(n) \in \mathcal{S}_n$  and  $\widehat{\Sigma}_n \in \mathcal{S}_n$ , we immediately have  $R(\Sigma^*(n)) \leq R(\widehat{\Sigma}_n)$  and  $\widehat{R}_n(\widehat{\Sigma}_n) \leq \widehat{R}_n(\Sigma^*(n))$ ; thus

$$\begin{aligned} 0 &\leq R(\widehat{\Sigma}_n) - R(\Sigma^*(n)) \\ &= R(\widehat{\Sigma}_n) - \widehat{R}_n(\widehat{\Sigma}_n) + \widehat{R}_n(\widehat{\Sigma}_n) - R(\Sigma^*(n)) \\ &\leq R(\widehat{\Sigma}_n) - \widehat{R}_n(\widehat{\Sigma}_n) + \widehat{R}_n(\Sigma^*(n)) - R(\Sigma^*(n)) \end{aligned}$$

Using the triangle inequality and  $\widehat{\Sigma}_n, \Sigma^*(n) \in \mathcal{S}_n$ ,

$$\begin{aligned} |R(\widehat{\Sigma}_n) - R(\Sigma^*(n))| &\leq \\ &|R(\widehat{\Sigma}_n) - \widehat{R}_n(\widehat{\Sigma}_n) + \widehat{R}_n(\Sigma^*(n)) - R(\Sigma^*(n))| \\ &\leq |R(\widehat{\Sigma}_n) - \widehat{R}_n(\widehat{\Sigma}_n)| + |\widehat{R}_n(\Sigma^*(n)) - R(\Sigma^*(n))| \\ &\leq 2 \sup_{\Sigma \in \mathcal{S}_n} |R(\Sigma) - \widehat{R}_n(\Sigma)|. \quad \text{Thus } \forall \epsilon > 0, \end{aligned}$$

the event  $\left\{|R(\widehat{\Sigma}_n) - R(\Sigma^*(n))| > \epsilon\right\}$  is contained in the event  $\left\{\sup_{\Sigma \in \mathcal{S}_n} |R(\Sigma) - \widehat{R}_n(\Sigma)| > \epsilon/2\right\}$ . Thus, for  $L_n = o((n/\log n)^{1/2})$ , and  $\forall \epsilon > 0$ , as  $n \rightarrow \infty$ ,

$$\begin{aligned} &\mathbf{P}\left(\left|R(\widehat{\Sigma}_n) - R(\Sigma^*(n))\right| > \epsilon\right) \leq \\ &\mathbf{P}\left(\sup_{\Sigma \in \mathcal{S}_n} |R(\Sigma) - \widehat{R}_n(\Sigma)| > \epsilon/2\right) \rightarrow 0. \quad \square \end{aligned}$$

## 4 Frobenius Norm Consistency

In this section, we show an explicit convergence rate in the Frobenius norm for estimating  $\Theta(t), \forall t$ , where  $p, |F|$  grow with  $n$ , so long as the covariances change smoothly over  $t$ . Note that certain smoothness assumptions on a matrix  $W$  would guarantee the corresponding smoothness conditions on its inverse  $W^{-1}$ , so long as  $W$  is non-singular, as we show in Section 6. We first write our time-varying estimator  $\hat{\Theta}_n(t)$  for  $\Sigma^{-1}(t)$  at time  $t \in [0, 1]$  as the minimizer of the  $\ell_1$  regularized negative smoothed log-likelihood over the entire set of positive definite matrices,

$$\hat{\Theta}_n(t) = \arg \min_{\Theta \succ 0} \{ \text{tr}(\Theta \hat{S}_n(t)) - \log |\Theta| + \lambda_n |\Theta|_1 \} \quad (15)$$

where  $\lambda_n$  is a non-negative regularization parameter, and  $\hat{S}_n(t)$  is the smoothed sample covariance matrix using a kernel function as defined in (2).

Now fix a point of interest  $t_0$ . In the following, we use  $\Sigma_0 = (\sigma_{ij}(t_0))$  to denote the true covariance matrix at this time. Let  $\Theta_0 = \Sigma_0^{-1}$  be its inverse matrix. Define the set  $S = \{(i, j) : \theta_{ij}(t_0) \neq 0, i \neq j\}$ . Then  $|S| = s$ . Note that  $|S|$  is twice the number of edges in the graph  $G(t_0)$ . We make the following assumptions.

**A4** Let  $p + s = o(n^{2/3}/\log n)$  and  $\varphi_{\min}(\Sigma_0) \geq \underline{k} > 0$ , hence  $\varphi_{\max}(\Theta_0) \leq 1/\underline{k}$ . For some sufficiently large constant  $M$ , let  $\varphi_{\min}(\Theta_0) = \Omega\left(2M\sqrt{\frac{(p+s)\log n}{n^{2/3}}}\right)$ .

The proof draws upon techniques from [RBLZ07], with modifications necessary to handle the fact that we penalize  $|\Theta|_1$  rather than  $|\Theta^\diamond|_1$  as in their case.

**Theorem 8** Let  $\hat{\Theta}_n(t)$  be the minimizer defined by (15). Suppose all conditions in Lemma 2 and A 4 hold. If

$$\lambda_n \asymp \sqrt{\frac{\log n}{n^{2/3}}}, \quad \text{then}$$

$$\|\hat{\Theta}_n(t) - \Theta_0\|_F = O_P\left(2M\sqrt{\frac{(p+s)\log n}{n^{2/3}}}\right). \quad (16)$$

**Proof:** Let  $\underline{0}$  be a matrix with all entries being zero. Let

$$\begin{aligned} Q(\Theta) &= \text{tr}(\Theta \hat{S}_n(t_0)) - \log |\Theta| + \lambda |\Theta| - \\ &\quad \text{tr}(\Theta_0 \hat{S}_n(t_0)) + \log |\Theta_0| - \lambda |\Theta_0|_1 \\ &= \text{tr}\left((\Theta - \Theta_0)(\hat{S}_n(t) - \Sigma_0)\right) - \\ &\quad (\log |\Theta| - \log |\Theta_0|) + \text{tr}((\Theta - \Theta_0)\Sigma_0) \\ &\quad + \lambda(|\Theta|_1 - |\Theta_0|_1). \end{aligned} \quad (17)$$

$\hat{\Theta}_n$  minimizes  $Q(\Theta)$ , or equivalently  $\hat{\Delta}_n = \hat{\Theta}_n - \Theta_0$  minimizes  $G(\Delta) \equiv Q(\Theta_0 + \Delta)$ . Hence  $G(\underline{0}) = 0$  and  $G(\hat{\Theta}_n) \leq G(\underline{0}) = 0$  by definition. Define for some constant  $C_1$ ,  $\delta_n = C_1\sqrt{\frac{\log n}{n^{2/3}}}$ . Now, let

$$\lambda_n = \frac{C_1}{\varepsilon} \sqrt{\frac{\log n}{n^{2/3}}} = \frac{\delta_n}{\varepsilon} \quad \text{for some } 0 < \varepsilon < 1. \quad (18)$$

Consider now the set

$$\mathcal{T}_n = \{\Delta : \Delta = B - \Theta_0, B, \Theta_0 \succ 0, \|\Delta\|_F = Mr_n\},$$

where

$$r_n = \sqrt{\frac{(p+s)\log n}{n^{2/3}}} \asymp \delta_n \sqrt{p+s} \rightarrow 0. \quad (19)$$

**Claim 9** Under A 4, for all  $\Delta \in \mathcal{T}_n$  such that  $\|\Delta\|_F = o(1)$  as in (19),  $\Theta_0 + v\Delta \succ 0, \forall v \in I \supset [0, 1]$ .

**Proof:** It is sufficient to show that  $\Theta_0 + (1 + \varepsilon)\Delta \succ 0$  and  $\Theta_0 - \varepsilon\Delta \succ 0$  for some  $1 > \varepsilon > 0$ . Indeed,  $\varphi_{\min}(\Theta_0 + (1 + \varepsilon)\Delta) \geq \varphi_{\min}(\Theta_0) - (1 + \varepsilon)\|\Delta\|_2 > 0$  for  $\varepsilon < 1$ , given that  $\varphi_{\min}(\Theta_0) = \Omega(2Mr_n)$  and  $\|\Delta\|_2 \leq \|\Delta\|_F = Mr_n$ . Similarly,  $\varphi_{\min}(\Theta_0 - \varepsilon\Delta) \geq \varphi_{\min}(\Theta_0) - \varepsilon\|\Delta\|_2 > 0$  for  $\varepsilon < 1$ .  $\square$

Thus we have that  $\log \det(\Theta_0 + v\Delta)$  is infinitely differentiable on the open interval  $I \supset [0, 1]$  of  $v$ . This allows us to use the Taylor's formula with integral remainder to obtain the following lemma:

**Lemma 10** With probability  $1 - 1/n^c$  for some  $c \geq 2$ ,  $G(\Delta) > 0$  for all  $\Delta \in \mathcal{T}_n$ .

**Proof:** Let us use  $A$  as a shorthand for

$$\text{vec}\Delta^T \left( \int_0^1 (1-v)(\Theta_0 + v\Delta)^{-1} \otimes (\Theta_0 + v\Delta)^{-1} dv \right) \text{vec}\Delta,$$

where  $\otimes$  is the Kronecker product (if  $W = (w_{ij})_{m \times n}$ ,  $P = (b_{kl})_{p \times q}$ , then  $W \otimes P = (w_{ij}P)_{mp \times nq}$ ), and  $\text{vec}\Delta \in \mathbf{R}^{p^2}$  is  $\Delta_{p \times p}$  vectorized. Now, the Taylor expansion gives

$$\begin{aligned} \log |\Theta_0 + \Delta| - \log |\Theta_0| &= \frac{d}{dv} \log |\Theta_0 + v\Delta|_{v=0} \Delta + \\ &\int_0^1 (1-v) \frac{d^2}{dv^2} \log \det(\Theta_0 + v\Delta) dv = \text{tr}(\Sigma_0 \Delta) + A, \end{aligned}$$

where by symmetry,  $\text{tr}(\Sigma_0 \Delta) = \text{tr}(\Theta - \Theta_0)\Sigma_0$ . Hence

$$G(\Delta) = \quad (20)$$

$$A + \text{tr}\left(\Delta(\hat{S}_n - \Sigma_0)\right) + \lambda_n (|\Theta_0 + \Delta|_1 - |\Theta_0|_1).$$

For an index set  $S$  and a matrix  $W = [w_{ij}]$ , write  $W_S \equiv (w_{ij}I((i, j) \in S))$ , where  $I(\cdot)$  is an indicator function. Recall  $S = \{(i, j) : \theta_{0ij} \neq 0, i \neq j\}$  and let  $S^c = \{(i, j) : \theta_{0ij} = 0, i \neq j\}$ . Hence  $\Theta = \Theta^\setminus + \Theta_S^\diamond + \Theta_{S^c}^\diamond, \forall \Theta$  in our notation. Note that we have  $\Theta_{0S^c}^\diamond = \underline{0}$ ,

$$|\Theta_0^\diamond + \Delta^\diamond|_1 = |\Theta_{0S}^\diamond + \Delta_S^\diamond|_1 + |\Delta_{S^c}^\diamond|_1,$$

$$|\Theta_0^\diamond|_1 = |\Theta_{0S}^\diamond|_1, \quad \text{hence}$$

$$|\Theta_0^\diamond + \Delta^\diamond|_1 - |\Theta_0^\diamond|_1 \geq |\Delta_{S^c}^\diamond|_1 - |\Delta_S^\diamond|_1,$$

$$|\Theta_0^\setminus + \Delta^\setminus|_1 - |\Theta_0^\setminus|_1 \geq -|\Delta^\setminus|_1,$$

where the last two steps follow from the triangle inequality. Therefore

$$\begin{aligned} |\Theta_0 + \Delta|_1 - |\Theta_0|_1 &= \\ &|\Theta_0^\diamond + \Delta^\diamond|_1 - |\Theta_0^\diamond|_1 + |\Theta_0^\setminus + \Delta^\setminus|_1 - |\Theta_0^\setminus|_1 \\ &\geq |\Delta_{S^c}^\diamond|_1 - |\Delta_S^\diamond|_1 - |\Delta^\setminus|_1. \end{aligned} \quad (21)$$

Now, from Lemma 2,  $\max_{j,k} |\widehat{S}_n(t, j, k) - \sigma(t, j, k)| = O_P\left(\frac{\sqrt{\log n}}{n^{1/3}}\right) = O_P(\delta_n)$ . By (9), with probability  $1 - \frac{1}{n^2}$

$$\begin{aligned} & \left| \text{tr}(\Delta(\widehat{S}_n - \Sigma_0)) \right| \leq \delta_n |\Delta|_1, \quad \text{hence by (21)} \\ \text{tr}(\Delta(\widehat{S}_n - \Sigma_0)) + \lambda_n (|\Theta_0 + \Delta|_1 - |\Theta_0|_1) & \\ \geq -\delta_n |\Delta^\setminus|_1 - \delta_n |\Delta_{S^c}^\diamond|_1 - \delta_n |\Delta_S^\diamond|_1 & \\ -\lambda_n |\Delta^\setminus|_1 + \lambda_n |\Delta_{S^c}^\diamond|_1 - \lambda_n |\Delta_S^\diamond|_1 & \\ \geq -(\delta_n + \lambda_n) (|\Delta^\setminus|_1 + |\Delta_S^\diamond|_1) + (\lambda_n - \delta_n) |\Delta_{S^c}^\diamond|_1 & \\ \geq -(\delta_n + \lambda_n) (|\Delta^\setminus|_1 + |\Delta_S^\diamond|_1), \quad \text{where} & \quad (22) \end{aligned}$$

$$\begin{aligned} & (\delta_n + \lambda_n) (|\Delta^\setminus|_1 + |\Delta_S^\diamond|_1) \\ & \leq (\delta_n + \lambda_n) (\sqrt{p} \|\Delta^\setminus\|_F + \sqrt{s} \|\Delta_S^\diamond\|_F) \\ & \leq (\delta_n + \lambda_n) (\sqrt{p} \|\Delta^\setminus\|_F + \sqrt{s} \|\Delta^\diamond\|_F) \\ & \leq (\delta_n + \lambda_n) \max\{\sqrt{p}, \sqrt{s}\} (\|\Delta^\setminus\|_F + \|\Delta^\diamond\|_F) \\ & \leq (\delta_n + \lambda_n) \max\{\sqrt{p}, \sqrt{s}\} \sqrt{2} \|\Delta\|_F \\ & \leq \delta_n \frac{1+\varepsilon}{\varepsilon} \sqrt{p+s} \sqrt{2} \|\Delta\|_F. \quad (23) \end{aligned}$$

Combining (20), (22), and (23), we have with probability  $1 - \frac{1}{n^c}$ , for all  $\Delta \in \mathcal{T}_n$ ,

$$\begin{aligned} G(\Delta) & \geq A - (\delta_n + \lambda_n) (|\Delta^\setminus|_1 + |\Delta_S^\diamond|_1) \\ & \geq \frac{k^2}{2+\tau} \|\Delta\|_F^2 - \delta_n \frac{1+\varepsilon}{\varepsilon} \sqrt{p+s} \sqrt{2} \|\Delta\|_F \\ & = \|\Delta\|_F^2 \left( \frac{k^2}{2+\tau} - \delta_n \frac{\sqrt{2}(1+\varepsilon)}{\varepsilon \|\Delta\|_F} \sqrt{p+s} \right) \\ & = \|\Delta\|_F^2 \left( \frac{k^2}{2+\tau} - \frac{\delta_n \sqrt{2}(1+\varepsilon)}{\varepsilon M r_n} \sqrt{p+s} \right) > 0 \end{aligned}$$

for  $M$  sufficiently large, where the bound on  $A$  comes from Lemma 11 by [RBLZ07].  $\square$

**Lemma 11** ([RBLZ07]) *For some  $\tau = o(1)$ , under A 4,  $\text{vec} \Delta^T \left( \int_0^1 (1-v)(\Theta_0 + v\Delta)^{-1} \otimes (\Theta_0 + v\Delta)^{-1} dv \right) \text{vec} \Delta$*   
 $\geq \|\Delta\|_F^2 \frac{k^2}{2+\tau}$ , for all  $\Delta \in \mathcal{T}_n$ .

We next show the following claim.

**Claim 12** *If  $G(\Delta) > 0, \forall \Delta \in \mathcal{T}_n$ , then  $G(\Delta) > 0$  for all  $\Delta$  in  $\mathcal{V}_n = \{\Delta : \Delta = D - \Theta_0, D \succ 0, \|\Delta\|_F > M r_n, \text{ for } r_n \text{ as in (19)}\}$ . Hence if  $G(\Delta) > 0, \forall \Delta \in \mathcal{T}_n$ , then  $G(\Delta) > 0$  for all  $\Delta \in \mathcal{T}_n \cup \mathcal{V}_n$ .*

**Proof:** Now by contradiction, suppose  $G(\Delta') \leq 0$  for some  $\Delta' \in \mathcal{V}_n$ . Let  $\Delta_0 = \frac{M r_n}{\|\Delta'\|_F} \Delta'$ . Thus  $\Delta_0 = \theta \underline{0} + (1-\theta)\Delta'$ , where  $0 < 1-\theta = \frac{M r_n}{\|\Delta'\|_F} < 1$  by definition of  $\Delta_0$ . Hence  $\Delta_0 \in \mathcal{T}_n$  given that  $\Theta_0 + \Delta_0 \succ 0$  by Claim 13. Hence by convexity of  $G(\Delta)$ , we have that  $G(\Delta_0) \leq \theta G(\underline{0}) + (1-\theta)G(\Delta') \leq 0$ , contradicting that  $G(\Delta_0) > 0$  for  $\Delta_0 \in \mathcal{T}_n$ .  $\square$

By Claim 12 and the fact that  $G(\widehat{\Delta}_n) \leq G(0) = 0$ , we have the following: If  $G(\Delta) > 0, \forall \Delta \in \mathcal{T}_n$ , then  $\widehat{\Delta}_n \notin (\mathcal{T}_n \cup \mathcal{V}_n)$ , that is,  $\|\widehat{\Delta}_n\|_F < M r_n$ , given that  $\widehat{\Delta}_n = \widehat{\Theta}_n - \Theta_0$ , where  $\widehat{\Theta}_n, \Theta_0 \succ 0$ . Therefore

$$\begin{aligned} \mathbf{P} \left( \|\widehat{\Delta}_n\|_F \geq M r_n \right) & = 1 - \mathbf{P} \left( \|\widehat{\Delta}_n\|_F < M r_n \right) \\ & \leq 1 - \mathbf{P} (G(\Delta) > 0, \forall \Delta \in \mathcal{T}_n) \\ & = \mathbf{P} (G(\Delta) \leq 0 \text{ for some } \Delta \in \mathcal{T}_n) < \frac{1}{n^c}. \end{aligned}$$

We thus establish that  $\|\widehat{\Delta}_n\|_F \leq O_P(M r_n)$ .  $\square$

**Claim 13** *Let  $B$  be a  $p \times p$  matrix. If  $B \succ 0$  and  $B + D \succ 0$ , then  $B + vD \succ 0$  for all  $v \in [0, 1]$ .*

**Proof:** We only need to check for  $v \in (0, 1)$ , where  $1-v > 0; \forall x \in \mathbf{R}^p$ , by  $B \succ 0$  and  $B + D \succ 0$ ,  $x^T B x > 0$  and  $x^T (B + D)x > 0$ ; hence  $x^T D x > -x^T B x$ . Thus  $x^T (B + vD)x = x^T B x + v x^T D x > (1-v)x^T B x > 0$ .  $\square$

## 5 Large Deviation Inequalities

Before we go on, we explain the notation that we follow throughout this section. We switch notation from  $t$  to  $x$  and form a regression problem for non-iid data. Given an interval of  $[0, 1]$ , the point of interest is  $x_0 = 1$ . We form a design matrix by sampling a set of  $n$   $p$ -dimensional Gaussian random vectors  $Z^t$  at  $t = 0, 1/n, 2/n, \dots, 1$ , where  $Z^t \sim N(0, \Sigma_t)$  are independently distributed. In this section, we index the random vectors  $Z$  with  $k = 0, 1, \dots, n$  such that  $Z_k = Z^t$  for  $k = nt$ , with corresponding covariance matrix denoted by  $\Sigma_k$ . Hence

$$Z_k = (Z_{k1}, \dots, Z_{kp})^T \sim N(0, \Sigma_k), \quad \forall k. \quad (24)$$

These are independent but not identically distributed. We will need to generalize the usual inequalities. In Section A, via a boxcar kernel function, we use moment generating functions to show that for  $\widehat{\Sigma} = \frac{1}{n} \sum_{k=1}^n Z_k Z_k^T$ ,

$$P^n(|\widehat{\Sigma}_{ij} - \Sigma_{ij}(x_0)| > \epsilon) < e^{-c n \epsilon^2} \quad (25)$$

where  $P^n = P_1 \times \dots \times P_n$  denotes the product measure. We look across  $n$  time-varying Gaussian vectors, and roughly, we compare  $\widehat{\Sigma}_{ij}$  with  $\Sigma_{ij}(x_0)$ , where  $\Sigma(x_0) = \Sigma_n$  is the covariance matrix in the end of the window for  $t_0 = n$ . Furthermore, we derive inequalities in Section 5.1 for a general kernel function.

### 5.1 Bounds For Kernel Smoothing

In this section, we derive large deviation inequalities for the covariance matrix based on kernel regression estimations. Recall that we assume that the symmetric nonnegative kernel function  $K$  has a bounded support  $[-1, 1]$  in A 1. This kernel has the property that:

$$2 \int_{-1}^0 v K(v) dv \leq 2 \int_{-1}^0 K(v) dv = 1 \quad (26)$$

$$2 \int_{-1}^0 v^2 K(v) dv \leq 1. \quad (27)$$

In order to estimate  $t_0$ , instead of taking an average of sample variances/covariances over the last  $n$  samples, we use the weighting scheme such that data close to  $t_0$  receives larger weights than those that are far away. Let  $\Sigma(x) = (\sigma_{ij}(x))$ . Let us define  $x_0 = \frac{t_0}{n} = 1$ , and  $\forall i = 1, \dots, n, x_i = \frac{t_0 - i}{n}$  and

$$\ell_i(x_0) = \frac{2}{nh} K\left(\frac{x_i - x_0}{h}\right) \approx \frac{K\left(\frac{x_i - x_0}{h}\right)}{\sum_{i=1}^n K\left(\frac{x_i - x_0}{h}\right)} \quad (28)$$

where the approximation is due to replacing the sum with the Riemann integral:

$$\sum_{i=1}^n \ell_i(x_0) = \sum_{i=1}^n \frac{2}{nh} K\left(\frac{x_i - x_0}{h}\right) \approx 2 \int_{-1}^0 K(v) dv = 1,$$

due to the fact that  $K(v)$  has compact support in  $[-1, 1]$  and  $h \leq 1$ . Let  $\Sigma_k = (\sigma_{ij}(x_k)), \forall k = 1, \dots, n$ , where  $\sigma_{ij}(x_k) = \text{cov}(Z_{ki}, Z_{kj}) = \rho_{ij}(x_k) \sigma_i(x_k) \sigma_j(x_k)$  and  $\rho_{ij}(x_k)$  is the correlation coefficient between  $Z_i$  and  $Z_j$  at time  $x_k$ . Recall that we have independent  $(Z_{ki} Z_{kj})$  for all  $k = 1, \dots, n$  such that  $\mathbf{E}(Z_{ki} Z_{kj}) = \sigma_{ij}(x_k)$ . Let

$$\Phi_1(i, j) = \frac{1}{n} \sum_{k=1}^n \frac{2}{h} K\left(\frac{x_k - x_0}{h}\right) \sigma_{ij}(x_k), \text{ hence}$$

$$\mathbf{E} \sum_{k=1}^n \ell_k(x_0) Z_{ki} Z_{kj} = \sum_{k=1}^n \ell_k(x_0) \sigma_{ij}(x_k) = \Phi_1(i, j).$$

We thus decompose and bound for point of interest  $x_0$

$$\begin{aligned} & \left| \sum_{k=1}^n \ell_k(x_0) Z_{ki} Z_{kj} - \sigma_{ij}(x_0) \right| \leq \\ & \left| \mathbf{E} \sum_{k=1}^n \ell_k(x_0) Z_{ki} Z_{kj} - \sigma_{ij}(x_0) \right| + \\ & \left| \sum_{k=1}^n \ell_k(x_0) Z_{ki} Z_{kj} - \mathbf{E} \sum_{k=1}^n \ell_k(x_0) Z_{ki} Z_{kj} \right| \quad (29) \\ & = \left| \sum_{k=1}^n \ell_k(x_0) Z_{ki} Z_{kj} - \Phi_1(i, j) \right| + |\Phi_1(i, j) - \sigma_{ij}(x_0)|. \end{aligned}$$

Before we start our analysis on large deviations, we first look at the bias term.

**Lemma 14** Suppose there exists  $C > 0$  such that

$$\max_{i,j} \sup_t |\sigma''(t, i, j)| \leq C. \text{ Then}$$

$$\forall t \in [0, 1], \max_{i,j} |\mathbf{E} \hat{S}_n(t, i, j) - \sigma_{ij}(t)| = O(h).$$

**Proof:** W.l.o.g, let  $t = t_0$ , hence  $\mathbf{E} \hat{S}_n(t, i, j) = \Phi_1(i, j)$ .

We use the Riemann integral to approximate the sum,

$$\begin{aligned} \Phi_1(i, j) &= \frac{1}{n} \sum_{k=1}^n \frac{2}{h} K\left(\frac{x_k - x_0}{h}\right) \sigma_{ij}(x_k) \\ &\approx \int_{x_n}^{x_0} \frac{2}{h} K\left(\frac{u - x_0}{h}\right) \sigma_{ij}(u) du \\ &= 2 \int_{-1/h}^0 K(v) \sigma_{ij}(x_0 + hv) dv. \end{aligned}$$

We now use Taylor's Formula to replace  $\sigma_{ij}(x_0 + hv)$  and obtain  $2 \int_{-1/h}^0 K(v) \sigma_{ij}(x_0 + hv) dv =$

$$\begin{aligned} & 2 \int_{-1}^0 K(v) \left( \sigma_{ij}(x_0) + hv \sigma'_{ij}(x_0) + \frac{\sigma''_{ij}(y(v))(hv)^2}{2} \right) dv \\ &= \sigma_{ij}(x_0) + 2 \int_{-1}^0 K(v) \left( hv \sigma'_{ij}(x_0) + \frac{C(hv)^2}{2} \right) dv, \end{aligned}$$

$$\text{where } 2 \int_{-1}^0 K(v) \left( hv \sigma'_{ij}(x_0) + \frac{C(hv)^2}{2} \right) dv$$

$$= 2h \sigma'_{ij}(x_0) \int_{-1}^0 v K(v) dv + \frac{Ch^2}{2} \int_{-1}^0 v^2 K(v) dv$$

$$\leq h \sigma'_{ij}(x_0) + \frac{Ch^2}{4}, \text{ where } y(v) - x_0 < hv.$$

Thus  $\Phi_1(i, j) - \sigma_{ij}(x_0) = O(h)$ .  $\square$

We now move on to the large deviation bound for all entries of the smoothed empirical covariance matrix.

**Lemma 15** For  $\epsilon < \frac{C_1 (\sigma_i^2(x_0) \sigma_j^2(x_0) + \sigma_{ij}^2(x_0))}{\max_{k=1, \dots, n} (2K(\frac{x_k - x_0}{h}) \sigma_i(x_k) \sigma_j(x_k))}$ ,

where  $C_1$  is defined in Claim 18, for some  $C > 0$ ,

$$\mathbf{P} \left( |\hat{S}_n(t, i, j) - \mathbf{E} \hat{S}_n(t, i, j)| > \epsilon \right) \leq \exp \{-Cnh\epsilon^2\}.$$

**Proof:** Let us define  $A_k = Z_{ki} Z_{kj} - \sigma_{ij}(x_k)$ .

$$\mathbf{P} \left( |\hat{S}_n(t, i, j) - \mathbf{E} \hat{S}_n(t, i, j)| > \epsilon \right)$$

$$= \mathbf{P} \left( \sum_{k=1}^n \ell_k(x_0) Z_{ki} Z_{kj} - \sum_{k=1}^n \ell_k(x_0) \sigma_{ij}(x_k) > \epsilon \right)$$

For every  $t > 0$ , we have by Markov's inequality

$$\begin{aligned} & \mathbf{P} \left( \sum_{k=1}^n n \ell_k(x_0) A_k > n\epsilon \right) \\ &= \mathbf{P} \left( e^{t \sum_{k=1}^n \frac{2}{h} K\left(\frac{x_i - x_0}{h}\right) A_k} > e^{nt\epsilon} \right) \\ &\leq \frac{\mathbf{E} e^{t \sum_{k=1}^n \frac{2}{h} K\left(\frac{x_i - x_0}{h}\right) A_k}}{e^{nt\epsilon}}. \quad (30) \end{aligned}$$

Before we continue, for a given  $t$ , let us first define the following quantities, where  $i, j$  are omitted from  $\Phi_1(i, j)$

- $a_k = \frac{2t}{h} K\left(\frac{x_k - x_0}{h}\right) (\sigma_i(x_k) \sigma_j(x_k) + \sigma_{ij}(x_k))$
- $b_k = \frac{2t}{h} K\left(\frac{x_k - x_0}{h}\right) (\sigma_i(x_k) \sigma_j(x_k) - \sigma_{ij}(x_k))$  thus
- $\Phi_1 = \frac{1}{n} \sum_{k=1}^n \frac{a_k - b_k}{2t}, \Phi_2 = \frac{1}{n} \sum_{k=1}^n \frac{a_k^2 + b_k^2}{4t^2}$
- $\Phi_3 = \frac{1}{n} \sum_{k=1}^n \frac{a_k^3 - b_k^3}{6t^3}, \Phi_4 = \frac{1}{n} \sum_{k=1}^n \frac{a_k^4 + b_k^4}{8t^4}$
- $M = \max_{k=1, \dots, n} \left( \frac{2}{h} K\left(\frac{x_k - x_0}{h}\right) \sigma_i(x_k) \sigma_j(x_k) \right)$

We now establish some convenient comparisons; see Section B.1 and B.2 for their proofs.

**Claim 16**  $\frac{\Phi_3}{\Phi_2} \leq \frac{4M}{3}$  and  $\frac{\Phi_4}{\Phi_2} \leq 2M^2$ , where both equalities are established at  $\rho_{ij}(x_k) = 1, \forall k$ .

**Lemma 17** For  $b_k \leq a_k \leq \frac{1}{2}, \forall k, \frac{1}{2} \sum_{k=1}^n \ln \frac{1}{(1-a_k)(1+b_k)} \leq nt\Phi_1 + nt^2\Phi_2 + nt^3\Phi_3 + \frac{9}{5}nt^4\Phi_4$ .

To show the following, we first replace the sum with a Riemann integral, and then use Taylor's Formula to approximate  $\sigma_i(x_k), \sigma_j(x_k)$ , and  $\sigma_{ij}(x_k), \forall k = 1, \dots, n$  with  $\sigma_i, \sigma_j, \sigma_{ij}$  and their first derivatives at  $x_0$  respectively, plus some remainder terms; see Section B.3 for details.

**Claim 18** For  $h = n^{-\epsilon}$  for some  $1 > \epsilon > 0$ , there exists some constant  $C_1 > 0$  such that

$$\Phi_2(i, j) = \frac{C_1(\sigma_i^2(x_0)\sigma_j^2(x_0) + \sigma_{ij}^2(x_0))}{h}.$$

Lemma 19 computes the moment generating function for  $\frac{2}{h}K\left(\frac{x_k-x_0}{h}\right)Z_{ki} \cdot Z_{kj}$ . The proof proceeds exactly as that of Lemma 21 after substituting  $t$  with  $\frac{2t}{h}K\left(\frac{x_k-x_0}{h}\right)$  everywhere.

**Lemma 19** Let  $\frac{2t}{h}K\left(\frac{x_k-x_0}{h}\right)(1+\rho_{ij}(x_k))\sigma_i(x_k)\sigma_j(x_k) < 1, \forall k$ . For  $b_k \leq a_k < 1$ .

$$\mathbf{E}e^{\frac{2t}{h}K\left(\frac{x_k-x_0}{h}\right)Z_{ki}Z_{kj}} = ((1-a_k)(1+b_k))^{-1/2}.$$

**Remark 20** Thus when we set  $t = \frac{\epsilon}{4\Phi_2}$ , the bound on  $\epsilon$  implies that  $b_k \leq a_k \leq 1/2, \forall k$ :

$$\begin{aligned} a_k &= t(1 + \rho_{ij}(x_k))\sigma_i(x_k)\sigma_j(x_k) \\ &\leq 2t\sigma_i(x_k)\sigma_j(x_k) = \frac{\epsilon\sigma_i(x_k)\sigma_j(x_k)}{2\Phi_2} \leq \frac{1}{2}. \end{aligned}$$

We can now finish showing the large deviation bound for  $\max_{i,j} |\hat{S}_{i,j} - \mathbf{E}S_{i,j}|$ . Given that  $A_1, \dots, A_n$  are independent, we have

$$\begin{aligned} \mathbf{E}e^{t\sum_{k=1}^n \frac{2}{h}K\left(\frac{x_k-x_0}{h}\right)A_k} &= \prod_{k=1}^n \mathbf{E}e^{\frac{2t}{h}K\left(\frac{x_k-x_0}{h}\right)A_k} \\ &= \prod_{k=1}^n \exp\left(-\frac{2t}{h}K\left(\frac{x_k-x_0}{h}\right)\sigma_{ij}(x_k)\right) \cdot \\ &\quad \prod_{k=1}^n \mathbf{E}e^{\frac{2t}{h}K\left(\frac{x_k-x_0}{h}\right)Z_{ki}Z_{kj}} \end{aligned} \quad (31)$$

By (30), (31), Lemma 19, for  $t \leq \frac{\epsilon}{4\Phi_2}$ ,

$$\begin{aligned} \mathbf{P}\left(\sum_{k=1}^n \frac{2}{h}K\left(\frac{x_k-x_0}{h}\right)A_k > n\epsilon\right) &\leq \frac{\mathbf{E}e^{t\sum_{k=1}^n \frac{2}{h}K\left(\frac{x_k-x_0}{h}\right)A_k}}{e^{-nt\epsilon}} = e^{-nt\epsilon} \cdot \\ &\prod_{k=1}^n e^{-\frac{2t}{h}K\left(\frac{x_k-x_0}{h}\right)\sigma_{ij}(x_k)} \cdot \mathbf{E}e^{\frac{2t}{h}K\left(\frac{x_k-x_0}{h}\right)Z_{ki}Z_{kj}} \\ &= e^{-nt\epsilon - nt\Phi_2(i,j) + \frac{1}{2}\sum_{k=1}^n \ln \frac{1}{(1-a_k)(1+b_k)}} \\ &\leq \exp\left(-nt\epsilon + nt^2\Phi_2 + nt^3\Phi_3 + \frac{9}{5}nt^4\Phi_4\right), \end{aligned}$$

where the last step is due to Remark 20 and Lemma 17. Now let us consider taking  $t$  that minimizes  $\exp(-nt\epsilon + nt^2\Phi_2 + nt^3\Phi_3 + \frac{9}{5}nt^4\Phi_4)$ ; Let  $t = \frac{\epsilon}{4\Phi_2}$ ;  $\frac{d}{dt}(-nt\epsilon + nt^2\Phi_2 + nt^3\Phi_3 + \frac{9}{5}nt^4\Phi_4) \leq -\frac{\epsilon}{40}$ ; Now given that  $\frac{\epsilon^2}{\Phi_2} < \frac{1}{M}$ , Claim 16 and 18:

$$\begin{aligned} &\mathbf{P}\left(\sum_{k=1}^n \frac{2}{h}K\left(\frac{x_k-x_0}{h}\right)A_k > n\epsilon\right) \\ &\leq \exp\left(-nt\epsilon + nt^2\Phi_2 + nt^3\Phi_3 + \frac{9}{5}nt^4\Phi_4\right) \\ &\leq \exp\left(\frac{-n\epsilon^2}{4\Phi_2} + \frac{n\epsilon^2}{16\Phi_2} + \frac{n\epsilon^2}{64\Phi_2} \frac{\epsilon\Phi_3}{\Phi_2^2} + \frac{9}{5} \frac{n\epsilon^2}{256\Phi_2} \frac{\epsilon^2\Phi_4}{\Phi_2^3}\right) \\ &\leq \exp\left(\frac{-3n\epsilon^2}{20\Phi_2}\right) \\ &\leq \exp\left(-\frac{3nh\epsilon^2}{20C_1(\sigma_i^2(x_0)\sigma_j^2(x_0) + \sigma_{ij}^2(x_0))}\right). \end{aligned}$$

Finally, let's check the requirement on  $\epsilon \leq \frac{\Phi_2}{M}$ ,

$$\begin{aligned} \epsilon &\leq \frac{C_1(1 + \rho_{ij}^2(x_0))\sigma_i^2(x_0)\sigma_j^2(x_0)/h}{\max_{k=1, \dots, n} \left(\frac{2}{h}K\left(\frac{x_k-x_0}{h}\right)\sigma_i(x_k)\sigma_j(x_k)\right)} \\ &= \frac{C_1(1 + \rho_{ij}^2(x_0))\sigma_i^2(x_0)\sigma_j^2(x_0)}{\max_{k=1, \dots, n} \left(2K\left(\frac{x_k-x_0}{h}\right)\sigma_i(x_k)\sigma_j(x_k)\right)}. \end{aligned}$$

□

For completeness, we compute the moment generating function for  $Z_{k,i}Z_{k,j}$ .

**Lemma 21** Let  $t(1 + \rho_{ij}(x_k))\sigma_i(x_k)\sigma_j(x_k) < 1, \forall k$ , so that  $b_k \leq a_k < 1$ , omitting  $x_k$  everywhere,

$$\mathbf{E}e^{tZ_{k,i}Z_{k,j}} = \left(\frac{1}{(1-t(\sigma_i\sigma_j + \sigma_{ij}))(1+t(\sigma_i\sigma_j - \sigma_{ij}))}\right)^{1/2}.$$

**Proof:** W.l.o.g., let  $i = 1$  and  $j = 2$ .

$$\begin{aligned} \mathbf{E}(e^{tZ_1Z_2}) &= \mathbf{E}(\mathbf{E}(e^{tZ_2Z_1}|Z_2)) \\ &= \mathbf{E}\exp\left(\left(\frac{t\rho_{12}\sigma_1}{\sigma_2} + \frac{t^2\sigma_1^2(1-\rho_{12}^2)}{2}\right)Z_2^2\right) \\ &= \left(1 - 2\left(\frac{t\rho_{12}\sigma_1}{\sigma_2} + \frac{t^2\sigma_1^2(1-\rho_{12}^2)}{2}\right)\sigma_2^2\right)^{-1/2} \\ &= \left(\frac{1}{1 - (2t\rho_{12}\sigma_1\sigma_2 + t^2\sigma_1^2\sigma_2^2(1-\rho_{12}^2))}\right)^{1/2} \\ &= \left(\frac{1}{(1-t(1+\rho_{12})\sigma_1\sigma_2)(1+t(1-\rho_{12})\sigma_1\sigma_2)}\right)^{1/2} \end{aligned}$$

where  $2t\rho_{12}\sigma_1\sigma_2 + t^2\sigma_1^2\sigma_2^2(1-\rho_{12}^2) < 1$ . This requires that  $t < \frac{1}{(1+\rho_{12})\sigma_1\sigma_2}$  which is equivalent to  $2t\rho_{12}\sigma_1\sigma_2 + t^2\sigma_1^2\sigma_2^2(1-\rho_{12}^2) - 1 < 0$ . One can check that if we require  $t(1+\rho_{12})\sigma_1\sigma_2 \leq 1$ , which implies that  $t\sigma_1\sigma_2 \leq 1 - t\rho_{12}\sigma_1\sigma_2$  and hence  $t^2\sigma_1^2\sigma_2^2 \leq (1 - t\rho_{12}\sigma_1\sigma_2)^2$ , the lemma holds. □

## 6 Smoothness and Sparsity of $\Sigma_t$ via $\Sigma_t^{-1}$

In this section we show that if we assume  $\Theta(x) = (\theta_{ij}(x))$  are smooth and twice differentiable functions of  $x \in [0, 1]$ , i.e.,  $\theta'_{ij}(x) < \infty$  and  $\theta''_{ij}(x) < \infty$  for  $x \in [0, 1]$ ,  $\forall i, j$ , and satisfy A 3, then the smoothness conditions of Lemma 2 are satisfied. The following is a standard result in matrix analysis.

**Lemma 22** *Let  $\Theta(t) \in R^{p \times p}$  has entries that are differentiable functions of  $t \in [0, 1]$ . Assuming that  $\Theta(t)$  is always non-singular, then*

$$\frac{d}{dt}[\Sigma(t)] = -\Sigma(t) \frac{d}{dt}[\Theta(t)]\Sigma(t).$$

**Lemma 23** *Suppose  $\Theta(t) \in R^{p \times p}$  has entries that each are twice differentiable functions of  $t$ . Assuming that  $\Theta(t)$  is always non-singular, then*

$$\frac{d^2}{dt^2}[\Sigma(t)] = \Sigma(t)D(t)\Sigma(t), \quad \text{where}$$

$$D(t) = 2 \frac{d}{dt}[\Theta(t)]\Sigma(t) \frac{d}{dt}[\Theta(t)] - \frac{d^2}{dt^2}[\Theta(t)].$$

**Proof:** The existence of the second order derivatives for entries of  $\Sigma(t)$  is due to the fact that  $\Sigma(t)$  and  $\frac{d}{dt}[\Theta(t)]$  are both differentiable  $\forall t \in [0, 1]$ ; indeed by Lemma 22,

$$\begin{aligned} \frac{d^2}{dt^2}[\Sigma(t)] &= \frac{d}{dt} \left[ -\Sigma(t) \frac{d}{dt}[\Theta(t)]\Sigma(t) \right] \\ &= -\frac{d}{dt}[\Sigma(t)] \frac{d}{dt}[\Theta(t)]\Sigma(t) - \Sigma(t) \frac{d}{dt} \left[ \frac{d}{dt}[\Theta(t)]\Sigma(t) \right] \\ &= -\frac{d}{dt}[\Sigma(t)] \frac{d}{dt}[\Theta(t)]\Sigma(t) - \Sigma(t) \frac{d^2}{dt^2}[\Theta(t)]\Sigma(t) - \\ &\quad \Sigma(t) \frac{d}{dt}[\Theta(t)] \frac{d}{dt}[\Sigma(t)] \\ &= \Sigma(t) \left( 2 \frac{d}{dt}[\Theta(t)]\Sigma(t) \frac{d}{dt}[\Theta(t)] - \frac{d^2}{dt^2}[\Theta(t)] \right) \Sigma(t), \end{aligned}$$

hence the lemma holds by the definition of  $D(t)$ .  $\square$

Let  $\Sigma(x) = (\sigma_{ij}(x)), \forall x \in [0, 1]$ . Let  $\Sigma(x) = (\Sigma_1(x), \Sigma_2(x), \dots, \Sigma_p(x))$ , where  $\Sigma_i(x) \in R^p$  denotes a column vector. By Lemma 23,

$$\sigma'_{ij}(x) = -\Sigma_i^T(x)\Theta'(x)\Sigma_j(x), \quad (32)$$

$$\sigma''_{ij}(x) = \Sigma_i^T(x)D(x)\Sigma_j(x), \quad (33)$$

where  $\Theta'(x) = (\theta'_{ij}(x)), \forall x \in [0, 1]$ .

**Lemma 24** *Given A 2 and A 3,  $\forall x \in [0, 1]$ ,*

$$|\sigma'_{ij}(x)| \leq S_0^2 \sqrt{S_1} < \infty.$$

**Proof:**  $|\sigma'_{ij}(x)| = |\Sigma_i^T(x)\Theta'(x)\Sigma_j(x)|$

$$\leq \max_{i=1, \dots, p} |\sigma_i^2(x)| \sum_{k=1}^p \sum_{\ell=1}^p |\theta'_{k\ell}(x)| \leq S_0^2 \sqrt{S_1}.$$

$\square$

We denote the elements of  $\Theta(x)$  by  $\theta_{jk}(x)$ . Let  $\theta'_\ell$  represent a column vector of  $\Theta'$ .

**Theorem 25** *Given A 2 and A 3,  $\forall i, j, \forall x \in [0, 1]$ ,*

$$\sup_{x \in [0, 1]} |\sigma''_{ij}(x)| < 2S_0^3 S_1 + S_0^2 S_2 < \infty.$$

**Proof:** By (33) and the triangle inequality,

$$\begin{aligned} |\sigma''_{ij}(x)| &= |\Sigma_i^T(x)D(x)\Sigma_j(x)| \\ &\leq \max_{i=1, \dots, p} |\sigma_i^2(x)| \sum_{k=1}^p \sum_{\ell=1}^p |D_{k\ell}(x)| \\ &\leq S_0^2 \sum_{k=1}^p \sum_{\ell=1}^p 2|\theta_k^{TT}(x)\Sigma(x)\theta'_\ell(x)| + |\theta''_{k\ell}(x)| \\ &= 2S_0^3 S_1 + S_0^2 S_2, \end{aligned}$$

where by A 3,  $\sum_{k=1}^p \sum_{\ell=1}^p |\theta''_{k\ell}(x)| \leq S_2$ , and

$$\begin{aligned} &\sum_{k=1}^p \sum_{\ell=1}^p |\theta_k^{TT}(x)\Sigma(x)\theta'_\ell(x)| \\ &= \sum_{k=1}^p \sum_{\ell=1}^p \sum_{i=1}^p \sum_{j=1}^p |\theta'_{ki}(x)\theta'_{\ell j}(x)\sigma_{ij}(x)| \\ &\leq \max_{i=1, \dots, p} |\sigma_i(x)| \sum_{k=1}^p \sum_{\ell=1}^p \sum_{i=1}^p \sum_{j=1}^p |\theta'_{ki}(x)\theta'_{\ell j}(x)| \\ &\leq S_0 S_1. \quad \square \end{aligned}$$

## 7 Some Implications of a Very Sparse $\Theta$

We use  $\mathcal{L}^1$  to denote Lebesgue measure on  $\mathbf{R}$ . The aim of this section is to prove some bounds that correspond to A 3, but only for  $\mathcal{L}^1$  a.e.  $x \in [0, 1]$ , based on a single sparsity assumption on  $\Theta$  as in A 5. We let  $E \subset [0, 1]$  represent the “bad” set with  $\mathcal{L}^1(E) = 0$ . and  $\mathcal{L}^1$  a.e.  $x \in [0, 1]$  refer to points in the set  $[0, 1] \setminus E$  such that  $\mathcal{L}^1([0, 1] \setminus E) = 1$ . When  $\|\Theta(x)\|_0 \leq s + p$  for all  $x \in [0, 1]$ , we immediately obtain Theorem 26, whose proof appears in Section 7.1. We like to point out that although we apply Theorem 26 to  $\Theta$  and deduce smoothness of  $\Sigma$ , we could apply it the other way around. In particular, it might be interesting to apply it to the correlation coefficient matrix  $(\rho_{ij})$ , where the diagonal entries remain invariant. We use  $\Theta'(x)$  and  $\Theta''(x)$  to denote  $(\theta'_{ij}(x))$  and  $(\theta''_{ij}(x))$  respectively  $\forall x$ .

**A5** *Assume that  $\|\Theta(x)\|_0 \leq s + p \forall x \in [0, 1]$ .*

**A6**  $\exists S_4, S_5 < \infty$  such that

$$S_4 = \max_{ij} \|\theta'_{ij}\|_\infty^2 \quad \text{and} \quad S_5 = \max_{ij} \|\theta''_{ij}\|_\infty. \quad (34)$$

We state a theorem, the proof of which is in Section 7.1 and a corollary.

**Theorem 26** *Under A 5, we have  $\|\Theta''(x)\|_0 \leq \|\Theta'(x)\|_0 \leq \|\Theta(x)\|_0 \leq s + p$  for  $\mathcal{L}^1$  a.e.  $x \in [0, 1]$ .*

**Corollary 27** Given A 2 and A 5, for  $\mathcal{L}^1$  a.e.  $x \in [0, 1]$

$$|\sigma'_{ij}(x)| \leq S_0^2 \sqrt{S_4}(s+p) < \infty. \quad (35)$$

**Proof:** By proof of Lemma 24,

$$|\sigma'_{ij}(x)| \leq \max_{i=1, \dots, p} \|\sigma_i^2\|_\infty \sum_{k=1}^p \sum_{\ell=1}^p |\theta'_{k\ell}(x)|.$$

Hence by Theorem 26, for  $\mathcal{L}^1$  a.e.  $x \in [0, 1]$ ,  $|\sigma'_{ij}(x)| \leq \max_{i=1, \dots, p} \|\sigma_i^2\|_\infty \sum_{k=1}^p \sum_{\ell=1}^p |\theta'_{k\ell}(x)| \leq S_0^2 \max_{k, \ell} \|\theta'_{k\ell}\|_\infty \|\Theta'(x)\|_0 \leq S_0^2 \sqrt{S_4}(s+p)$ .  $\square$

**Lemma 28** Under A 5 and 6, for  $\mathcal{L}^1$  a.e.  $x \in [0, 1]$ ,

$$\sum_{k=1}^p \sum_{\ell=1}^p \sum_{i=1}^p \sum_{j=1}^p |\theta'_{ki}(x)\theta'_{\ell j}(x)| \leq (s+p)^2 \max_{ij} \|\theta''_{ij}\|_\infty^2$$

$$\sum_{k=1}^p \sum_{\ell=1}^p \theta''_{k\ell} \leq (s+p) \max_{ij} \|\theta''_{ij}\|_\infty, \text{ hence}$$

$$\text{ess sup}_{x \in [0,1]} \sigma''_{ij}(x) \leq 2S_0^3(s+p)^2 S_4 + S_0^2(s+p)S_5.$$

**Proof:** By the triangle inequality, for  $\mathcal{L}^1$  a.e.  $x \in [0, 1]$ ,

$$|\sigma''_{ij}(x)| = |\Sigma_i^T D \Sigma_j|$$

$$= \left| \sum_{k=1}^p \sum_{\ell=1}^p \sigma_{ik}(x)\sigma_{j\ell}(x)D_{k\ell}(x) \right|$$

$$\leq \max_{i=1, \dots, p} \|\sigma_i^2\|_\infty \sum_{k=1}^p \sum_{\ell=1}^p |D_{k\ell}(x)|$$

$$\leq 2S_0^2 \sum_{k=1}^p \sum_{\ell=1}^p |\theta_k^T \Sigma \theta'_\ell| + S_0^2 \sum_{k=1}^p \sum_{\ell=1}^p |\theta''_{k\ell}|$$

$$= 2S_0^3(s+p)^2 S_4 + S_0^2(s+p)S_5,$$

where for  $\mathcal{L}^1$  a.e.  $x \in [0, 1]$ ,

$$\sum_{k=1}^p \sum_{\ell=1}^p |\theta_k^T \Sigma \theta'_\ell| \leq \sum_{k=1}^p \sum_{\ell=1}^p \sum_{i=1}^p \sum_{j=1}^p |\theta'_{ki}\theta'_{\ell j}\sigma_{ij}|$$

$$\leq \max_{i=1, \dots, p} \|\sigma_i\|_\infty \sum_{k=1}^p \sum_{\ell=1}^p \sum_{i=1}^p \sum_{j=1}^p |\theta'_{ki}\theta'_{\ell j}|$$

$$\leq S_0(s+p)^2 S_4$$

and  $\sum_{k=1}^p \sum_{\ell=1}^p |\theta''_{k\ell}| \leq (s+p)S_5$ . The first inequality is due to the following observation: at most  $(s+p)^2$  elements in the sum of  $\sum_k \sum_i \sum_\ell \sum_j |\theta'_{ki}(x)\theta'_{\ell j}(x)|$  for  $\mathcal{L}^1$  a.e.  $x \in [0, 1]$ , that is, except for  $E$ , are non-zero, due to the fact that for  $x \in [0, 1] \setminus N$ ,  $\|\Theta'(x)\|_0 \leq \|\Theta(x)\|_0 \leq s+p$  as in Theorem 26. The second inequality is obtained similarly using the fact that for  $\mathcal{L}^1$  a.e.  $x \in [0, 1]$ ,  $\|\Theta''(x)\|_0 \leq \|\Theta(x)\|_0 \leq s+p$ .  $\square$

**Remark 29** For the bad set  $E \subset [0, 1]$  with  $\mathcal{L}^1(E) = 0$ ,  $\sigma'_{ij}(x)$  is well defined as shown in Lemma 22, but it can only be loosely bounded by  $O(p^2)$ , as  $\|\Theta'(x)\|_0 = O(p^2)$ , instead of  $s+p$ , for  $x \in E$ ; similarly,  $\sigma''_{ij}(x)$  can only be loosely bounded by  $O(p^4)$ .

By Lemma 28, using the Lebesgue integral, we can derive the following corollary.

**Corollary 30** Under A 2, A 5, and A 6,

$$\int_0^1 (\sigma''_{ij}(x))^2 dx \leq 2S_0^3 S_4 s + p^2 + S_0^2 S_5 (s+p) < \infty.$$

### 7.1 Proof of Theorem 26.

Let  $\|\Theta(x)\|_0 \leq s+p$  for all  $x \in [0, 1]$ .

**Lemma 31** Let a function  $u : [0, 1] \rightarrow \mathbf{R}$ . Suppose  $u$  has a derivative on  $F$  (finite or not) with  $\mathcal{L}^1(u(F)) = 0$ . Then  $u'(x) = 0$  for  $\mathcal{L}^1$  a.e.  $x \in F$ .

Take  $F = \{x \in [0, 1] : \theta_{ij}(x) = 0\}$  and  $u = \theta_{ij}$ . For  $\mathcal{L}^1$  a.e.  $x \in F$ , that is, except for a set  $N_{ij}$  of  $\mathcal{L}^1(N_{ij}) = 0$ ,  $\theta'_{ij}(x) = 0$ . Let  $N = \bigcup_{ij} N_{ij}$ . By Lemma 31,

**Lemma 32** If  $x \in [0, 1] \setminus N$ , where  $\mathcal{L}^1(N) = 0$ , if  $\theta_{ij}(x) = 0$ , then  $\theta'_{ij}(x) = 0$  for all  $i, j$ .

Let  $v_{ij} = \theta'_{ij}$ . Take  $F = \{x \in [0, 1] : v_{ij}(x) = 0\}$ . For  $\mathcal{L}^1$  a.e.  $x \in F$ , that is, except for a set  $N_{ij}^1$  with  $\mathcal{L}(N_{ij}^1) = 0$ ,  $v'_{ij}(x) = 0$ . Let  $N_1 = \bigcup_{ij} N_{ij}^1$ . By Lemma 31,

**Lemma 33** If  $x \in [0, 1] \setminus N_1$ , where  $\mathcal{L}^1(N_1) = 0$ , if  $\theta'_{ij}(x) = 0$ , then  $\theta''_{ij}(x) = 0, \forall i, j$ .

Thus this allows to conclude that

**Lemma 34** If  $x \in [0, 1] \setminus N \cup N_1$ , where  $\mathcal{L}^1(N \cup N_1) = 0$ , if  $\theta_{ij}(x) = 0$ , then  $\theta'_{ij}(x) = 0$  and  $\theta''_{ij}(x) = 0, \forall i, j$ .

Thus for all  $x \in [0, 1] \setminus N \cup N_1$ ,  $\|\Theta''(x)\|_0 \leq \|\Theta'(x)\|_0 \leq \|\Theta(x)\|_0 \leq (s+p)$ .  $\square$

## 8 Examples

In this section, we demonstrate the effectiveness of the method in a simulation. Starting at time  $t = t_0$ , the original graph is as shown at the top of Figure 1. The graph evolves according to a type of Erdős-Rényi random graph model. Initially we set  $\Theta = 0.25I_{p \times p}$ , where  $p = 50$ . Then, we randomly select 50 edges and update  $\Theta$  as follows: for each new edge  $(i, j)$ , a weight  $a > 0$  is chosen uniformly at random from  $[0.1, 0.3]$ ; we subtract  $a$  from  $\theta_{ij}$  and  $\theta_{ji}$ , and increase  $\theta_{ii}, \theta_{jj}$  by  $a$ . This keeps  $\Sigma$  positive definite. When we later delete an existing edge from the graph, we reverse the above procedure with its weight. Weights are assigned to the initial 50 edges, and then we change the graph structure periodically as follows: Every 200 discrete time steps, five existing edges are deleted, and five new edges are added. However, for each of the five new edges, a target weight is chosen, and the weight on the edge is gradually changed over the ensuing 200 time steps in order ensure smoothness. Similarly, for each of the five edges to be deleted, the weight gradually decays to zero over the ensuing 200 time steps. Thus, almost always, there are 55 edges in the graph and 10 edges have weights that are varying smoothly.

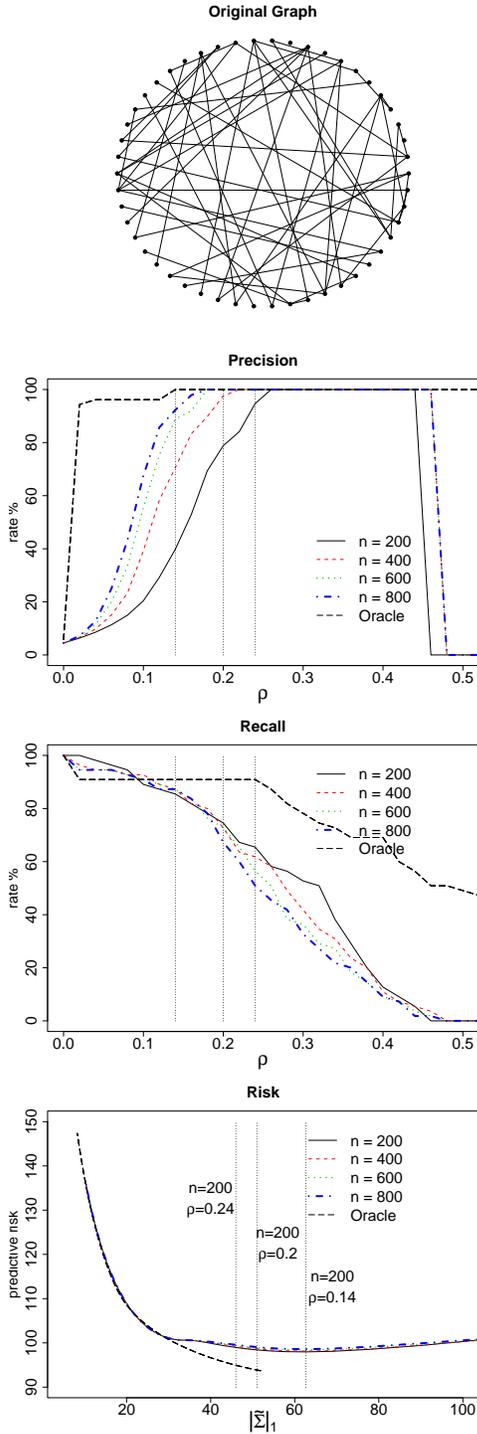


Figure 1: Plots from top to bottom show that as the penalization parameter  $\rho$  increases, precision goes up, and then down as no edges are predicted in the end. Recall goes down as the estimated graphs are missing more and more edges. The oracle  $\Sigma^*$  performs the best, given the same value for  $|\widehat{\Sigma}_n(t_0)|_1 = |\Sigma^*|_1, \forall n$ .

### 8.1 Regularization Paths

We increase the sample size from  $n = 200$ , to 400, 600, and 800 and use a Gaussian kernel with bandwidth  $h = \frac{5.848}{n^{1/3}}$ . We use the following metrics to evaluate model consistency risk for (3) and predictive risk (4) in Figure 1 as the  $\ell_1$  regularization parameter  $\rho$  increases.

- Let  $\widehat{F}_n$  denote edges in estimated  $\widehat{\Theta}_n(t_0)$  and  $F$  denote edges in  $\Theta(t_0)$ . Let us define

$$\text{precision} = 1 - \frac{|\widehat{F}_n \setminus F|}{|\widehat{F}_n|} = \frac{|\widehat{F}_n \cap F|}{|\widehat{F}_n|},$$

$$\text{recall} = 1 - \frac{|F \setminus \widehat{F}_n|}{|F|} = \frac{|\widehat{F}_n \cap F|}{|F|}.$$

Figure 1 shows how they change with  $\rho$ .

- Predictive risks in (4) are plotted for both the oracle estimator (6) and empirical estimators (7) for each  $n$ . They are indexed with the  $\ell_1$  norm of various estimators vectorized; hence  $|\cdot|_1$  for  $\widehat{\Sigma}_n(t_0)$  and  $\Sigma^*(t_0)$  are the same along a vertical line. Note that  $|\Sigma^*(t_0)|_1 \leq |\Sigma(t_0)|_1, \forall \rho \geq 0$ ; for every estimator  $\widehat{\Sigma}$  (the oracle or empirical),  $|\widehat{\Sigma}|_1$  decreases as  $\rho$  increases, as shown in Figure 1 for  $|\widehat{\Sigma}_{200}(t_0)|_1$ .

Figure 2 shows a subsequence of estimated graphs as  $\rho$  increases for sample size  $n = 200$ . The original graph at  $t_0$  is shown in Figure 1.

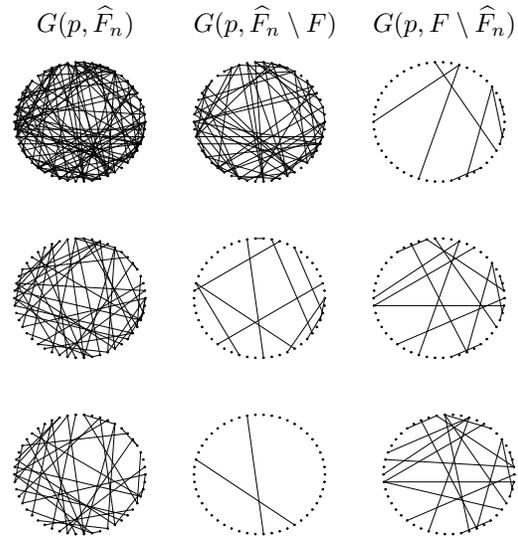


Figure 2:  $n = 200$  and  $h = 1$  with  $\rho = 0.14, 0.2, 0.24$  indexing each row. The three columns show sets of edges in  $\widehat{F}_n$ , extra edges, and missing edges with respect to the true graph  $G(p, F)$ . This array of plots show that  $\ell_1$  regularization is effective in selecting the subset of edges in the true model  $\Theta(t_0)$ , even when the samples before  $t_0$  were from graphs that evolved over time.

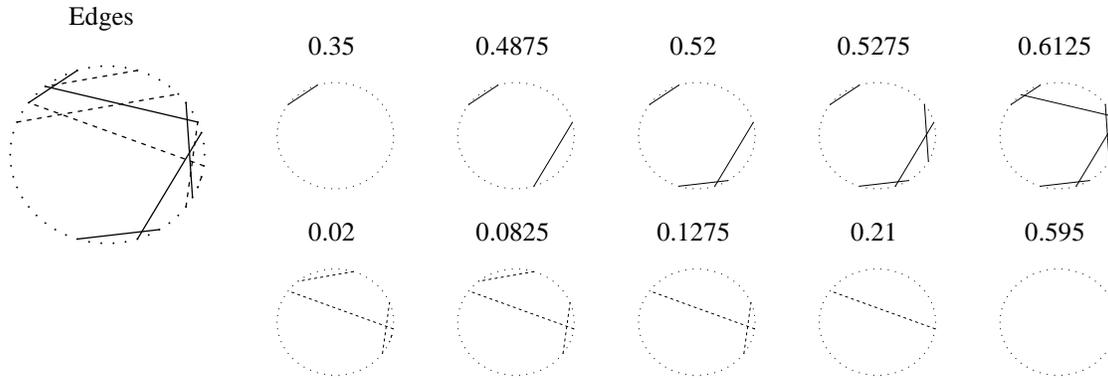


Figure 3: There are 400 discrete steps in  $[0, 1]$  such that the edge set  $F(t)$  remains unchanged before or after  $t = 0.5$ . This sequence of plots shows the times at which each of the new edges added at  $t = 0$  appears in the estimated graph (top row), and the times at which each of the old edges being replaced is removed from the estimated graph (bottom row), where the weight decreases from a positive value in  $[0.1, 0.3]$  to zero during the time interval  $[0, 0.5]$ . Solid and dashed lines denote new and old edges respectively.

## 8.2 Chasing the Changes

Finally, we show how quickly the smoothed estimator using GLASSO [FHT07] can include the edges that are being added in the beginning of interval  $[0, 1]$ , and get rid of edges being replaced, whose weights start to decrease at  $x = 0$  and become 0 at  $x = 0.5$  in Figure 3.

## 9 Conclusions and Extensions

We have shown that if the covariance changes smoothly over time, then minimizing an  $\ell_1$ -penalized kernel risk function leads to good estimates of the covariance matrix. This, in turn, allows estimation of time varying graphical structure. The method is easy to apply and is feasible in high dimensions.

We are currently addressing several extensions to this work. First, with stronger conditions we expect that we can establish *sparsistency*, that is, we recover the edges with probability approaching one. Second, we can relax the smoothness assumption using nonparametric changepoint methods [GH02] which allow for jumps. Third, we used a very simple time series model; extensions to more general time series models are certainly feasible.

## References

- [BGd08] O. Banerjee, L. E. Ghaoui, and A. d’Aspremont. Model selection through sparse maximum likelihood estimation. *Journal of Machine Learning Research*, 9:485–516, March 2008.
- [BL08] P.J. Bickel and E. Levina. Covariance regularization by thresholding. *The Annals of Statistics*, 2008. To appear.
- [DP04] M. Drton and M.D. Perlman. Model selection for gaussian concentration graphs. *Biometrika*, 91(3):591–602, 2004.

- [FHT07] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostat*, 2007.
- [GH02] G. Grégoire and Z. Hamrouni. Change point estimation by local linear smoothing. *J. Multivariate Anal.*, 83:56–83, 2002.
- [GR04] E. Greenshtein and Y. Ritov. Persistency in high dimensional linear predictor-selection and the virtue of over-parametrization. *Journal of Bernoulli*, 10:971–988, 2004.
- [LF07] Clifford Lam and Jianqing Fan. Sparsistency and rates of convergence in large covariance matrices estimation, 2007. arXiv:0711.3933v1.
- [MB06] N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.
- [RBLZ07] A.J. Rothman, P.J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation, 2007. Technical report 467, Dept. of Statistics, Univ. of Michigan.

## A Large Deviation Inequalities for Boxcar Kernel Function

In this section, we prove the following lemma, which implies the i.i.d case as in the corollary.

**Lemma 35** *Using a boxcar kernel that weighs uniformly over  $n$  samples  $Z_k \sim N(0, \Sigma(k)), k = 1, \dots, n$ , that are independently but not identically distributed, we have for  $\epsilon$  small enough, for some  $c_2 > 0$ ,*

$$\mathbf{P} \left( |\widehat{S}_n(t, i, j) - \mathbf{E}\widehat{S}_n(t, i, j)| > \epsilon \right) \leq \exp \{-c_2 n \epsilon^2\}.$$

**Corollary 36** *For the i.i.d. case, for some  $c_3 > 0$ ,*

$$\mathbf{P} \left( |\widehat{S}_n(i, j) - \mathbf{E}\widehat{S}_n(i, j)| > \epsilon \right) \leq \exp \{-c_3 n \epsilon^2\}.$$

Lemma 35 is implied by Lemma 37 for diagonal entries, and Lemma 38 for non-diagonal entries.

### A.1 Inequalities for Squared Sum of Independent Normals with Changing Variances

Throughout this section, we use  $\sigma_i^2$  as a shorthand for  $\sigma_{ii}$  as before. Hence  $\sigma_i^2(x_k) = \text{Var}(Z_{k,i}) = \sigma_{ii}(x_k), \forall k = 1, \dots, n$ . Ignoring the bias term as in (29), we wish to show that each of the diagonal entries of  $\widehat{\Sigma}_{ii}$  is close to  $\sigma_i^2(x_0), \forall i = 1, \dots, p$ . For a boxcar kernel that weighs uniformly over  $n$  samples, we mean strictly  $\ell_k(x_0) = \frac{1}{n}, \forall k = 1, \dots, n$ , and  $h = 1$  for (28) in this context. We omit the mention of  $i$  or  $t$  in all symbols from here on. The following lemma might be of its independent interest; hence we include it here. We omit the proof due to its similarity to that of Lemma 15.

**Lemma 37** *We let  $z_1, \dots, z_n$  represent a sequence of independent Gaussian random variables such that  $z_k \sim N(0, \sigma^2(x_k))$ . Let  $\sigma^2 = \frac{1}{n} \sum_{k=1}^n \sigma^2(x_k)$ . Using a boxcar kernel that weighs uniformly over  $n$  samples,  $\forall \epsilon < c\sigma^2$ , for some  $c \geq 2$ , we have*

$$\mathbf{P} \left( \left| \frac{1}{n} \sum_{k=1}^n z_k^2 - \sigma^2 \right| > \epsilon \right) \leq \exp \left\{ \frac{-(3c-5)n\epsilon^2}{3c^2\sigma^2\sigma_{\max}^2} \right\},$$

where  $\sigma_{\max}^2 = \max_{k=1, \dots, n} \{\sigma^2(x_k)\}$ .

### A.2 Inequalities for Independent Sum of Products of Correlated Normals

The proof of Lemma 38 follows that of Lemma 15.

**Lemma 38** *Let  $\Psi_2 = \frac{1}{n} \sum_{k=1}^n \frac{(\sigma_i^2(x_k)\sigma_j^2(x_k) + \sigma_{ij}^2(x_k))}{2}$  and  $c_4 = \frac{3}{20\Psi_2}$ . Using a boxcar kernel that weighs uniformly over  $n$  samples, for  $\epsilon \leq \frac{\Psi_2}{\max_k(\sigma_i(x_k)\sigma_j(x_k))}$ ,*

$$\mathbf{P} \left( |\widehat{S}_n(t, i, j) - \mathbf{E}\widehat{S}_n(t, i, j)| > \epsilon \right) \leq \exp \{-c_4 n \epsilon^2\}.$$

## B Proofs for Large Deviation Inequalities

### B.1 Proof of Claim 16

We show one inequality; the other one is bounded similarly.  $\forall k$ , we compare the  $k^{\text{th}}$  elements  $\Phi_{2,k}, \Phi_{4,k}$  that appear in the sum for  $\Phi_2$  and  $\Phi_4$  respectively:

$$\begin{aligned} \frac{\Phi_{4,k}}{\Phi_{2,k}} &= \frac{(a_k^4 + b_k^4)4t^2}{(a_k^2 + b_k^2)4t^4} \\ &= \left( \frac{2}{h} K \left( \frac{x_k - x_0}{h} \right) \sigma_i(x_k) \sigma_j(x_k) \right)^2 \cdot \frac{2((1 + \rho_{ij}(x_k))^4 + (1 - \rho_{ij}(x_k))^4)}{8(1 + \rho_{ij}^2(x_k))} \\ &\leq \max_k \left( \frac{2}{h} K \left( \frac{x_k - x_0}{h} \right) \sigma_i(x_k) \sigma_j(x_k) \right)^2 \cdot \max_{0 \leq \rho \leq 1} \frac{(1 + \rho)^4 + (1 - \rho)^4}{4(1 + \rho^2)} = 2M^2. \quad \square \end{aligned}$$

### B.2 Proof of Lemma 17

We first use the Taylor expansions to obtain:

$$\ln(1 - a_k) = -a_k - \frac{a_k^2}{2} - \frac{a_k^3}{3} - \frac{a_k^4}{4} - \sum_{l=5}^{\infty} \frac{(a_k)^l}{l},$$

where,

$$\sum_{l=5}^{\infty} \frac{(a_k)^l}{l} \leq \frac{1}{5} \sum_{l=5}^{\infty} (a_k)^5 = \frac{a_k^5}{5(1 - a_k)} \leq \frac{2a_k^5}{5} \leq \frac{a_k^4}{5}$$

for  $a_k < 1/2$ ; Similarly,

$$\ln(1 + b_k) = \sum_{n=1}^{\infty} \frac{(-1)^{l-1}(b_k)^l}{l}, \quad \text{where}$$

$$\sum_{l=4}^{\infty} \frac{(-1)^l(b_k)^l}{l} > 0 \quad \text{and} \quad \sum_{l=5}^{\infty} \frac{(-1)^n(b_k)^l}{l} < 0.$$

Hence for  $b_k \leq a_k \leq \frac{1}{2}, \forall k$ ,

$$\begin{aligned} &\frac{1}{2} \sum_{k=1}^n \ln \frac{1}{(1 - a_k)(1 + b_k)} \\ &\leq \sum_{k=1}^n \frac{a_k - b_k}{2} + \frac{a_k^2 + b_k^2}{4} + \frac{a_k^3 - b_k^3}{6} + \frac{9}{5} \frac{a_k^4 + b_k^4}{8} \\ &= nt\Phi_1 + nt^2\Phi_2 + nt^3\Phi_3 + \frac{9}{5}nt^4\Phi_4. \quad \square \end{aligned}$$

### B.3 Proof of Claim 18

We replace the sum with the Riemann integral, and then use Taylor's Formula to replace  $\sigma_i(x_k), \sigma_j(x_k)$ , and  $\sigma_{ij}(x_k)$ ,

$$\begin{aligned} \Phi_2(i, j) &= \frac{1}{n} \sum_{k=1}^n \frac{2}{h^2} K^2 \left( \frac{x_k - x_0}{h} \right) (\sigma_i^2(x_k)\sigma_j^2(x_k) + \sigma_{ij}^2(x_k)) \\ &\approx \int_{x_n}^{x_0} \frac{2}{h^2} K^2 \left( \frac{u - x_0}{h} \right) (\sigma_i^2(u)\sigma_j^2(u) + \sigma_{ij}^2(u)) du \\ &= \frac{2}{h} \int_{-\frac{1}{h}}^0 K^2(v) (\sigma_i^2(x_0 + hv)\sigma_j^2(x_0 + hv) + \sigma_{ij}^2(x_0 + hv)) dv \\ &= \frac{2}{h} \int_{-1}^0 K^2(v) \left( \sigma_i(x_0) + hv\sigma'_i(x_0) + \frac{\sigma_i''(y_1)(hv)^2}{2} \right)^2 \\ &\quad \left( \sigma_j(x_0) + hv\sigma'_j(x_0) + \frac{\sigma_j''(y_2)(hv)^2}{2} \right)^2 + \\ &\quad \left( \sigma_{ij}(x_0) + hv\sigma'_{ij}(x_0) + \frac{\sigma_{ij}''(y_3)(hv)^2}{2} \right)^2 dv \\ &= \frac{2}{h} \int_{-1}^0 K^2(v) ((1 + \rho_{ij}^2(x_0))\sigma_i^2(x_0)\sigma_k^2(x_0)) dv + \\ &\quad C_2 \int_{-1}^0 vK^2(v)dv + O(h) \\ &= \frac{C_1(1 + \rho_{ij}^2(x_0))\sigma_i^2(x_0)\sigma_j^2(x_0)}{h} \end{aligned}$$

where  $y_0, y_1, y_2 \leq hv + x_0$  and  $C_1, C_2$  are some constants chosen so that all equalities hold.  $\square$

---

# Learning in the Limit with Adversarial Disturbances

---

Constantine Caramanis\* and Shie Mannor<sup>†‡</sup>

## Abstract

We study distribution-dependent, data-dependent, learning in the limit with adversarial disturbance. We consider an optimization-based approach to learning binary classifiers from data under worst-case assumptions on the disturbance. The learning process is modeled as a decision-maker who seeks to minimize generalization error, given access only to possibly maliciously corrupted data. Two models for the nature of the disturbance are considered: disturbance in the labels of a certain fraction of the data, and disturbance that also affects the position of the data points. We provide distribution-dependent bounds on the amount of error as a function of the noise level for the two models, and describe the optimal strategy of the decision-maker, as well as the worst-case disturbance.

## 1 Introduction

Most of the work on learning in the presence of malicious noise has been within the PAC framework, focusing on *a priori*, distribution independent bounds on generalization error and sample complexity. This work has not fully addressed the question of what a decision-maker must do when faced with a particular realization of the data, and perhaps some knowledge of the underlying distribution and the corrupting disturbance. The main contribution of this paper is the development of a robust optimization-based, algorithmic data-dependent, distribution-dependent approach to minimizing error of learning subject to adversarial disturbance.

In the adversarial PAC setup, a decision-maker has access to IID samples from some source, only that a fraction of these points are altered by an adversary. There are several models for the noise which we discuss below. The decision-maker is given  $\epsilon > 0$  and  $\delta > 0$  and attempts to learn an  $\epsilon$ -optimal classifier with probability of at least  $1 - \delta$ . The emphasis in [KL93], as well as in several follow-up works

(e.g., [BEK02, ACB98, CBDF<sup>+</sup>99, Ser03]) is on the sample complexity of learning in such setups and on particularly bad data sources.

The algorithmic issue of the decision-maker's optimal strategy when faced with a certain disturbance level, i.e., a certain amount of possible data corruption, and a realization of the data has not been adequately explored; see [Lai88] for an initial discussion. While there are quite a few possible disturbance models that differ on the precise setup (what the adversary know, what the adversary can do and in which order), we focus on the strongest disturbance model where the adversary has access to the actual distribution and can modify it adversarially within a constraint on the disturbance level. This "learning in the information limit" model is used to abstract other issues such as finite sample or limited adversary (see [CBDF<sup>+</sup>99] for a discussion on some relevant models). In this paper we consider two different noise models, with the intention of addressing the algorithmic aspects and the effect of the disturbance level. We note that we use the term disturbance rather than noise because in our model data are corrupted in a possibly adversarial way and the probabilistic aspect is essentially not relevant.

We deviate from the traditional learning setup in three major assumptions. First, we focus on the question of how the decision-maker should minimize error, rather than following PAC-style results of computing *a priori* bounds on that error. Moreover, our analysis is distribution specific and we do not focus on particularly bad data sources. Second, the noise level is not assumed small and the decision-maker has to incur error in all but trivial problems (this has been studied in the malicious noise setup; see [CBDF<sup>+</sup>99]). Third, we do not ask how many samples are needed to obtain low generalization error, instead we assume that the distribution of the samples is provided to the decision-maker (equivalently, one may think of this as considering the large sample or "information theoretic" limit). However, this distribution is corrupted by potentially persistent noise; we may consider it as first tampered with by an adversary. After observing the modified distribution, the decision-maker has to commit to a single classifier from some predefined set  $\mathcal{H}$ . The performance of the classifier chosen by the decision-maker is measured on the original, true distribution (this is similar to the agnostic setup of [KSS92]). The question is what should the decision-maker do? And how much error will he incur in the worst case?

In order to answer these questions we adopt a robust

---

\*Department of Electrical and Computer Engineering, The University of Texas at Austin, cmcaram@ece.utexas.edu

<sup>†</sup>Department of Electrical and Computer Engineering, McGill University, shie.mannor@mcgill.ca

<sup>‡</sup>This work was partially supported by NSF Grants CNS-0721532, EFRI-0735905, and the Canada Research Chairs Program

optimization-theoretic perspective, where we regard our decision-maker as trying to make optimal decisions while facing an adversary. Our aim is to provide an analysis by identifying optimal strategies, and quantify the error as a function of the adversary’s strategy, i.e., the nature of the corrupting disturbance. We refer to the disturbance as selected by an adversary merely as a conceptual device, and not in strict analogy to game theory. In particular, the decision-maker does not assume that the corrupting noise is chosen with any specific aim; rather, the decision-maker selects a strategy to protect himself in the worst-case scenario.

The true probability distribution is defined over the input space and on the labels. We focus on the case of proper learning, where this amounts to a distribution and the true classifier. Then the adversary modifies the distribution of the input points and the labels. The decision-maker observes the modified distribution and chooses a classifier in  $\mathcal{H}$  to minimize the *worst-case* error. We note the relationship with [KSS92] who use a slightly different model. In their model, the decision maker chooses a classifier in  $\mathcal{H}$  knowing that the true classifier is in some “touchstone” class  $T \subseteq \mathcal{H}$ . They say that an algorithm facilitates learning (with respect to a loss function) if it learns a function from  $\mathcal{H}$  that is close to a function from  $T$  in the usual PAC sense (i.e., with high probability and small error after observing a number of samples polynomial in one over the error, and one over the confidence). As opposed to [KSS92] and most subsequent works, we do not focus on small noise and we ignore the sample complexity aspect altogether. Instead, we focus on the policy chosen by the decision maker and on the informational limits. In that respect, our work is most related to [CBDF<sup>+</sup>99] who considered the case of substantial noise. Their proposed strategy that deals with noise, however, is based on randomizing two strategies or using majority vote (phase 2 of the randomized Algorithm SIH in [CBDF<sup>+</sup>99]). We propose a more principled approach to handling adversarial noise, leading to improved results.

If the noise level and characteristics are unlimited, the decision-maker cannot hope to do better than randomly guessing. We therefore limit the noise, and allow the adversary to change only a given fraction of the distribution, which we refer to as “the power of the adversary”. An alternative view, which is common in robust optimization [BTN99], is to consider the power of the adversary as a *design parameter*. According to this view, the decision-maker tries to be resilient to a specified amount of uncertainty in the parameters of the problem.

The paper is structured as follows. In Section 2 we describe the setup. We define two types of adversaries: one that can only flip a fraction of the points, and one that can also move the points to another location. In Section 3 we consider the optimal solution pairs for the two different set-ups. We characterize the strategy of both the decision-maker and the adversary as a function of the level of noise (the power of the adversary) and the specific distribution that generates the data. Taking such a distribution-dependent perspective allows us to characterize the decision-maker’s optimal strategy as the solution to a linear program if the adversary can only flip labels, or a robust optimization problem in the case of the more powerful adversary that can also modify the measure.

We further bound the error that may be incurred and show that in the worst case, both adversaries can cause an error twice their power. In Section 4 we show how performance degrades with the increase of this power. A technical proof along with a somewhat surprising worked out example are deferred to the online appendix [CM08].

## 2 Setup and Definitions

In this section we give the basic definitions of the noisy learning setup. Also, we formulate the optimization problem which characterizes the optimal policy of the decision-maker, and the worst-case noise. The decision-maker, after observing the noisy data, and knowing the power of the adversary, outputs a decision in the classifier space. The disagreement with the true classifier, is the generalization error. The decision-maker’s goal is to minimize this, in the worst case. We allow our decision-maker to output a so-called mixed strategy.<sup>1</sup>

Throughout this paper we focus on proper learning. We let  $\mathcal{H}$  denote a predefined set of classifiers from which the true classifier is drawn, and from which the decision-maker must choose. Moreover, we assume that  $\mathcal{H}$  is finite for the sake of simplicity and to avoid some (involved but straightforward) technicalities. Indeed, there are three natural extensions to our work that we postpone, primarily due to space limitations. First, while we focus on the proper learning setup, the non-proper setup (as in [KSS92]) seems to naturally follow our framework. Second, the case of an infinite set of classifiers  $\mathcal{H}$  could be resolved by eliminating classifiers that are “close” according to the observed measure. This is particularly useful for the flip-only setup where the adversary cannot make two classifiers substantially different. Finally, while we do not consider sample complexity, such results should not be too difficult to derive by imitating the arguments in [CBDF<sup>+</sup>99].

### 2.1 The Learning Model

In this paper, we deviate from the PAC learning setup, and consider an *a priori* fixed underlying distribution  $\mu$ , that generates the location (not the labels) of the training data. Thus the error calculations we make are a function of the power of the adversary and also of the fixed probability measure  $\mu$ . We use the symbol  $\mu$  throughout this paper, exclusively in reference to the true probability distribution which generates the location (not the label) of the points, and hence, is used to determine the generalization error. Given a particular classifier  $\hat{h}$ , a true classifier  $h_{\text{true}}$ , and the underlying probability measure  $\mu$ , the generalization error is given by the error function

$$\mathcal{E}_\mu(h_{\text{true}}; \hat{h}) \triangleq \mu\{x : h_{\text{true}}(x) \neq \hat{h}(x)\}.$$

We can extend this definition to a probability measure over  $\mathcal{H}$ , or, in the game-theory terminology, a mixed strategy over  $\mathcal{H}$ , given by a weighting vector  $\alpha = (\alpha_1, \alpha_2, \dots)$  where  $\sum_i \alpha_i = 1$  and  $\alpha_i \geq 0$ . In that case, denoting the space of mixed strategies by  $\Delta_{\mathcal{H}}$ , and a particular mixed strategy by

<sup>1</sup>That is, rather than commit to a single classifier, our decision-maker can commit to a randomized strategy, involving possibly multiple classifiers.

$\alpha \in \Delta_{\mathcal{H}}$ , we have

$$\mathcal{E}_{\mu}(h_{\text{true}}; \alpha) \triangleq \sum_i \alpha_i \mathcal{E}_{\mu}(h_{\text{true}}; h_i).$$

We note that the mixing is often referred to as ‘‘probabilistic concepts’’ or ‘‘probabilistic hypotheses’’ in machine learning. In the context of learning with adversarial noise see [CBDF<sup>+</sup>99].

## 2.2 The Noise Model and The Decision-Maker

We next define the possible actions of the adversary and of the decision-maker. As discussed above, in this paper we do not consider sample complexity, and effectively consider the situation where the training sample is infinite in size (the information theoretic limit). We model this situation by assuming that rather than training samples, the decision-maker receives a distribution for each of the two labels. Since the adversary modifies this object in various ways (noise is added to the observations) we make some formal definitions which facilitate discussion of this in the sequel.

Let  $\mathcal{X}$  denote the space in which the training data exist. In the typical, finite training data model, the decision-maker has access to a collection of labelled points,  $\{(x_i, l_i)\}$ , where  $x_i \in \mathcal{X}$ , and  $l_i \in \{+, -\}$ . In our case then, the decision-maker receives a probability measure over this space  $\sigma \in \mathcal{M}(\mathcal{X} \times \{+, -\})$  ( $\mathcal{M}$  denotes the space of probability measures). We can represent such a measure  $\sigma$  by a triple  $(\lambda, \mu_+, \mu_-)$ , where  $\mu_+, \mu_-$  are probability measures on  $\mathcal{X}$ , and represent the distribution of the positive and negative-labelled points respectively, and  $\lambda \in [0, 1]$  is the weight (or probability) of the positively labelled region, and  $(1 - \lambda)$  that of the negatively labelled region. The interpretation is that a point-label pair is generated by first choosing a label ‘+’ or ‘-’ with probability  $\lambda$  or  $1 - \lambda$ , respectively, and then a point is generated according to the corresponding distribution,  $\mu_+$  or  $\mu_-$ . Thus, the underlying distribution  $\mu$  generating the location of the points (not the labels) is given by  $(\lambda\mu_+ + (1 - \lambda)\mu_-)$ . Thus, if  $h_{\text{true}}$  is the true classifier, then in the absence of any noise, we would observe  $\sigma = (\lambda, \mu_+, \mu_-)$ , where  $\mu_+$  is the scaled restriction of  $\mu$  to the region  $h_{\text{true}}(+)$   $\triangleq \{x : h_{\text{true}}(x) = +\}$ , and similarly for  $\mu_-$ :

$$\lambda = \mu(h_{\text{true}}(+)); \quad \mu_+ = \frac{\mu \cdot \chi_{\{h_{\text{true}}(+)\}}}{\lambda};$$

$$\mu_- = \frac{\mu \cdot \chi_{\{h_{\text{true}}(-)\}}}{1 - \lambda},$$

where if  $\lambda = 0$  there is no  $\mu_+$ , and if  $\lambda = 1$  there is no  $\mu_-$ . Indeed, the triple  $(\lambda, \mu_+, \mu_-)$  is completely defined by  $\mu$  and the true classifier  $h_{\text{true}}$ . Since  $\mu$  is fixed, we write  $(\lambda, \mu_+, \mu_-)_{h_{\text{true}}}$  to denote the triple determined by  $\mu$  and  $h_{\text{true}}$ .

Using this terminology, the adversary’s action is a map

$$T : \mathcal{M}(\mathcal{X} \times \{+, -\}) \longrightarrow \mathcal{M}(\mathcal{X} \times \{+, -\})$$

$$(\lambda, \mu_+, \mu_-) \longmapsto (\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-).$$

We use the hat symbol, ‘ $\hat{\cdot}$ ’ throughout, to denote the observation of the decision-maker. Therefore, while the true probability measure generating the point location is given, as above, by  $\mu = \lambda\mu_+ + (1 - \lambda)\mu_-$ , the decision-maker

observes an underlying probability measure of the form  $\hat{\mu} = \hat{\lambda}\hat{\mu}_+ + (1 - \hat{\lambda})\hat{\mu}_-$ .

The restrictions on this map determine the nature and level of noise. We consider two models for the noise, i.e., two adversaries. First, we have a ‘flip-only’ adversary, corresponding to the noise model where the adversary can flip some fixed fraction of the labels. We also consider a stronger ‘move-and-flip’ adversary who can not only flip a constant fraction of the points, but may also change their location. For the flip-only adversary the underlying measure  $\mu$  is the same as the observed measure  $\hat{\mu}$ . Therefore the decision-maker minimizes the worst-case error where the worst case is over all possible  $h \in \mathcal{H}$ . This need not be true for the move-and-flip adversary. In this case, the decision-maker has only partial information of the measure  $\mu$  against which generalization error is computed, and hence the decision-maker must protect himself against the worst-case error, considering all possible classifiers  $h \in \mathcal{H}$ , as well as all possible underlying measures  $\tilde{\mu}$  consistent with the observations  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$ .

We do not intend measurability questions to be an issue in this paper. Therefore we assume throughout that all measures (and images under the adversary’s action) are measurable with respect to some natural  $\sigma$ -field  $\mathcal{G}$ .

In each of the two cases above, the level of noise is determined by how different the output probability measure  $T(\lambda, \mu_+, \mu_-) = (\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$  can be from the true probability measure  $(\lambda, \mu_+, \mu_-)$ . A natural measure for this is the notion of total variation. The distance, in total variation, between measures  $\nu_1, \nu_2$  is defined as

$$\|\nu_1 - \nu_2\|_{TV} = \frac{1}{2} \sup_{\substack{k, A_1, \dots, A_k \in \mathcal{G} \\ \text{s.t. } A_i \cap A_j = \emptyset \text{ for } i \neq j}} \sum_{i=1}^k |\nu_1(A_i) - \nu_2(A_i)|.$$

This definition also holds for unnormalized measures. We extend this definition to triples  $(\lambda, \mu_+, \mu_-)$  by

$$\|(\lambda, \mu_+, \mu_-) - (\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)\|_{TV} \triangleq \|\lambda\mu_+ - \hat{\lambda}\hat{\mu}_+\|_{TV} + \|(1 - \lambda)\mu_- - (1 - \hat{\lambda})\hat{\mu}_-\|_{TV}.$$

Therefore, we have:

**Definition 1** *An adversary using policy  $T$  (either flip-only, or move-and-flip) has power  $\eta$  if given any triple  $(\lambda, \mu_+, \mu_-)$ , his policy  $T$  satisfies  $\|T(\lambda, \mu_+, \mu_-) - (\lambda, \mu_+, \mu_-)\|_{TV} \leq \eta$ . We abbreviate this, and simply write  $\|T\| \leq \eta$ .*

We can now define the two notions of adversary introduced above.

**Definition 2** *A flip-only adversary of power  $\eta$  can choose any policy  $T$  such that  $\|T\| \leq \eta$ , and  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-) = T(\lambda, \mu_+, \mu_-)$  satisfies*

$$\mu = \lambda\mu_+ + (1 - \lambda)\mu_- = \hat{\lambda}\hat{\mu}_+ + (1 - \hat{\lambda})\hat{\mu}_- = \hat{\mu}.$$

**Definition 3** *A move-and-flip adversary of power  $\eta$  can choose any policy  $T$  such that  $\|T\| \leq \eta$ .*

The decision-maker must base his decision on the ‘noisy observations’ he receives, in other words, on the triple  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$  =  $T(\lambda, \mu_+, \mu_-)$  which he sees. His goal is to minimize the worst-case generalization error, where the worst case is taken over consistent  $h \in \mathcal{H}$ , and also over consistent measures  $\tilde{\mu}$ . We allow our decision-maker to play a so-called mixed strategy, and rather than output a single classifier  $h \in \mathcal{H}$ , to output a randomized strategy,  $\alpha$ , interpreted to mean that classifier  $h_i$  is chosen with probability  $\alpha_i$ . We denote the set of these mixed strategies by  $\Delta_{\mathcal{H}}$ , and a particular mixed strategy by  $\alpha \in \Delta_{\mathcal{H}}$ . Then, the decision-maker’s strategy is a map:

$$D_{\eta, \mathcal{H}} : \mathcal{M}(\mathcal{X} \times \{+, -\}) \longrightarrow \Delta_{\mathcal{H}} \\ (\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-) \longmapsto \alpha.$$

The idea is that if the decision-maker can eliminate some elements of  $\mathcal{H}$ , but cannot identify a unique optimal choice, then the resulting strategy  $D_{\eta, \mathcal{H}}$  will output some measure supported over the ambiguous elements of  $\mathcal{H}$ . We explicitly assume that the decision-maker’s policy is a function of  $\eta$ , the power of the adversary. In a worst-case formulation, a decision-maker without knowledge of  $\eta$  is necessarily powerless. We also assume that the decision-maker knows whether the adversary has flip-only, or move-and-flip power. We do not assume that the decision-maker has any knowledge of the underlying distribution  $\mu$  that generates the location of the points. For the flip-only adversary, the decision-maker receives exact knowledge ‘for free’ since by ignoring the  $\{+, -\}$ -labels, he obtains the true underlying distribution  $\mu$ . Therefore in this case there is only a single consistent underlying measure, namely, the correct measure  $\mu$ , and the decision-maker need only protect against the worst-case  $h \in \mathcal{H}$ . In the case of the move-and-flip adversary, however, the decision-maker receives only partial knowledge of the probability measure that generates the location of the points.

Given a strategy  $D$  of the decision maker and a rule  $T$  for the adversary, we define the error for a given measure  $\mu$  and a true classifier  $h_{\text{true}}$  as:

$$\text{Error}(\mu, h_{\text{true}}, \eta, D, T) \triangleq [\mathcal{E}_{\mu}(h_{\text{true}}; D(T((\lambda, \mu_+, \mu_-)_{h_{\text{true}}})))] \quad (2.1)$$

### 2.3 An Optimization-Based Characterization

In this section we characterize the optimal policy of the decision-maker, and also the worst-case policy of the adversary, i.e., the worst-case noise, given the policy of the decision-maker. The noise-selecting adversary has access to the true triple  $(\lambda, \mu_+, \mu_-)$ , and seeks to maximize the true error incurred. The decision-maker sees only the corrupted version  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$ , and minimizes the worst-case error, where the worst case is taken over all possible, or consistent triples  $(\tilde{\lambda}, \tilde{\mu}_+, \tilde{\mu}_-)$  that the particular adversary with power  $\eta$  (flip-only, or move-and-flip) could, under any policy, map to the observed triple  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$ .<sup>2</sup>

For the flip-only adversary, any consistent triple  $(\tilde{\lambda}, \tilde{\mu}_+, \tilde{\mu}_-)$  the decision-maker considers must satisfy  $\tilde{\lambda}\tilde{\mu}_+ + (1 - \tilde{\lambda})\tilde{\mu}_- =$

$\mu$ . Therefore the worst case over all consistent triples becomes a worst case over all consistent classifiers.

When facing the move-and-flip adversary, it may no longer be true that  $\tilde{\lambda}\tilde{\mu}_+ + (1 - \tilde{\lambda})\tilde{\mu}_- = \mu$ . Therefore the decision-maker must consider the worst case over all consistent classifiers, and also over all consistent underlying measures  $\nu$  such that  $\nu = \tilde{\lambda}\tilde{\mu}_+ + (1 - \tilde{\lambda})\tilde{\mu}_-$  for some possible  $(\tilde{\lambda}, \tilde{\mu}_+, \tilde{\mu}_-)$  with total variation at most  $\eta$  from  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$ . We refer to this set of consistent underlying measures as

$$\Phi \triangleq \Phi(\eta, (\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)).$$

We define the following two setups for a fixed measure  $\mu$  on  $\mathcal{X}$ ,  $h_{\text{true}} \in \mathcal{H}$ , and a value  $\eta$  for the power of the adversary.

(S1) The flip-only setup:

$$D_1 \triangleq \underset{D_{\eta, \mathcal{H}}}{\text{argmin}} \left[ \max_{T: \|T\| \leq \eta} \left[ \max_{h \in \mathcal{H}} \text{Error}(\mu, h, \eta, D, T) \right] \right] \quad (2.2)$$

$$T_1 \triangleq \underset{\substack{T: \|T\| \leq \eta \\ T \text{ flip-only}}}{\text{argmax}} [\text{Error}(\mu, h_{\text{true}}, \eta, D_1, T)].$$

The decision-maker knows  $\eta$  and  $\mathcal{H}$ , and can infer  $\mu$  since the adversary is flip-only. Thus he chooses  $D_1$  to minimize the worst-case error, where the worst case is over classifiers  $h \in \mathcal{H}$ . The adversary has prior knowledge of  $\mu$ ,  $h_{\text{true}}$  and  $\mathcal{H}$ , and of course  $\eta$ , and chooses his strategy to maximize the true error, i.e., the error with respect to  $h_{\text{true}}$  and  $\mu$ .

(S2) The move-and-flip setup:

$$D_2 \triangleq \underset{D_{\eta, \mathcal{H}}}{\text{argmin}} \left[ \max_{T: \|T\| \leq \eta} \left[ \max_{\substack{\nu \in \Phi \\ h \in \mathcal{H}}} \text{Error}(\nu, h, \eta, D, T) \right] \right] \quad (2.3)$$

$$T_2 \triangleq \underset{T: \|T\| \leq \eta}{\text{argmax}} [\text{Error}(\mu, h_{\text{true}}, \eta, D_2, T)].$$

Here the adversary is no longer constrained to pick  $T$  so that  $\hat{\mu} = \mu$ . In this case the decision-maker must choose a policy  $D_2$  to minimize the worst-case generalization error, with respect to  $h \in \mathcal{H}$  and also measures  $\nu \in \Phi$ . The adversary again tries to maximize the true error w.r.t.  $h_{\text{true}}$  and  $\mu$ .

We use  $\text{Error}^i$  ( $i = 1, 2$ ) to denote the error in S1 and S2 when  $\mu, h_{\text{true}}$ , and  $\eta$  are clear from the context, i.e.,  $\text{Error}^i = \text{Error}(\eta, h_{\text{true}}, \eta, D_i, T_i)$ . We show below that the max and min in both (2.2) and (2.3) are attained, and can be computed by solving appropriate optimization problems. We interpret the argmin/argmax as selecting an arbitrary optimal solution if there are more than one.

The fact that the max and min in both (2.2) and (2.3) are attained by some rule requires a proof. We show below that this is indeed the case for both setups since the respective rules can be computed by solving appropriate optimization problems.

<sup>2</sup>We remark again that unlike the game-theoretic setup, the decision-maker does not assume a rational adversary. We consider this case elsewhere.

### S1 and S2 are not equivalent.

We first show by example that the “flip only” setup and the “move and flip” setup are not equivalent. This is the case even for two classifiers. Indeed, consider the case  $\mathcal{X} = [-5, 5] \subseteq \mathbb{R}$ , with threshold classifiers  $\mathcal{H} = \{h_1, h_2\}$  with  $h_1(+) = [0, 5]$  and  $h_2(+) = [1, 5]$ . Then the disagreement region is  $[0, 1)$ . Suppose  $h_1$  is the true classifier, and that the true underlying measure  $\mu$  is uniform on  $[-5, 5]$ , so that  $\mu([0, 1)) = 10\%$ . For  $\eta < 5\%$ ,  $\text{Error}^1 = \text{Error}^2 = 0$ . For  $\eta \geq 5\%$ , however, both the flip-only and move-and-flip adversaries can cause error. Suppose  $\eta = 10\%$ . In S1, the decision-maker knows the true  $\mu$ , and hence knows that  $\mu([0, 1)) = \eta = 10\%$ . Thus regardless of the action of the adversary, the decision-maker’s optimal strategy is  $(\alpha_1, \alpha_2) = (1/2, 1/2)$ , and the error is therefore  $\text{Error}^1 = 10/2 = 5\%$ . In S2, however, the optimal strategy of the adversary is unique: flip the labels of all the points in  $[0, 1)$ . The decision-maker sees  $\hat{\mu}([0, 1)) = 10\%$ , but because the adversary has move-power, the decision-maker does not know  $\mu$  exactly. His goal is to minimize the error in the worst case, where now the worst case is over classifiers, and also over possible underlying measures. From his observations, the decision-maker can only conclude that if  $h_{\text{true}} = h_1$  then  $0\% \leq \mu([0, 1)) \leq 10\%$ , and if  $h_{\text{true}} = h_2$ , then  $0\% \leq \mu([0, 1)) \leq 20\%$ . The worst-case error corresponding to a strategy  $(\alpha_1, \alpha_2)$  is therefore  $\max\{10\alpha_1; 20\alpha_2\}$ . Minimizing this objective function subject to  $\alpha_1 + \alpha_2 = 1$ , and  $\alpha_1, \alpha_2 \geq 0$ , we find  $(\alpha_1, \alpha_2) = (1/3, 2/3)$ , and the true error (as opposed to the worst-case error) is  $\text{Error}^2 = (1/3) \cdot 0 + (2/3) \cdot 10 = 20/3$ , which is greater than  $\text{Error}^1$ .

## 3 Optimal Strategy and Worst-Case Noise

In this section we consider S1 and S2, and determine optimal strategies for the decision-maker, and the optimal strategy for the adversary, i.e., the worst-case noise.

### 3.1 The Decision-Maker in S1

First we consider the decision-maker’s optimal strategy for S1, i.e., in the face of the flip-only adversary. The decision-maker outputs a mixed strategy  $\alpha \in \Delta_{\mathcal{H}}$ . The support of the weight vector  $\alpha$  is the subset  $\mathcal{F}$  of ‘feasible’ classifiers in  $\mathcal{H}$ , which incur at most error  $\eta$ . This set is often referred to as the “version space”.

**Definition 4** Given the output  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-) = T(\lambda, \mu_+, \mu_-)$  of a flip-only adversary with power  $\eta$ , the set of feasible, and hence ambiguous classifiers,  $\mathcal{F} \triangleq \mathcal{F}_{\eta}(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-) \subseteq \mathcal{H}$ , is given by

$$\mathcal{F} \triangleq \{h \in \mathcal{H} : \hat{\lambda}\hat{\mu}_+(h(-)) + (1 - \hat{\lambda})\hat{\mu}_-(h(+)) \leq \eta\}. \quad (3.4)$$

Here we define  $h(+)$  to be the positively labelled region, and  $h(-)$  the negatively labelled region, so that  $\hat{\lambda}\hat{\mu}_+(h(-))$  is the measure of the positive labels observed in the region  $h(-)$ . The measure of the region where the true classifier disagrees with the observed measure can be at most  $\eta$ . That is,

$$\hat{\lambda}\hat{\mu}_+(h_{\text{true}}(-)) + (1 - \hat{\lambda})\hat{\mu}_-(h_{\text{true}}(+)) \leq \eta.$$

This follows by our assumption that the adversary has power  $\eta$ , and because  $\lambda\mu_+(h_{\text{true}}(-)) + (1 - \lambda)\mu_-(h_{\text{true}}(+)) = 0$ . Therefore,  $\mathcal{F}$  is the set of classifiers in  $\mathcal{H}$  that could possibly be equal to  $h_{\text{true}}$  and thus Definition 4 above indeed gives the set of feasible, and therefore ambiguous, classifiers. In particular, under the assumption of proper learning,  $h_{\text{true}} \in \mathcal{F}$ .

Next, the decision-maker must compute the value of  $\alpha_h$  for every  $h \in \mathcal{F}$ , the feasible subset of classifiers. For any mixed strategy (this is sometimes referred to as a “probabilistic hypothesis”)  $\alpha \in \Delta_{\mathcal{H}}$  that the decision-maker might choose, the error incurred is

$$\mathcal{E}_{\mu}(h_{\text{true}}; \alpha) = \sum_{h \neq h_{\text{true}}} \alpha_h \mu(\blacktriangle(h, h_{\text{true}})), \quad (3.5)$$

where for any two classifiers  $h', h''$ , we define  $\blacktriangle(h', h'') \triangleq \{x : h'(x) \neq h''(x)\}$  to be the region where they differ.

The decision-maker, however, does not know  $h_{\text{true}}$ , and hence his optimal strategy is the one that minimizes the worst-case error,  $\max_{h_{\text{true}} \in \mathcal{H}} \mathcal{E}_{\mu}(h_{\text{true}}; \alpha)$ . In the case of the flip-only adversary, the decision-maker sees the probability measure  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$ , and since he knows that  $\mu = \hat{\mu}$ , he can correctly compute the value  $\mu(\blacktriangle(h', h''))$  for any two classifiers  $h', h''$ . In other words, the decision-maker knows the true weight of any region where two classifiers disagree, and therefore we can state the following result which is a restatement of the above.

**Proposition 5** The optimal policy of the decision-maker in S1 is given by computing the minimizer of:

$$\min_{\alpha} \max_{h_{\text{true}} \in \mathcal{F}} \sum_{h \neq h_{\text{true}}} \alpha_h \mu(\blacktriangle(h, h_{\text{true}})). \quad (3.6)$$

Enumerating the set  $\mathcal{F}$  as  $\{h_1, \dots, h_k\}$ , the optimal  $\alpha$  is computed by solving the following linear optimization problem:

$$\begin{aligned} \min : & u \\ \text{s.t.} : & u \geq \sum_{i \neq j} \alpha_i \mu(\blacktriangle(h_i, h_j)) \quad j = 1, \dots, k \\ & \sum_i \alpha_i = 1 \\ & \alpha_i \geq 0 \quad i = 1, \dots, k. \end{aligned}$$

PROOF. The proof follows directly from the definition of the error associated to any mixed strategy  $\alpha$ , given in (3.5).  $\square$

We note that in [CBDF<sup>+</sup>99] the question of how to choose the best probabilistic hypothesis was considered. The solution there was to randomize between two (maximally apart) classifiers or to choose a majority vote. We now explain why this is suboptimal. Consider three linear classifiers in general position in the plane  $\mathcal{H} = \{h_1, h_2, h_3\}$  and let’s suppose that there are 7 regions in the plane according to the agreement of the classifiers (assume that  $h_1(+) \cap h_2(+) \cap h_3(+) \neq \emptyset$ ). Suppose that the decision maker observes that  $\hat{\mu}_+$  has support only on  $h_1(+) \cap h_2(+) \cap h_3(+) (assume that  $\hat{\lambda} = 1 - 3\eta$  and that  $\eta < 1/4$ ) and that  $\hat{\mu}_-$  has equal support of  $\eta$  on  $h_1(-) \cap h_2(-) \cap h_3(+)$ ,  $h_1(-) \cap h_2(+) \cap h_3(-)$  and  $h_1(+) \cap h_2(-) \cap h_3(-)$ . The example is constructed so that choosing any one classifier, in the worst case can lead to an error of  $2\eta$ . It is easy to see that a majority vote would lead to$

a worst case error of  $2\eta$ . Mixing between any two classifiers would lead to a worst case error of  $2\eta$  as well. Mixing between the 3 classifiers, which is suggested by Proposition 5 leads to a worst case error of  $4\eta/3$  since we will get the classifier right with probability  $1/3$  and incur the  $2\eta$  loss with probability  $2/3$ .

### 3.2 The Decision-Maker in $S2$

Next we consider the setup  $S2$ , with the more powerful move-and-flip adversary. Again, the goal of the decision-maker is to pick a mixed strategy  $\alpha \in \Delta_{\mathcal{H}}$ , that minimizes the error given in (3.5). The set  $\mathcal{F}$  of ambiguous classifiers is as defined in (3.4). In this case, however, in addition to not knowing  $h_{\text{true}}$ , the decision-maker also does not know the underlying measure  $\mu$ , and hence the values  $\mu(\blacktriangle(h', h''))$ , exactly.

As introduced in Section 2.3, we use  $\Phi \triangleq \Phi(\eta, \hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$  to denote the set of measures consistent with  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$ . Thus the decision-maker seeks to minimize the worst-case error, now over  $\mathcal{H}$  and  $\Phi$ .

Any points that have the wrong label w.r.t.  $h$  could have been both moved and flipped. Therefore, to compute the worst case possible values of  $\mu(\blacktriangle(h', h''))$ , for each classifier  $h$  the decision-maker considers, he must consider the observed measure of the points that have the *correct* label, and the *wrong* label, with respect to  $h$ . Thus we define:

$$\begin{aligned} \hat{\mu}_h^{\text{wrong}}(\blacktriangle(h', h'')) &\triangleq \hat{\lambda}\hat{\mu}_+(\blacktriangle(h', h'') \cap h(-)) + \\ &\quad (1 - \hat{\lambda})\hat{\mu}_-(\blacktriangle(h', h'') \cap h(+)) \\ \hat{\mu}_h^{\text{correct}}(\blacktriangle(h', h'')) &\triangleq \hat{\mu}(\blacktriangle(h', h'')) - \hat{\mu}_h^{\text{wrong}}(\blacktriangle(h', h'')). \end{aligned} \quad (3.7)$$

In Proposition 6 below, the decision-maker uses these quantities to compute his optimal strategy that protects against the worst-case consistent classifier  $h \in \mathcal{F}$ , and underlying measure  $\nu \in \Phi$ . The worst-case classifier  $h$  and measure  $\nu$  may depend on the action  $\alpha$  the decision-maker chooses. Thus, the decision-maker must solve a min max linear program. In doing so, he implicitly computes the worst-case measure  $\nu$  as well, by computing a saddle point.

**Proposition 6 (a)** *The decision-maker's optimal policy, is to compute the set  $\mathcal{F}$ , and then compute the optimal weight-vector  $\alpha$  that is the minimizer of*

$$\min_{\alpha} \max_{\substack{\nu \in \Phi \\ h_{\text{true}} \in \mathcal{H}}} \mathcal{E}_{\nu}(h_{\text{true}}; \alpha) = \min_{\alpha} \max_{\substack{\nu \in \Phi \\ h_{\text{true}} \in \mathcal{H}}} \sum_{h \neq h_{\text{true}}} \alpha_h \nu(\blacktriangle(h, h_{\text{true}})), \quad (3.8)$$

where the max is over  $\mathcal{H}$  and  $\Phi$ . The min and the max are both attained.

**(b)** *Moreover, the optimal strategy of the decision-maker is obtained as the solution to a robust linear optimization problem, which we reformulate as a single linear optimization.*

Recall that in  $S2$ , in addition to the labels, the underlying measure  $\mu$  is also corrupted. Therefore the decision-maker must compute the strategy  $\alpha$  with respect to the worst-case feasible classifier, and the worst-case consistent values for

$\mu(\blacktriangle(h', h''))$ , i.e., the worst-case values for  $\nu(\blacktriangle(h', h''))$  for  $\nu \in \Phi$ .

The worst case over  $\nu$  depends on the worst case over  $h \in \mathcal{H}$ . That is, if  $h_1$  is the true classifier, then the worst-case values for  $\nu(\blacktriangle(h', h''))$  may be different from the worst-case value if  $h_2$  is the true classifier.

The worst-case values are computed using  $\hat{\mu}_h^{\text{wrong}}(\blacktriangle(h', h''))$  and  $\hat{\mu}_h^{\text{correct}}(\blacktriangle(h', h''))$ . The idea is as follows: if some  $h$  is the true classifier, then any measure in the region  $\blacktriangle(h', h'')$  that is incorrectly labelled with respect to  $h$  may have also been moved from some other region. Therefore in the case that  $h = h_{\text{true}}$ , the weight of any particular region  $\blacktriangle(h', h)$  could be as large as the weight of the correctly labeled points under  $\hat{\mu}$ ,  $\hat{\mu}_h^{\text{correct}}(\blacktriangle(h', h''))$ , plus the weight (again under  $\hat{\mu}$ ) of the mislabelled points with respect to  $h$  in all other regions, plus the additional weight that could be moved to  $\blacktriangle(h', h)$  using any 'unused' power of the adversary. The weight of the mislabelled points is

$$\hat{\lambda}\hat{\mu}_+(h(-)) + (1 - \hat{\lambda})\hat{\mu}_-(h(+)).$$

The unused power is

$$\eta - \hat{\lambda}\hat{\mu}_+(h(-)) + (1 - \hat{\lambda})\hat{\mu}_-(h(+)).$$

Therefore the weight (under  $\hat{\mu}$ ) of the mislabelled points with respect to any  $h$ , plus the unused power, must be exactly  $\eta$ .

If  $h = h_{\text{true}}$ , consider some region  $\blacktriangle(h', h)$ . The reasoning above tells us that the worst-case measure of this region is  $\hat{\mu}_h^{\text{correct}}(\blacktriangle(h', h)) + \eta$ . The following lemma makes this intuition precise, and shows that this is indeed the case.

**Lemma 7** *Assume that  $\blacktriangle(h, h') \neq \emptyset$  for any  $h \neq h'$ . Then, if  $h = h_{\text{true}}$ , we have*

$$\mu(\blacktriangle(h, h')) \leq \hat{\mu}_h^{\text{correct}}(\blacktriangle(h, h')) + \eta.$$

*This bound is tight in the sense that there is a measure  $(\tilde{\lambda}, \tilde{\mu}_+, \tilde{\mu}_-)$  with total variation at most  $\eta$  from the observations, that attains the upper bound.*

**PROOF.** We exhibit the following triple  $(\tilde{\lambda}, \tilde{\mu}_+, \tilde{\mu}_-)$  that satisfies  $\|(\tilde{\lambda}, \tilde{\mu}_+, \tilde{\mu}_-) - (\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)\|_{TV} \leq \eta$ : Assume, without loss of generality, that  $\blacktriangle(h, h') \subseteq h(+)$ . Let  $\theta$  be any probability measure over  $\mathcal{X}$ , supported on  $\blacktriangle(h, h')$ . Then, define:

$$\begin{aligned} \tilde{\lambda} &= \hat{\lambda} + (1 - \hat{\lambda})\hat{\mu}_-(h(+)), \\ \tilde{\mu}_- &= \hat{\mu}_- - \hat{\mu}_- \Big|_{h(+)}, \\ \tilde{\mu}_+ &= \frac{\left( \hat{\lambda} \left( \hat{\mu}_+ - \hat{\mu}_+ \Big|_{h(-)} \right) + \kappa \theta \right)}{\hat{\lambda} + (1 - \hat{\lambda})\hat{\mu}_-(h(+))}, \end{aligned}$$

where  $\kappa = \left( (1 - \hat{\lambda})\hat{\mu}_-(h(+)) + \hat{\lambda}\hat{\mu}_+(h(-)) \right)$ . For the triple  $(\tilde{\lambda}, \tilde{\mu}_+, \tilde{\mu}_-)$ , there exists a move-and-flip policy  $T$  with  $\|T\| \leq \eta$ , such that  $T(\tilde{\lambda}, \tilde{\mu}_+, \tilde{\mu}_-) = (\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$ , hence the scalar upper bound is attainable.  $\square$

For the vector case, if  $\blacktriangle(h, h_1) \cap \dots \cap \blacktriangle(h, h_k) \neq \emptyset$ , there exists a triple  $(\tilde{\lambda}, \tilde{\mu}_+, \tilde{\mu}_-)$  that satisfies

$$(\mu(\blacktriangle(h, h_1)), \dots, \mu(\blacktriangle(h, h_k))) = (\hat{\mu}^{\text{correct}}(\blacktriangle(h, h_1)), \dots, \hat{\mu}^{\text{correct}}(\blacktriangle(h, h_k))) + \eta(1, \dots, 1).$$

This follows by replacing  $\blacktriangle(h, h')$  by  $\blacktriangle(h, h_1) \cap \dots \cap \blacktriangle(h, h_k)$  in the proof above. In general, however, the tightness result does not hold simultaneously for many classifiers. That is to say, given classifiers  $\{h_1, \dots, h_k\}$  different from some  $h$ , if  $\blacktriangle(h, h_1) \cap \dots \cap \blacktriangle(h, h_k) = \emptyset$  (as is in general the case), then, while the lemma tells us that  $\mu(\blacktriangle(h, h_i)) \leq \hat{\mu}^{\text{correct}}(\blacktriangle(h, h_i)) + \eta$  for each  $i$ , there will be no measure  $\nu \in \Phi$  which realizes these upper bounds simultaneously. Moreover, the worst-case values then will depend on the decision-maker's particular choice of  $\alpha$ . The  $\alpha$ -dependent worst-case consistent values for  $\mu(\blacktriangle(h', h''))$  are computed implicitly in the robust LP below.

With this intuition, and the result of the lemma, we can now prove the proposition, and explicitly give the LP that yields the optimal strategy of the decision-maker.

**PROOF. (of Proposition 6)** The proof proceeds in three main steps:

- (i) First we show that the error, and hence the optimal strategy of the decision-maker, depends only on a finite dimensional equivalence class of measures  $\nu \in \Phi$ . The first part of the proof is to characterize this finite dimensional set.
- (ii) Next, we establish the connection to robust optimization, and write a robust optimization problem that we claim yields the decision-maker's optimal strategy. Proving this claim is the second part of the proof.
- (iii) Finally, we show that the robust optimization problem may in fact be rewritten as a single LP, using duality theory of linear programming.

For  $\mathcal{F}$  the set of ambiguous classifiers, the decision-maker's policy is given by

$$\min_{\alpha} \max_{\substack{\nu \in \Phi \\ h_{\text{true}} \in \mathcal{F}}} \mathcal{E}_{\nu}(h_{\text{true}}; \alpha) = \min_{\alpha} \max_{\substack{\nu \in \Phi \\ h_{\text{true}} \in \mathcal{F}}} \sum_{h \neq h_{\text{true}}} \alpha_h \nu(\blacktriangle(h, h_{\text{true}})),$$

where  $\alpha$  is supported on  $\mathcal{F}$ . While the worst case is over classifiers  $h \in \mathcal{F}$  and all measures  $\nu \in \Phi$ , the worst-case error incurred for any particular strategy  $\alpha$  in fact can only depend on the values of  $\nu(\blacktriangle(h', h''))$  for every  $h', h'' \in \mathcal{F}$ . Therefore we can consider equivalence classes of measures in  $\Phi$  that have the same values  $\nu(\blacktriangle(h', h''))$ . This reduces the inner maximization to a finite dimensional one. Enumerate the set  $\mathcal{F}$  as  $\{h_1, \dots, h_k\}$ . Then for any fixed  $h_j \in \mathcal{F}$ , if  $h_{\text{true}} = h_j$ , then the regions whose measure is important for the error computation, are those that can be written as

$$\left( \bigcap_{i \in S} \blacktriangle(h_i, h_j) \right) \cap \left( \bigcap_{i \notin S} \blacktriangle(h_i, h_j)^c \right),$$

for some  $S \subseteq \{1, \dots, k\}$ . We use  $\blacktriangle(h_i, h_j)^c$  to denote the complement of the set. We define a variable  $\hat{\xi}_{S,j}$  to represent the amount of mass that can be added (in the worst case) to the region  $(\bigcap_{i \in S} \blacktriangle(h_i, h_j)) \cap (\bigcap_{i \notin S} \blacktriangle(h_i, h_j)^c)$  in the case where  $h_j$  is the true classifier. We can consider these as components of a vector in  $\mathbb{R}^{2^k - 1}$ , indexed by nonempty subsets  $S \subseteq \{1, \dots, k\}$ . Any such vector corresponds to an equivalence class of measures  $\nu \in \Phi$ , that are indistinguishable to the decision-maker, in the sense that they induce precisely the same error. Given such a vector, the weight of the region  $\blacktriangle(h_i, h_j)$  is then  $\left[ \hat{\mu}_{h_j}^{\text{correct}}(\blacktriangle(h_i, h_j)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j} \right]$  and thus for a given  $\alpha$ , the error would be

$$\sum_{i \neq j} \alpha_i \left[ \hat{\mu}_{h_j}^{\text{correct}}(\blacktriangle(h_i, h_j)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j} \right].$$

For any fixed  $j$ , the collection of variables  $(\hat{\xi}_{S,j})_{S \subseteq \{1, \dots, k\}}$  must satisfy four properties in order to correspond to some measure  $\nu \in \Phi$ . The variables must be nonnegative, and the sum over  $S$  of  $\hat{\xi}_{S,j}$  must be at most  $\eta$ . This follows since the total amount of mass moved or flipped must be at most  $\eta$ , by definition of the power of the adversary. Third, if the set  $(\bigcap_{i \in S} \blacktriangle(h_i, h_j)) \cap (\bigcap_{i \notin S} \blacktriangle(h_i, h_j)^c)$  is empty, then the corresponding variable  $\hat{\xi}_{S,j}$  must be zero. Finally, the weight of each region  $\blacktriangle(h_i, h_j)$  can be at most 100%, and thus we must have

$$\left[ \hat{\mu}_{h_j}^{\text{correct}}(\blacktriangle(h_i, h_j)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j} \right] \leq 100\%.$$

Therefore, if  $h_j = h_{\text{true}}$ , the possible values of  $\hat{\xi}_{\cdot,j} \in \mathbb{R}^{2^k - 1}$  are given by:

$$\Xi(j) = \left\{ (\hat{\xi}_{\cdot,j}) : \begin{cases} \sum_S \hat{\xi}_{S,j} \leq \eta, \\ \hat{\xi}_{S,j} \geq 0, \forall S \subseteq \{1, \dots, k\}, S \neq \emptyset, \\ \hat{\xi}_{S,j} = 0, \forall S : (\bigcap_{i \in S} \blacktriangle(h_i, h_j)) \cap (\bigcap_{i \notin S} \blacktriangle(h_i, h_j)^c) = \emptyset, \\ \sum_{S \ni i} \hat{\xi}_{S,j} \leq 100 - \hat{\mu}_{h_j}^{\text{correct}}(\blacktriangle(h_i, h_j)) \\ \forall i \neq j. \end{cases} \right\}$$

For every  $j$ , the set  $\Xi(j)$  is a polytope. The decision-maker must choose some  $\alpha$  that minimizes the worst-case error, where the worst case is over possible  $h_{\text{true}} \in \mathcal{F} = \{h_1, \dots, h_k\}$ , and then once that  $h_j$  is fixed, the worst case over all possible  $(\hat{\xi}_{\cdot,j}) \in \Xi(j)$ . Therefore the optimal strategy  $\alpha$  of the decision-maker is the solution to the following robust opti-

mization problem:

$$\begin{aligned}
\min : & \quad u \\
\text{s.t.} : & \quad u \geq \max_{\{(\hat{\xi}_{S,j})_{S \subseteq \{1, \dots, k\}} \in \Xi(j)\}} \sum_{i \neq j} \alpha_i \\
& \quad \left[ \begin{array}{c} \text{correct} \\ \hat{\mu}_{h_j}(\blacktriangle(h_i, h_j)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j} \end{array} \right], j = 1, \dots, k \\
& \quad \sum_i \alpha_i = 1, \quad \alpha_i \geq 0, \\
\Xi(j) = & \quad \left\{ (\hat{\xi}_{\cdot,j}) : \begin{array}{l} \sum_S \hat{\xi}_{S,j} \leq \eta, \\ \hat{\xi}_{S,j} \geq 0, \forall S \subseteq \{1, \dots, k\}, S \neq \emptyset, \\ \hat{\xi}_{S,j} = 0, \forall S : \left( \bigcap_{i \in S} \blacktriangle(h_i, h_j) \right) \cap \\ \left( \bigcap_{i \notin S} \blacktriangle(h_i, h_j)^c \right) = \emptyset, \\ \sum_{S \ni i} \hat{\xi}_{S,j} \leq 100 - \hat{\mu}_{h_j}^{\text{correct}}(\blacktriangle(h_i, h_j)) \\ \forall i \neq j. \end{array} \right\}
\end{aligned}$$

First we prove that this robust optimization indeed yields the strategy of the decision-maker that minimizes the worst-case effort. The proof of this follows by a combination of the methods used to prove Proposition 5 and Lemma 7. Certainly, for any  $h_j \in \mathcal{F}$  and  $\nu \in \Phi$ , there exists a vector  $(\hat{\xi}_{\cdot,j}) \in \Xi(j)$  such that

$$\nu(\blacktriangle(h_i, h_j)) = \left[ \begin{array}{c} \text{correct} \\ \hat{\mu}_{h_j}(\blacktriangle(h_i, h_j)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j} \end{array} \right], \forall i \neq j.$$

The technique of Lemma 7 establishes the converse, namely, for any feasible vector  $(\hat{\xi}_{\cdot,j}) \in \Xi(j)$  there exists a measure  $\nu \in \Phi$  that is consistent with the observed measure, and such that for any  $i \in \{1, \dots, k\}$ ,

$$\nu(\blacktriangle(h_i, h_j)) = \hat{\mu}_{h_j}^{\text{correct}}(\blacktriangle(h_i, h_j)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j}.$$

Thus we have shown that the sets  $\Xi(j)$  are indeed the sets we should be considering. Next we show that the optimization we write down is the correct one. The proof of this follows that of Proposition 5. Let  $\alpha^*$  be the minimizer of the expression above, and let  $u^*$  be the optimal value of the optimization. If the decision-maker chooses some mixed strategy  $\rho$  that is not a minimizer of the above, then there must exist some  $r \in \{1, \dots, k\}$ , corresponding to some  $h_{\text{true}} \in \mathcal{F}$ , and also a vector  $(\hat{\xi}_{\cdot,r}) \in \Xi(r)$  feasible for the above linear optimization, for which

$$\sum_{i \neq r} \rho_i \left[ \begin{array}{c} \text{correct} \\ \hat{\mu}_{h_r}(\blacktriangle(h_i, h_r)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,r} \end{array} \right] > u^*.$$

Thus, we must have

$$\begin{aligned}
& \sum_{i \neq r} \rho_i \left[ \begin{array}{c} \text{correct} \\ \hat{\mu}_{h_r}(\blacktriangle(h_i, h_r)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,r} \end{array} \right] > \\
& \max_{\substack{j \in \{1, \dots, k\} \\ \hat{\xi}_{S,j} \in \Xi(j)}} \sum_{i \neq j} \alpha_i^* \left[ \begin{array}{c} \text{correct} \\ \hat{\mu}_{h_j}(\blacktriangle(h_i, h_j)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j} \end{array} \right].
\end{aligned}$$

But then there must exist a measure  $\nu \in \Phi$  consistent with the observed measure, for which

$$\nu(\blacktriangle(h_i, h_j)) = \hat{\mu}_{h_j}^{\text{correct}}(\blacktriangle(h_i, h_j)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j},$$

and thus we have:

$$\begin{aligned}
\max_{\substack{\mu \in \Phi \\ h_{\text{true}} \in \mathcal{H}}} \mathcal{E}_\mu(h_{\text{true}}; \rho) & \geq \mathcal{E}_\nu(h_r; \rho) \\
& > \max_{\substack{\nu \in \Phi \\ h_{\text{true}} \in \mathcal{H}}} \mathcal{E}_\nu(h_{\text{true}}; \alpha^*).
\end{aligned}$$

Therefore, if  $\nu$  is indeed the true probability measure generating the location of the points, and if  $h_r$  is the true classifier, then the error incurred by using strategy  $\rho$  is strictly greater than the error incurred using strategy  $\alpha^*$ . Since both  $\nu$  and  $h_r$  are consistent with the observed probability measure and labels, respectively, the mixed strategy  $\rho$  does not minimize the worst-case error.

On the other hand, by similar reasoning, if  $\rho$  is not an optimal strategy, i.e., if it does not minimize the worst-case error as given in (3.8), then it is a strictly suboptimal solution to the linear optimization. This completes the proof that the robust optimization above indeed yields the strategy of the adversary which minimizes the worst-case error, where the worst case is over  $h \in \mathcal{F}$  and also  $\nu \in \Phi$ . This concludes the proofs of parts (i) and (ii).

We have left to prove the second part of the proposition, and part (iii) in the outline, namely, that we can rewrite the robust optimization problem as a single LP. First, we remark that for each  $j$ , the set  $\Xi(j)$  is a polytope. The problem then, is a robust linear optimization problem. Using standard results from duality theory [BTN99], this can be reformulated as an ordinary linear optimization problem.

We have the robust linear optimization problem:

$$\begin{aligned}
\min : & \quad u \\
\text{s.t.} : & \quad u \geq \max_{\{(\hat{\xi}_{S,1})_{S \subseteq \{1, \dots, k\}} \in \Xi(1)\}} \sum_{i \neq 1} \alpha_i \left[ \begin{array}{c} \text{correct} \\ \hat{\mu}_{h_1}(\blacktriangle(h_i, h_1)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,1} \end{array} \right] \\
& \quad u \geq \max_{\{(\hat{\xi}_{S,2})_{S \subseteq \{1, \dots, k\}} \in \Xi(2)\}} \sum_{i \neq 2} \alpha_i \left[ \begin{array}{c} \text{correct} \\ \hat{\mu}_{h_2}(\blacktriangle(h_i, h_2)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,2} \end{array} \right] \\
& \quad \vdots \\
& \quad u \geq \max_{\{(\hat{\xi}_{S,k})_{S \subseteq \{1, \dots, k\}} \in \Xi(k)\}} \sum_{i \neq k} \alpha_i \left[ \begin{array}{c} \text{correct} \\ \hat{\mu}_{h_k}(\blacktriangle(h_i, h_k)) + \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,k} \end{array} \right] \\
& \quad \sum_i \alpha_i = 1, \quad \alpha_i \geq 0.
\end{aligned}$$

Note that the robustification here is constraintwise-rectangular, that is, the uncertainty set has the form

$$\Xi = \Xi(1) \times \dots \times \Xi(k).$$

Therefore, we can consider each constraint individually. Indeed, each inequality can be rewritten as

$$\begin{aligned}
u - \sum_{i \neq j} \alpha_i \text{correct} \hat{\mu}_{h_j}(\blacktriangle(h_i, h_j)) \geq \\
\max_{\{(\hat{\xi}_{S,j})_{S \subseteq \{1, \dots, k\}} \in \Xi(j)\}} \sum_{i \neq j} \alpha_i \left[ \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j} \right],
\end{aligned}$$

and thus we can consider the linear optimization:

$$\begin{aligned}
\max : & \quad \sum_{i \neq j} \alpha_i \left[ \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j} \right] \\
\text{s.t.} : & \quad (\hat{\xi}_{S,j})_{S \subseteq \{1, \dots, k\}} \in \Xi(j).
\end{aligned} \tag{3.9}$$

The objective function is bilinear in both  $\alpha_i$  and  $\hat{\xi}_{S,j}$ . We have

$$\sum_{i \neq j} \alpha_i \left[ \sum_{\substack{S \subseteq \{1, \dots, k\} \\ i \in S}} \hat{\xi}_{S,j} \right] = \sum_{S \subseteq \{1, \dots, k\}} \hat{\xi}_{S,j} \left[ \sum_{i \in S} \alpha_i \right],$$

and hence defining the vector  $c$  by  $c_S = \sum_{i \in S} \alpha_i$  we can write the objective function as  $c' \hat{\xi}_{\cdot, j}$ . The polytope  $\Xi(j)$  is

defined by equalities and inequalities among the variables. Writing these in vector form, we have:

$$\begin{aligned}
-[I] \hat{\xi}_{\cdot, j} & \leq 0, \\
[Q^{(j)}] \hat{\xi}_{\cdot, j} & = 0, \\
(1, 1, \dots, 1)' \hat{\xi}_{\cdot, j} & \leq \eta, \\
[R^{(j)}] \hat{\xi}_{\cdot, j} & \leq (100 - \text{correct} \hat{\mu}_{h_j}(\blacktriangle(h_1, h_j)), \dots, \\
& \quad 100 - \text{correct} \hat{\mu}_{h_j}(\blacktriangle(h_k, h_j))).
\end{aligned} \tag{3.10}$$

Here,  $I$  is the identity matrix,  $Q^{(j)}$  is a subset of the identity matrix corresponding to the sets  $S$  for which we have  $(\bigcap_{i \in S} \blacktriangle(h_i, h_j)) \cap (\bigcap_{i \notin S} \blacktriangle(h_i, h_j)^c) = \emptyset$ , and the generic row of  $R^{(j)}$  contains a 1 in every index containing a particular  $i$ . Writing the equality as  $[Q^{(j)}] \hat{\xi}_{\cdot, j} \leq 0$ , and  $-[Q^{(j)}] \hat{\xi}_{\cdot, j} \leq 0$ , we can express the constraints defining  $\Xi(j)$  more compactly as

$$\Xi(j) = \left\{ (\hat{\xi}_{S,j})_{S \subseteq \{1, \dots, k\}} : A^{(j)} \hat{\xi}_{\cdot, j} \leq b \right\}.$$

The matrices  $A^{(j)}$ , and the vector  $b$ , are given by the vector inequalities in (3.10) above:

$$A^{(j)} = \begin{bmatrix} -I \\ Q^{(j)} \\ -Q^{(j)} \\ R^{(j)} \\ 1 \ 1 \ \dots \ 1 \ 1 \end{bmatrix} \quad b = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 100 - \text{correct} \hat{\mu}_{h_j}(\blacktriangle(h_1, h_j)) \\ \vdots \\ 100 - \text{correct} \hat{\mu}_{h_j}(\blacktriangle(h_k, h_j)) \\ \eta \end{pmatrix}.$$

Note that while the vector  $c^{(j)}$  is a linear function of  $\alpha$ , the matrices  $A^{(j)}$  and the vector  $b$  are constant. We can then rewrite the linear optimization (3.9) as

$$\begin{aligned}
\max : & \quad c' \hat{\xi}_{\cdot, j} \\
\text{s.t.} : & \quad A^{(j)} \hat{\xi}_{\cdot, j} \leq b.
\end{aligned}$$

The linear programming dual to this program is then

$$\begin{aligned}
\min : & \quad (b)' p^{(j)} \\
\text{s.t.} : & \quad (p^{(j)})' A^{(j)} = c \\
& \quad p_S^{(j)} \geq 0, \quad \forall S \subseteq \{1, \dots, k\}.
\end{aligned}$$

Recalling that  $c_S = \sum_{i \in S} \alpha_i$ , the robust linear optimization problem determining the optimal strategy of the decision-maker can now be rewritten:

$$\begin{aligned}
\min : & \quad u \\
\text{s.t.} : & \quad \left( u - \sum_{i \neq j} \alpha_i \text{correct} \hat{\mu}_{h_j}(\blacktriangle(h_i, h_j)) \right) \geq (b)' p^{(j)}, \\
& \quad j = 1, \dots, k \\
& \quad [(p^{(j)})' A^{(j)}]_S = \sum_{i \in S} \alpha_i, \quad \forall S, \quad j = 1, \dots, k \\
& \quad p_S^{(j)} \geq 0, \quad \forall S, \quad j = 1, \dots, k \\
& \quad \sum_i \alpha_i = 1 \\
& \quad \alpha_i \geq 0.
\end{aligned}$$

The variables of optimization are  $\{u, \alpha_i, p_S^{(j)}\}$ . The matrices  $A^{(j)}$  and the vector  $b$  are constants, determined by (3.10). Therefore this is indeed a linear optimization. Thus the proof of parts (i), (ii) and (iii) is complete, as is the proof of the proposition.  $\square$

From a complexity perspective the linear program is exponential in the size of  $\mathcal{F}$  since all subsets are considered. This complexity is not an added feature of our development, as even linear classification over non-separable data becomes combinatorial. Still, in spite of this exponential nature the linear program can consider several approximation schemes such as constraint sampling. Moreover, pruning can be used for the classifiers in  $\mathcal{F}$ ; this is pursued elsewhere.

We have thus derived the optimal policy of the decision-maker for both  $S1$  and  $S2$ . We denote these as  $D_1^*$  and  $D_2^*$ , respectively.

### 3.3 Bounding the Decision-Maker's Error

As defined above, the decision-maker's policy is a mixed strategy – a randomized policy. In the setting of the worst-case analysis which we consider, the decision-maker stands to benefit from the randomization. For example, suppose  $\mathcal{H} = \{h_1, h_2\}$ , and  $\mu(\blacktriangle(h_1, h_2)) = 2\eta$ , where the adversary's power is  $\eta$ . We consider the general optimal strategy for the adversary in the next section. In this case, however, it is clear that the optimal strategy for both the flip-only and the move-and-flip adversary, is to flip half of the 'points', or measure, in  $\blacktriangle(h_1, h_2)$ . Then the decision-maker cannot distinguish between  $h_1$  and  $h_2$ , and the optimal policy is  $\frac{1}{2}h_1 + \frac{1}{2}h_2$ . The expected worst-case error is  $\frac{1}{2}\mu(\blacktriangle(h_1, h_2)) = \eta$ . If not for randomization, the worst-case error would have been  $2\eta$ . Thus there is a concrete benefit to randomization. The next proposition quantifies this benefit (this is similar to Proposition 4.1 from [CBDF<sup>+</sup>99], but has a slightly tighter lower bound<sup>3</sup>), and obtains bounds on the error an adversary with power  $\eta$  can obtain in any possible setup.

**Proposition 8** *In both  $S1$  and  $S2$ , for an adversary with power  $\eta \leq 1/2$ , there is a setup where  $\text{Error}^i \geq (1 - \eta)2\eta$  for  $i = 1, 2$ . On the other hand, we always have  $\text{Error}^i \leq 2\eta$  for  $i = 1, 2$  and if  $\mathcal{F}$  is finite we have  $\text{Error}^i \leq (1 - 1/|\mathcal{F}|)2\eta$  for  $i = 1, 2$ .*

**PROOF.** We need to show that the lower bound can be approached arbitrarily closely in the case of the weaker adversary (flip-only), and the upper bound can never be exceeded by the more powerful adversary (move-and-flip). Let  $\mathcal{X}$  be the unit circle in  $\mathbb{R}^2$ , with  $\mu$  the uniform measure on the disk. If  $\eta = p/q$  is rational, divide the disk into  $q$  equal, numbered wedges, and define  $q$  classifiers, so that classifier  $i$  assigns positive labels to wedges  $\{i, i + 1, \dots, i + p - 1\} \bmod q$ , and negative labels to the remaining  $q - p$  wedges. As in Figure 1 suppose the true classifier is  $h_1$ . The optimal action of the adversary with power  $\eta$  is to flip all positive labels. Now all classifiers are indistinguishable, and thus the decision-maker's optimal strategy is the uniform measure over all  $\{h_i\}$ . The probability of full overlap with  $h_{\text{true}}$

<sup>3</sup>The lower bound of Proposition 4.1 from [CBDF<sup>+</sup>99] translates to  $\text{Error}^i \geq \eta/(2 - \eta)$  which is smaller than the bound of Proposition 8.

is  $1/q$ , the probability of no overlap is  $(q - 2p + 1)/q$ , and of overlap  $r$  for  $0 < r < p$  is  $2r/q$ . Computing the expectation, we have  $\text{Error}^1 = (2p(q - p))/q^2 = (1 - \eta)2\eta$ , as claimed. For  $\eta$  irrational, we can approximate it arbitrarily closely with a rational number. In this case we can approach the lower bound arbitrarily closely.

Next we show that even the more powerful move-and-flip adversary can never exceed the upper bound. Observe that if the power of the adversary is  $\eta$ , then for any two classifiers  $h_i$  and  $h_j$ , we must have  $\mu(\blacktriangle(h_i, h_j)) \leq 2\eta$ . Then, if the decision-maker uses the possibly sub-optimal strategy of choosing  $\alpha = (1/n, 1/n, \dots, 1/n)$  (where  $n = |\mathcal{F}|$ ), then since by definition  $\blacktriangle(h, h) = \emptyset$  for all  $h$ , from expression (3.5) above, it follows that the expected error will never exceed  $(1 - 1/n)2\eta$ .

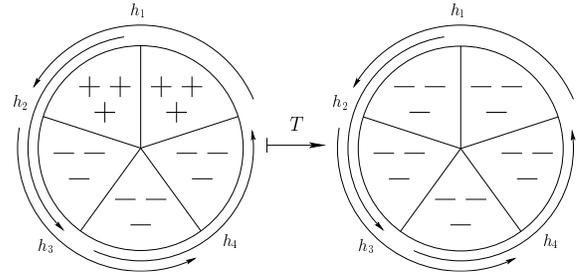


Figure 1: Here we have  $\eta = 2/5$ , so  $p = 2$  and  $q = 5$ . The figure on the left shows the correct labels. The adversary flips all  $+$ -labels to  $-$ . In the figure on the right, all classifiers are indistinguishable to the decision-maker. The decision-maker, therefore, outputs a randomized strategy that is uniform over all  $n$  classifiers (here  $n = 5$ ).

### 3.4 The Adversary

First we consider  $S1$  and the flip-only adversary. From Proposition 5, the optimal strategy of the decision-maker is specified by the subset of ambiguous classifiers,  $\mathcal{F}$ . We call this  $\alpha(\mathcal{F})$ . Therefore the true error is also a function of  $\mathcal{F}$ . By an abuse of notation, we can denote this by  $\mathcal{E}(\mathcal{F}) \triangleq \sum_{h \neq h_{\text{true}}} \alpha_h(\mathcal{F}) \mu(\blacktriangle(h, h_{\text{true}}))$ . Then the optimal strategy of the adversary is to create an ambiguous set  $\mathcal{F}$  with as large an error as possible. Given any legal strategy  $T$  of the adversary, we denote by  $\mathcal{F}_T$  the resulting set of ambiguous classifiers. Therefore we have:

**Proposition 9** *In  $S1$ , the adversary's optimal strategy is to maximize  $\mathcal{E}(\mathcal{F})$ :*

$$T_1^* = \arg \max_{\substack{T: |\mathcal{F}_T| \leq \eta \\ T \text{ flip-only}}} \mathcal{E}(\mathcal{F}_T). \quad (3.11)$$

The max here is attained since there are only finitely many different sets  $\mathcal{F}$ . If there are more than one (as in general there will be) maps  $T$  corresponding to the optimal  $\mathcal{F}$ , we arbitrarily choose one. Therefore  $T_1^*$  is well-defined, and is the optimal strategy for the adversary in  $S1$ , and the proposition follows.

Next we consider  $S2$ , and the case of the move-and-flip adversary. From Proposition 6, the decision-maker's optimal

action is given by an LP that is a function of the ambiguity set  $\mathcal{F}$ , and the values  $\{\hat{\mu}_{h'}^{\text{correct}}(\blacktriangle(h'', h'))\}$  for  $h', h'' \in \mathcal{F}$ . As above, we denote this optimal solution by  $\beta \triangleq \beta^{\text{correct}}(\hat{\mu}_{h'}^{\text{correct}}(\blacktriangle(h'', h')))$ , and the associated true generalization error is then  $\mathcal{E}_\mu(h_{\text{true}}; \beta)$ . For a given triple  $(\lambda, \mu_+, \mu_-)$ , and power  $\eta$  of the adversary, not all ambiguity sets  $\mathcal{F}$ , and values for  $\{\hat{\mu}_{h'}^{\text{correct}}(\blacktriangle(h'', h'))\}$  are attainable. We define the set of such attainable values.

**Definition 10** Let  $\mathcal{A}$  be the set of values  $\{\hat{\mu}_{h'}^{\text{correct}}(\blacktriangle(h'', h'))\}$ , for  $h', h'' \in \mathcal{F}$  for some  $\mathcal{F}$ , such that there exists a triple  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$  that meets three conditions:

- (a)  $\mathcal{F}$  must be the ambiguity set corresponding to  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$ , as in (3.4).
- (b) The triple must satisfy

$$\begin{aligned} \hat{\mu}_h^{\text{wrong}}(\blacktriangle(h', h'')) &= \hat{\lambda}\hat{\mu}_+(\blacktriangle(h', h'') \cap h(-)) + \\ &\quad (1 - \hat{\lambda})\hat{\mu}_-(\blacktriangle(h', h'') \cap h(+)) \\ \hat{\mu}_h^{\text{correct}}(\blacktriangle(h', h'')) &= \hat{\mu}(\blacktriangle(h', h'')) - \hat{\mu}_h^{\text{wrong}}(\blacktriangle(h', h'')). \end{aligned}$$

- (c) We must have  $\|(\lambda, \mu_+, \mu_-) - (\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)\|_{TV} \leq \eta$ .

**Lemma 11 (a)** The set  $\mathcal{A}$  is a finite union of polyhedral sets, and it is compact.

- (b) The function  $\mathcal{E}_\mu(h_{\text{true}}; \beta(\hat{\mu}_h^{\text{correct}}(\blacktriangle(h', h''))))$  is piecewise continuous, with finitely many discontinuities.

We defer the proof of this lemma to [CM08]. The next proposition gives the optimal policy of the adversary for  $S_2$ :

**Proposition 12** The adversary's optimal strategy  $T_2^*$  maps  $(\lambda, \mu_+, \mu_-)$  to a triple  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$  that matches the values  $\hat{\mu}_h^{\text{correct}}(\blacktriangle(h', h''))$  from the solution to the (nonlinear) program:

$$\begin{aligned} \max : & \mathcal{E}_\mu(h_{\text{true}}; \beta(\hat{\mu}_h^{\text{correct}}(\blacktriangle(h', h'')))) \\ \text{s.t. :} & \{\hat{\mu}_h^{\text{correct}}(\blacktriangle(h', h''))\} \in \mathcal{A}. \end{aligned} \quad (3.12)$$

**PROOF.** By Lemma 11,  $\mathcal{A}$  is compact, and  $\mathcal{E}_\mu(h_{\text{true}}; \beta)$  is piecewise continuous. Therefore, the optimal value is attained for some ambiguity set  $\mathcal{F}$ , and corresponding element  $\{\hat{\mu}_h^{\text{correct}}(\blacktriangle(h', h''))\}$  of  $\mathcal{A}$ . By the definition of  $\mathcal{A}$ , there exists at least one such map  $T_2^*$ , with  $\|T_2^*\| \leq \eta$ , that attains this value.  $\square$

Thus the optimal policies for the decision-maker and adversary are each given by respective optimization problems. In summary, we have:

**Theorem 13** The pair of strategies  $(D_i^*, T_i^*)$  ( $i = 1, 2$ ) for the decision-maker and the adversary, gives optimal solutions to  $S_1$  and  $S_2$ , respectively.

## 4 Error and the Power of the Adversary

While we treat the noise as generated by an adversary, we may also consider it to be a design parameter chosen according to how we care to trade off optimality for robustness. Indeed, upon seeing some realization  $(\hat{\lambda}, \hat{\mu}_+, \hat{\mu}_-)$ , the decision-maker may have partial knowledge of the level  $\eta$  of noise. Equally, the decision-maker may specifically be interested in choosing a solution appropriate for some particular level  $\tilde{\eta}$  of noise. For any fixed level  $\tilde{\eta}$ , from the results in Section 3, the decision-maker obtains the resulting optimal policy. When  $\tilde{\eta} = 0$ , the optimal strategy of the decision-maker is to deterministically choose the single classifier that minimizes the empirical error. If indeed  $\eta = 0$ , then this is the optimal strategy. As  $\tilde{\eta}$  grows, the optimal strategy of the decision-maker becomes increasingly random, and in the limit as  $\tilde{\eta} \rightarrow 100\%$ , the optimal policy approaches the uniform distribution over all classifiers.

For a fixed measure  $\mu$ ,  $\mathcal{H}$ , and  $h_{\text{true}} \in \mathcal{H}$  we consider the error as a function of  $\eta$ . Graphing this function allows the decision-maker, in the scenario described above, to consider the tradeoff of robustness and optimality, and thus may choose the desirable design parameter  $\tilde{\eta}$ , with respect to which the optimal mixed strategy is obtained. In addition, this graph provides other information that is of interest. The graph of the error is not continuous. Rather, it is piecewise continuous (not necessarily linear), with certain break points. The location of these break points is important, and it is a function of the structure of  $\mathcal{H}$ . A particular solution  $\alpha$  of the decision-maker might be optimal for any  $\tilde{\eta}$  in some interval  $[\eta_1, \eta_2)$ , but not optimal for  $\tilde{\eta} \geq \eta_2$ .

We consider the example from the end of Section 2.3 where  $h_1$  is the true classifier. There, the move-and-flip adversary is strictly more powerful than the flip-only adversary when  $\eta > 5$ , and hence the setups  $S_1$  and  $S_2$  are not equivalent. The graphs in Figure 2 show  $\text{Error}^i(\mu, h_{\text{true}}, \eta, T_i^*, D_i^*)$  for fixed  $\mu$  and  $h_{\text{true}}$ , and varying values of  $\eta$ . In the left side of Figure 2 we have the superimposed graphs for this example, for  $S_1$  and  $S_2$  for  $0 \leq \eta \leq 11$ . In the right side of Figure 2 we show the full graph of the true error  $\text{Error}^2$ , for  $0 \leq \eta \leq 100$ .

The graph for  $S_2$  is obtained by using the results of Propositions 6 and 12. The optimal policy of the move-and-flip adversary differs for the three regions  $0 \leq \eta < 5$ ,  $5 \leq \eta \leq 10$ ,  $10 \leq \eta \leq 100$ . In the first region, the adversary is powerless regardless of his action. In the second region, the optimal strategy is to flip  $\eta\%$  of the labels in  $\blacktriangle(h_1, h_2)$ . For  $10 \leq \eta \leq 100$ , the adversary's optimal strategy is to flip all the points in  $\blacktriangle(h_1, h_2)$ , and also move and label '−' a  $(\eta - 10)$  fraction of the mass into  $\blacktriangle(h_1, h_2)$ , so that  $\hat{\mu}(\blacktriangle(h_1, h_2)) = \eta$ .

The decision-maker's policy, as given by Proposition 6, protects the decision-maker against the worst possible (consistent) triple  $(\tilde{\lambda}, \tilde{\mu}_+, \tilde{\mu}_-)$ . Solving the robust LP from the proposition reveals both the true error, and the worst-case error. Both of these quantities may be of interest. In [CM08] we show, for this example, both the true error, and the worst-case error, for all values of  $\eta$ . The true error exhibits numerous interesting properties. For instance, as shown in the figure, the true error is *not monotonic* in the power of the

adversary (the worst-case error over measures and classifiers is, of course, monotonic). This is a direct consequence of Proposition 6. In [CM08] we pay particular attention to this, and other properties of the graph. Also, we give the details of the computations.

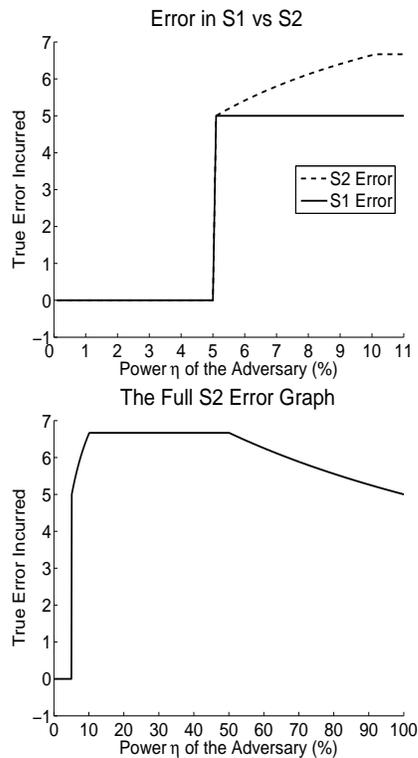


Figure 2: The graph above shows the error incurred in  $G1$  on the same axes as the error incurred in  $S2$ , for  $0 \leq \eta \leq 11$ . As soon as  $\eta > 5$ , we see that the move-and-flip adversary is more powerful. Note that  $\text{Error}^2$  grows sublinearly for  $\eta \geq 5$ . In the graph on the right we show the error graph for the more powerful adversary for  $0 \leq \eta \leq 100$ . The true error is not monotonic, as it decreases (non-linearly) for  $\eta \geq 50\%$ .

## 5 Discussion

This work takes a learning in the information theoretic limit view of learning with adversarial disturbance. Our main contribution is the introduction of an optimization-theoretic algorithmic framework for finding a classifier in the presence of such disturbance. We characterized the optimal policy of the decision-maker (as a function of the data) in terms of a tractable and easily solved optimization problem. This is a first step in developing the theory for a range of setups. For example, the Bayesian setup may be of interest. Here, the decision-maker has a prior over the possible classifiers, and instead of minimizing generalization error with respect to the worst-case consistent classifier and (in  $S2$ ) underlying measure  $\tilde{\mu}$ , he considers minimizing expected (under the Bayesian posterior) error. Extending this algorithmic approach to the game-theoretic setup, where the decision-maker plays against a rational adversary, is also of interest, and allows the possibility of more complex information

structures.

Considering the noise level as a design parameter and viewing the resulting error as a function of it yielded surprising results that show how counterintuitive the mini-max formulation of learning with adversarial noise could be. We showed for a simple example that while the worst-case error is monotone in the power of the adversary, the actual error (which depends on the particular underlying true probability measure) may not be monotone in the power of the adversary! This is because even though the adversary is more powerful, the decision maker is also better prepared.

There are three natural extensions to our work that we did not pursue here mostly due to space limits. First, while we considered the proper learning setup, the non-proper setup (as in [KSS92]) seems to naturally follow our framework. Second, the case of infinite set of classifier  $\mathcal{H}$  could be resolved by eliminating classifiers that are “close” according to the observed measure. This is particularly useful for the flip-only setup where the adversary cannot make two classifiers substantially different. Finally, while we do not consider sample complexity, such results should not be too difficult to derive by imitating the arguments in [CBDF<sup>+</sup>99].

## References

- [ACB98] P. Auer and N. Cesa-Bianchi. On-line learning with malicious noise and the closure algorithm. *Annals of AI and Mathematics*, 23(1):83–99, 1998.
- [BEK02] N. H. Bshouty, N. Eiron, and E. Kushilevitz. PAC learning with nasty noise. *Theoretical Computer Science*, 288(2):255–275, 2002.
- [BTN99] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, August 1999.
- [CBDF<sup>+</sup>99] N. Cesa-Bianchi, E. Dichterman, P. Fischer, E. Shamir, and H. Ulrich Simon. Sample-efficient strategies for learning in the presence of noise. *Journal of the ACM*, 46(5):684–719, 1999.
- [CM08] C. Caramanis and S. Mannor. Beyond PAC: A robust optimization approach for learning in the presence of noise: Online appendix. Available from <http://users.ece.utexas.edu/~cmcaram/pubs/RobustLearningOnlineApp.pdf>, 2008.
- [KL93] M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.
- [KSS92] Michael J. Kearns, Robert E. Schapire, and Linda Sellie. Toward efficient agnostic learning. In *Computational Learning Theory*, pages 341–352, 1992.
- [Lai88] P. D. Laird. *Learning from good and bad data*. Kluwer Academic Publishers, Norwell, MA, USA, 1988.
- [Ser03] R. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003.

---

# On the Margin Explanation of Boosting Algorithms

---

Liwei Wang<sup>\*1</sup>, Masashi Sugiyama<sup>2</sup>, Cheng Yang<sup>1</sup>, Zhi-Hua Zhou<sup>3</sup>, and Jufu Feng<sup>1</sup>

<sup>1</sup> Key Laboratory of Machine Perception, MOE, School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, P.R.China. {wanglw,yangch,fjf}@cis.pku.edu.cn

<sup>2</sup> Department of Computer Science, Tokyo Institute of Technology, 2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan. sugi@cs.titech.ac.jp

<sup>3</sup> National Key Laboratory for Novel Software Technology, Nanjing University Nanjing 210093, P.R. China. zhoush@nju.edu.cn

## Abstract

Much attention has been paid to the theoretical explanation of the empirical success of AdaBoost. The most influential work is the margin theory, which is essentially an upper bound for the generalization error of any voting classifier in terms of the margin distribution over the training data. However, Breiman raised important questions about the margin explanation by developing a boosting algorithm *arc-gv* that provably generates a larger minimum margin than AdaBoost. He also gave a sharper bound in terms of the minimum margin, and argued that the minimum margin governs the generalization. In experiments however, *arc-gv* usually performs worse than AdaBoost, putting the margin explanation into serious doubts. In this paper, we try to give a complete answer to Breiman's critique by proving a bound in terms of a new margin measure called Equilibrium margin (Emargin). The Emargin bound is uniformly sharper than Breiman's minimum margin bound. This result suggests that the minimum margin is not crucial for the generalization error. We also show that a large Emargin implies good generalization. Experimental results on benchmark datasets demonstrate that AdaBoost usually has a larger Emargin and a smaller test error than *arc-gv*, which agrees well with our theory.

## 1 Introduction

The AdaBoost algorithm [FS96, FS97] has achieved great success in the past ten years. It has demonstrated excellent experimental performance both on benchmark datasets and real applications [BK99, Die00, VJ01]. It is observed in experiments that the test error of a combined voting classifier usually keeps decreasing as its size becomes very large and even after the training error is zero [Bre98, Qui96]. This fact, on the first sight, obviously violates Occam's razor.

---

<sup>\*</sup>This work was supported by NSFC(60775005, 60635030, 60721002) and Global COE Program of Tokyo Institute of Technology.

Schapire et al. [SFBL98] tried to explain this phenomenon in terms of the margins of the training examples. Roughly speaking, the margin of an example with respect to a classifier is a measure of the confidence of the classification result. Schapire et al. [SFBL98] proved an upper bound for the generalization error of a voting classifier that does not depend on how many classifiers were combined, but only on the margin distribution over the training set, the number of the training examples and the size (the VC dimension for example) of the set of base classifiers. They also demonstrate that AdaBoost has the ability to produce a good margin distribution. This theory indicates that producing a good margin distribution is the key to the success of AdaBoost and explains well the surprising phenomenon observed in experiments.

Soon after that however, Breiman [Bre99] cast serious doubts on this margin explanation. He developed a boosting-type algorithm called *arc-gv*, which provably generates a larger minimum margin than AdaBoost<sup>1</sup> (Minimum margin is the smallest margin over all the training examples, see Section 2 for the formal definition). Then he gave an upper bound for the generalization error of a voting classifier in terms of the minimum margin, as well as the number of training examples and the size of the set of base classifiers. This bound is sharper than the bound based on the margin distribution given by Schapire et al.

Breiman argued that if the bound of Schapire et al. implied that the margin distribution is the key to the generalization error, his bound implied more strongly that the minimum margin is the key to the generalization error, and the *arc-gv* algorithm would achieve the best performance among all boosting-type algorithms. In experiments, even though *arc-gv* always produces larger minimum margins than AdaBoost, its test error is consistently higher. Breiman also investigated the margin distributions generated by AdaBoost and *arc-gv*, and found that *arc-gv* actually produced uniformly better margin distributions than AdaBoost. Thus he concluded that neither the minimum margin nor the margin distribution determined the generalization error and a new theoretical explanation is needed.

---

<sup>1</sup>Actually, the minimum margin of *arc-gv* converges to the largest possible value among all voting classifiers.

Breiman’s argument seems convincing and put the margin explanation into serious doubts. Recently however, Reyzin and Schapire [RS06] gained important discovery after a careful study on Breiman’s arc-gv algorithm. Note first that the bounds of both Breiman and Schapire et al. state that the generalization error also depends on the complexity of the set of base classifiers as well as the minimum margin or the margin distribution. To investigate how the margin affects the generalization error, one has to keep the complexity of the base classifiers fixed. In Breiman’s experiments, he tried to control this by always using CART trees [BFOS84] of a fixed number of leaves as the base classifier. Reyzin and Schapire re-conducted Breiman’s experiments and found that the trees produced by arc-gv were much deeper than those produced by AdaBoost. Since deeper trees are more complex even though the number of leaves is the same, arc-gv uses base classifiers of higher complexity than AdaBoost in Breiman’s experiments. Thus it was not a fair comparison.

In order to study the margin explanation in a fair manner, a more controlled setting is needed. Reyzin and Schapire then compared arc-gv and AdaBoost by using the decision stump, whose complexity is fixed, as the base classifier. Experiments showed that arc-gv produced larger minimum margins yet still a higher error rate. But this time, the margin distribution generated by arc-gv is not as “good” as that AdaBoost generated (see Fig.7 in [RS06]). So they argued that according to the Schapire et al. bound in terms of the margin distribution, the empirical observation, i.e., the inferior performance of arc-gv, could be explained.

From a more critical point of view however, Breiman’s doubt has not been fully answered by the above results. First of all, Breiman backed up his argument with a sharper bound in terms of the minimum margin. In Reyzin and Schapire’s experiment with the decision stumps, arc-gv still produced larger minimum margin and had worse performance. Even though AdaBoost generates a “better” margin distribution than arc-gv, it would not disprove Breiman’s critique unless we could show a bound in terms of the margin distribution and is uniformly sharper than Breiman’s minimum margin bound. Another problem is how to measure the “goodness” of a margin distribution. The statement that AdaBoost generates “better” margin distributions than arc-gv is vague. Reyzin and Schapire used the average margin as a measure to compare margin distributions produced by AdaBoost and arc-gv. But the average margin does not explicitly appear in the bound of Schapire et al. Thus a larger average margin does not necessarily imply a smaller generalization error in theory.

In this paper, we try to give a complete answer to Breiman’s doubt by solving the two problems mentioned above. We first propose a novel upper bound for the generalization error of voting classifiers. This bound is uniformly sharper than Breiman’s bound. The key factor in this bound is a new margin notion which we refer to as the Equilibrium margin (Emargin). The Emargin

can be viewed as a measure of how good a margin distribution is. In fact, the Emargin depends, in a complicated way, on the margin distribution, and has little relation to the minimum margin. Experimental results show that AdaBoost usually produces a larger Emargin than arc-gv when the complexity of the base classifier is well controlled. Our results thus explain the inferior performance of arc-gv and give Breiman’s doubt a negative answer.

The rest of this paper is organized as follows: In Section 2 we briefly describe the margin theory of Schapire et al. and Breiman’s argument. Our main results are given in Section 3. We provide further explanation of the main bound in Section 4. All the proofs can be found in Section 5. We provide experimental justification in Section 6 and conclude in Section 7.

## 2 Background and Related Work

In this section we briefly review the existing margin bounds and the two boosting algorithms.

Consider binary classification problems. Examples are drawn independently according to an underlying distribution  $D$  over  $X \times \{-1, +1\}$ , where  $X$  is an instance space. Let  $H$  denote the space from which the base hypotheses are chosen. A base hypothesis  $h \in H$  is a mapping from  $X$  to  $\{-1, +1\}$ . A voting classifier  $f(x)$  is of the form

$$f(x) = \sum \alpha_i h_i(x),$$

where

$$\sum \alpha_i = 1, \quad \alpha_i \geq 0.$$

An error occurs on an example  $(x, y)$  if and only if

$$yf(x) \leq 0.$$

We use  $P_D(A(x, y))$  to denote the probability of the event  $A$  when an example  $(x, y)$  is chosen randomly according to the distribution  $D$ . Therefore,  $P_D(yf(x) \leq 0)$  is the generalization error which we want to bound. We also use  $P_S(A(x, y))$  to denote the probability with respect to choosing an example  $(x, y)$  uniformly at random from the training set  $S$ .

For an example  $(x, y)$ , the value of  $yf(x)$  reflects the confidence of the prediction. Since each base classifier outputs  $-1$  or  $+1$ , one has

$$yf(x) = \sum_{i:y=h_i(x)} \alpha_i - \sum_{i:y \neq h_i(x)} \alpha_i.$$

Hence  $(yf(x))$  is the difference between the weights assigned to those base classifiers that correctly classify  $(x, y)$  and the weights assigned to those that misclassify the example.  $yf(x)$  is called the *margin* for  $(x, y)$  with respect to  $f$ . If we consider the margins over the whole set of training examples, we can regard  $P_S(yf(x) \leq \theta)$  as a distribution over  $\theta$  ( $-1 \leq \theta \leq 1$ ), since  $P_S(yf(x) \leq \theta)$  is the fraction of training examples whose margin is at most  $\theta$ . This distribution is referred to as the *margin distribution*. The *minimum margin* of  $f$ , which is the smallest margin over the training examples, then can

**Input:**  $S = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$   
 where  $x_i \in X, y_i \in \{-1, 1\}$ .

**Initialization:**  $D_1(i) = 1/n$ .

**for**  $t = 1$  **to**  $T$  **do**

1. Train base learner using distribution  $D_t$ .
2. Get base classifier  $h_t : X \rightarrow \{-1, 1\}$ .
3. Choose  $\alpha_t$ .
4. Update:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t},$$

where  $Z_t$  is a normalization factor chosen so that  $D_{t+1}$  will be a distribution.

**end**

**Output:** The final Classifier

$$H(x) = \text{sgn} \left( \sum_{t=1}^T \alpha_t h_t(x) \right).$$

**Algorithm 1:** A unified description of AdaBoost and arc-gv.

be equivalently represented by the maximum value of  $\theta$  such that  $P_S(yf(x) \leq \theta) = 0$ .

A unified description of AdaBoost and arc-gv is shown in Algorithm 1. The only difference of the two algorithms is the choice of  $\alpha_t$ . AdaBoost sets  $\alpha_t$  as

$$\alpha_t = \frac{1}{2} \log \frac{1 + \gamma_t}{1 - \gamma_t},$$

where  $\gamma_t$  is the *edge* of the base classifier  $h_t$ , defined as:

$$\gamma_t = \sum_{i=1}^n D_t(i) y_i h_t(x_i).$$

The edge  $\gamma_t$  is an affine transformation of the error rate of  $h_t$  with respect to the distribution  $D_t$ .

Arc-gv chooses  $\alpha_t$  in a different way. It takes into consideration of the minimum margin of the composite classifier up to the current round. Denote by  $\rho_t$  the minimum margin of the voting classifier of round  $t - 1$ , that is,

$$\rho_t = \min_i \left( y_i \frac{\sum_{s=1}^{t-1} \alpha_s h_s(x_i)}{\sum_{s=1}^{t-1} \alpha_s} \right).$$

Let

$$\beta_t = \frac{1}{2} \log \frac{1 + \gamma_t}{1 - \gamma_t} - \frac{1}{2} \log \frac{1 + \rho_t}{1 - \rho_t}.$$

Arc-gv sets  $\alpha_t$  as [Bre99]:

$$\alpha_t = \begin{cases} 1 & : \beta_t > 1, \\ \beta_t & : 0 \leq \beta_t \leq 1, \\ 0 & : \beta_t < 0. \end{cases}$$

The first margin explanation of the AdaBoost algorithm [SFBL98] is to upper bound the generalization error of voting classifiers in terms of the margin distribution, the number of training examples and the complexity of the set from which the base classifiers are

chosen. The theory contains two bounds: one applies to the case that the base classifier set  $H$  is finite, and the other applies to the general case that  $H$  has a finite VC dimension.

**Theorem 1** [SFBL98] For any  $\delta > 0$ , with probability at least  $1 - \delta$  over the random choice of the training set  $S$  of  $n$  examples, every voting classifier  $f$  satisfies the following bounds:

$$P_D(yf(x) \leq 0) \leq \inf_{\theta \in (0,1)} \left[ P_S(yf(x) \leq \theta) + O \left( \frac{1}{\sqrt{n}} \left( \frac{\log n \log |H|}{\theta^2} + \log \frac{1}{\delta} \right)^{1/2} \right) \right],$$

if  $|H| < \infty$ . And

$$P_D(yf(x) \leq 0) \leq \inf_{\theta \in (0,1)} \left[ P_S(yf(x) \leq \theta) + O \left( \frac{1}{\sqrt{n}} \left( \frac{d \log^2(n/d)}{\theta^2} + \log \frac{1}{\delta} \right)^{1/2} \right) \right],$$

where  $d$  is the VC dimension of  $H$ .

The theorem states that if the voting classifier generates a good margin distribution, that is, most training examples have large margins so that  $P_S(yf(x) \leq \theta)$  is small for not too small  $\theta$ , then the generalization error is also small. In [SFBL98] it has also been shown that for the AdaBoost algorithm,  $P_S(yf(x) \leq \theta)$  decreases to zero exponentially fast with respect to the number of boosting iterations if  $\theta$  is not too large. These results imply that the excellent performance of AdaBoost is due to its good margin distribution.

Breiman's doubts on the margin explanation came from the arc-gv algorithm. It can be shown that the minimum margin generated by arc-gv converges to the largest possible value among all voting classifiers. In practice, arc-gv has larger minimum margins than AdaBoost in most cases for a finite number of boosting iterations. Breiman also proved an upper bound for the generalization error of voting classifiers. This bound depends only on the minimum margin, not on the entire margin distribution.

**Theorem 2** [Bre99] Let  $\theta_0$  be the minimum margin defined as

$$\theta_0 = \min \{ yf(x) : (x, y) \in S \}, \quad (1)$$

where  $S$  is the training set. If

$$\begin{aligned} |H| &< \infty, \\ \theta_0 &> 4 \sqrt{\frac{2}{|H|}}, \\ R &= \frac{32 \log(2|H|)}{n\theta_0^2} \leq 2n, \end{aligned}$$

then for any  $\delta > 0$ , with probability at least  $1 - \delta$  over the random choice of the training set  $S$  of  $n$  examples, every voting classifier  $f$  satisfies the following bounds:

$$P_D(yf(x) \leq 0) \leq R \left( \log(2n) + \log \frac{1}{R} + 1 \right) + \frac{1}{n} \log \left( \frac{|H|}{\delta} \right). \quad (2)$$

Breiman pointed out that his bound is sharper than the margin distribution bound of Schapire et al. If  $\theta$  in Theorem 1 is taken to be the minimum margin  $\theta_0$ , the bound in Theorem 2 is about the square of the bound in terms of the margin distribution, since the bound in Theorem 2 is  $O(\log n/n)$  and the bound in Theorem 1 is  $O(\sqrt{\log n/n})$ . Breiman then argued that compared to the margin distribution explanation, his bound implied more strongly that the minimum margin governs the generalization error. However, arc-gv performs almost consistently worse than AdaBoost in experiments<sup>2</sup>. These empirical results contradict what the margin theory predicts and therefore put the margin explanation into serious doubts.

A lot of efforts have been made on providing better explanation of the boosting algorithms in recent years [MBG02, KP02, KP05, AKLL02]. Koltchinskii and Panchanko [KP02, KP05] proved a number of bounds in terms of the margin distribution which are sharper than Theorem 1. However, it is difficult to compare the minimum margin bound to these bounds since they contain unspecified constants. Nevertheless, these results imply that the margin distribution might be more important than the minimum margin for the generalization error of voting classifiers.

### 3 Main Results

In this section we propose upper bounds in terms of the Emargin. The bound is uniformly sharper than Breiman's minimum margin bound.

First let us introduce some notions. Consider the Bernoulli relative entropy function  $D(q||p)$  defined as

$$D(q||p) = q \log \frac{q}{p} + (1 - q) \log \frac{1 - q}{1 - p}, \quad 0 \leq p, q \leq 1.$$

For a fixed  $q$ ,  $D(q||p)$  is a monotone increasing function of  $p$  for  $q \leq p \leq 1$ . It is easy to check that

$$D(q||p) = 0 \quad \text{when } p = q,$$

and

$$D(q||p) \rightarrow \infty \quad \text{as } p \rightarrow 1.$$

Thus one can define the inverse function of  $D(q||p)$  for fixed  $q$  as  $D^{-1}(q, u)$ , such that

$$D(q||D^{-1}(q, u)) = u \quad \text{for all } u \geq 0 \text{ and } D^{-1}(q, u) \geq q.$$

See also [Lan05].

<sup>2</sup>Actually, the inferior performance has also been observed when using other voting classifiers that maximize the minimum margin (see also [GS98, RW02]).

The next theorem is our main result: the Emargin bound. Here we consider the case that the base classifier set  $H$  is finite. For the case that  $H$  is infinite but has a finite VC dimension, the bound is more complicated and will be given in Theorem 8. All the proofs can be found in Section 5.

**Theorem 3** *If  $|H| < \infty$ , then for any  $\delta > 0$ , with probability at least  $1 - \delta$  over the random choice of the training set  $S$  of  $n$  examples, every voting classifier  $f$  satisfies the following bound:*

$$P_D(yf(x) \leq 0) \leq \frac{\log |H|}{n} + \inf_{q \in \{0, \frac{1}{n}, \frac{2}{n}, \dots, 1\}} D^{-1} \left( q, u \left[ \hat{\theta}(q) \right] \right), \quad (3)$$

where

$$u \left[ \hat{\theta}(q) \right] = \frac{1}{n} \left( \frac{8}{\hat{\theta}^2(q)} \log \left( \frac{2n^2}{\log |H|} \right) \log |H| + \log |H| + \log \frac{n}{\delta} \right),$$

and  $\hat{\theta}(q)$  is given by

$$\hat{\theta}(q) = \sup \left\{ \theta \in \left( \sqrt{8/|H|}, 1 \right] : P_S(yf(x) \leq \theta) \leq q \right\}. \quad (4)$$

Clearly the key factors in this bound are the optimal  $q$  and the corresponding  $\hat{\theta}(q)$ .

**Definition 4** *Let  $q^*$  be the optimal  $q$  in Eq.(3), and denote*

$$\theta^* = \hat{\theta}(q^*).$$

*We call  $\theta^*$  the Equilibrium margin (**Emargin**).*

The name *equilibrium* is due to the following fact.

**Proposition 5**  *$q^*$  is the empirical error at the Emargin  $\theta^*$ .*

$$P_S(yf(x) < \theta^*) = q^*. \quad (5)$$

With Definition 4, the Emargin bound (3) can be simply written as

$$P_D(yf(x) \leq 0) \leq \frac{\log |H|}{n} + D^{-1} \left( q^*, u(\theta^*) \right). \quad (6)$$

Theorem 3 then states that the generalization error of a voting classifier depends on its Emargin and the empirical error at the Emargin.

Our Emargin bound has a similar flavor to Theorem 1. Note that the Emargin depends, in a complicated way, on the whole margin distribution. Roughly, if most training examples have large margins, then  $\theta^*$  is large and  $q^*$  is small. The minimum margin is only a special case of the Emargin. From Eq.(4) one can see that  $\hat{\theta}(0)$  is the minimum margin. Hence the Emargin is

equal to the minimum margin if and only if the optimal  $q^*$  is zero.

We next compare our Emargin bound to Breiman's minimum margin bound. We show that the Emargin bound is uniformly sharper than the minimum margin bound.

**Theorem 6** *The bound given in Theorem 3 is uniformly sharper than the minimum margin bound in Theorem 2. That is*

$$\begin{aligned} & \frac{\log |H|}{n} + D^{-1}(q^*, u(\theta^*)) \\ & \leq R \left( \log(2n) + \log \frac{1}{R} + 1 \right) + \frac{1}{n} \log \frac{|H|}{\delta}, \end{aligned}$$

where

$$R = \frac{32 \log(2|H|)}{n\theta_0^2} \leq 2n.$$

According to this theorem, the minimum margin is not crucial for the generalization error, i.e., a larger minimum margin does not necessarily imply a smaller test error. Thus arc-gv does not necessarily have better performance than AdaBoost. Our new bound implies that it is the Emargin  $\theta^*$  and the empirical error  $q^*$  at  $\theta^*$  that govern the performance of the classifier. The following theorem describes how the Emargin  $\theta^*$  and the Emargin error  $q^*$  affect the generalization ability. It states that a larger Emargin and a smaller Emargin error result in a lower generalization error.

**Theorem 7** *Let  $f_1, f_2$  be two voting classifiers. Denote by  $\theta_1, \theta_2$  the Emargin and by  $q_1, q_2$  the empirical error at  $\theta_1, \theta_2$  of  $f_1, f_2$  respectively. That is*

$$q_i = P_S(yf_i(x) < \theta_i), \quad i = 1, 2.$$

Also denote by  $B_1, B_2$  the Emargin upper bound of the generalization error of  $f_1, f_2$  (i.e. the right-hand side of Eq.(3)). Then

$$B_1 \leq B_2,$$

if

$$\theta_1 \geq \theta_2 \quad \text{and} \quad q_1 \leq q_2.$$

Theorem 7 suggests that the Emargin and the Emargin error can be used as measures of the goodness of a margin distribution. A large Emargin and a small Emargin error indicate a good margin distribution. Experimental results in Section 6 show that AdaBoost usually has larger Emargins and smaller Emargin errors than arc-gv.

The last theorem of this section is the Emargin bound for the case that the set of base classifiers has a finite VC dimension.

**Theorem 8** *Suppose the set of base classifiers  $H$  has VC dimension  $d$ . Then for any  $\delta > 0$ , with probability at least  $1 - \delta$  over the random choice of the training set*

*$S$  of  $n$  examples, every voting classifier  $f$  satisfies the following bounds:*

$$\begin{aligned} & P_D(yf(x) \leq 0) \\ & \leq \frac{d^2 + 1}{n} + \inf_{q \in \{0, \frac{1}{n}, \frac{2}{n}, \dots, 1\}} \frac{n}{n-1} \cdot D^{-1}(q, u[\hat{\theta}(q)]), \end{aligned}$$

where

$$\begin{aligned} u[\hat{\theta}(q)] = \frac{1}{n} & \left( \frac{16d}{\hat{\theta}^2(q)} \log \frac{n}{d} \log \frac{en^2}{d} \right. \\ & \left. + 3 \log \left( \frac{16}{\hat{\theta}^2(q)} \log \frac{n}{d} + 1 \right) + \log \frac{2n}{\delta} \right), \end{aligned}$$

and  $\hat{\theta}(q)$  is

$$\hat{\theta}(q) = \sup \left\{ \theta \in (0, 1] : P_S(yf(x) \leq \theta) \leq q \right\}. \quad (7)$$

## 4 Explanation of the Emargin Bound

In Theorem 3, we adopt the partial inverse of the relative entropy to upper bound the generalization error. The key term in the Emargin bound is  $\inf_q D^{-1}(q, u[\hat{\theta}(q)])$ . To better understand the bound, we make use of three different upper bounds of  $\inf_q D^{-1}(q, u)$  to obtain simpler forms of the Emargin bound. We list in the following lemma the upper bounds of  $\inf_q D^{-1}(q, u[\hat{\theta}(q)])$ .

**Lemma 9** *The following bounds holds.*

1.

$$\begin{aligned} \inf_q D^{-1}(q, u[\hat{\theta}(q)]) & \leq D^{-1}(0, u[\hat{\theta}(0)]) \\ & \leq u[\hat{\theta}(0)]. \end{aligned}$$

2.

$$\inf_q D^{-1}(q, u[\hat{\theta}(q)]) \leq \inf_q \left( q + \left( \frac{u[\hat{\theta}(q)]}{2} \right)^{1/2} \right).$$

3.

$$\begin{aligned} \inf_q D^{-1}(q, u[\hat{\theta}(q)]) & \leq \inf_{q \leq C u[\hat{\theta}(q)]} D^{-1}(q, u[\hat{\theta}(q)]) \\ & \leq \inf_{q \leq C u[\hat{\theta}(q)]} C' u[\hat{\theta}(q)], \end{aligned}$$

where  $C > 0$  is any constant and  $C' = \max(2C, 8)$ .

Note from Theorem 3 that

$$u[\hat{\theta}(q)] = O \left( \frac{1}{n} \left( \frac{\log n \log |H|}{\hat{\theta}(q)^2} + \log \frac{1}{\delta} \right) \right),$$

and

$$q = P_S(yf(x) \leq \hat{\theta}(q)).$$

Thus we can derive the following three bounds from the Emargin bound by using the three inequalities in Lemma 9 respectively.

**Corollary 10** *If  $|H| < \infty$ , then for any  $\delta > 0$ , with probability at least  $1 - \delta$  over the random choice of the training set  $S$  of  $n$  examples, every voting classifier  $f$  satisfies the following bounds:*

1.

$$P_D(yf(x) \leq 0) \leq O\left(\frac{1}{n} \left(\frac{\log n \log |H|}{\theta_0^2} + \log \frac{1}{\delta}\right)\right),$$

where  $\theta_0$  is the minimum margin.

2.

$$P_D(yf(x) \leq 0) \leq \inf_{\theta \in (0,1]} \left[ P_S(yf(x) \leq \theta) + O\left(\frac{1}{\sqrt{n}} \left(\frac{\log n \log |H|}{\theta^2} + \log \frac{1}{\delta}\right)^{1/2}\right) \right],$$

3.

$$P_D(yf(x) \leq 0) \leq O\left(\frac{1}{n} \left(\frac{\log n \log |H|}{\theta^2} + \log \frac{1}{\delta}\right)\right),$$

for all  $\theta$  such that

$$P_S(yf(x) \leq \theta) \leq O\left(\frac{1}{n} \left(\frac{\log n \log |H|}{\theta^2} + \log \frac{1}{\delta}\right)\right).$$

The first bound in the Corollary has the same order of magnitude as the minimum margin bound. The second bound is the same as Theorem 1. So essentially, previous bounds can be derived from the Emargin bound. The third bound in the Corollary is new. It states that the generalization error is  $O\left(\frac{\log n \log |H|}{n\theta^2}\right)$  even in the non-zero error case, provided the margin error  $P_S(yf(x) \leq \theta)$  is small enough.

## 5 Proofs

In this section, we give proofs of the theorems, lemmas and corollaries.

### 5.1 Proof of Theorem 3

The proof uses the tool developed in [SFBL98]. The difference is that we do not bound the deviation of the generalization error from the empirical margin error directly, instead we consider the difference of the generalization error to a zero-one function of a certain empirical measure. This allows us to unify the zero-error and nonzero-error cases and it results in a sharper bound. For the sake of convenience, we follow the convention in [SFBL98].

Let  $C(H)$  denote the convex hull of  $H$ . Also let  $C_N(H)$  denote the set of unweighted averages over  $N$  elements from the base classifier set  $H$ . Formally,

$$C_N(H) = \left\{ g : g = \frac{1}{N} \sum_{j=1}^N h_j, h_j \in H \right\}.$$

For any voting classifier

$$f = \sum \beta_i h_i \in C(H),$$

where

$$\sum \beta_i = 1, \beta_i \geq 0,$$

there can be associated with a distribution over  $H$  by the coefficients  $\{\beta_i\}$ . We denote this distribution as  $\tilde{Q}(f)$ . By choosing  $N$  elements independently and randomly from  $H$  according to  $\tilde{Q}(f)$ , we can generate a classifier  $g \in C_N(H)$ . The distribution of  $g$  is denoted by  $Q(f)$ . For any fixed  $\alpha$  ( $0 < \alpha < 1$ )

$$\begin{aligned} P_D(yf(x) \leq 0) &\leq P_{D,g \sim Q(f)}(yg(x) \leq \alpha) \\ &\quad + P_{D,g \sim Q(f)}(yg(x) > \alpha, yf(x) \leq 0) \\ &\leq P_{D,g \sim Q(f)}(yg(x) \leq \alpha) + \exp\left(-\frac{N\alpha^2}{2}\right). \end{aligned} \quad (8)$$

We next bound the first term on the right-hand side of the inequality. For any fixed  $g \in C_N(H)$ , and for any positive number  $\varepsilon$  and nonnegative integer  $k$  such that  $k \leq n\varepsilon$ , we consider the probability (over the random draw of  $n$  training examples) that the training error at margin  $\alpha$  is less than  $k/n$ , while the true error of  $g$  at margin  $\alpha$  is larger than  $\varepsilon$ . A compact representation of this probability is

$$\Pr_{S \sim D^n} \left( P_D(yg(x) \leq \alpha) > I \left[ P_S \left( yg(x) \leq \alpha \right) > \frac{k}{n} \right] + \varepsilon \right)$$

where  $\Pr_{S \sim D^n}$  denotes the probability over  $n$  training samples chosen independently at random according to  $D$ , and  $I$  is the indicator function. Note that

$$\begin{aligned} \Pr_{S \sim D^n} \left( P_D(yg(x) \leq \alpha) > I \left[ P_S(yg(x) \leq \alpha) > \frac{k}{n} \right] + \varepsilon \right) &\leq \Pr_{S \sim D^n} \left( P_S(yg(x) \leq \alpha) \leq \frac{k}{n} \mid P_D(yg(x) \leq \alpha) > \varepsilon \right) \\ &\leq \sum_{r=0}^k \binom{n}{r} \varepsilon^r (1-\varepsilon)^{n-r}. \end{aligned}$$

Then applying the relative entropy Chernoff bound to the Bernoulli trials, we further have

$$\sum_{r=0}^k \binom{n}{r} \varepsilon^r (1-\varepsilon)^{n-r} \leq \exp\left(-nD\left(\frac{k}{n} \parallel \varepsilon\right)\right).$$

We thus obtain

$$\begin{aligned} \Pr_{S \sim D^n} \left( P_D(yg(x) \leq \alpha) > I \left[ P_S(yg(x) \leq \alpha) > \frac{k}{n} \right] + \varepsilon \right) &\leq \exp\left(-nD\left(\frac{k}{n} \parallel \varepsilon\right)\right). \end{aligned} \quad (9)$$

We only consider  $\alpha$  at the values in the set

$$U = \left\{ \frac{1}{|H|}, \frac{2}{|H|}, \dots, 1 \right\}.$$

There are no more than  $|H|^N$  elements in  $C_N(H)$ . Using the union bound we get

$$\begin{aligned} & \Pr_{S \sim D^n} \left( \exists g \in C_N(H), \exists \alpha \in U, P_D(yg(x) \leq \alpha) \right. \\ & \quad \left. > I \left[ P_S(yg(x) \leq \alpha) > \frac{k}{n} \right] + \varepsilon \right) \\ & \leq |H|^{(N+1)} \exp \left( -nD \left( \frac{k}{n} \parallel \varepsilon \right) \right). \end{aligned}$$

Note that

$$\begin{aligned} & E_{g \sim Q(f)} P_D(yg(x) \leq \alpha) \\ & = P_{D, g \sim Q(f)}(yg(x) \leq \alpha), \\ & E_{g \sim Q(f)} I \left[ P_S(yg(x) \leq \alpha) > \frac{k}{n} \right] \\ & = P_{g \sim Q(f)} \left( P_S(yg(x) \leq \alpha) > \frac{k}{n} \right). \end{aligned}$$

We have

$$\begin{aligned} & \Pr_{S \sim D^n} \left( \exists f \in C(H), \exists \alpha \in U, P_{D, g \sim Q(f)}(yg(x) \leq \alpha) \right. \\ & \quad \left. > P_{g \sim Q(f)} \left( P_S(yg(x) \leq \alpha) > \frac{k}{n} \right) + \varepsilon \right) \\ & \leq |H|^{(N+1)} \exp \left( -nD \left( \frac{k}{n} \parallel \varepsilon \right) \right). \end{aligned}$$

Let

$$\delta = |H|^{(N+1)} \exp \left( -nD \left( \frac{k}{n} \parallel \varepsilon \right) \right),$$

then

$$\varepsilon = D^{-1} \left( \frac{k}{n}, \frac{1}{n} \left[ (N+1) \log |H| + \log \frac{1}{\delta} \right] \right).$$

We obtain that with probability at least  $1 - \delta$  over the draw of the training samples, for all  $f \in C(H)$ , all  $\alpha \in U$ ,

$$\begin{aligned} & P_{D, g \sim Q(f)}(yg(x) \leq \alpha) \\ & \leq P_{g \sim Q(f)} \left( P_S(yg(x) \leq \alpha) > \frac{k}{n} \right) \\ & \quad + D^{-1} \left( \frac{k}{n}, \frac{1}{n} \left[ (N+1) \log |H| + \log \frac{1}{\delta} \right] \right). \end{aligned}$$

Using the union bound over  $k = 0, 1, \dots, n$ , then with probability at least  $1 - \delta$  over the draw of the training samples, for all  $f \in C(H)$ , all  $\alpha \in U$ , and all  $k$

$$\begin{aligned} & P_{D, g \sim Q(f)}(yg(x) \leq \alpha) \\ & \leq P_{g \sim Q(f)} \left( P_S(yg(x) \leq \alpha) > \frac{k}{n} \right) \\ & \quad + D^{-1} \left( \frac{k}{n}, \frac{1}{n} \left[ (N+1) \log |H| + \log \frac{n}{\delta} \right] \right). \quad (10) \end{aligned}$$

We next bound the first term in the right-hand side of Eq.(10). Using the same argument for deriving Eq.(8), we have for any  $\theta > \alpha$

$$\begin{aligned} & P_{g \sim Q(f)} \left( P_S(yg(x) \leq \alpha) > \frac{k}{n} \right) \\ & \leq I \left[ P_S(yf(x) \leq \theta) > \frac{k}{n} \right] \\ & \quad + P_{g \sim Q(f)} \left( P_S(yg(x) > \alpha) > \frac{k}{n}, \right. \\ & \quad \left. P_S(yf(x) \leq \theta) \leq \frac{k}{n} \right). \quad (11) \end{aligned}$$

Note that the last term in Eq.(11) can be further bounded by

$$\begin{aligned} & P_{g \sim Q(f)} \left( \exists (x_i, y_i) \in S : y_i g(x_i) \leq \alpha \text{ and } y_i f(x_i) > \theta \right) \\ & \leq n \exp \left( -\frac{N(\theta - \alpha)^2}{2} \right). \quad (12) \end{aligned}$$

Combining (8), (10), (11) and (12), we have that with probability at least  $1 - \delta$  over the draw of training examples, for all  $f \in C(H)$ , all  $\alpha \in U$ , all  $\theta > \alpha$ , and all  $k$ , but fixed  $N$

$$\begin{aligned} & P_D(yf(x) \leq 0) \\ & \leq \exp \left( -\frac{N\alpha^2}{2} \right) + n \exp \left( -\frac{N(\theta - \alpha)^2}{2} \right) \\ & \quad + I \left[ P_S(yf(x) \leq \theta) > \frac{k}{n} \right] \\ & \quad + D^{-1} \left( \frac{k}{n}, \frac{1}{n} \left[ (N+1) \log |H| + \log \frac{n}{\delta} \right] \right). \end{aligned}$$

Let

$$\alpha = \frac{\theta}{2} - \frac{\eta}{|H|} \in U,$$

where  $0 \leq \eta < 1$ . It is easy to check that the sum of the first two terms on the right-hand side of the above inequality can be bounded by

$$\max \left( 2n, \exp \left( \frac{N}{2|H|} \right) \right) \exp \left( -\frac{N\theta^2}{8} \right).$$

Let

$$\delta_N = \delta \cdot 2^{-N},$$

we can get a union bound over all  $N$ . Put

$$N = \frac{8}{\theta^2} \log \left( \frac{2n^2}{\log |H|} \right),$$

note that if

$$\theta > \sqrt{\frac{8}{|H|}},$$

then

$$2n > \exp \left( \frac{N}{2|H|} \right).$$

We obtain

$$P_D(yf(x) \leq 0) \leq \frac{\log |H|}{n} + \inf_{0 \leq k < n} \left( I \left[ P_S(yf(x) \leq \theta) > \frac{k}{n} \right] + D^{-1} \left( \frac{k}{n}, u \right) \right),$$

where

$$u = \frac{1}{n} \left( \frac{8}{\theta^2} \log \left( \frac{2n^2}{\log |H|} \right) \log |H| + \log |H| + \log \frac{n}{\delta} \right).$$

The theorem follows.  $\blacksquare$

## 5.2 Proof of Proposition 5

Let  $M$  be the set defined as

$$M = \left\{ q : \hat{\theta}(q) = \hat{\theta}(q^*) = \theta^* \right\}.$$

Let  $q_0$  be the minimal  $q$  in  $M$ . We will show that

$$q^* = q_0, \quad (13)$$

and

$$P_S(yf(x) < \theta^*) = q_0. \quad (14)$$

To show  $q^* = q_0$ , note that  $D^{-1}(q, u)$  is an increasing function of  $q$  for fixed  $u$ . Since  $q^*$  is the optimal value such that  $D^{-1}(q, u(\hat{\theta}(q)))$  achieves the minimum, one must have  $q^* = q_0$ .

To show

$$P_S(yf(x) < \theta^*) = q_0,$$

first note that

$$P_S(yf(x) < \theta^*) \in M.$$

For every  $q \in M$ , by the definition of  $\hat{\theta}(q)$ , one has

$$P_S(yf(x) < \theta^*) \leq q.$$

This implies

$$P_S(yf(x) < \theta^*) = q_0.$$

This completes the proof.  $\blacksquare$

## 5.3 Proof of Theorem 6

The following lemma will be used to prove Theorem 6.

**Lemma 11**  $D^{-1}(0, p) \leq p$  for  $p \geq 0$ .

**Proof of Lemma 11.** We only need to show

$$D(0|p) \geq p,$$

since  $D(q|p)$  is a monotonic increasing function of  $p$  for  $p \geq q$ . By the Taylor expansion

$$D(0|p) = -\log(1-p) = p + \frac{p^2}{2} + \frac{p^3}{3} + \dots \geq p. \quad \blacksquare$$

**Proof of Theorem 6.** The right-hand side of the Emargin bound (3) is the minimum over all  $q \in$

$\{0, \frac{1}{n}, \frac{2}{n}, \dots, 1\}$ . Take  $q = 0$ , it is clear that  $\hat{\theta}(0)$  is the minimum margin. By Lemma 9, the Emargin bound can be relaxed to

$$P_D(yf(x) \leq 0) \leq \frac{1}{n} \left( \frac{8}{\theta_0^2} \log \left( \frac{2n^2}{\log |H|} \right) \log |H| + 2 \log |H| + \log \frac{n}{\delta} \right). \quad (15)$$

We show that this relaxed bound is sharper than Theorem 2. For the minimum margin bound, we only consider the case that  $R \leq 1$ , since otherwise the bound is larger than one. Simple calculations show that the right-hand side of (15) is smaller than the minimum margin bound. The theorem then follows.  $\blacksquare$

## 5.4 Proof of Theorem 7

According to Proposition 5, we have that  $q_i = P_S(yf_i(x) < \theta_i)$  is also the optimal  $q^*$  in the Emargin bound. Thus we only need to show

$$D^{-1}(q_1, u(\theta_1)) \leq D^{-1}(q_2, u(\theta_2)).$$

Note that if  $\theta_1 \geq \theta_2$ , then  $u(\theta_1) \leq u(\theta_2)$ . So

$$D^{-1}(q_2, u(\theta_2)) \geq D^{-1}(q_2, u(\theta_1)),$$

since  $D^{-1}(q, u)$  is an increasing function of  $u$  for fixed  $q$ . Also  $D^{-1}(q, u)$  is an increasing function of  $q$  for fixed  $u$ , we have

$$D^{-1}(q_2, u(\theta_1)) \geq D^{-1}(q_1, u(\theta_1))$$

since  $q_1 \leq q_2$ . This completes the proof.  $\blacksquare$

## 5.5 Proof of Theorem 8

The next lemma is a modified version of the uniform convergence result of [VC71, Vap98] and its refinement [Dev82]. It will be used for proving Theorem 8.

**Lemma 12** Let  $\mathcal{A}$  be a class of subsets of a space  $Z$ . Let  $N^{\mathcal{A}}(z_1, z_2, \dots, z_n)$  be the number of different sets in

$$\left\{ \{z_1, z_2, \dots, z_n\} \cap A : A \in \mathcal{A} \right\}.$$

Define

$$s(\mathcal{A}, n) = \max_{(z_1, z_2, \dots, z_n) \in Z^n} N^{\mathcal{A}}(z_1, z_2, \dots, z_n).$$

Then for any fixed integer  $k$

$$\begin{aligned} \Pr_{S \sim D^n} \left( \exists A \in \mathcal{A} : P_D(A) > I \left[ P_S(A) > \frac{k}{n} \right] + \varepsilon \right) \\ \leq 2 \cdot s(\mathcal{A}, n^2) \exp \left( -nD \left( \frac{k}{n} \middle| \varepsilon' \right) \right), \end{aligned}$$

where

$$\varepsilon' = \frac{n}{n-1} \varepsilon - \frac{1}{n}.$$

**Proof of Lemma 12.** The proof is the standard argument. We first show that for any  $0 < \alpha < 1$ ,  $\varepsilon > 0$ , and any integer  $n'$

$$\begin{aligned} & \Pr_{S \sim D^n} \left( \exists A \in \mathcal{A} : P_D(A) > I \left[ P_S(A) > \frac{k}{n} \right] + \varepsilon \right) \\ & \leq \left( \frac{1}{1 - e^{-2n'\alpha^2\varepsilon^2}} \right) \Pr_{S \sim D^n, S' \sim D^{n'}} \left( \exists A \in \mathcal{A} : P_{S'}(A) \right. \\ & \quad \left. > I \left[ P_S(A) > \frac{k}{n} \right] + (1 - \alpha)\varepsilon \right). \end{aligned}$$

Or equivalently,

$$\begin{aligned} & \Pr_{S \sim D^n} \left( \sup_{A \in \mathcal{A}} \left( P_D(A) - I \left[ P_S(A) > \frac{k}{n} \right] \right) > \varepsilon \right) \\ & \leq \left( \frac{1}{1 - e^{-2n'\alpha^2\varepsilon^2}} \right) \Pr_{S \sim D^n, S' \sim D^{n'}} \left( \sup_{A \in \mathcal{A}} \left( P_{S'}(A) \right. \right. \\ & \quad \left. \left. - I \left[ P_S(A) > \frac{k}{n} \right] \right) > (1 - \alpha)\varepsilon \right). \quad (16) \end{aligned}$$

Let  $V$  denote the event

$$\sup_{A \in \mathcal{A}} \left( P_D(A) - I \left[ P_S(A) > \frac{k}{n} \right] \right) > \varepsilon.$$

Let  $A^*$  be (one of) the optimal  $A$  so that

$$P_D(A) - I \left[ P_S(A) > \frac{k}{n} \right]$$

achieves the maximum. Note that the following two events

$$P_{S'}(A^*) \geq P_D(A^*) - \alpha\varepsilon$$

and

$$P_D(A^*) - I \left[ P_S(A^*) > \frac{k}{n} \right] > \varepsilon$$

imply that

$$P_{S'}(A^*) - I \left[ P_S(A^*) > \frac{k}{n} \right] > (1 - \alpha)\varepsilon.$$

Then

$$\begin{aligned} & \Pr_{S \sim D^n, S' \sim D^{n'}} \left( \sup_{A \in \mathcal{A}} \left( P_{S'}(A) - I \left[ P_S(A) > \frac{k}{n} \right] \right) \right. \\ & \quad \left. > (1 - \alpha)\varepsilon \right) \\ & = \int dP \int I \left[ \sup_{A \in \mathcal{A}} \left( P_{S'}(A) - I \left[ P_S(A) > \frac{k}{n} \right] \right) \right. \\ & \quad \left. > (1 - \alpha)\varepsilon \right] dP' \\ & \geq \int_V dP \int I \left[ \sup_{A \in \mathcal{A}} \left( P_{S'}(A) - I \left[ P_S(A) > \frac{k}{n} \right] \right) \right. \\ & \quad \left. > (1 - \alpha)\varepsilon \right] dP' \\ & \geq \int_V dP \int I \left[ P_{S'}(A^*) - I \left[ P_S(A^*) > \frac{k}{n} \right] \right. \\ & \quad \left. > (1 - \alpha)\varepsilon \right] dP' \\ & \geq \int_V dP \int I \left[ P_{S'}(A^*) \geq P_D(A^*) - \alpha\varepsilon \right] dP' \\ & \geq \left( 1 - e^{-2n'\alpha^2\varepsilon^2} \right) \int_V dP \\ & = \left( 1 - e^{-2n'\alpha^2\varepsilon^2} \right) \\ & \quad \times \Pr_{S \sim D^n} \left( \sup_{A \in \mathcal{A}} \left( P_D(A) - I \left[ P_S(A) > \frac{k}{n} \right] \right) > \varepsilon \right). \end{aligned}$$

This completes the proof of (16).

Take

$$\begin{aligned} n' &= n^2 - n, \\ \alpha &= \frac{1}{(n-1)\varepsilon}, \end{aligned}$$

we have

$$\begin{aligned} & \Pr_{S \sim D^n} \left( \exists A \in \mathcal{A} : P_D(A) > I \left[ P_S(A) > \frac{k}{n} \right] + \varepsilon \right) \\ & \leq 2 \Pr_{S \sim D^n, S' \sim D^{n'}} \left( \exists A \in \mathcal{A} : P_{S'}(A) \right. \\ & \quad \left. > I \left[ P_S(A) > \frac{k}{n} \right] + \left( \varepsilon - \frac{1}{n-1} \right) \right). \end{aligned}$$

Proceeding as [Dev82] and using the relative entropy Hoeffding inequality, the theorem follows.  $\blacksquare$

**Proof of Theorem 8.** The proof is the same as Theorem 3 until we have Eq.(9). Let  $\alpha = \frac{\theta}{2}$ , we need to

bound

$$\Pr_{S \sim D^n} \left( \exists g \in C_N(H), \exists \theta > 0, P_D(yg(x) \leq \frac{\theta}{2}) > I \left[ P_S(yg(x) \leq \frac{\theta}{2}) > \frac{k}{n} \right] + \varepsilon \right).$$

Note that we only need to consider  $\theta = 0, \frac{1}{N}, \frac{2}{N}, \dots, 1$ . Let

$$A(g) = \left\{ (x, y) \in X \times \{-1, 1\} : yg(x) \leq \frac{\theta}{2} \right\},$$

and

$$\mathcal{A} = \{A(g) : g \in C_N(H)\}.$$

By Sauer's lemma [Sau72] it is easy to see that

$$s(\mathcal{A}, n) \leq \left(\frac{en}{d}\right)^{Nd},$$

where  $d$  is the VC dimension of  $H$ . By Lemma 12, we have

$$\begin{aligned} \Pr_{S \sim D^n} \left( \exists g \in C_N(H), \exists \theta > 0, P_D(yg(x) \leq \frac{\theta}{2}) > I \left[ P_S(yg(x) \leq \frac{\theta}{2}) > \frac{k}{n} \right] + \varepsilon \right) \\ \leq 2(N+1) \left(\frac{en^2}{d}\right)^{Nd} \exp\left(-nD\left(\frac{k}{n} \parallel \varepsilon'\right)\right), \end{aligned}$$

where

$$\varepsilon' = \frac{n}{n-1} \varepsilon - \frac{1}{n}.$$

Using the argument as Theorem 3, the theorem follows.  $\blacksquare$

**Proof of Lemma 9.** The first inequality has already been proved in Lemma 11.

For the second inequality, we only need to show

$$D^{-1}(q, u) \leq q + \sqrt{u/2},$$

or equivalently

$$D(q, q + \sqrt{u/2}) \geq u,$$

since  $D$  is an increasing function in the second parameter. But this is immediate by a well known result [Hoe63]:

$$D(q, q + \delta) \geq 2\delta^2.$$

For the third inequality we first show that for all  $0 < q < 1$

$$D^{-1}\left(\frac{q}{2}, \frac{q}{8}\right) \leq q, \quad (17)$$

which is equivalent to

$$D\left(\frac{q}{2} \parallel q\right) \geq \frac{q}{8}.$$

For fixed  $q$ , let  $\phi(x) = D(qx \parallel q)$ ,  $0 < x \leq 1$ . Note that

$$\phi(1) = \phi'(1) = 0,$$

and

$$\phi''(x) = \frac{q}{x(1-qx)} \geq q,$$

we have

$$D\left(\frac{q}{2} \parallel q\right) = \phi\left(\frac{1}{2}\right) \geq \frac{q}{8}.$$

This completes the proof of Eq. (17).

Now if  $q \leq C'u[\hat{\theta}(q)]$ , recall that  $C' = \max(2C, 8)$ , and note  $D^{-1}$  is increasing function on its first and second parameter respectively. We have

$$\begin{aligned} D^{-1}\left(q, u[\hat{\theta}(q)]\right) &\leq D^{-1}\left(\frac{C'}{2}u[\hat{\theta}(q)], u[\hat{\theta}(q)]\right) \\ &\leq D^{-1}\left(\frac{C'}{2}u[\hat{\theta}(q)], \frac{C'}{8}u[\hat{\theta}(q)]\right) \\ &\leq C'u[\hat{\theta}(q)]. \end{aligned}$$

The lemma then follows.  $\blacksquare$

## 6 Experiments

In this section we provide experimental results to verify our theory. We compare AdaBoost and arc-gv in terms of their Emargin, Emargin error and the generalization error. Theorem 7 indicates that if a voting classifier  $f_1$  has a larger Emargin and a smaller Emargin error than another classifier  $f_2$ , then  $f_1$  would have better performance on the test data. The goal of the experiment is to see whether the empirical results agree with the theoretical prediction.

The experiments are conducted on 10 benchmark datasets described in Table 1. Except the USPS which contains handwritten digits, all datasets are from the UCI repository [AN07]. If the data is multiclass, we group them into two classes, since we study the binary classification problem. For instance, the "letter" dataset has 26 classes, we use the first 13 as the positive and the others as the negative. In the preprocessing stage, each feature is normalized to  $[0, 1]$ . All datasets are used in a five-fold cross validation manner. For the USPS which originally has a training set and a test set, we merge them and regenerate the cross validation data.

In all experiments, decision stumps are adopted as the base learner, so the complexity of the base classifiers is well controlled. We use a finite set of possible decision stumps. Specifically, for each feature we consider 100 thresholds uniformly distributed on  $[0, 1]$ . Therefore the size of the base classifier set is  $2 \times 100 \times k$ , where  $k$  denotes the number of features.

We run AdaBoost and arc-gv for 500 rounds, then calculate the Emargin, Emargin error, test error as well as the minimum margin of them respectively. The results are described in Table 2. AdaBoost has a larger or equal Emargin and a smaller Emargin error than arc-gv on all the datasets except *German* and *Ionosphere*. According to our theory, it predicts that AdaBoost would have a lower generalization error. The experiments show that among these eight datasets, AdaBoost outperforms arc-gv on six datasets, ties on one dataset, and loses

Table 1: Description of the datasets

Dataset	# Examples	# Features	Dataset	# Examples	# Features
Breast	683	9	Letter	20000	16
Diabetes	768	8	Satimage	6435	36
German	1000	24	USPS	9298	256
Image	2310	16	Vehicle	846	20
Ionosphere	351	34	Wdbc	569	30

Table 2: Margin measures and performances of AdaBoost and arc-gv. For the datasets in bold-face, AdaBoost generates larger Emargins and smaller Emargin errors than arc-gv. AdaBoost outperforms arc-gv on all these datasets except the *Image* dataset.

		Emargin	Emargin Error	Test Error	Minimum margin
<b>Breast</b>	AdaBoost	<b>0.313</b>	<b>0.803</b>	<b>0.052</b>	0.005
	arc-gv	0.281	0.909	0.057	0.008
<b>Diabetes</b>	AdaBoost	<b>0.110</b>	<b>0.748</b>	<b>0.255</b>	-0.064
	arc-gv	0.049	0.759	0.256	-0.017
German	AdaBoost	0.157	0.824	0.258	-0.118
	arc-gv	0.034	0.780	0.261	-0.026
<b>Image</b>	AdaBoost	<b>0.196</b>	<b>0.610</b>	0.023	-0.009
	arc-gv	0.195	0.705	<b>0.021</b>	-0.003
Ionosphere	AdaBoost	0.323	0.800	0.100	0.084
	arc-gv	0.131	0.577	0.106	0.061
<b>Letter</b>	AdaBoost	<b>0.078</b>	<b>0.645</b>	<b>0.174</b>	-0.165
	arc-gv	0.063	0.958	0.178	-0.034
<b>Satimage</b>	AdaBoost	<b>0.133</b>	<b>0.521</b>	<b>0.053</b>	-0.054
	arc-gv	0.133	0.956	0.057	-0.019
<b>USPS</b>	AdaBoost	<b>0.108</b>	<b>0.972</b>	<b>0.450</b>	-0.142
	arc-gv	0.053	0.990	0.460	-0.024
<b>Vehicle</b>	AdaBoost	<b>0.129</b>	<b>0.737</b>	<b>0.297</b>	-0.117
	arc-gv	0.052	0.794	0.304	-0.033
<b>Wdbc</b>	AdaBoost	<b>0.350</b>	<b>0.581</b>	<b>0.035</b>	-0.130
	arc-gv	0.350	0.710	<b>0.035</b>	-0.100

only on one dataset. These results agree well with our theory.

Note also that on all the datasets except *Ionosphere*, arc-gv has a larger minimum margin than AdaBoost, but arc-gv has a lower test error than AdaBoost only on one dataset. This verifies that the minimum margin is not crucial for the generalization error.

## 7 Conclusions

In this paper we tried to give a complete answer to Breiman’s doubt on the margin explanation of the AdaBoost algorithm. We proposed a bound in terms of a new margin measure called the Emargin, which depends on the whole margin distribution. This bound is uniformly sharper than the minimum margin bound used by Breiman to back up his argument. According to our theory, arc-gv does not necessarily outperform AdaBoost even though it generates larger minimum margins.

Our bounds also imply that the Emargin and the

Emargin error are the key to the generalization error of a voting classifier—a larger Emargin and a smaller Emargin error result in better generalization ability. Experiments on benchmark datasets agree well with our theory.

A future work is to study why AdaBoost generates larger Emargins and smaller Emargin errors, i.e., better margin distributions, than arc-gv. Can we find a strategy that optimizes the margin distribution? If such an algorithm exists, it would be a good test of our theory to see whether it has better performance than AdaBoost as we predict.

## References

- [AKLL02] A. Antos, B. Kégl, T. Linder, and G. Lugosi. Data-dependent margin-based generalization bounds for classification. *Journal of Machine Learning Research*, 3:73–98, 2002.
- [AN07] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

- [BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.
- [BK99] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, 36:105–139, 1999.
- [Bre98] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26:801–849, 1998.
- [Bre99] L. Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11:1493–1517, 1999.
- [Dev82] L. Devroye. Bounds for the uniform deviation of empirical measures. *Journal of Multivariate Analysis*, 12:72–79, 1982.
- [Die00] T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning*, 40:139–157, 2000.
- [FS96] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, 1996.
- [FS97] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.
- [GS98] A. J. Grove and D. Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *National Conference on Artificial Intelligence*, 1998.
- [Hoe63] W. Hoeffding. Probability inequalities for sum of bounded random variables. *Journal of American Statistical Society*, 58:13–30, 1963.
- [KP02] V. Koltchinskii and D. Panchanko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30:1–50, 2002.
- [KP05] V. Koltchinskii and D. Panchanko. Complexities of convex combinations and bounding the generalization error in classification. *Annals of Statistics*, 33:1455–1496, 2005.
- [Lan05] J. Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, 2005.
- [MBG02] L. Mason, P. Bartlett, and M. Golea. Generalization error of combined classifiers. *Journal of Computer and System Sciences*, 65:415–438, 2002.
- [Qui96] J. R. Quinlan. Bagging, boosting, and c4.5. In *13th International Conference on Artificial Intelligence*, 1996.
- [RS06] L. Reyzin and R. E. Schapire. How boosting the margin can also boost classifier complexity. In *International Conference on Machine Learning*, 2006.
- [RW02] G. Rätsch and M. Warmuth. Maximizing the margin with boosting. In *15th Annual Conference on Computational Learning Theory*, 2002.
- [Sau72] N. Sauer. On the density of family of sets. *Journal of Combinatorial Theory, Series A*, 13:145–147, 1972.
- [SFBL98] R. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26:1651–1686, 1998.
- [Vap98] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., 1998.
- [VC71] V. N. Vapnik and A. YA. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16:264–280, 1971.
- [VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

---

# Adaptive Hausdorff Estimation of Density Level Sets

---

Aarti Singh and Robert D. Nowak\*

University of Wisconsin - Madison, USA

singh@cae.wisc.edu, nowak@enr.wisc.edu

Clayton D. Scott

University of Michigan - Ann Arbor, USA

cscott@eecs.umich.edu

## Abstract

Hausdorff accurate estimation of density level sets is relevant in applications where a spatially uniform mode of convergence is desired to ensure that the estimated set is close to the target set at all points. The minimax optimal rate of error convergence for the Hausdorff metric is known to be  $(n/\log n)^{-1/(d+2\alpha)}$  for level sets with boundaries that have a Lipschitz functional form, and where the parameter  $\alpha$  characterizing the regularity of the density around the level of interest is known. Thus, all previous work is non-adaptive to the density regularity and assumes knowledge of the regularity parameter  $\alpha$ . Moreover, the estimators proposed in previous work achieve the minimax optimal rate for rather restricted classes of sets (for example, the boundary fragment and star-shaped sets) that effectively reduce the set estimation problem to a function estimation problem. This characterization precludes level sets with multiple connected components, which are fundamental to many applications. This paper presents a fully data-driven procedure that is adaptive to unknown local density regularity, and achieves minimax optimal Hausdorff error control for a class of level sets with very general shapes and multiple connected components.

## 1 Introduction

Density level sets provide useful summaries of a density function for many applications including clustering [Har75, Stu03], anomaly detection [SHS05, SN06, VV06], and data ranking [LPS99]. In practice, however, the density function itself is unknown a priori and only a finite number of observations from the density are available. Let  $X_1, \dots, X_n$  be independent, identically distributed observations drawn from an unknown probability measure  $P$ , having density  $f$  with respect to the Lebesgue measure, and defined on the domain  $\mathcal{X} \subseteq \mathbb{R}^d$ . Given a desired density level  $\gamma$ , consider the  $\gamma$ -level set of the density  $f$

$$G_\gamma^* := \{x \in \mathcal{X} : f(x) \geq \gamma\}.$$

---

\*This work was partially supported by the National Science Foundation grants CNS-0519824 and ECS-0529381.

The goal of the density level set estimation problem is to generate an estimate  $\hat{G}$  of the level set based on the  $n$  observations  $\{X_i\}_{i=1}^n$ , such that the error between the estimate  $\hat{G}$  and the target set  $G_\gamma^*$ , as assessed by some performance measure which gauges the closeness of the two sets, is small.

Most literature available on level set estimation [SHS05, SN06, SD07, WN07, KT93, Tsy97, Pol95, RV06] considers global error measures related to the symmetric set difference. However some applications may need a more local or spatially uniform error measure as provided by the Hausdorff metric, for example, to ensure robustness to outliers or preserve topological properties of the level set. The Hausdorff error metric is defined as follows between two non-empty sets:

$$d_\infty(G_1, G_2) = \max\left\{\sup_{x \in G_2} \rho(x, G_1), \sup_{x \in G_1} \rho(x, G_2)\right\}$$

where  $\rho(x, G) = \inf_{y \in G} \|x - y\|$ , the smallest Euclidean distance of a point in  $G$  to the point  $x$ . If  $G_1$  or  $G_2$  is empty, then let  $d_\infty(G_1, G_2)$  be defined as the largest distance between any two points in the domain. Control of this error measure provides a uniform mode of convergence as it implies control of the deviation of a single point from the desired set. A symmetric set difference based estimator may not provide such a uniform control as it is easy to see that a set estimate can have very small symmetric difference error but large Hausdorff error. Conversely, as long as the set boundary is not space-filling, small Hausdorff error implies small symmetric difference error.

Existing results pertaining to nonparametric level set estimation using the Hausdorff metric [KT93, Tsy97, Cav97] focus on rather restrictive classes of level sets (for example, the boundary fragment and star-shaped set classes). These restrictions, which effectively reduce the set estimation problem to a boundary function estimation problem (in rectangular or polar coordinates, respectively), are typically not met in practical applications. In particular, the characterization of level set estimation as a boundary function estimation problem precludes level sets with multiple connected components, which are fundamental to many applications. Moreover, the estimation techniques proposed in [KT93, Tsy97, Cav97] require precise knowledge of the local regularity of the distribution (quantified by the parameter  $\alpha$ , to be defined below) in the vicinity of the desired level set in order to achieve minimax optimal rates of convergence. Such prior knowledge is unavailable in most practical ap-

plications. Recently, a plug-in method based on sup-norm density estimation was put forth in [CMC06] that can handle more general classes than boundary fragments or star-shaped sets, however sup-norm based methods require global smoothness assumptions on the density to ensure that the density estimate is good everywhere. Also, the method only deals with a special case of the density regularity condition considered in this paper ( $\alpha = 1$ ), and is therefore not adaptive to unknown density regularity.

In this paper, we propose a plug-in procedure based on a regular histogram partition that can adaptively achieve minimax optimal rates of Hausdorff error convergence over a broad class of level sets with very general shapes and multiple connected components, without assuming *a priori* knowledge of the density regularity parameter  $\alpha$ . Adaptivity is achieved by a new data-driven procedure for selecting the histogram resolution. The procedure is specifically designed for the level set estimation problem and only requires local regularity of the density in the vicinity of the desired level.

## 2 Density assumptions

In this paper, we assume that the density  $f$  is supported on the unit hypercube in  $d$ -dimensions, that is  $\mathcal{X} = [0, 1]^d$ , and is bounded with range  $[0, f_{\max}]$ . Controlling the Hausdorff accuracy of level set estimates also requires some smoothness assumptions. The most crucial assumption is the first, which characterizes the relationship between distances and changes in density. The last two are topological assumptions on the level set and essentially generalize the notion of Lipschitz functions to closed hypersurfaces.

Here we define an  $\epsilon$ -ball centered at a point  $x$  as  $B(x, \epsilon) = \{y \in \mathcal{X} : \|x - y\| \leq \epsilon\}$ , where  $\|\cdot\|$  denotes the Euclidean distance. Also, an *inner*  $\epsilon$ -cover of a set  $G$  is defined as the union of all  $\epsilon$ -balls contained in  $G$ . Formally,  $\mathcal{I}_\epsilon(G) = \bigcup_{x: B(x, \epsilon) \subseteq G} B(x, \epsilon)$ .

**[A] Local density regularity:** The density is  $\alpha$ -regular around the  $\gamma$ -level set,  $0 < \alpha < \infty$  and  $\gamma < f_{\max}$ , if there exist constants  $C_2 > C_1 > 0$  and  $\delta_0, \delta_1 > 0$  such that

$$C_1 \rho(x, \partial G_\gamma^*)^\alpha \leq |f(x) - \gamma| \leq C_2 \rho(x, \partial G_\gamma^*)^\alpha$$

for all  $x \in \mathcal{X}$  with  $|f(x) - \gamma| \leq \delta_0$ , where  $\partial G_\gamma^*$  is the boundary of the true level set  $G_\gamma^*$ . And there exists  $y_0 \in \partial G_\gamma^*$  such that for all  $x \in B(y_0, \delta_1)$ ,  $|f(x) - \gamma| \leq \delta_0$ .

This assumption is similar to the one used in [Tsy97, Cav97] (we elaborate on the differences later on). The regularity parameter  $\alpha$  determines the rate of error convergence for level set estimation. Accurate estimation is more difficult at levels where the density is relatively flat (large  $\alpha$ ), as intuition would suggest. In this paper, we do not assume knowledge of  $\alpha$  unlike previous investigations into Hausdorff accurate level set estimation [KT93, Tsy97, Cav97, CMC06]. Therefore, here the assumption simply states that there is a relationship between distance and density level, but the precise nature of the relationship is unknown.

**[B] Level set regularity:** There exist constants  $\epsilon_o > 0$  and  $C_3 > 0$  such that for all  $\epsilon \leq \epsilon_o$ ,  $\mathcal{I}_\epsilon(G_\gamma^*) \neq \emptyset$  and  $\rho(x, \mathcal{I}_\epsilon(G_\gamma^*)) \leq C_3 \epsilon$  for all  $x \in \partial G_\gamma^*$ .

This assumption states that the level set is not arbitrarily narrow anywhere. It precludes features like cusps and arbitrarily thin ribbons, as well as connected components of arbitrarily small size. This condition is necessary since arbitrarily small features cannot be detected and resolved from a finite sample. However, from a practical perspective, if the assumption fails to hold then it simply means that it is not possible to theoretically guarantee that such small features will be recovered.

**[C] Level set boundary dimension:** There exists a constant  $C_4 > 0$  such that for all  $x \in \partial G_\gamma^*$  and all  $\epsilon, \delta$  such that  $0 < \delta \leq \epsilon$ , the minimum number of  $\delta$ -balls required to cover  $\partial G_\gamma^* \cap B(x, \epsilon)$  is  $\leq C_4 (\delta/\epsilon)^{-(d-1)}$ .

This assumption is related to the box-counting dimension [Fal90] of the boundary of the level set. It essentially says that, at any scale, the boundary behaves locally like a  $(d - 1)$ -dimensional surface in the  $d$ -dimensional domain and is not space-filling. This condition is not restrictive since the Hausdorff error itself is inappropriate for space-filling curves, and in fact it is not required if the density regularity parameter  $\alpha$  is known. However, the condition is needed to achieve adaptivity using the proposed method, as we shall discuss later.

Let  $\mathcal{F}_1^*(\alpha)$  denotes the class of densities satisfying assumptions **[A, B]**, and  $\mathcal{F}_2^*(\alpha)$  denotes the class of densities satisfying assumptions **[A, B, C]**. The dependence on other parameters is omitted as these do not influence the minimax optimal rate of convergence. The classes  $\mathcal{F}_1^*(\alpha), \mathcal{F}_2^*(\alpha)$  are a generalization of the Lipschitz boundary fragments or star-shaped sets considered in [KT93, Tsy97, Cav97] since assumptions **[B, C]** basically imply that the boundary looks locally like a Lipschitz function. In fact assumptions **[B, C]** are satisfied by a Lipschitz boundary fragment or star-shaped set; please refer to Section 5.3 for a formal proof. However, there is a slight difference between the upper bound of assumption **[A]** here and that employed in [Tsy97, Cav97]. The upper bound assumption in [Tsy97, Cav97] only requires that the set  $\{x : |f(x) - \gamma| \leq \delta_0\}$  be non-empty. So as long as there is at least one point on the boundary where the density regularity assumption **[A]** holds, this determines the complexity of the class. Our assumption requires the density regularity to hold for an open neighborhood about at least one point on the boundary. This is necessary for adaptivity since a procedure cannot sense the regularity as characterized by  $\alpha$  unless the regularity holds in a region with positive measure.

In [Tsy97], Tsybakov established a minimax lower bound of  $(n/\log n)^{-\frac{1}{d+2\alpha}}$  for the class of Lipschitz star-shaped sets, which satisfy our assumptions **[B, C]** (see Section 5.3) and the slightly modified version of assumption **[A]**, as discussed above. His proof uses Fano's lemma to derive the lower bound for a discrete subset of densities from this class. It is easy to see that the discrete subset of densities

used in his construction also satisfy our form of assumption [A]. Hence, the same lower bound holds for the classes  $\mathcal{F}_1^*(\alpha)$  and  $\mathcal{F}_2^*(\alpha)$  under consideration as well and we have the following proposition. Here  $\mathbb{E}$  denotes expectation with respect to the random data sample.

**Proposition 1** *There exists  $c > 0$  such that*

$$\inf_{\widehat{G}_n} \sup_{f \in \mathcal{F}_1^*(\alpha)} \mathbb{E}[d_\infty(\widehat{G}_n, G_\gamma^*)] \geq \inf_{\widehat{G}_n} \sup_{f \in \mathcal{F}_2^*(\alpha)} \mathbb{E}[d_\infty(\widehat{G}_n, G_\gamma^*)] \\ \geq c \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}}.$$

Here the inf is taken over all possible set estimators  $\widehat{G}_n$ .

In the paper, we present a method that achieves this minimax lower bound for the class  $\mathcal{F}_1^*(\alpha)$ , given knowledge of the density regularity parameter  $\alpha$ . We also extend the method to achieve adaptivity to  $\alpha$  for the class  $\mathcal{F}_2^*(\alpha)$  under the additional assumption [C], while preserving the minimax optimal rate of convergence.

### 3 Proposed method

In this section, we propose a plug-in level set estimator that is based on a regular histogram. The histogram resolution is adaptively selected in a purely data-driven way without assuming knowledge of the local density regularity.

Let  $\mathcal{A}_j$  denote the collection of cells in a regular partition of  $[0, 1]^d$  into hypercubes of dyadic sidelength  $2^{-j}$ , where  $j$  is a non-negative integer. The estimator at this resolution is given as

$$\widehat{G}_j = \{A \in \mathcal{A}_j : \widehat{f}(A) \geq \gamma\}. \quad (1)$$

Here  $\widehat{f}(A) = \widehat{P}(A)/\mu(A)$ , where  $\widehat{P}(A) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{\{X_i \in A\}}$  denotes the empirical probability of an observation occurring in  $A$  and  $\mu$  is the Lebesgue measure.

Our first result shows that, if the density regularity parameter  $\alpha$  is known, then the correct resolution can be chosen (as in [Tsy97, Cav97]), and the corresponding estimator achieves near minimax optimal rate over the class of densities given by  $\mathcal{F}_1^*(\alpha)$ . Here  $\mathbb{E}$  denotes expectation with respect to the random data sample. We introduce the notation  $a_n \asymp b_n$  to denote that  $a_n = O(b_n)$  and  $b_n = O(a_n)$ .

**Theorem 1** *Assume that the local density regularity  $\alpha$  is known. Pick resolution  $j$  such that  $2^{-j} \asymp s_n(n/\log n)^{-\frac{1}{(d+2\alpha)}}$ , where  $s_n$  is a monotone diverging sequence. Then*

$$\sup_{f \in \mathcal{F}_1^*(\alpha)} \mathbb{E}[d_\infty(\widehat{G}_j, G_\gamma^*)] \leq C s_n \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}}$$

for all  $n$ , where  $C \equiv C(C_1, C_3, \epsilon_0, f_{\max}, \delta_0, d, \alpha) > 0$  is a constant.

The proof is given in Section 5.1.

Theorem 1 provides an upper bound on the Hausdorff error of our estimate. If  $s_n$  is slowly diverging, for example if  $s_n = (\log n)^\epsilon$  where  $\epsilon > 0$ , this upper bound agrees with the minimax lower bound of Proposition 1 up to a  $(\log n)^\epsilon$

factor. Hence the proposed estimator can achieve near minimax optimal rates, given knowledge of the density regularity. We would like to point out that if the parameter  $\delta_0$  characterizing assumption [A] and the density bound  $f_{\max}$  are also known, then the appropriate resolution can be chosen as  $j = \lfloor \log_2(c^{-1}(n/\log n)^{1/(d+2\alpha)}) \rfloor$ , where the constant  $c \equiv c(\delta_0, f_{\max})$ . With this choice, the optimal sidelength scales as  $2^{-j} \asymp (n/\log n)^{-1/(d+2\alpha)}$ , and the estimator  $\widehat{G}_j$  exactly achieves the minimax optimal rate.

#### 3.1 Adaptivity to unknown local density regularity

In this section we present a procedure that automatically selects the appropriate resolution  $j$  without prior knowledge of  $\alpha$ . The selected resolution needs to be adapted to the local regularity of the density around the level of interest. To achieve this, we propose the following *vernier*:

$$\mathcal{V}_{\gamma,j} = \min_{A \in \mathcal{A}_j} \max_{A' \in \mathcal{A}_{j'} \cap A} |\gamma - \bar{f}(A')|.$$

Here  $\bar{f}(A) = P(A)/\mu(A)$ , and  $j' = \lfloor j + \log_2 s_n \rfloor$ , where  $s_n$  is a slowly diverging monotone sequence, for example  $\log n$ ,  $\log \log n$ , etc. Hence  $\mathcal{A}_{j'} \cap A$  denotes the collection of subcells with sidelength  $2^{-j'} \in [2^{-j}/s_n, 2^{-j+1}/s_n)$  within the cell  $A$ . The vernier focuses on cells  $A$  at resolution  $j$  that intersect the boundary (have smallest density deviation from the desired level  $\gamma$ ), and then evaluates the deviation in average density within subcells of  $A$  to judge whether or not the density is uniformly close to  $\gamma$  over the cell. Thus, the vernier is sensitive to the local density regularity in the vicinity of the desired level and in fact minimizing the vernier leads to selection of the appropriate resolution adapted to the unknown density regularity parameter  $\alpha$ . By choosing  $s_n$  with arbitrarily slow divergence, it is possible to get arbitrarily close to the optimal rate of convergence in the Hausdorff sense. However, note that the vernier may not function properly if the boundary of  $G_\gamma^*$  passes through every subcell of  $A$  (since then the subcell averages may be arbitrarily close to  $\gamma$  irrespective of the density regularity). Assumption [C] precludes this possibility at sufficiently high resolutions.

Since  $\mathcal{V}_{\gamma,j}$  requires knowledge of the unknown probability measure, we must work with the empirical version, defined analogously as:

$$\widehat{\mathcal{V}}_{\gamma,j} = \min_{A \in \mathcal{A}_j} \max_{A' \in \mathcal{A}_{j'} \cap A} |\gamma - \widehat{f}(A')|.$$

We propose a complexity regularization scheme wherein the empirical vernier  $\widehat{\mathcal{V}}_{\gamma,j}$  is balanced by a penalty term:

$$\Psi_{j'} := \max_{A \in \mathcal{A}_{j'}} \sqrt{8 \frac{\log(2^{j'(d+1)} 16)}{\delta} \max \left( \widehat{f}(A), 8 \frac{\log(2^{j'(d+1)} 16)}{\delta} \right) n \mu(A)}$$

where  $0 < \delta < 1$  is a confidence parameter, and  $\mu(A) = 2^{-j'd}$ . Notice that the penalty is computable from the given observations. The precise form of  $\Psi$  is chosen so that minimizing the empirical vernier plus penalty provides control over the true vernier (refer to Section 5.2 for a formal proof). The final level set estimate is given by

$$\widehat{G} = \widehat{G}_{\widehat{j}} \quad (2)$$

where

$$\hat{j} = \arg \min_{0 \leq j \leq J} \left\{ \widehat{\mathcal{V}}_{\gamma, j} + \Psi_{j'} \right\} \quad (3)$$

Thus the search is focused on regular partitions of dyadic sidelength  $2^{-j}$ ,  $j \in \{0, 1, \dots, J\}$ . The choice of  $J$  will be specified below. Observe that the value of the empirical vernier decreases with increasing resolution as better approximations to the true level are available. On the other hand, the penalty is designed to increase with resolution to penalize high complexity estimates that might overfit the given sample of data. Thus, the above procedure chooses the appropriate resolution automatically by balancing these two terms.

We now establish that our complexity penalized procedure leads to minimax optimal rates of convergence without requiring prior knowledge of any parameters.

**Theorem 2** Pick  $J \equiv J(n)$  such that  $2^{-J} \asymp s_n(n/\log n)^{-\frac{1}{d}}$ , where  $s_n$  is a monotone diverging sequence. Let  $\hat{j}$  denote the resolution chosen by the complexity penalized method as given by Eq. (3), and  $\widehat{G}$  denote the final estimate of Eq. (2). Then with probability at least  $1 - 3/n$ , for all densities in the class  $\mathcal{F}_2^*(\alpha)$ ,

$$c_1 s_n^{\frac{d}{d+2\alpha}} \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}} \leq 2^{-\hat{j}} \leq c_2 s_n^{\frac{d}{d+2\alpha}} \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}}$$

for  $n$  large enough (so that  $s_n > c(C_4, d)$ ), where  $c_1, c_2 > 0$  are constants. In addition,

$$\sup_{f \in \mathcal{F}_2^*(\alpha)} \mathbb{E}[d_\infty(\widehat{G}, G_\gamma^*)] \leq C s_n^2 \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}}$$

for all  $n$ , where  $C \equiv C(C_1, C_2, C_3, C_4, \epsilon_0, f_{\max}, \delta_0, \delta_1, d, \alpha) > 0$  is a constant.

The proof is given in Section 5.2.

The maximum resolution  $2^J \asymp s_n^{-1}(n/\log n)^{\frac{1}{d}}$  can be easily chosen, based only on  $n$ , and allows the optimal resolution for any  $\alpha$  to lie in the search space. Observe that by appropriate choice of  $s_n$ , for example  $s_n = (\log n)^{\epsilon/2}$  with  $\epsilon$  a small number  $> 0$ , the bound of Theorem 2 matches the minimax lower bound of Proposition 1, except for an additional  $(\log n)^\epsilon$  factor. Hence our method *adaptively* achieves near minimax optimal rates of convergence for the class  $\mathcal{F}_2^*(\alpha)$ .

## 4 Concluding Remarks

In this paper, we propose a Hausdorff accurate level set estimation method that is adaptive to unknown local density regularity and achieves minimax optimal rates of error convergence over a very general classes of level sets. The analysis in this paper assumes  $\alpha > 0$ , however the case  $\alpha \geq 0$  that allows jumps in the density can also be handled (see [SSN07]), but is omitted here to keep the presentation and proofs simpler. Also, this paper considers locally Lipschitz boundaries, however extensions to additional boundary smoothness (for example, Hölder regularity  $> 1$ ) may be possible in the proposed framework using techniques such as wedgelets [Don99] or curvelets [CD99]. The earlier work on Hausdorff accurate level set estimation [KT93, Tsy97, Cav97] does address higher smoothness of the boundary but that follows as a

straightforward consequence of assuming a functional form for the boundary. We would also like to comment that while we only addressed the density level set problem in this paper, extensions to general regression level set estimation should be possible using a similar approach.

The complexity regularization approach (Eq. 3) based on the vernier is similar in spirit to the so-called Lepski methods (for example, [LMS97]) for function estimation which are spatially adaptive bandwidth selectors, however the vernier focuses on cells close to the desired level and thus is specifically tailored to the level set problem. The vernier provides the key to achieve adaptivity while requiring only local regularity of the density in the vicinity of the desired level.

In this paper, we assume that the density regularity is the same everywhere along the level set. This might be somewhat restrictive, particularly if the level set consists of multiple components. Adaptivity to spatial variations in the density regularity can be achieved using a spatially adapted partition instead of a regular histogram partition. This might be possible by developing a tree-based approach or a modified Lepski method, and is the subject of current research.

## 5 Proofs

Before proceeding to the proofs, we establish two lemmas that are used throughout. The first one establishes a bound on the deviation of true and empirical density averages.

**Lemma 1** Consider  $0 < \delta < 1$ . With probability at least  $1 - \delta$ , the following is true for all dyadic resolutions  $j$ :

$$\max_{A \in \mathcal{A}_j} |\bar{f}(A) - \widehat{f}(A)| \leq \Psi_j.$$

**Proof:** The proof relies on a pair of VC inequalities (See [DL01] Chapter 3) that bound the *relative* deviation of true and empirical probabilities. For the collection  $\mathcal{A}_j$  with shatter coefficient bounded by  $2^{jd}$ , the relative VC inequalities state that for any  $\epsilon > 0$

$$P \left( \sup_{A \in \mathcal{A}_j} \frac{P(A) - \widehat{P}(A)}{\sqrt{P(A)}} > \epsilon \right) \leq 4 \cdot 2^{jd} e^{-n\epsilon^2/4}$$

and

$$P \left( \sup_{A \in \mathcal{A}_j} \frac{\widehat{P}(A) - P(A)}{\sqrt{\widehat{P}(A)}} > \epsilon \right) \leq 4 \cdot 2^{jd} e^{-n\epsilon^2/4}.$$

Also observe that

$$\widehat{P}(A) \leq P(A) + \epsilon \sqrt{\widehat{P}(A)} \implies \widehat{P}(A) \leq 2 \max(P(A), 2\epsilon^2)$$

$$P(A) \leq \widehat{P}(A) + \epsilon \sqrt{P(A)} \implies P(A) \leq 2 \max(\widehat{P}(A), 2\epsilon^2).$$

To see the first statement, consider 1)  $\widehat{P}(A) \leq 4\epsilon^2$  - The statement is obvious. 2)  $\widehat{P}(A) > 4\epsilon^2$  - This gives a bound on  $\epsilon$ , which implies  $\widehat{P}(A) \leq P(A) + \widehat{P}(A)/2$  and hence  $\widehat{P}(A) \leq 2P(A)$ . The second statement follows similarly.

Using these statements and the relative VC inequalities for the collection  $\mathcal{A}_j$ , we have: With probability  $> 1 - 8 \cdot 2^{jd} e^{-n\epsilon^2/4}$ ,  $\forall A \in \mathcal{A}_j$  both

$$P(A) - \widehat{P}(A) \leq \epsilon \sqrt{P(A)} \leq \epsilon \sqrt{2 \max(\widehat{P}(A), 2\epsilon^2)}$$

and

$$\widehat{P}(A) - P(A) \leq \epsilon \sqrt{\widehat{P}(A)} \leq \epsilon \sqrt{2 \max(\widehat{P}(A), 2\epsilon^2)}$$

Setting  $\epsilon = \sqrt{4 \log(2^{jd} 8 / \delta_j) / n}$ , we have with probability  $> 1 - \delta_j, \forall A \in \mathcal{A}_j$

$$\begin{aligned} |P(A) - \widehat{P}(A)| \\ \leq \sqrt{8 \frac{\log(2^{jd} 8 / \delta_j)}{n} \max\left(\widehat{P}(A), 8 \frac{\log(2^{jd} 8 / \delta_j)}{n}\right)} \end{aligned}$$

The result follows by dividing the result by  $\mu(A)$ , setting  $\delta_j = \delta 2^{-(j+1)}$  and taking union bound. ■

The next lemma states how the density deviation bound or penalty  $\Psi_j$  scales with resolution. This will be used to derive rates of convergence.

**Lemma 2** For all resolutions such that  $2^j = O((n/\log n)^{\frac{1}{\alpha}})$ , there exist constants  $c_3, c_4 \equiv c_4(f_{\max}, d) > 0$  such that for all  $n$ , with probability at least  $1 - 1/n$ ,

$$c_3 \sqrt{2^{jd} \frac{\log n}{n}} \leq \Psi_j \leq c_4 \sqrt{2^{jd} \frac{\log n}{n}}$$

**Proof:** The lower bound follows by observing that

$$1 = \sum_{A \in \mathcal{A}_j} \widehat{P}(A) \leq \max_{A \in \mathcal{A}_j} \widehat{P}(A) \times |\mathcal{A}_j| = \max_{A \in \mathcal{A}_j} \frac{\widehat{P}(A)}{\mu(A)} = \max_{A \in \mathcal{A}_j} \widehat{f}(A)$$

and using  $\delta = 1/n, j \geq 0$  and  $\mu(A) = 2^{-jd}$ .

To get an upper bound, using the same arguments as in proof of Lemma 1 based on the relative VC inequality it follows that [SSN07] with probability  $> 1 - \delta_j$ , for all  $A \in \mathcal{A}_j$

$$\widehat{P}(A) \leq 2 \max\left(P(A), 8 \frac{\log(2^{jd} 8 / \delta_j)}{n}\right).$$

Dividing by  $\mu(A) = 2^{-jd}$ , using density bound  $f_{\max}$ , setting  $\delta_j = \delta 2^{-(j+1)}$  and taking union bound, we have with probability  $> 1 - \delta$ , for all dyadic resolutions  $j$

$$\max_{A \in \mathcal{A}_j} \widehat{f}(A) \leq 2 \max\left(f_{\max}, 2^{jd} 8 \frac{\log(2^{j(d+1)} 16 / \delta)}{n}\right).$$

This implies the upper bound using  $\delta = 1/n$  and  $2^j = O((n/\log n)^{1/d})$ . ■

### 5.1 Proof of Theorem 1

The proof relies on the following lemma that will also be used in the proof of Theorem 2.

**Lemma 3** Consider densities satisfying assumptions [A] and [B]. Then for all resolutions such that  $2^j = O(s_n^{-1} (n/\log n)^{\frac{1}{\alpha}})$ , where  $s_n$  is a monotone diverging sequence, and  $n \geq n_0 \equiv n_0(f_{\max}, d, \delta_0, \epsilon_o, C_1, \alpha)$  with probability at least  $1 - 3/n$

$$d_\infty(\widehat{G}_j, G_\gamma^*) \leq \max(2C_3 + 3, 8\sqrt{d}\epsilon_o^{-1}) \left[ \left( \frac{\Psi_j}{C_1} \right)^{\frac{1}{\alpha}} + \sqrt{d} 2^{-j} \right].$$

**Proof:** Let  $J_0 = \lceil \log_2 4\sqrt{d}/\epsilon_o \rceil$ , where  $\epsilon_o$  is as defined in assumption [B]. Also define

$$\epsilon_j := \left[ \left( \frac{\Psi_j}{C_1} \right)^{\frac{1}{\alpha}} + \sqrt{d} 2^{-j} \right].$$

Consider two cases:

I.  $j < J_0$ .

For this case, since the domain  $\mathcal{X} = [0, 1]^d$ , we use the trivial bound

$$d_\infty(\widehat{G}_j, G_\gamma^*) \leq \sqrt{d} \leq 2^{J_0} (\sqrt{d} 2^{-j}) \leq 8\sqrt{d}\epsilon_o^{-1}\epsilon_j.$$

The last step follows by choice of  $J_0$  and since  $\Psi_j, C_1 > 0$ .

II.  $j \geq J_0$ .

Observe that assumption [B] implies that  $G_\gamma^*$  is not empty since  $G_\gamma^* \supseteq \mathcal{I}_\epsilon(G_\gamma^*) \neq \emptyset$  for  $\epsilon \leq \epsilon_o$ . We will show that for large enough  $n$ , with high probability,  $\widehat{G}_j \cap G_\gamma^* \neq \emptyset$  for  $j \geq J_0$  and hence  $\widehat{G}_j$  is not empty. Thus the Hausdorff error is given as

$$d_\infty(\widehat{G}_j, G_\gamma^*) = \max\left\{ \sup_{x \in \widehat{G}_j} \rho(x, \widehat{G}_j), \sup_{x \in G_\gamma^*} \rho(x, G_\gamma^*) \right\}, \quad (4)$$

and we need bounds on the two terms in the right hand side.

We now prove that  $\widehat{G}_j$  is not empty and obtain bounds on the two terms in the Hausdorff error. Towards this end, we establish two propositions. The first proposition proves that for large enough  $n$ , with high probability, the distance of all points that are erroneously excluded or included in the level set estimate, from the true set boundary is bounded by  $\epsilon_j$ . Notice that, if  $\widehat{G}_j$  is non-empty, this provides an upper bound on the second term of the Hausdorff error (Eq. 4). The second proposition establishes that, for large enough  $n$  and  $j \geq J_0$ ,  $2\epsilon_j \leq \epsilon_o$  and hence the inner cover  $\mathcal{I}_{2\epsilon_j}(G_\gamma^*)$  is not empty. And using the first proposition, with high probability,  $\mathcal{I}_{2\epsilon_j}(G_\gamma^*)$  contains points that are correctly included in the level set estimate and lie in  $\widehat{G}_j \cap G_\gamma^*$ . Thus  $\widehat{G}_j$  is not empty. Further, along with assumption [B], this provides a bound of  $\epsilon_j$  on the distance of any point in  $G_\gamma^*$  from the estimate  $\widehat{G}_j$ , thus bounding the first term of the Hausdorff error (Eq. 4).

We end the proof of the two propositions with a white box □ to indicate that these propositions are included within the proof of Lemma 3, and do not signify end of the proof of Lemma 3.

**Proposition 2** If  $\widehat{G}_j \Delta G_\gamma^* \neq \emptyset$ , then for resolutions satisfying  $2^j = O(s_n^{-1} (n/\log n)^{1/d})$  and  $n \geq n_1(f_{\max}, d, \delta_0)$  with probability at least  $1 - 2/n$

$$\sup_{x \in \widehat{G}_j \Delta G_\gamma^*} \rho(x, \partial G_\gamma^*) \leq \left( \frac{\Psi_j}{C_1} \right)^{1/\alpha} + \sqrt{d} 2^{-j} = \epsilon_j.$$

**Proof:** Since by assumption  $\widehat{G}_j \Delta G_\gamma^* \neq \emptyset$ , consider  $x \in \widehat{G}_j \Delta G_\gamma^*$ . Let  $A_x \in \mathcal{A}_j$  denote the cell containing  $x$  at resolution  $j$ . Consider two cases:

(i)  $A_x \cap \partial G_\gamma^* \neq \emptyset$ . This implies that

$$\rho(x, \partial G_\gamma^*) \leq \sqrt{d}2^{-j}.$$

(ii)  $A_x \cap \partial G_\gamma^* = \emptyset$ . Since  $x \in \widehat{G}_j \Delta G_\gamma^*$ , it is erroneously included or excluded from the level set estimate  $\widehat{G}_j$ . Therefore, if  $\bar{f}(A_x) \geq \gamma$ , then  $\widehat{f}(A_x) < \gamma$  otherwise if  $\bar{f}(A_x) < \gamma$ , then  $\widehat{f}(A_x) \geq \gamma$ . This implies that  $|\gamma - \bar{f}(A_x)| \leq |\bar{f}(A_x) - \widehat{f}(A_x)|$ . Using Lemma 1, we get  $|\gamma - \bar{f}(A_x)| \leq \Psi_j$  with probability at least  $1 - \delta$ .

Now let  $x_0$  be any point in  $A_x$  such that  $|\gamma - f(x_0)| \leq |\gamma - \bar{f}(A_x)|$  (Notice that at least one such point must exist in  $A_x$  since this cell does not intersect the boundary). As argued above,  $|\gamma - \bar{f}(A_x)| \leq \Psi_j$  with probability at least  $1 - 1/n$  (for  $\delta = 1/n$ ) and using Lemma 2,  $\Psi_j$  decreases with  $n$  for resolutions satisfying  $2^j = O(s_n^{-1}(n/\log n)^{1/d})$  with probability at least  $1 - 1/n$ . So for large enough  $n \geq n_1(f_{\max}, d, \delta_0)$ ,  $\Psi_j \leq \delta_0$  and hence  $|\gamma - f(x_0)| \leq \delta_0$ . Thus, the density regularity assumption **[A]** holds at  $x_0$  with probability  $> 1 - 2/n$  and we have

$$\begin{aligned} \rho(x_0, \partial G_\gamma^*) &\leq \left( \frac{|\gamma - f(x_0)|}{C_1} \right)^{\frac{1}{\alpha}} \\ &\leq \left( \frac{|\gamma - \bar{f}(A_x)|}{C_1} \right)^{\frac{1}{\alpha}} \leq \left( \frac{\Psi_j}{C_1} \right)^{\frac{1}{\alpha}}. \end{aligned}$$

Since  $x, x_0 \in A_x$ ,  $\rho(x, \partial G_\gamma^*) \leq \rho(x_0, \partial G_\gamma^*) + \sqrt{d}2^{-j}$ . Therefore,

$$\rho(x, \partial G_\gamma^*) \leq \left( \frac{\Psi_j}{C_1} \right)^{1/\alpha} + \sqrt{d}2^{-j}.$$

So for both cases, we can say that for resolutions satisfying  $2^j = O(s_n^{-1}(n/\log n)^{1/d})$  and  $n \geq n_1(f_{\max}, d, \delta_0)$  with probability at least  $1 - 2/n$ ,  $\forall x \in \widehat{G}_j \Delta G_\gamma^*$

$$\rho(x, \partial G_\gamma^*) \leq \left( \frac{\Psi_j}{C_1} \right)^{1/\alpha} + \sqrt{d}2^{-j} = \epsilon_j. \quad \square$$

**Proposition 3** Recall assumption **[B]** and denote the inner cover of  $G_\gamma^*$  with  $2\epsilon_j$ -balls,  $\mathcal{I}_{2\epsilon_j}(G_\gamma^*) \equiv \mathcal{I}_{2\epsilon_j}$ . For resolutions satisfying  $2^j = O(s_n^{-1}(n/\log n)^{1/d})$ ,  $j \geq J_0$  and  $n \geq n_0 \equiv n_0(f_{\max}, d, \delta_0, \epsilon_o, C_1, \alpha)$ , with probability at least  $1 - 3/n$ ,

$$\widehat{G}_j \neq \emptyset \quad \text{and} \quad \sup_{x \in \mathcal{I}_{2\epsilon_j}} \rho(x, \widehat{G}_j \cap G_\gamma^*) \leq \epsilon_j.$$

**Proof:** Observe that for  $j \geq J_0$ ,  $2\sqrt{d}2^{-j} \leq 2\sqrt{d}2^{-J_0} \leq \epsilon_o/2$ . And using Lemma 2 for large enough  $n \geq n_2 \equiv n_2(\epsilon_o, f_{\max}, C_1, \alpha)$ ,

$2(\Psi_j/C_1)^{1/\alpha} \leq \epsilon_o/2$  with probability at least  $1 - 1/n$ . Therefore for all  $j \geq J_0$  and  $n \geq n_2$ ,  $2\epsilon_j \leq \epsilon_o$  with probability at least  $1 - 1/n$  and hence  $\mathcal{I}_{2\epsilon_j} \neq \emptyset$ . Now consider any  $2\epsilon_j$ -ball in  $\mathcal{I}_{2\epsilon_j}$ . Then the distance of all points in the interior of the concentric  $\epsilon_j$ -ball from the boundary of  $\mathcal{I}_{2\epsilon_j}$ , and hence from the boundary of  $G_\gamma^*$  is greater than  $\epsilon_j$ . As per Proposition 2 for  $n \geq n_0 = \max(n_1, n_2)$ , with probability  $> 1 - 3/n$ , none of these points can lie in  $\widehat{G}_j \Delta G_\gamma^*$ , and hence must lie in  $\widehat{G}_j \cap G_\gamma^*$  since they are in  $\mathcal{I}_{2\epsilon_j} \subseteq G_\gamma^*$ . Therefore,

$$\widehat{G}_j \neq \emptyset \quad \text{and} \quad \sup_{x \in \mathcal{I}_{2\epsilon_j}} \rho(x, \widehat{G}_j \cap G_\gamma^*) \leq \epsilon_j. \quad \square$$

Now since  $G_\gamma^*$  and  $\widehat{G}_j$  are non-empty sets, we bound the two terms that contribute to the Hausdorff error

$$\sup_{x \in G_\gamma^*} \rho(x, \widehat{G}_j) \quad \text{and} \quad \sup_{x \in \widehat{G}_j} \rho(x, G_\gamma^*).$$

For this, we will use Propositions 2 and 3, hence all the following statements will hold for resolutions satisfying  $2^j = O(s_n^{-1}(n/\log n)^{1/d})$ ,  $j \geq J_0$  and  $n \geq n_0 \equiv n_0(f_{\max}, d, \delta_0, \epsilon_o, C_1, \alpha)$ , with probability at least  $1 - 3/n$ .

To bound the second term, observe that

- (i) If  $\widehat{G}_j \setminus G_\gamma^* = \emptyset$ , then  $\sup_{x \in \widehat{G}_j} \rho(x, G_\gamma^*) = 0$ .
- (ii) If  $\widehat{G}_j \setminus G_\gamma^* \neq \emptyset$ , it implies that  $\widehat{G}_j \Delta G_\gamma^* \neq \emptyset$ . Hence, using Proposition 2, we have

$$\begin{aligned} \sup_{x \in \widehat{G}_j} \rho(x, G_\gamma^*) &= \sup_{x \in \widehat{G}_j \setminus G_\gamma^*} \rho(x, G_\gamma^*) \\ &= \sup_{x \in \widehat{G}_j \setminus G_\gamma^*} \rho(x, \partial G_\gamma^*) \\ &\leq \sup_{x \in \widehat{G}_j \Delta G_\gamma^*} \rho(x, \partial G_\gamma^*) \leq \epsilon_j. \end{aligned}$$

Thus, for either case

$$\sup_{x \in \widehat{G}_j} \rho(x, G_\gamma^*) \leq \epsilon_j. \quad (5)$$

To bound the first term, observe that

- (i) If  $G_\gamma^* \setminus \widehat{G}_j = \emptyset$ , then  $\sup_{x \in G_\gamma^*} \rho(x, \widehat{G}_j) = 0$ .
- (ii) If  $G_\gamma^* \setminus \widehat{G}_j \neq \emptyset$ , we proceed as follows:

$$\begin{aligned} \sup_{x \in G_\gamma^*} \rho(x, \widehat{G}_j) &\leq \sup_{x \in G_\gamma^*} \rho(x, \widehat{G}_j \cap G_\gamma^*) \\ &= \max\left\{ \sup_{x \in \mathcal{I}_{2\epsilon_j}} \rho(x, \widehat{G}_j \cap G_\gamma^*), \right. \\ &\quad \left. \sup_{x \in G_\gamma^* \setminus \mathcal{I}_{2\epsilon_j}} \rho(x, \widehat{G}_j \cap G_\gamma^*) \right\} \\ &\leq \max\left\{ \epsilon_j, \sup_{x \in G_\gamma^* \setminus \mathcal{I}_{2\epsilon_j}} \rho(x, \widehat{G}_j \cap G_\gamma^*) \right\}. \end{aligned}$$

The last step follows using Proposition 3.

Now consider any  $x \in G_\gamma^* \setminus \mathcal{I}_{2\epsilon_j}$ . Then using triangle inequality,  $\forall y \in \partial G_\gamma^*$  and  $\forall z \in \mathcal{I}_{2\epsilon_j}$ ,

$$\begin{aligned} \rho(x, \widehat{G}_j \cap G_\gamma^*) &\leq \rho(x, y) + \rho(y, z) + \rho(z, \widehat{G}_j \cap G_\gamma^*) \\ &\leq \rho(x, y) + \rho(y, z) + \\ &\quad \sup_{z \in \mathcal{I}_{2\epsilon_j}} \rho(z, \widehat{G}_j \cap G_\gamma^*) \\ &\leq \rho(x, y) + \rho(y, z) + \epsilon_j. \end{aligned}$$

The last step follows using Proposition 3. This implies that  $\forall y \in \partial G_\gamma^*$ ,

$$\begin{aligned} \rho(x, \widehat{G}_j \cap G_\gamma^*) &\leq \rho(x, y) + \inf_{z \in \mathcal{I}_{2\epsilon_j}} \rho(y, z) + \epsilon_j \\ &= \rho(x, y) + \rho(y, \mathcal{I}_{2\epsilon_j}) + \epsilon_j \\ &\leq \rho(x, y) + \sup_{y \in \partial G_\gamma^*} \rho(y, \mathcal{I}_{2\epsilon_j}) + \epsilon_j \\ &\leq \rho(x, y) + 2C_3\epsilon_j + \epsilon_j. \end{aligned}$$

Here the last step invokes assumption **[B]**. This in turn implies that

$$\begin{aligned} \rho(x, \widehat{G}_j \cap G_\gamma^*) &\leq \inf_{y \in \partial G_\gamma^*} \rho(x, y) + (2C_3 + 1)\epsilon_j \\ &\leq 2\epsilon_j + (2C_3 + 1)\epsilon_j. \end{aligned}$$

The second step is true for  $x \in G_\gamma^* \setminus \mathcal{I}_{2\epsilon_j}$ . If it was not true, then  $\forall y \in \partial G_\gamma^*$ ,  $\rho(x, y) > 2\epsilon_j$  and hence there exists a closed  $2\epsilon_j$ -ball around  $x$  that is in  $G_\gamma^*$ . This contradicts the fact that  $x \notin \mathcal{I}_{2\epsilon_j}$ . Therefore, we have:

$$\sup_{x \in G_\gamma^* \setminus \mathcal{I}_{2\epsilon_j}} \rho(x, \widehat{G}_j \cap G_\gamma^*) \leq (2C_3 + 3)\epsilon_j.$$

And going back to the start of case (ii) we get:

$$\sup_{x \in G_\gamma^*} \rho(x, \widehat{G}_j) \leq (2C_3 + 3)\epsilon_j.$$

Therefore, for either case we have

$$\sup_{x \in G_\gamma^*} \rho(x, \widehat{G}_j) \leq (2C_3 + 3)\epsilon_j. \quad (6)$$

From Eq. (5) and (6), we have that for all densities satisfying assumptions **[A, B]**, for resolutions satisfying  $2^j = O(s_n^{-1}(n/\log n)^{1/d})$ ,  $j \geq J_0$  and  $n \geq n_0 \equiv n_0(f_{\max}, d, \delta_0, \epsilon_o, C_1, \alpha)$ , with probability  $> 1 - 3/n$ ,

$$\begin{aligned} d_\infty(\widehat{G}_j, G_\gamma^*) &= \max\left\{ \sup_{x \in G_\gamma^*} \rho(x, \widehat{G}_j), \sup_{x \in \widehat{G}_j} \rho(x, G_\gamma^*) \right\} \\ &\leq (2C_3 + 3)\epsilon_j. \end{aligned}$$

And addressing both Case I ( $j < J_0$ ) and Case II ( $j \geq J_0$ ), we finally have that for all densities satisfying assumptions **[A, B]**, for resolutions satisfying  $2^j = O(s_n^{-1}(n/\log n)^{\frac{1}{d}})$  and  $n \geq n_0 \equiv n_0(f_{\max}, d, \delta_0, \epsilon_o, C_1, \alpha)$ , with probability  $> 1 - 3/n$ ,

$$d_\infty(\widehat{G}_j, G_\gamma^*) \leq \max(2C_3 + 3, 8\sqrt{d}\epsilon_o^{-1})\epsilon_j. \quad \blacksquare$$

Since the chosen resolution  $2^{-j} \asymp s_n(n/\log n)^{-\frac{1}{(d+2\alpha)}}$  satisfies conditions of Lemma 3, proof of Theorem 1 now follows using the bound on  $\Psi_j$  from Lemma 2. Let  $\Omega$  denote the event such that the bounds of Lemma 2 and Lemma 3 hold. Then for  $n \geq n_0$ ,  $P(\bar{\Omega}) \leq 4/n$ . Hence for all  $n$ ,  $P(\bar{\Omega}) \leq \max(4, n_0)/n$ . So  $\forall f \in \mathcal{F}_1^*(\alpha)$ : (Here  $C$  may denote a different constant from line to line. Explanation for each step is provided after the equations.)

$$\begin{aligned} \mathbb{E}[d_\infty(\widehat{G}_j, G_\gamma^*)] &= P(\Omega)\mathbb{E}[d_\infty(\widehat{G}_j, G_\gamma^*)|\Omega] + P(\bar{\Omega})\mathbb{E}[d_\infty(\widehat{G}_j, G_\gamma^*)|\bar{\Omega}] \\ &\leq \mathbb{E}[d_\infty(\widehat{G}_j, G_\gamma^*)|\Omega] + P(\bar{\Omega})\sqrt{d} \\ &\leq C \left[ \left( \frac{\Psi_j}{C_1} \right)^{1/\alpha} + \sqrt{d}2^{-j} + \frac{\sqrt{d}}{n} \right] \\ &\leq C \max \left\{ \left( 2^{jd} \frac{\log n}{n} \right)^{\frac{1}{2\alpha}}, 2^{-j}, \frac{1}{n} \right\} \\ &\leq C(C_1, C_3, \epsilon_o, f_{\max}, \delta_0, d, \alpha) s_n \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}}. \end{aligned}$$

The second step follows using the trivial bounds  $P(\Omega) \leq 1$  and since the domain  $\mathcal{X} = [0, 1]^d$ ,  $\mathbb{E}[d_\infty(\widehat{G}_j, G_\gamma^*)|\bar{\Omega}] \leq \sqrt{d}$ . The third step follows from Lemma 3 and the fourth one using Lemma 2. The last step follows since the chosen resolution  $2^{-j} \asymp s_n(n/\log n)^{-\frac{1}{(d+2\alpha)}}$ .

## 5.2 Proof of Theorem 2

To analyze the resolution chosen by the complexity penalized procedure of Eq. (3) based on the vernier, we first establish two results regarding the vernier. Using Lemma 1, we have the following corollary that bounds the deviation of true and empirical vernier.

**Corollary 1** Consider  $0 < \delta < 1$ . With probability at least  $1 - \delta$ , the following is true for all dyadic resolutions  $j$ :

$$|\mathcal{V}_{\gamma, j} - \widehat{\mathcal{V}}_{\gamma, j}| \leq \Psi_{j'}.$$

**Proof:** Let  $A_0 \in \mathcal{A}_j$  denote the cell achieving the min defining  $\mathcal{V}_{\gamma, j}$  and  $A_1 \in \mathcal{A}_j$  denote the cell achieving the min defining  $\widehat{\mathcal{V}}_{\gamma, j}$ . Also let  $A'_0$  and  $A'_1$  denote the subcells at resolution  $j'$  within  $A_0$  and  $A_1$ , respectively, that have maximum average density deviation from  $\gamma$ . Similarly, let  $\widehat{A}'_0$  and  $\widehat{A}'_1$  denote the subcells at resolution  $j'$  within  $A_0$  and  $A_1$ , respectively, that have maximum empirical density deviation from  $\gamma$ . Then we have: (Explanation for the steps are given after the equations.)

$$\begin{aligned} \mathcal{V}_{\gamma, j} - \widehat{\mathcal{V}}_{\gamma, j} &= |\gamma - \bar{f}(A'_0)| - |\gamma - \widehat{f}(\widehat{A}'_1)| \\ &\leq |\gamma - \bar{f}(A'_1)| - |\gamma - \widehat{f}(\widehat{A}'_1)| \\ &\leq |\bar{f}(A'_1) - \widehat{f}(\widehat{A}'_1)| \\ &= \max\{\bar{f}(A'_1) - \widehat{f}(\widehat{A}'_1), \widehat{f}(\widehat{A}'_1) - \bar{f}(A'_1)\} \\ &\leq \max\{\bar{f}(A'_1) - \widehat{f}(\widehat{A}'_1), \widehat{f}(\widehat{A}'_1) - \bar{f}(\widehat{A}'_1)\} \\ &\leq \max_{A \in \mathcal{A}_{j'}} |\bar{f}(A) - \widehat{f}(A)| \\ &\leq \Psi_{j'} \end{aligned}$$

The first inequality invokes definition of  $A_0$ , the third inequality invokes definitions of the subcells  $A'_1, \widehat{A}'_1$ , and the last one follows from Lemma 1. Similarly,

$$\begin{aligned}\widehat{\mathcal{V}}_{\gamma,j} - \mathcal{V}_{\gamma,j} &= |\gamma - \widehat{f}(\widehat{A}'_1)| - |\gamma - \bar{f}(A'_0)| \\ &\leq |\gamma - \widehat{f}(\widehat{A}'_0)| - |\gamma - \bar{f}(A'_0)| \\ &\leq |\bar{f}(A'_0) - \widehat{f}(\widehat{A}'_0)|\end{aligned}$$

Here the first inequality invokes definition of  $A_1$ . The rest follows as above, considering cell  $A_0$  instead of  $A_1$ . ■

The second result establishes that the vernier is sensitive to the resolution and density regularity.

**Lemma 4** Consider densities satisfying assumptions [A] and [C]. Recall that  $j' = \lfloor j + \log_2 s_n \rfloor$ , where  $s_n$  is a monotone diverging sequence. Then for all dyadic resolutions  $j$

$$\min(\delta_0, C_1)2^{-j'\alpha} \leq \mathcal{V}_{\gamma,j} \leq C(\sqrt{d}2^{-j})^\alpha$$

holds for  $n$  large enough such that  $s_n > 4C_46^d$ . Here  $C \equiv C(C_2, f_{\max}, \delta_1, \alpha) > 0$ .

**Proof:** We first establish the upper bound. Recall assumption [A] and consider the cell  $A \in \mathcal{A}_j$  that contains the point  $y_0$ . Then  $A \cap \partial G_\gamma^* \neq \emptyset$ . Let  $A'$  denote the subcell at resolution  $j'$  within  $A$  that has maximum average density deviation from  $\gamma$ . Consider two cases:

- (i) If the resolution is large enough so that  $\sqrt{d}2^{-j} \leq \delta_1$ , then the density regularity assumption [A] holds  $\forall x \in A$  since  $A \subset B(y_0, \delta_1)$ , the  $\delta_1$ -ball around  $y_0$ . The same holds also for the subcell  $A'$ . Hence

$$|\gamma - \bar{f}(A')| \leq C_2(\sqrt{d}2^{-j})^\alpha$$

- (ii) If the resolution is not large enough and  $\sqrt{d}2^{-j} > \delta_1$ , the following trivial bound holds:

$$|\gamma - \bar{f}(A')| \leq f_{\max} \leq \frac{f_{\max}}{\delta_1^\alpha}(\sqrt{d}2^{-j})^\alpha$$

The last step holds since  $\sqrt{d}2^{-j} > \delta_1$ .

Hence we can say for all  $j$  there exists  $A \in \mathcal{A}_j$  such that

$$|\gamma - \bar{f}(A')| \leq \max\left(C_2, \frac{f_{\max}}{\delta_1^\alpha}\right)(\sqrt{d}2^{-j})^\alpha$$

This yields the upper bound on the vernier:

$$\mathcal{V}_{\gamma,j} \leq \max\left(C_2, \frac{f_{\max}}{\delta_1^\alpha}\right)(\sqrt{d}2^{-j})^\alpha := C(\sqrt{d}2^{-j})^\alpha$$

where  $C \equiv C(C_2, f_{\max}, \delta_1, \alpha)$ .

For the lower bound, consider a cell  $A \in \mathcal{A}_j$ . We will show that assumption [C] on the level set boundary dimension basically implies that the boundary does not intersect all subcells at resolution  $j'$  within the cell  $A$  at resolution  $j$ . And in fact for large enough  $n$  (so that  $2^{-j'}$  is small enough, recall that  $j' = \lfloor j + \log_2 s_n \rfloor$  where  $s_n$  is a monotone diverging sequence), there exists at least one subcell  $A'_o \in A \cap \mathcal{A}_{j'}$  such that  $\forall x \in A'_o$ ,

$$\rho(x, \partial G_\gamma^*) \geq 2^{-j'}.$$

We establish this statement formally later on, but for now assume that it holds. The local density regularity condition [A] now gives that for all  $x \in A'_o$ ,  $|\gamma - f(x)| \geq \min(\delta_0, C_1)2^{-j'\alpha}$ . So we have:

$$\max_{A' \in A \cap \mathcal{A}_{j'}} |\gamma - \bar{f}(A')| \geq |\gamma - \bar{f}(A'_o)| \geq \min(\delta_0, C_1)2^{-j'\alpha}$$

Since this is true for any  $A \in \mathcal{A}_j$ , in particular, this is true for the cell achieving the min defining  $\mathcal{V}_{\gamma,j}$ . Hence, the lower bound on the vernier  $\mathcal{V}_{\gamma,j}$  follows.

We now formally prove that assumption [C] on the level set boundary dimension implies that for large enough  $n$  (so that  $s_n > 4C_46^d$ ),  $\exists A'_o \in A \cap \mathcal{A}_{j'}$  s.t.  $\forall x \in A'_o$ ,

$$\rho(x, \partial G_\gamma^*) \geq 2^{-j'}.$$

Observe that it suffices to show that for large enough  $n$ ,  $\exists A'' \in A \cap \mathcal{A}_{j'-2}$  s.t.  $A'' \cap \partial G_\gamma^* = \emptyset$ . To prove this last statement, consider two cases:

- (i)  $A \cap \partial G_\gamma^* = \emptyset$ . For  $s_n \geq 8$ ,  $j' - 2 \geq j$  (recall definition of  $j'$ ), and since  $A$  does not intersect the boundary, clearly  $\exists A'' \in A \cap \mathcal{A}_{j'-2}$  s.t.  $A'' \cap \partial G_\gamma^* = \emptyset$ .

- (ii)  $A \cap \partial G_\gamma^* \neq \emptyset$ . Let  $x \in A \cap \partial G_\gamma^*$ . Consider  $\epsilon = \sqrt{d}2^{-j}$  (the diagonal length of a cell), then  $A \subseteq B(x, \epsilon)$ . Also let  $\delta = \sqrt{d}2^{-(j'-2)}/2$  (the choice will be justified below). For  $s_n \geq 4$ ,  $0 < \delta \leq \epsilon$  and using assumption [C], the minimum number of  $\delta$ -balls required to cover  $\partial G_\gamma^* \cap B(x, \epsilon)$  is  $\leq C_4(\delta/\epsilon)^{-(d-1)}$ . Since  $A \subseteq B(x, \epsilon)$ , the minimum number of  $\delta$ -balls required to cover  $\partial G_\gamma^* \cap A$  is also  $\leq C_4(\delta/\epsilon)^{-(d-1)}$ . Now consider a uniform partition of the cell  $A$  into subcells of sidelength  $2\delta/\sqrt{d} = 2^{-(j'-2)}$ . Since the diagonal length of a subcell  $\sqrt{d}2^{-(j'-2)} = 2\delta$ , this choice of  $\delta$  implies that a subcell at resolution  $2^{-(j'-2)}$  is inscribed within an aligned  $\delta$ -ball. Observe that at this resolution, in  $d$ -dim, an unaligned  $\delta$ -ball can intersect up to  $3^d - 1$  subcells (number of neighbors of any hypercube). Therefore, the number of subcells in  $A \cap \mathcal{A}_{j'-2}$  that intersect the boundary can be no more than

$$\begin{aligned}3^d C_4 (\delta/\epsilon)^{-(d-1)} &= 3^d C_4 \left( \frac{\sqrt{d}2^{-(j'-2)}}{2\sqrt{d}2^{-j}} \right)^{-(d-1)} \\ &= \frac{C_4 6^d}{2} 2^{(j'-2-j)d} 2^{-(j'-2-j)} \\ &< \frac{4C_4 6^d}{s_n} 2^{(j'-2-j)d}\end{aligned}$$

where the last step uses the fact  $2^{-j'} < 2^{-j+1}/s_n$ . For  $s_n > 4C_4 6^d$ , the number of subcells within  $A$  at resolution  $j' - 2$  that intersect the boundary is less than the total number of subcells within  $A$  at that resolution. Therefore,  $\exists A'' \in A \cap \mathcal{A}_{j'-2}$  s.t.  $A'' \cap \partial G_\gamma^* = \emptyset$ .

This in turn implies that for  $n$  large enough (so that  $s_n > 4C_4 6^d$ ),  $\exists A'_o \in A \cap \mathcal{A}_{j'}$  such that  $\forall x \in A'_o$ ,  $\rho(x, \partial G_\gamma^*) \geq 2^{-j'}$ . ■

We are now ready to prove Theorem 2. Observe that Lemmas 2, 3 and Corollary 1 hold together with probability at least  $1 - 5/n$  (taking  $\delta = 1/n$ ). Using these lemmas, we will show that for the resolution  $\hat{j}$  chosen by Eq. (3), both  $\mathcal{V}_{\gamma, \hat{j}}$  and  $\Psi_{\hat{j}}$  are upper bounded by  $C s_n^{\frac{d\alpha}{d+2\alpha}} (n/\log n)^{-\frac{\alpha}{d+2\alpha}}$ , where  $C \equiv C(C_2, f_{\max}, \delta_1, d, \alpha) > 0$ . If this holds, then using Lemma 4 and the definition of  $j'$ , we have the following upper bound on the sidelength: For  $s_n > 4C_4 6^d$

$$\begin{aligned} 2^{-\hat{j}} \leq s_n 2^{-\hat{j}'} &\leq s_n \left( \frac{\mathcal{V}_{\gamma, \hat{j}}}{\min(\delta_0, C_1)} \right)^{\frac{1}{\alpha}} \\ &\leq c_2 s_n s_n^{\frac{d}{d+2\alpha}} \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}}, \end{aligned}$$

where  $c_2 \equiv c_2(C_1, C_2, f_{\max}, \delta_0, \delta_1, d, \alpha) > 0$ . Also notice that since  $2^J \asymp s_n^{-1} (n/\log n)^{1/d}$ , we have  $2^{j'} \leq 2^{J'}$   $\leq s_n 2^J \asymp (n/\log n)^{1/d}$ , and hence Lemma 2 can be used to provide a lower bound on the sidelength:

$$\begin{aligned} 2^{-\hat{j}} > \frac{s_n}{2} 2^{-\hat{j}'} &\geq \frac{s_n}{2} \left( \frac{\Psi_{\hat{j}'}^2}{c_3^2 \log n} n \right)^{-\frac{1}{d}} \\ &\geq c_1 s_n \left( s_n^{\frac{2d\alpha}{d+2\alpha}} \left( \frac{n}{\log n} \right)^{-\frac{2\alpha}{d+2\alpha}} \frac{n}{\log n} \right)^{-\frac{1}{d}} \\ &= c_1 s_n^{\frac{d}{d+2\alpha}} \left( \frac{n}{\log n} \right)^{\frac{-1}{d+2\alpha}}, \end{aligned}$$

where  $c_1 \equiv c_1(C_2, f_{\max}, \delta_1, d, \alpha) > 0$ . So we have for  $s_n > 4C_4 6^d$ , with probability at least  $1 - 5/n$ ,

$$c_1 s_n^{\frac{d}{d+2\alpha}} \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}} \leq 2^{-\hat{j}} \leq c_2 s_n s_n^{\frac{d}{d+2\alpha}} \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}}. \quad (7)$$

Hence the automatically chosen resolution behaves as desired.

Let us now derive the claimed bounds on  $\mathcal{V}_{\gamma, \hat{j}}$  and  $\Psi_{\hat{j}}$ . Using Corollary 1 and Eq. (3), we have the following oracle inequality:

$$\begin{aligned} \mathcal{V}_{\gamma, \hat{j}} &\leq \hat{\mathcal{V}}_{\gamma, \hat{j}} + \Psi_{\hat{j}} \\ &= \min_{0 \leq j \leq J} \left\{ \hat{\mathcal{V}}_{\gamma, j} + \Psi_{j'} \right\} \leq \min_{0 \leq j \leq J} \left\{ \mathcal{V}_{\gamma, j} + 2\Psi_{j'} \right\} \end{aligned}$$

Lemma 4 provides an upper bound on the vernier  $\mathcal{V}_{\gamma, j}$ , and Lemma 2 provides an upper bound on the penalty  $\Psi_{j'}$ . We now plug these bounds into the oracle inequality. Here  $C$  may denote a different constant from line to line.

$$\begin{aligned} \mathcal{V}_{\gamma, \hat{j}} &\leq \hat{\mathcal{V}}_{\gamma, \hat{j}} + \Psi_{\hat{j}} \leq C \min_{0 \leq j \leq J} \left\{ 2^{-j\alpha} + \sqrt{2^{j'd} \frac{\log n}{n}} \right\} \\ &\leq C \min_{0 \leq j \leq J} \left\{ \max \left( 2^{-j\alpha}, \sqrt{2^{j'd} s_n^d \frac{\log n}{n}} \right) \right\} \\ &\leq C s_n^{\frac{d\alpha}{d+2\alpha}} \left( \frac{n}{\log n} \right)^{-\frac{\alpha}{d+2\alpha}}. \end{aligned}$$

Here  $C \equiv C(C_2, f_{\max}, \delta_1, d, \alpha)$ . The second step uses the definition of  $j'$  and the last step follows by balancing the two

terms for optimal resolution  $2^{-j^*} \asymp s_n^{\frac{d}{d+2\alpha}} \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}}$ . This establishes the desired bounds on  $\mathcal{V}_{\gamma, \hat{j}}$  and  $\Psi_{\hat{j}}$ .

Now we can invoke Lemma 3 to derive the rate of convergence for the Hausdorff error. Consider large enough  $n \geq n_1(C_4, d)$  so that  $s_n > 4C_4 6^d$ . Also, recall that the condition of Lemma 3 requires that  $n \geq n_0(f_{\max}, d, \delta_0, \epsilon_o, C_1, \alpha)$ . Pick  $n \geq \max(n_0, n_1)$  and let  $\Omega$  denote the event such that the bounds of Lemma 2, Lemma 3 and Corollary 1 hold with  $\delta = 1/n$ . Then, we have  $P(\bar{\Omega}) \leq 5/n$  for  $n \geq \max(n_0, n_1)$ , or for all  $n$ ,  $P(\bar{\Omega}) \leq \max(5, n_0, n_1)/n$ . So  $\forall f \in \mathcal{F}_2^*(\alpha)$ , we have: (Here  $C$  may denote a different constant from line to line. Explanation for each step is provided after the equations.)

$$\begin{aligned} \mathbb{E}[d_\infty(\hat{G}, G_\gamma^*)] &= P(\Omega) \mathbb{E}[d_\infty(\hat{G}, G_\gamma^*) | \Omega] + P(\bar{\Omega}) \mathbb{E}[d_\infty(\hat{G}, G_\gamma^*) | \bar{\Omega}] \\ &\leq \mathbb{E}[d_\infty(\hat{G}, G_\gamma^*) | \Omega] + P(\bar{\Omega}) \sqrt{d} \\ &\leq C \left[ \left( \frac{\Psi_{\hat{j}}}{C_1} \right)^{1/\alpha} + \sqrt{d} 2^{-\hat{j}} + \frac{\sqrt{d}}{n} \right] \\ &\leq C \max \left\{ \left( 2^{\hat{j}d} \frac{\log n}{n} \right)^{\frac{1}{2\alpha}}, 2^{-\hat{j}}, \frac{1}{n} \right\} \\ &\leq C s_n s_n^{\frac{d}{d+2\alpha}} \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}} \leq C s_n^2 \left( \frac{n}{\log n} \right)^{-\frac{1}{d+2\alpha}}. \end{aligned}$$

Here  $C \equiv C(C_1, C_2, C_3, C_4, \epsilon_o, f_{\max}, \delta_0, \delta_1, d, \alpha)$ . The second step follows by observing the trivial bounds  $P(\Omega) \leq 1$  and since the domain  $\mathcal{X} = [0, 1]^d$ ,  $\mathbb{E}[d_\infty(\hat{G}, G_\gamma^*) | \bar{\Omega}] \leq \sqrt{d}$ . The third step follows from Lemma 3 and the fourth one from Lemma 2. The last step follows using the upper and lower bounds established on  $2^{-\hat{j}}$  in Eq. (7).

### 5.3 Star-shaped sets satisfy assumptions [B] and [C]

Recall the definition of  $\mathcal{F}_{SL}$  as defined in [Tsy97]. The class corresponds to densities bounded above by  $f_{\max}$ , satisfying a slightly modified form of the local density regularity assumption [A]:

[A'] *Local density regularity:* The density is  $\alpha$ -regular around the  $\gamma$ -level set,  $0 < \alpha < \infty$  and  $\gamma < f_{\max}$ , if there exist constants  $C_2 > C_1 > 0$  and  $\delta_0 > 0$  such that

$$C_1 \rho(x, \partial G_\gamma^*)^\alpha \leq |f(x) - \gamma| \leq C_2 \rho(x, \partial G_\gamma^*)^\alpha$$

for all  $x \in \mathcal{X}$  with  $|f(x) - \gamma| \leq \delta_0$ , where  $\partial G_\gamma^*$  is the boundary of the true level set  $G_\gamma^*$ , and the set  $\{x : |f(x) - \gamma| \leq \delta_0\}$  is non-empty.

and the densities have  $\gamma$  level sets of the form

$$G_\gamma^* = \{(r, \phi); \phi \in [0, \pi)^{d-2} \times [0, 2\pi), 0 \leq r \leq g(\phi) \leq R\},$$

where  $(r, \phi)$  denote the polar/hyperspherical coordinates and  $R > 0$  is a constant.  $g$  is a periodic Lipschitz function that satisfies  $g(\phi) \geq h$ , where  $h > 0$  is a constant, and

$$|g(\phi) - g(\theta)| \leq L \|\phi - \theta\|_1, \quad \forall \phi, \theta \in [0, \pi)^{d-2} \times [0, 2\pi).$$

Here  $L > 0$  is the Lipschitz constant, and  $\|\cdot\|_1$  denotes the  $\ell_1$  norm.

We set  $R = 1/2$  in the definition of the star-shaped set so that the domain is a subset of  $[-1/2, 1/2]^d$ . With this domain, the following lemma shows that the level set  $G_\gamma^*$  of a density  $f \in \mathcal{F}_{SL}$  satisfies **[B]** and **[C]**.

**Lemma 5** Consider the  $\gamma$  level set  $G_\gamma^*$  of a density  $f \in \mathcal{F}_{SL}$ . Then  $G_\gamma^*$  satisfies the assumptions **[B]** and **[C]** on the level set regularity and the level set boundary dimension, respectively.

**Proof:** We first present a sketch of the main ideas, and then provide a detailed proof. Consider the  $\gamma$ -level set  $G_\gamma^*$  of a density  $f \in \mathcal{F}_{SL}$ . To see that it satisfies **[B]**, divide the star-shaped set  $G_\gamma^*$  into sectors of width  $\asymp \epsilon$  so that each sector contains at least one  $\epsilon$ -ball and the inner cover  $\mathcal{I}_\epsilon(G_\gamma^*)$  touches the boundary at some point(s) in each sector. Now one can argue that, in each sector, all other points on the boundary are  $O(\epsilon)$  from the inner cover since the boundary is Lipschitz. Since this is true for each sector, we have  $\forall x \in \partial G_\gamma^*, \rho(x, \mathcal{I}_\epsilon(G_\gamma^*)) = O(\epsilon)$ . To see that  $G_\gamma^*$  satisfies **[C]**, consider any sector of width  $\asymp \epsilon$  and divide it into sub-sectors of width  $O(\delta)$ ,  $0 < \delta \leq \epsilon$ . Since the boundary is Lipschitz, a constant number of  $\delta$ -balls can cover the boundary in each sub-sector. Thus, the minimum number of  $\delta$ -balls needed to cover the boundary in all sub-sectors is of the order of the minimum number of sub-sectors, that is,  $O((\epsilon/\delta)^{d-1})$ . Hence, the result follows. We now present the proof in detail.

To see that  $G_\gamma^*$  satisfies **[B]**, fix  $\epsilon_o \leq h/3$ . Then for all  $\epsilon \leq \epsilon_o$ ,  $B(0, \epsilon) \subseteq G_\gamma^*$  (since  $g(\phi) \geq h > \epsilon_o$ ), and hence  $\mathcal{I}_\epsilon(G_\gamma^*) \neq \emptyset$ . We also need to show that  $\exists C_3 > 0$  such that for all  $x \in \partial G_\gamma^*, \rho(x, \mathcal{I}_\epsilon(G_\gamma^*)) \leq C_3\epsilon$ . For this, divide  $G_\gamma^*$  into  $M^{d-1}$  sectors indexed by  $\mathbf{m} = (m_1, m_2, \dots, m_{d-1}) \in \{1, \dots, M\}^{d-1}$

$$S_{\mathbf{m}} = \left\{ (r, \phi) : 0 \leq r \leq g(\phi), \right.$$

$$\left. \begin{aligned} \frac{\pi(m_i - 1)}{M} \leq \phi_i < \frac{\pi m_i}{M}, i = 1, \dots, d-2, \\ \frac{2\pi(m_{d-1} - 1)}{M} \leq \phi_{d-1} < \frac{2\pi m_{d-1}}{M} \end{aligned} \right\},$$

where  $\phi = (\phi_1, \phi_2, \dots, \phi_{d-1})$ . Let

$$M = \left\lceil \frac{\pi}{2 \sin^{-1} \frac{\epsilon}{h - \epsilon_o}} \right\rceil$$

This choice of  $M$  implies that:

- (i) There exists an  $\epsilon$ -ball within  $S_{\mathbf{m}} \cap B(0, h)$  for every  $\mathbf{m} \in \{1, \dots, M\}^{d-1}$ , and hence within each sector  $S_{\mathbf{m}}$ . This follows because the minimum angular width of a sector with radius  $h$  required to fit an  $\epsilon$ -ball within is

$$2 \sin^{-1} \frac{\epsilon}{h - \epsilon} \leq 2 \sin^{-1} \frac{\epsilon}{h - \epsilon_o} \leq \frac{\pi}{M}.$$

- (ii) The angular-width of the sectors scales as  $O(\epsilon)$ .

$$\begin{aligned} \frac{\pi}{M} &< \frac{\pi}{2 \sin^{-1} \frac{\pi}{h - \epsilon_o} - 1} = \frac{1}{2 \sin^{-1} \frac{1}{h - \epsilon_o} - \frac{1}{\pi}} \\ &\leq 3 \sin^{-1} \frac{\epsilon}{h - \epsilon_o} \leq 6 \frac{\epsilon}{h - \epsilon_o} \leq \frac{9}{h} \epsilon \end{aligned}$$

The second inequality follows as

$$\frac{1}{\pi} \leq \frac{1}{6 \sin^{-1} \frac{\epsilon}{h - \epsilon_o}}$$

since  $\frac{\epsilon}{h - \epsilon_o} \leq \frac{\epsilon_o}{h - \epsilon_o} \leq \frac{1}{2}$  by choice of  $\epsilon_o \leq h/3$ . The third inequality is true since  $\sin^{-1}(z/2) \leq z$  for  $0 \leq z \leq \pi/2$ . The last step follows by choice of  $\epsilon_o \leq h/3$ .

Now from (i) above, each sector contains at least one  $\epsilon$ -ball. Consider any  $\mathbf{m} \in \{1, \dots, M\}^{d-1}$ . We claim that there exists a point  $x_{\mathbf{m}} \in \partial G_\gamma^* \cap S_{\mathbf{m}}$ ,  $x_{\mathbf{m}} = (g(\theta), \theta)$  for some  $\theta \in [0, \pi)^{d-2} \times [0, 2\pi)$ , such that  $\rho(x_{\mathbf{m}}, \mathcal{I}_\epsilon(G_\gamma^*)) = 0$ . Suppose not. Then one can slide the  $\epsilon$ -ball within the sector towards the periphery and never touch the boundary, implying that the set  $G_\gamma^*$  is unbounded. This is a contradiction by the definition of the class  $\mathcal{F}_{SL}$ . So now we have,  $\forall y \in \partial G_\gamma^* \cap S_{\mathbf{m}}, y = (g(\phi), \phi)$

$$\begin{aligned} \rho(y, \mathcal{I}_\epsilon(G_\gamma^*)) &\leq \rho(y, x_{\mathbf{m}}) = \|y - x_{\mathbf{m}}\| \\ &= \|(g(\phi), \phi) - (g(\theta), \theta)\| \\ &\leq |g(\phi) - g(\theta)| + 2\sqrt{g(\phi)g(\theta)} \cdot \\ &\quad \sum_{i=1}^{d-1} \left| \sin \frac{\phi_i - \theta_i}{2} \right| \\ &\leq L \|\phi - \theta\|_1 + \sum_{i=1}^{d-1} \frac{|\phi_i - \theta_i|}{2} \\ &= (L + 1/2) \sum_{i=1}^{d-1} |\phi_i - \theta_i| \\ &\leq (L + 1/2) d \frac{\pi}{M} \\ &\leq \frac{9d(L + 1/2)}{h} \epsilon := C_3\epsilon \end{aligned}$$

The third step follows using simple algebra (see [SSN07]), the fourth step follows by the Lipschitz condition on  $g(\cdot)$ ,  $g(\cdot) \leq R = 1/2$  and since  $|\sin(z)| \leq |z|$ . The sixth step follows since  $x, y \in S_{\mathbf{m}}$  and hence  $|\phi_i - \theta_i| \leq \pi/M$  for  $i = 1, \dots, d-2$  and  $|\phi_{d-1} - \theta_{d-1}| \leq 2\pi/M$ . The last step invokes (ii) above. Therefore, we have for all  $y \in \partial G_\gamma^* \cap S_{\mathbf{m}}, \rho(y, \mathcal{I}_\epsilon(G_\gamma^*)) \leq C_3\epsilon$ . And since the result is true for any sector, condition **[B]** is satisfied by any level set  $G_\gamma^*$  with density  $f \in \mathcal{F}_{SL}$ .

To see that  $G_\gamma^*$  satisfies **[C]**, consider  $x \in \partial G_\gamma^*$ . Let  $x = (g(\phi_0), \phi_0)$ . Also let  $\phi_i^{(1)} = \min\{\phi_i : (g(\phi), \phi) \in B(x, \epsilon)\}$  and  $\phi_i^{(2)} = \max\{\phi_i : (g(\phi), \phi) \in B(x, \epsilon)\}$ . Define the sector

$$S_\epsilon^x = \left\{ (r, \phi) : 0 \leq r \leq g(\phi), \right. \\ \left. \phi_i^{(1)} \leq \phi_i \leq \phi_i^{(2)}, \forall i = 1, \dots, d-1 \right\}$$

Observe that if  $\epsilon \leq \pi h/4 < h$ , the width of  $S_\epsilon^x$  in the  $i^{\text{th}}$  coordinate,  $\Delta\phi_i = \phi_i^{(2)} - \phi_i^{(1)} \leq 2 \sin^{-1} \frac{\epsilon}{g(\phi_0)}$  by construction. Since  $g(\cdot) \geq h$ , we have  $\Delta\phi_i \leq 2 \sin^{-1} \frac{\epsilon}{h} \leq 4\epsilon/h$ , where the last step follows since for  $0 \leq z \leq \pi/2$ ,  $\sin^{-1}(z/2) \leq z$ . If  $\epsilon > \pi h/4$ , then use the trivial bound

$\Delta\phi_i \leq 2\pi \leq 8\epsilon/h$ . Equivalently, we can say for all  $\epsilon$  and all  $i$ ,

$$\Delta\phi_i \leq 8\epsilon/h. \quad (8)$$

Further subdivide  $S_\epsilon^x$  into  $M^{d-1}$  sub-sectors indexed by  $\mathbf{m} = (m_1, \dots, m_{d-1})$

$$S_{\mathbf{m}} = \left\{ (r, \phi) : 0 \leq r \leq g(\phi), \phi_i^{(1)} + \frac{(m_i - 1)\Delta\phi_i}{M} \leq \phi_i < \phi_i^{(1)} + \frac{m_i\Delta\phi_i}{M}, \forall i = 1, \dots, d-1 \right\}$$

Pick  $M$  such that for all coordinates, the sub-sector width  $\frac{\Delta\phi_i}{M} \leq \frac{2\delta}{(d-1)(L+1/2)}$ , where  $0 < \delta \leq \epsilon$ . With this choice of sub-sector width,  $S_{\mathbf{m}} \cap \partial G_\gamma^*$  can be covered by a  $\delta$ -ball. To see this, consider two points in  $S_{\mathbf{m}} \cap \partial G_\gamma^*$  -  $(g(\phi), \phi)$  and  $(g(\theta), \theta)$ . Proceeding as before, we have:

$$\begin{aligned} \|(g(\phi), \phi) - (g(\theta), \theta)\| &\leq (L+1/2) \sum_{i=1}^{d-1} |\phi_i - \theta_i| \\ &\leq (L+1/2) \sum_{i=1}^{d-1} \frac{\Delta\phi_i}{M} \leq 2\delta. \end{aligned}$$

Since each sub-sector can be covered by a  $\delta$ -ball, the minimum number of  $\delta$ -balls needed to cover  $B(x, \epsilon) \cap \partial G_\gamma^*$  is equal to the minimum number of sub-sectors needed ( $M^{d-1}$ ). This corresponds to the smallest  $M$  such that  $\max_i \frac{\Delta\phi_i}{M} \leq \frac{2\delta}{(d-1)(L+1/2)}$ . Therefore, minimum number of  $\delta$ -balls needed to cover  $B(x, \epsilon) \cap \partial G_\gamma^*$  is equal to

$$\begin{aligned} &\left( \left\lceil \frac{(d-1)(L+1/2) \max_i \Delta\phi_i}{2\delta} \right\rceil \right)^{d-1} \\ &\leq \left( \frac{(d-1)(L+1/2) \max_i \Delta\phi_i}{2\delta} + 1 \right)^{d-1} \\ &\leq \left( \frac{2(d-1)(2L+1)\epsilon}{h} \frac{\epsilon}{\delta} + \frac{\epsilon}{\delta} \right)^{d-1} \\ &\leq \left( \frac{2(d-1)(2L+1)}{h} + 1 \right)^{d-1} \left( \frac{\epsilon}{\delta} \right)^{d-1} \\ &:= C_4 \left( \frac{\epsilon}{\delta} \right)^{d-1} \end{aligned}$$

The second inequality follows since from Eq. (8),  $\Delta\phi_i \leq \frac{8\epsilon}{h}$  for all  $i$ , and since  $\delta \leq \epsilon$ . Therefore, any level set  $G_\gamma^*$  with density  $f \in \mathcal{F}_{SL}$  also satisfies [C]. ■

## Acknowledgements

The authors would like to thank Rui Castro for helpful discussions and carefully reviewing the paper.

## References

- [Cav97] L. Cavalier. Nonparametric estimation of regression level sets. *Statistics*, 29:131–160, 1997.  
[CD99] E. Candés and D. L. Donoho. Curvelets: A surprisingly effective nonadaptive representation for

objects with edges. *Curves and Surfaces*, Larry Schumaker et al., Ed. Vanderbilt University Press, Nashville, TN, 1999.

- [CMC06] A. Cuevas, W. G. Manteiga, and A. R. Casal. Plug-in estimation of general level sets. *Aust. N. Z. J. Stat.*, 48(1):7–19, 2006.  
[DL01] L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer, NY, 2001.  
[Don99] D. L. Donoho. Wedgelets: Nearly-minimax estimation of edges. *Ann. Statist.*, 27:859–897, 1999.  
[Fal90] K. Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. Wiley, West Sussex, England, 1990.  
[Har75] J. A. Hartigan. *Clustering Algorithms*. Wiley, NY, 1975.  
[KT93] A. P. Korostelev and A. B. Tsybakov. *Minimax Theory of Image Reconstruction*. Springer, NY, 1993.  
[LMS97] O. V. Lepski, E. Mammen, and V. G. Spokoiny. Optimal spatial adaptation to inhomogeneous smoothness: An approach based on kernel estimates with variable bandwidth selectors. *Ann. Statist.*, 25(3):929–947, 1997.  
[LPS99] R. Y. Liu, J. M. Parelus, and K. Singh. Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *Ann. Statist.*, 27(3):783–858, 1999.  
[Pol95] W. Polonik. Measuring mass concentrations and estimating density contour cluster-an excess mass approach. *Ann. Statist.*, 23(3):855–881, 1995.  
[RV06] Philippe Rigollet and Regis Vert. Fast rates for plug-in estimators of density level sets, available at <http://www.citebase.org/abstract?id=oi:arxiv.org:math/0611473>, 2006.  
[SD07] C. Scott and M. Davenport. Regression level set estimation via cost-sensitive classification. *IEEE Trans. Signal Process.*, 55(6):2752–2757, 2007.  
[SHS05] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *J. Mach. Learn. Res.*, 6:211–232, 2005.  
[SN06] C. Scott and R. Nowak. Learning minimum volume sets. *J. Mach. Learn. Res.*, 7:665–704, 2006.  
[SSN07] A. Singh, C. Scott, and R. D. Nowak. Adaptive hausdorff estimation of density level sets. Technical Report ECE-07-06, University of Wisconsin - Madison, ECE Dept., available at [www.cae.wisc.edu/~singh/TR\\_Hausdorff.pdf](http://www.cae.wisc.edu/~singh/TR_Hausdorff.pdf), 2007.  
[Stu03] W. Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *J. Classification*, 20(5):25–47, 2003.  
[Tsy97] A. B. Tsybakov. On nonparametric estimation of density level sets. *Ann. Statist.*, 25:948–969, 1997.  
[VV06] R. Vert and J.-P. Vert. Consistency and convergence rates of one-class svms and related algorithms. *J. Mach. Learn. Res.*, 7:817–854, 2006.  
[WN07] R. Willett and R. Nowak. Minimax optimal level set estimation. *IEEE Trans. Image Proc.*, 16(12):2965–2979, 2007.



---

# Density estimation in linear time

---

Satyaki Mahalanabis\* and Daniel Štefankovič

Department of Computer Science

University of Rochester

Rochester, NY 14627

{smahalan, stefanko}@cs.rochester.edu

## Abstract

We consider the problem of choosing a density estimate from a set of densities  $\mathcal{F}$ , minimizing the  $L_1$ -distance to an unknown distribution. Devroye and Lugosi [DL01] analyze two algorithms for the problem: Scheffé tournament winner and minimum distance estimate. The Scheffé tournament estimate requires fewer computations than the minimum distance estimate, but has strictly weaker guarantees than the latter.

We focus on the computational aspect of density estimation. We present two algorithms, both with the same guarantee as the minimum distance estimate. The first one, a modification of the minimum distance estimate, uses the same number (quadratic in  $|\mathcal{F}|$ ) of computations as the Scheffé tournament. The second one, called “efficient minimum loss-weight estimate,” uses only a linear number of computations, assuming that  $\mathcal{F}$  is preprocessed. We then apply our algorithms to bandwidth selection for kernel estimates and bin-width selection for histogram estimates, yielding efficient procedures for these problems.

We also give examples showing that the guarantees of the algorithms cannot be improved and explore randomized algorithms for density estimation.

## 1 Introduction

We study the following density estimation problem considered in [DL96, DL01, DGL02]. There is an unknown distribution  $g$  and we are given  $n$  (not necessarily independent) samples which define empirical distribution  $h$ . Given a finite class  $\mathcal{F}$  of densities, our objective is to output  $f \in \mathcal{F}$  such that the error  $\|f - g\|_1$  is minimized. The use of the  $L_1$ -norm is well justified because it has many useful properties, for example, scale invariance and the fact that approximate identification of a

distribution in the  $L_1$ -norm gives an estimate for the probability of every event.

The following two parameters influence the error of a possible estimate: the distance of  $g$  from  $\mathcal{F}$  and the empirical error. The first parameter is required since we have no control over  $\mathcal{F}$ , and hence we cannot select a density which is better than the “optimal” density in  $\mathcal{F}$ , that is, the one closest to  $g$  in  $L_1$ -norm. It is not obvious how to define the second parameter—the error of  $h$  with respect to  $g$ . We follow the definition of [DL01], which is inspired by [Yat85] (see Section 1.1 for a precise definition).

Devroye and Lugosi [DL01] analyze two algorithms in this setting: Scheffé tournament winner and minimum distance estimate. The minimum distance estimate, defined by Yatracos [Yat85], is a special case of the minimum distance principle, formalized by Wolfowitz in [Wol57]. It is a general density estimation tool which has been applied, for example, by [DL96, DL97] to the bandwidth selection problem for kernels and by [DL04, DL01] to bin-width selection for histograms. The minimum distance estimate also finds application in hypothesis testing [DGL02].

The Scheffé tournament winner algorithm requires fewer computations than the minimum distance estimate, but it has strictly weaker guarantees (in terms of the two parameters mentioned above) than the latter. Our main contribution are two procedures for selecting an estimate from  $\mathcal{F}$ , both of which have the same guarantees as the minimum distance estimate, but are computationally more efficient. The first has a quadratic (in  $|\mathcal{F}|$ ) cost, matching the cost of the Scheffé tournament winner algorithm. The second one is even faster, using *linearly* many (in  $|\mathcal{F}|$ ) computations (after preprocessing  $\mathcal{F}$ ).

We also apply our estimation procedures to the problem of bandwidth selection for kernels and to that of bin-width selection for histograms, following [DL01, DL96, DL97, DL04]. We show that in each of these applications “efficient minimum loss-weight estimate” is faster than our “modified minimum distance estimate,” which in turn is faster than the minimum distance estimate.

Now we outline the rest of the paper. In Section 1.1 we give the required definitions and introduce the notion of a test-function (a variant of Scheffé set). Then, in Section 1.2, we restate the previous density estimation

---

\*Supported by NSF grant IIS-0546554

algorithms (Scheffé tournament winner and the minimum distance estimate) using test-functions. Next, in Section 2, we present our algorithms. In Section 3 we discuss two widely studied nonparametric estimation problems where the computational cost of efficient minimum loss-weight estimate (including preprocessing) is much smaller than that of both the modified minimum distance and the minimum distance estimates. In Section 4 we explore randomized density estimation algorithms. In the final Section 5, we give examples showing tightness of the theorems stated in the previous sections.

Throughout this paper we focus on the case when  $\mathcal{F}$  is finite, in order to compare the computational costs of our estimates to previous ones. However our results generalize in a straightforward way to infinite classes as well if we ignore computational complexity.

### 1.1 Definitions and Notations

Throughout the paper  $g$  will be the unknown distribution. We will use  $h$  to denote the empirical distribution, which given samples  $X_1, X_2, \dots, X_n$ , is defined for each set  $A \subseteq \Omega$  as

$$h(A) = \frac{1}{n} \sum_{i=1}^n 1_{[X_i \in A]}$$

Let  $\mathcal{F}$  be a set of densities. We will assume that  $\mathcal{F}$  is finite. Let  $d_1(g, \mathcal{F})$  be the  $L_1$ -distance of  $g$  from  $\mathcal{F}$ , that is,  $\min_{f \in \mathcal{F}} \|f - g\|_1$ .

Given two functions  $f_i, f_j$  on  $\Omega$  (in this context, densities) we define a *test-function*  $T_{ij} : \Omega \rightarrow \{-1, 0, 1\}$  to be the function  $T_{ij}(x) = \text{sgn}(f_i(x) - f_j(x))$ . Note that  $T_{ij} = -T_{ji}$ . We also define  $\mathcal{T}_{\mathcal{F}}$  to be the set of all test-functions for  $\mathcal{F}$ , that is,

$$\mathcal{T}_{\mathcal{F}} = \{T_{ij} ; f_i, f_j \in \mathcal{F}\}.$$

Let  $\cdot$  be the inner product for the functions on  $\Omega$ , defined for any 2 functions  $f, f'$  as  $f \cdot f' = \int f f'$ . Note that

$$(f_i - f_j) \cdot T_{ij} = \|f_i - f_j\|_1.$$

We use the inner product of the empirical distribution  $h$  with the test-functions to choose an estimate, which is a density from  $\mathcal{F}$ .

In this paper we only consider algorithms which make their decisions purely on inner products of the test-functions with  $h$  and members of  $\mathcal{F}$ . It is reasonable to assume that the computation of the inner product will take significant time. Hence we measure the *computational cost* of an algorithm is by the number of inner products used.

We say that  $f_i$  wins against  $f_j$  if

$$(f_i - h) \cdot T_{ij} < (f_j - h) \cdot T_{ji}. \quad (1)$$

Note that either  $f_i$  wins against  $f_j$ , or  $f_j$  wins against  $f_i$ , or there is a draw (that is, there is equality in (1)). We will say that  $f_i$  loses to  $f_j$  if

$$(f_i - h) \cdot T_{ij} \geq (f_j - h) \cdot T_{ji}.$$

The algorithms choose an estimate  $f \in \mathcal{F}$  using the empirical distribution  $h$ . The  $L_1$ -distance of the estimates from the unknown distribution  $g$  will depend on

the following measure of distance between the empirical and the unknown distribution:

$$\Delta := \max_{T \in \mathcal{T}_{\mathcal{F}}} (g - h) \cdot T. \quad (2)$$

Now we discuss how test-functions can be viewed as a reformulation of Scheffé sets, defined by Devroye and Lugosi [DL01] (inspired by [Sch47] and implicit in [Yat85]), as follows. The Scheffé set of densities  $f_i, f_j$  is

$$A_{ij} = \{x ; f_i(x) > f_j(x)\}.$$

Devroye and Lugosi say that  $f_i$  wins against  $f_j$  if

$$\left| \int_{A_{ij}} f_i - h(A_{ij}) \right| < \left| \int_{A_{ij}} f_j - h(A_{ij}) \right|. \quad (3)$$

The advantage of using Scheffé sets is that for a concrete set  $\mathcal{F}$  of densities one can immediately use the theory of Vapnik-Chervonenkis dimension [VC71] for the family of Scheffé sets of  $\mathcal{F}$  (this family is called the *Yatracos class* of  $\mathcal{F}$ ), to obtain a bound on the empirical error.

If  $h, f_i, f_j$  are non-negative and integrate to 1 then the condition (1) is *equivalent* to (3) (to see this recall that  $T_{ij} = -T_{ji}$ , and add  $(f_i - h) \cdot \mathbf{1} = (h - f_j) \cdot \mathbf{1}$  to (1), where  $\mathbf{1}$  is the constant one function on  $\Omega$ ). Thus, in our algorithms the test-functions can be replaced by Scheffé sets and VC dimension arguments can be applied.

We chose to use test-functions for two reasons: first, they allow us to give succinct proofs of our theorems (especially Theorem 8), and second, they immediately extend to the case when the members of  $\mathcal{F}$  do not correspond to distributions (cf, e.g., Exercise 6.2, in [DL01]).

**Remark 1** Note that our value of  $\Delta$ , defined in terms of  $\mathcal{T}_{\mathcal{F}}$ , is at most twice the  $\Delta$  used in [DL01], which is defined in terms of Scheffé sets.

### 1.2 Previous Estimates

In this section we restate the two algorithms for density estimation from Chapter 6 of [DL01] using test-functions. The first algorithm requires less computation but has worse guarantees than the second algorithm.

**Algorithm 1** - SCHEFFÉ TOURNAMENT WINNER.  
Output  $f \in \mathcal{F}$  with the most wins (tie broken arbitrarily).

**Theorem 2 ([DL01], Theorem 6.2)** Let  $f_1 \in \mathcal{F}$  be the density output by Algorithm 1. Then

$$\|f_1 - g\|_1 \leq 9 d_1(g, \mathcal{F}) + 8\Delta.$$

The number of inner products used by Algorithm 1 is  $\Theta(|\mathcal{F}|^2)$ .

**Algorithm 2** - MINIMUM DISTANCE ESTIMATE.  
Output  $f \in \mathcal{F}$  that minimizes

$$\max \{ |(f - h) \cdot T_{ij}| ; f_i, f_j \in \mathcal{F} \}. \quad (4)$$

**Theorem 3 ([DL01], Theorem 6.3)** Let  $f_1$  be the density output by Algorithm 2. Then

$$\|f_1 - g\|_1 \leq 3 d_1(g, \mathcal{F}) + 2\Delta.$$

The number of inner products used by Algorithm 2 is  $\Theta(|\mathcal{F}|^3)$ .

Let us point out that Theorems 6.2 and 6.3 in [DL01] require that each  $f \in \mathcal{F}$  corresponds to a distribution, that is,  $\int f = 1$ . Since we use test-functions in the algorithms instead of Scheffé set based comparisons, the assumption  $\int f = 1$  is not actually needed in the proofs of Theorems 6.2 and 6.3 (we skip the proof), and is not used in the proofs of Theorems 4, 8.

## 2 Our estimators

### 2.1 A variant of the minimum distance estimate

The following modified minimum distance estimate uses only  $O(|\mathcal{F}|^2)$  computations as compared to  $O(|\mathcal{F}|^3)$  computations used by Algorithm 2 (equation (5) takes minimum of  $O(|\mathcal{F}|)$  terms, whereas equation (4) takes minimum of  $O(|\mathcal{F}|^2)$  terms), but as we show in Theorem 4, it gives us the same guarantee as the minimum distance estimate.

**Algorithm 3 - MODIFIED MINIMUM DISTANCE ESTIMATE.**

Output  $f_i \in \mathcal{F}$  that minimizes

$$\max \{ |(f_i - h) \cdot T_{ij}| ; f_j \in \mathcal{F} \}. \quad (5)$$

**Theorem 4** Let  $f_1 \in \mathcal{F}$  be the density output by Algorithm 3. Then

$$\|f_1 - g\|_1 \leq 3 d_1(g, \mathcal{F}) + 2\Delta.$$

The number of inner products used by Algorithm 3 is  $\Theta(|\mathcal{F}|^2)$ .

**Proof :**

Let  $f_1 \in \mathcal{F}$  be the function output by Algorithm 3. Let  $f_2 = \operatorname{argmin}_{f \in \mathcal{F}} \|f - g\|_1$ . By the triangle inequality we have

$$\|f_1 - g\|_1 \leq \|f_1 - f_2\|_1 + \|f_2 - g\|_1. \quad (6)$$

We bound  $\|f_1 - f_2\|_1$  as follows:

$$\begin{aligned} \|f_1 - f_2\|_1 &= (f_1 - f_2) \cdot T_{12} \\ &\leq |(f_1 - h) \cdot T_{12}| + |(f_2 - h) \cdot T_{12}| \\ &\leq |(f_1 - h) \cdot T_{12}| + \max_{f_j \in \mathcal{F}} |(f_2 - h) \cdot T_{2,j}| \end{aligned}$$

where in the last inequality we used the fact that  $T_{12} = -T_{21}$ .

By the criteria of selecting  $f_1$  we have  $|(f_1 - h) \cdot T_{12}| \leq \max_{f_j \in \mathcal{F}} |(f_2 - h) \cdot T_{2,j}|$  (since otherwise  $f_2$  would be selected). Hence

$$\begin{aligned} \|f_1 - f_2\|_1 &\leq 2 \max_{f_j \in \mathcal{F}} |(f_2 - h) \cdot T_{2,j}| \\ &\leq 2 \max_{f_j \in \mathcal{F}} |(f_2 - g) \cdot T_{2,j}| \\ &\quad + 2 \max_{f_j \in \mathcal{F}} |(g - h) \cdot T_{2,j}| \\ &\leq 2\|(f_2 - g)\|_1 + 2 \max_{T \in \mathcal{T}_{\mathcal{F}}} |(g - h) \cdot T| \\ &= 2\|f_2 - g\|_1 + 2\Delta. \end{aligned}$$

Combining the last inequality with (6) we obtain

$$\|f_1 - g\|_1 \leq 3\|f_2 - g\|_1 + 2\Delta. \quad \blacksquare$$

**Remark 5** Note that one can modify the Lemma to only require that  $g$  and  $h$  be “close” with respect to the test functions for the “best” function in the class, that is, only  $|(g - h) \cdot T_{2,j}|$  need to be small (where  $f_2$  is  $\operatorname{argmin}_{f \in \mathcal{F}} \|f - g\|_1$ ).

One can ask whether the observation in Remark 5 can lead to improved density estimation algorithms for concrete sets of densities. The bounds on  $\Delta$  (which is given by (2)) are often based on the VC-dimension of the Yatracos class of  $\mathcal{F}$ . Recall that the Yatracos class  $Y$  is the set of  $A_{ij} = \{x; f_i(x) > f_j(x)\}$  for all  $f_i, f_j \in \mathcal{F}$ . Remark 5 implies that instead of the Yatracos class it is enough to consider the set  $Y_i = \{A_{ij}; f_j \in \mathcal{F}\}$  for  $f_i \in \mathcal{F}$ . Is it possible that the VC-dimension of each set  $Y_i$  is smaller the VC-dimension of the Yatracos class  $Y$ ? The following (artificial) example shows that this can, indeed, be the case. Let  $\Omega = \{0, \dots, n\}$ . For each  $(n + 1)$ -bit binary string  $a_0, a_1, \dots, a_n$ , let us consider the distribution

$$P(k) = \frac{1}{4n} (1 + (1/2 - a_0)(1/2 - a_k)) 2^{-\sum_{j=1}^n a_j 2^j},$$

for  $k \in \{1, \dots, n\}$  (with  $P(0)$  chosen to make  $P$  into a distribution). For this family of  $2^{n+1}$  distributions the VC-dimension of the Yatracos class is  $n$ , whereas each  $Y_i$  has VC-dimension 1 (since a pair of distributions  $f_i, f_j$  has a non-trivial set  $A_{ij}$  if and only if their binary strings differ only in the first bit).

### 2.2 An even more efficient estimator - minimum loss-weight

In this section we present an estimator which, after pre-processing  $\mathcal{F}$ , uses only  $O(|\mathcal{F}|)$  inner products to obtain a density estimate. The guarantees of the estimate are the same as for Algorithms 2 and 3.

The algorithm uses the following quantity to choose the estimate:

$$\begin{aligned} \text{loss-weight}(f) &= \max \{ \|f - f'\|_1 ; f \text{ loses} \\ &\quad \text{to } f' \in \mathcal{F} \}. \end{aligned}$$

Intuitively a good estimate should have small loss-weight (ideally the loss-weight of the estimate would be  $-\infty = \max\{\}$ , that is, the estimate would not lose at all). Thus the following algorithm would be a natural candidate for a good density estimator (and, indeed, it has a guarantee matching Algorithms 2 and 3), but, unfortunately, we do not know how to implement it using  $O(|\mathcal{F}|)$  inner products.

**Algorithm 4a - MINIMUM LOSS-WEIGHT ESTIMATE.**

Output  $f \in \mathcal{F}$  that minimizes  $\text{loss-weight}(f)$ .

The next algorithm, seems less natural than algorithm 4a, but its condition can be implemented using only  $O(|\mathcal{F}|)$  inner products.

**Algorithm 4b** - EFFICIENT MINIMUM LOSS-WEIGHT ESTIMATE.

Output  $f \in \mathcal{F}$  such that for every  $f'$  to which  $f$  loses we have

$$\|f - f'\|_1 \leq \text{loss-weight}(f'). \quad (7)$$

Before we delve into the proof of (8) let us see how Algorithm 4b can be made to use  $|\mathcal{F}|-1$  inner products. We preprocess  $\mathcal{F}$  by computing  $L_1$ -distances between all pairs of densities in  $\mathcal{F}$  and store the distances in an list sorted in decreasing order. When the algorithm is presented with the empirical distribution  $h$ , all it needs to do is perform comparison between select pairs of densities. The advantage is that we preprocess  $\mathcal{F}$  only once and, for each new empirical distribution we only compute inner products necessary for the comparisons.

We will compute the estimate as follows.

**input** : family of densities  $\mathcal{F}$ , list  $L$  of all pairs  $\{f_i, f_j\}$  sorted in decreasing order by  $\|f_i - f_j\|_1$ , oracle for computing inner products  $h \cdot T_{ij}$ .  
**output** :  $f \in \mathcal{F}$  such that:  $(\forall f') f$  loses to  $f'$   
 $\implies \|f - f'\|_1 \leq \text{loss-weight}(f')$ .

```

1  $S \leftarrow \mathcal{F}$ 
2 repeat
3   pick the first edge  $\{f_i, f_j\}$  in  $L$ 
4   if  $f_i$  loses to  $f_j$  then  $f' \leftarrow f_i$  else  $f' \leftarrow f_j$  fi
5   remove  $f'$  from  $S$ 
6   remove pairs containing  $f'$  from  $L$ 
7 until  $|S| = 1$ 
8 output the density in  $S$ 

```

**Detailed version of algorithm 4b - using  $O(|\mathcal{F}|)$  inner products.**

Note that while Algorithm 4b uses only  $O(|\mathcal{F}|)$  inner products its running time is actually  $\Theta(|\mathcal{F}|^2)$ , since it traverses a list of length  $\Theta(|\mathcal{F}|^2)$ . Are we cheating? There are two answers to this question: practical and theoretical. As we will see in applications the inner products dominate the computation, justifying our focus on just the inner products (of which there are linearly many). Theoretically, if we are willing to spend exponential time for the preprocessing, we can build the complete decision tree corresponding to Algorithm 4b and obtain a linear-time density selection procedure. We find the following question interesting: Is it possible to achieve linear running time using only polynomial-time preprocessing?

**Question 6 (Tournament Revelation Problem)**

We are given a weighted undirected complete graph on  $n$  vertices. Assume that the edge-weights are distinct. We preprocess the weighted graph and then play the following game with an adversary until only one vertex remains: we report the edge with the largest weight and the adversary chooses one of the endpoints of the edge and removes it from the graph (together with all the adjacent edges).

Our goal is to make the computational cost during the game linear-time (in  $n$ ) in the worst-case (over the

adversary's moves). Is it possible to achieve this goal with polynomial-time preprocessing?

We now show that the detailed version of algorithm 4b outputs  $f$  satisfying the required condition.

**Lemma 7** *The estimate  $f$  output by the detailed version of algorithm 4b satisfies (7) for every  $f'$  to which  $f$  loses.*

**Proof :**

We show, using induction, that the following invariant is always satisfied on line 2. For any  $f \in S$  and any  $f' \in \mathcal{F} \setminus S$  we have that if  $f$  loses to  $f'$  then  $\|f - f'\|_1 \leq \text{loss-weight}(f')$ . Initially,  $\mathcal{F} \setminus S$  is empty and the invariant is trivially true. For the inductive step, let  $f'$  be the density most recently removed from  $S$ . To prove the induction step we only need to show that for every  $f \in S$  we have that if  $f$  loses to  $f'$  then  $\|f - f'\|_1 \leq \text{loss-weight}(f')$ . Let  $W$  be the  $L_1$ -distance between two densities in  $S \cup \{f'\}$ . Then  $\text{loss-weight}(f') \geq W$  (since  $f'$  lost), and  $\|f - f'\|_1 \leq W$  (by the definition of  $W$ ). ■

**Theorem 8** *Let  $f_1 \in \mathcal{F}$  be the density output by Algorithm 4a (or Algorithm 4b). Then*

$$\|f_1 - g\|_1 \leq 3 d_1(g, \mathcal{F}) + 2\Delta. \quad (8)$$

Assume that we are given  $L_1$ -distances between every pair in  $\mathcal{F}$ . The number of inner products used by Algorithm 4b is  $\Theta(|\mathcal{F}|)$ .

**Proof of Theorem 8:**

Let  $f_4 = g$ . Let  $f_2$  be the function  $f \in \mathcal{F}$  minimizing  $\|g - f\|_1$ . We can reformulate our goal (8) as follows:

$$(f_1 - f_4) \cdot T_{14} \leq 2\Delta + 3(f_2 - f_4) \cdot T_{24}. \quad (9)$$

Let  $f_3 \in \mathcal{F}$  be the function  $f' \in \mathcal{F}$  such that  $f_2$  loses against  $f'$  and  $\|f_2 - f'\|_1$  is maximal (there must be at least one function to which  $f_2$  loses, otherwise the algorithm would pick  $f_2$  and we would be done). Note that  $f_1, f_2, f_3 \in \mathcal{F}$ , but  $f_4$  does need to be in  $\mathcal{F}$ .

We know that  $f_2$  loses against  $f_3$ , that is, we have (see (1))

$$2h \cdot T_{23} \leq f_2 \cdot T_{23} + f_3 \cdot T_{23}, \quad (10)$$

and, since  $f_1$  satisfied (7), we also have

$$(f_1 - f_2) \cdot T_{12} \leq (f_2 - f_3) \cdot T_{23}. \quad (11)$$

By (2) we have

$$2(f_4 - h) \cdot T_{23} \leq 2\Delta. \quad (12)$$

Adding (10), (11), and (12) we obtain

$$2(f_2 - f_4) \cdot T_{23} + (f_2 - f_1) \cdot T_{12} + 2\Delta \geq 0. \quad (13)$$

Note that for any  $i, j, k, \ell$  we have:

$$(f_i - f_j) \cdot (T_{ij} - T_{k\ell}) \geq 0, \quad (14)$$

since if  $f_i(x) > f_j(x)$  then  $T_{ij} - T_{k\ell} \geq 0$ , if

$f_i(x) < f_j(x)$  then  $T_{ij} - T_{k\ell} \leq 0$ , and if  $f_i(x) = f_j(x)$  then the contribution of that  $x$  is zero. By applying (14) four times we obtain

$$(f_2 - f_4) \cdot (3T_{24} - 2T_{23} - T_{14}) + (f_1 - f_2) \cdot (T_{12} - T_{14}) \geq 0. \quad (15)$$

Finally, adding (13) and (15) yields (9). ■

**Remark 9** Note that Remark 5 also applies to Algorithms 4a and 4b, since (12) is the only inequality in which  $\Delta$  is used.

**Lemma 10** *If the condition (7) of Algorithm 4b is relaxed to*

$$\|f - f'\|_1 \leq C \cdot \text{loss-weight}(f'), \quad (16)$$

for some  $C \geq 1$ , an analogue of Theorem 8 with (8) replaced by

$$\|f_1 - g\|_1 \leq (1 + 2C) d_1(g, \mathcal{F}) + 2C\Delta \quad (17)$$

holds.

**Proof :**

The proof is almost identical to the proof of Theorem 8. Let  $f_4 = g$ . Let  $f_2$  be the function  $f \in \mathcal{F}$  minimizing  $\|g - f\|_1$ . We can reformulate our goal (17) as follows:

$$(f_1 - f_4) \cdot T_{14} \leq 2C\Delta + (1 + 2C)(f_2 - f_4) \cdot T_{24}. \quad (18)$$

Let  $f_3 \in \mathcal{F}$  be the function  $f' \in \mathcal{F}$  such that  $f_2$  loses against  $f'$  and  $\|f_2 - f'\|_1$  is maximal (there must be at least one function to which  $f_2$  loses, otherwise the algorithm would pick  $f_2$  and we would be done). Note that  $f_1, f_2, f_3 \in \mathcal{F}$ , but  $f_4$  does need to be in  $\mathcal{F}$ .

Equations (10) and (12) from proof of Theorem 8 are satisfied here as well. Since  $f_1$  satisfies (16), we also have

$$(f_1 - f_2) \cdot T_{12} \leq C(f_2 - f_3) \cdot T_{23}. \quad (19)$$

Adding (10) multiplied by  $C$ , (19), and (12) multiplied by  $C$  we obtain

$$2C(f_2 - f_4) \cdot T_{23} + (f_2 - f_1) \cdot T_{12} + 2C\Delta \geq 0. \quad (20)$$

By applying (14) four times we obtain

$$(f_2 - f_4) \cdot ((1 + 2C)T_{24} - 2CT_{23} - T_{14}) + (f_1 - f_2) \cdot (T_{12} - T_{14}) \geq 0. \quad (21)$$

Finally, adding (20) and (21) yields (18).  $\blacksquare$

Lemma 10 allows us to run Algorithm 4b with distances between the densities computed approximately with relative error  $(1 \pm \varepsilon)$  and obtain analogue of Theorem 8.

**Corollary 11** *Assume that we are given approximate  $L_1$ -distances between every pair in  $\mathcal{F}$  with relative error  $(1 \pm \varepsilon)$ . Let  $f_1 \in \mathcal{F}$  be the density output by Algorithm 4a (or Algorithm 4b), where the algorithm uses the approximate distances (instead of the true distances). Then*

$$\|f_1 - g\|_1 \leq \frac{3 + \varepsilon}{1 - \varepsilon} d_1(g, \mathcal{F}) + \frac{2 + 2\varepsilon}{1 - \varepsilon} \Delta. \quad (22)$$

The number of inner products used by Algorithm 4b is  $\Theta(|\mathcal{F}|)$ .

**Proof :**

Let  $D(f, f')$  be the approximate  $L_1$ -distance between  $f$  and  $f'$  given to the algorithm (for every pair  $f, f' \in \mathcal{F}$ ). Let

$$\text{loss-weight}'(f) = \max \{ D(f, f') ; f \text{ loses to } f' \in \mathcal{F} \}.$$

The proof of Lemma 7 yields that the estimate  $f$  output by the detailed version of algorithm 4b satisfies the following inequality

$$D(f, f') \leq \text{loss-weight}'(f').$$

for every  $f'$  to which  $f$  loses. Now using the fact that  $D(f, f')$  is an  $(1 \pm \varepsilon)$  approximation of  $\|f - f'\|_1$  we obtain that the estimate  $f$  output by algorithm 4b satisfies the following

$$\|f - f'\|_1 \leq \frac{1 + \varepsilon}{1 - \varepsilon} \cdot \text{loss-weight}(f').$$

for every  $f'$  to which  $f$  loses.  $\blacksquare$

### 3 Applications

We now describe two nonparametric density estimation problems where our estimates can be used to obtain efficient algorithms. The first of these problems is that of selecting the optimal smoothing factor for kernel estimates (Section 3.1) while the second one is that of finding an optimal bin-width for 1-dimensional histograms (Section 3.3).

#### 3.1 Bandwidth selection for kernel estimates

We are going to show that our estimates give fast algorithms for the bandwidth selection problem for uniform kernels on  $\mathbb{R}$ .

Given  $n$  i.i.d samples  $x_1, \dots, x_n \in \mathbb{R}$  drawn from an unknown distribution  $g$  the kernel estimate for  $g$  is the density

$$f_{n,s}(x) = \frac{1}{ns} \sum_{i=1}^n K\left(\frac{x - x_i}{s}\right)$$

where  $K$ , the kernel, is a function (usually nonnegative) with  $\int K = 1$  and  $\int |K| < \infty$ , and  $s > 0$  is called the smoothing factor. For us  $K$  will be the uniform distribution on  $[-1, 1]$ .

Given  $x_1, \dots, x_n$  the bandwidth selection problem is to select an  $s^* > 0$  such that  $\|f_{n,s^*} - g\|_1$  is close to  $\inf_{s>0} \|f_{n,s} - g\|_1$  [DL01, DL96, DL97]. The data splitting approach to bandwidth selection uses  $n - m$  ( $n \gg m > 0$ ) samples  $x_1, \dots, x_{n-m}$  to define the kernel estimate  $f_{n-m,s}$  and remaining  $m$  samples  $x_{n-m+1}, \dots, x_n$  as a test set which defines an empirical measure  $h$ . Devroye and Lugosi ([DL96]) use the minimum distance estimate to give an algorithm for selecting  $s^*$ . Given  $n > 0$  samples, they select  $s$  from an interval  $[a_n, b_n]$  (where, e.g.,  $a_n = e^{-n}, b_n = e^n$ ). They discretize  $[a_n, b_n]$  by defining  $s_1 = a_n, s_2 = a_n(1 + \delta_n), \dots, s_i = a_n(1 + \delta_n)^{i-1}, \dots, s_N = a_n(1 + \delta_n)^{N-1}$  where  $N = \lceil \ln(b_n/a_n) / \ln(1 + \delta_n) \rceil$  and  $\delta_n > 0$  is a parameter. They now select  $s^*$  to be  $s_i$  such that  $f_{n-m,s_i}$  is the minimum distance estimate for  $\{f_{n-m,s_i} ; 1 \leq i \leq N\}$  and measure  $h$ . Their main theorem is the following.

**Theorem 12** ([DL96]) *Let  $K$  be nonnegative, Lipschitz and nonzero only in  $[-1, 1]$ . Let  $a_n, b_n$  be such that  $na_n \rightarrow 0, b_n \rightarrow \infty$  as  $n \rightarrow \infty$ . Assume that  $\delta_n = \frac{c}{\sqrt{n}}$*

and that  $\ln \frac{b_n}{a_n} \leq c'n^a$  where  $c, c', a > 0$  are constants. If

$$\frac{m}{n} \rightarrow 0 \text{ and } \frac{m}{n^{4/5} \ln n} \rightarrow \infty \text{ as } n \rightarrow \infty,$$

then the estimate  $f_{n-m, s^*}$  satisfies

$$\sup_g \limsup_{n \rightarrow \infty} \frac{E[\|f_{n-m, s^*} - g\|_1]}{\inf_{s > 0} E[\|f_{n, s} - g\|_1]} \leq 3. \quad (23)$$

**Observation 13** For  $a_n, b_n, \delta_n, a$  as in Theorem 12,  $N = \Theta(n^{1/2+a})$ .

We can replace minimum distance with the minimum loss-weight estimate (Algorithm 4b) in this setting. Simply define  $\hat{s}$  to be  $s_i$  ( $1 \leq i \leq N$ ) such that  $f_{n-m, s_i}$  is the efficient minimum loss-weight estimate for  $\{f_{n-m, s_i}; 1 \leq i \leq N\}$  and measure  $h$ . This requires the computation of  $L_1$  distances between all  $O(N^2)$  pairs of densities. Assume however that the kernel  $K$  is such that we are able to compute approximate estimates  $D_{i,j}$ ,  $1 \leq i, j \leq N$  such that with probability at least  $1 - \delta$ ,

$$\forall i, j, (1 - \varepsilon)D_{ij} \leq \|f_{n-m, s_i} - f_{n-m, s_j}\|_1 \leq (1 + \varepsilon)D_{ij} \quad (24)$$

We can now define the approximate minimum loss-weight estimate  $\hat{s}'$  in the same way we defined  $\hat{s}$ . In other words,  $\hat{s}'$  is  $s_i$  such that Algorithm 4b outputs  $f_{n-m, s_i}$  for the class  $\{f_{n-m, s_i}; 1 \leq i \leq N\}$  and the measure  $h$ , except that it uses  $D_{ij}$  instead of  $\|f_{n-m, s_i} - f_{n-m, s_j}\|_1$  for each  $i, j$ . The following theorem is the analogue of Theorem 12 for both  $\hat{s}$  and  $\hat{s}'$ .

**Theorem 14** Let  $K, a_n, b_n, \delta_n, a > 0, m$  be as in Theorem 12. Then  $\hat{s}$  satisfies

$$\sup_g \limsup_{n \rightarrow \infty} \frac{E[\|f_{n-m, \hat{s}} - g\|_1]}{\inf_{s > 0} E[\|f_{n, s} - g\|_1]} \leq 3. \quad (25)$$

Moreover, if

$$\varepsilon \rightarrow 0 \text{ and } \frac{\delta}{n^{-2/5}} \rightarrow 0 \text{ as } n \rightarrow \infty$$

then  $\hat{s}'$  satisfies

$$\sup_g \limsup_{n \rightarrow \infty} \frac{E[\|f_{n-m, \hat{s}'} - g\|_1]}{\inf_{s > 0} E[\|f_{n, s} - g\|_1]} \leq 3. \quad (26)$$

The proof of Theorem 14 is identical to that of Theorem 12, except that the use of Theorem 3 needs to be replaced by Theorem 8 for (25), and by Corollary 11 for (26).

Finally we state a lemma which shows, using ideas from [Ind06] and [LHC07], that it is indeed possible to efficiently compute approximate estimates  $D_{ij}$  satisfying (24) (with confidence  $\delta$ ) when the kernel  $K$  is the uniform distribution on  $[-1, 1]$ .

**Lemma 15** Let the kernel  $K$  be the uniform distribution on  $[-1, 1]$ . Let  $\varepsilon, \delta \in (0, 1)$ . Then there is a randomized algorithm which in time  $O((1/\varepsilon)^2(nN + N^2) \log(nN/\delta))$  computes  $D_{ij}$  for  $i, j \in [N]$  such that with probability  $\geq 1 - \delta$  we have that for all  $i, j \in [N]$

$$(1 - \varepsilon)D_{ij} \leq \|f_{n-m, s_i} - f_{n-m, s_j}\|_1 \leq (1 + \varepsilon)D_{ij}.$$

**Proof :**

Follows immediately from Lemma 17. ■

Let us analyze the time required for computing  $\hat{s}'$  for the uniform kernel. Let  $T_{ij}$  denote the test function for  $f_{n-m, s_i}, f_{n-m, s_j}$ . If we sort  $x_1, \dots, x_{n-m}$  (using  $O(n \log n)$  time) in the preprocessing step then computing the inner product  $f_{n-m, s_i} \cdot T_{ij}$  for any  $i, j$  requires only  $O(n)$  time. Computing  $T_{ij}$  at any point in  $\mathbb{R}$  takes  $O(\log n)$  time (using a single binary search). Hence computing the inner product  $h \cdot T_{ij}$  can be done in  $O(m \log n)$  time.

So the preprocessing time

$$O((1/\varepsilon)^2(nN + N^2) \log(nN/\delta) + n \log n)$$

dominates the running time of the rest of the procedure, which is

$$O((n + m \log n)N).$$

Choosing  $\varepsilon = 1/\log n$  and  $\delta = 1/\sqrt{n}$  yields a running time of  $O((nN + N^2)\text{polylog}(n))$ . In contrast, modified minimum distance requires  $N^2(m \log n + n)$  time while the minimum distance estimate requires  $N^3(m \log n + n)$  time, both of which are much slower since in Theorem 12,  $m = \Omega(n^{4/5})$ .

### 3.2 Efficient approximation of $L_1$ -distances using projections.

Our main tool will be the following result of [LHC07] (for related work see also [Ind06]).

**Lemma 16 (Lemma 8 of [LHC07])** Let  $v_1, \dots, v_N \in \mathbb{R}^M$ . Let  $\varepsilon, \delta \in (0, 1)$ . Let

$$d \geq 11(2 \log N - \log \delta)/\varepsilon^2$$

be an integer. Let  $R$  be an  $d \times M$  matrix whose entries are i.i.d. from the Cauchy distribution  $C(0, 1)$ . Let  $w_i = Rv_i$  for  $i \in [N]$ . Let  $D_{ij}$  be the geometric mean of the coordinates of  $|w_i - w_j|$ . With probability  $\geq 1 - \delta$  (over the choice of the entries in  $R$ ) we have for all pairs  $i, j \in [N]$

$$(1 - \varepsilon)D_{ij} \leq \|v_i - v_j\|_1 \leq (1 + \varepsilon)D_{ij}. \quad (27)$$

As an immediate consequence of Lemma 16 we obtain an efficient algorithm for approximating all pairwise  $L_1$ -distances between  $N$  densities each of which is a mixture of  $n$  uniform distributions on intervals.

**Lemma 17** Let  $n$  and  $N$  be positive integers. Let  $\varepsilon, \delta \in (0, 1)$ . For each  $i \in [N]$  let  $f_i$  be a mixture of  $n$  uniform densities on intervals ( $f_i$  is given by a set of  $n$  mixture coefficients  $\alpha_{i,1}, \dots, \alpha_{i,n}$  and  $n$  disjoint intervals  $[a_{i,1}, b_{i,1}], \dots, [a_{i,n}, b_{i,n}]$ ). There is a randomized algorithm which in time  $O((1/\varepsilon)^2(nN + N^2) \log(nN/\delta))$  computes  $D_{ij}$  (for  $i, j \in [N]$ ) such that with probability  $\geq 1 - \delta$  we have that for all  $i, j \in [N]$

$$(1 - \varepsilon)D_{ij} \leq \|f_i - f_j\|_1 \leq (1 + \varepsilon)D_{ij}. \quad (28)$$

**Proof :**

Let  $S = s_0 < s_1 < \dots < s_M$  be the sequence obtained by sorting the set

$$\{a_{i,j}; i \in [N], j \in [n]\} \cup \{b_{i,j}; i \in [N], j \in [n]\}.$$

Note that  $M < 2Nn$ . Let  $v_i \in \mathbb{R}^M$  be the vector whose  $j$ -th coordinate is the measure of  $[s_{j-1}, s_j]$  under  $f_i$ . We have  $\|f_i - f_j\|_1 = \|v_i - v_j\|_1$  for all  $i, j \in [N]$ . Now we will apply Lemma 16 to  $v_1, \dots, v_N$ .

Let  $d = \lceil 11(2 \log 2nN - \log \delta) / \varepsilon^2 \rceil$ . Let  $R$  be an  $d \times M$  matrix whose entries are i.i.d. from the Cauchy distribution  $C(0, 1)$ . We can compute  $R$  in time  $O(dM)$ . Suppose that we computed  $w_i = Rv_i$  for  $i \in [N]$ . Then we can compute  $D_{ij}$ , the coordinate mean of  $|w_i - w_j|$  for all  $i, j \in [N]$  in time  $O(N^2d)$ . The equation (27) and the fact that  $\|f_i - f_j\|_1 = \|v_i - v_j\|_1$  implies (28). It remains to show how to compute  $w_i = Rv_i$  efficiently.

The  $j$ -th coordinate of  $v_i$  is the measure of  $[s_{j-1}, s_j]$  under  $f_i$  which is  $(s_j - s_{j-1})$  times the density of  $f_i$  on the interval  $[s_{j-1}, s_j]$  (the density of  $f_i$  is constant on this interval). Let  $R'$  be obtained from matrix  $R$  by multiplying  $j$ -th column by  $(s_j - s_{j-1})$  for  $j \in [M]$ . We can obtain  $R'$  from  $R$  in time  $O(dM)$ . Let  $R''$  be the matrix with  $R''_{ij} = R'_{i1} + R'_{i2} + \dots + R'_{ij}$  (again we can compute  $R''$  from  $R'$  in time  $O(dM)$ ). We have

$$(Rv_i)_k = \sum_{j=1}^n \frac{\alpha_{ij}}{b_{ij} - a_{ij}} \left( R''_{k,r(b_{ij})} - R''_{k,r(a_{ij})-1} \right). \quad (29)$$

Using equation (29) we can compute all  $v_i$  in time  $O(nNd)$ . ■

**Remark 18** In a forthcoming paper [MŠ08] we generalize Lemma 17 to piecewise polynomial densities. For each  $i \in [N]$ , let density  $f_i$  be specified by  $n$  disjoint intervals

$$[a_{i,1}, b_{i,1}), \dots, [a_{i,n}, b_{i,n}),$$

and in interval  $[a_{i,j}, b_{i,j})$  for each  $j \in [n]$  by coefficients  $\alpha_{i,j}^{(0)}, \alpha_{i,j}^{(1)}, \dots, \alpha_{i,j}^{(d)}$  such that

$$(\forall x \in [a_{i,j}, b_{i,j})) f(x) = \alpha_{i,j}^{(0)} + \alpha_{i,j}^{(1)}x + \dots + \alpha_{i,j}^{(d)}x^d.$$

Theorem 5.1 of [MŠ08] states that there is a randomized algorithm which takes  $O(N(N+n)(\frac{d}{\varepsilon})^3 \log \frac{N}{\delta})$  time and outputs  $D_{ij}$ ,  $1 \leq i < j \leq N$  such that with probability at least  $1 - \delta$ , for each  $1 \leq i < j \leq N$

$$(1 - \varepsilon)D_{ij} \leq \|f_i - f_j\|_1 \leq (1 + \varepsilon)D_{ij}.$$

### 3.3 Bin-width selection for histogram estimates

Here we show how the efficient minimum loss-weight estimate yields a fast algorithm for finding the optimal bin-width of 1-dimensional histograms. The set of densities arising in this problem will be such that for any subset of them it will be trivial to determine the pair whose  $L_1$ -distance is maximal.

Given a bin-width  $s > 0$ , define  $A_t$  for each integer  $t$  to be the interval  $[ts, (t+1)s)$ . Given  $n$  sample points  $x_1, \dots, x_n \in \mathbb{R}$  drawn from a distribution  $g$ , a regular

histogram estimate  $f_{n,s}$  is defined as the density such that for each  $t$  and each  $x \in A_t$

$$f_{n,s}(x) = \frac{|\{x_i; x_i \in A_t\}|}{ns}. \quad (30)$$

Devroye and Lugosi [DL01, DG85] consider the problem of finding  $L_1$ -optimal histogram estimates. As in the case of kernel estimates, they use the first  $n - m$  sample points  $x_1, \dots, x_{n-m}$  to define the histogram estimate  $f_{n-m,s}$ , and the remaining points  $x_{n-m+1}, \dots, x_n$  to define the empirical distribution  $h$ . Now, given a set  $\Theta$  to choose from,  $s^*$  is defined to be the bin-width such that  $f_{n-m,s^*}$  is the minimum distance estimate for  $\{f_{n-m,s}; s \in \Theta\}$  and  $h$ . If each width in  $\Theta$  is  $2^k$  for some integer  $k$ , Devroye and Lugosi [DL01] prove the following about  $s^*$ .

**Theorem 19** ([DL01], Theorem 10.3 and Lemma 10.5) *If  $\Theta \subseteq \{2^i; i \in \mathbb{Z}\}$  then for all  $n$  and  $m$ , with  $0 < m \leq n/2$ ,*

$$E[\|f_{n-m,s^*} - g\|_1] \leq 3 \inf_{s \in \Theta} E[\|f_{n,s} - g\|_1] \left( 1 + \frac{2m}{n-m} + 8\sqrt{\frac{m}{n}} \right) + 8\sqrt{\frac{\log(2(m+1)n^2)}{m}} + \frac{3}{n}.$$

Once again, like kernel estimates, we can simply use efficient minimum loss-weight instead of minimum distance. Now, define  $\hat{s}$  to be such that  $f_{n-m,\hat{s}}$  is the efficient minimum loss-weight estimate (Algorithm 4b) for  $\{f_{n-m,s}; s \in \Theta\}$  and  $h$ .

We state below the analogue of Theorem 19 for the efficient minimum loss-weight estimate. The proof is the same, except, one uses Theorem 8 instead of Theorem 3.

**Theorem 20** *If  $\Theta$  is as in Theorem 19 then for all  $n$  and  $m$  with  $0 < m \leq n/2$ ,*

$$E[\|f_{n-m,\hat{s}} - g\|_1] \leq 3 \inf_{s \in \Theta} E[\|f_{n,s} - g\|_1] \left( 1 + \frac{2m}{n-m} + 8\sqrt{\frac{m}{n}} \right) + 8\sqrt{\frac{\log(2(m+1)n^2)}{m}} + \frac{3}{n}.$$

Let us now consider the computational cost. For each  $n$ , let's say we choose  $\Theta$  to be  $\{2^i; -N \leq i \leq N\}$  (where, e. g.,  $N = n$  is a possible choice) so that we have  $2N + 1$  densities to select from. Define  $s_i = 2^{-N+i}$  for each  $0 \leq i \leq 2N$ . The following lemma shows that we need not actually pre-compute pairwise  $L_1$ -distances in the preprocessing step of Algorithm 4b.

**Lemma 21** *For any  $i \leq k \leq \ell \leq j$ ,*

$$\|f_{n,s_\ell} - f_{n,s_k}\|_1 \leq \|f_{n,s_j} - f_{n,s_i}\|_1.$$

**Proof :**

We first prove that for any  $n$  and  $i < j$ ,

$$\|f_{n,s_j} - f_{n,s_{i+1}}\|_1 \leq \|f_{n,s_j} - f_{n,s_i}\|_1, \quad (31)$$

and

$$\|f_{n,s_{j-1}} - f_{n,s_i}\|_1 \leq \|f_{n,s_j} - f_{n,s_i}\|_1. \quad (32)$$

In order to prove (31), consider any bin

$$A_t = [ts_{i+1}, (t+1)s_{i+1}] = [2ts_i, 2(t+1)s_i].$$

Denote the density of  $f_{n,s_j}$  in this bin by  $\mu$ , and that of  $f_{n,s_i}$  in  $[2ts_i, (2t+1)s_i], [(2t+1)s_i, (2t+2)s_i]$  respectively by  $\mu_1, \mu_2$ . Clearly the density of  $f_{n,s_{i+1}}$  in  $A_t$  is  $\frac{\mu_1 + \mu_2}{2}$ . However,

$$\begin{aligned} \int_{A_t} |f_{n,s_j} - f_{n,s_i}| &= s_i(|\mu - \mu_1| + |\mu - \mu_2|) \\ &\geq 2s_i \left| \mu - \frac{\mu_1 + \mu_2}{2} \right| \\ &= \int_{A_t} |f_{n,s_j} - f_{n,s_{i+1}}|. \end{aligned}$$

Thus

$$\begin{aligned} \|f_{n,s_j} - f_{n,s_i}\|_1 &= \sum_t \int_{A_t} |f_{n,s_j} - f_{n,s_i}| \geq \\ \sum_t \int_{A_t} |f_{n,s_j} - f_{n,s_{i+1}}| &= \|f_{n,s_j} - f_{n,s_{i+1}}\|_1. \end{aligned}$$

The proof of (32) is similar. The lemma now follows by induction.  $\blacksquare$

So in each iteration of Algorithm 4b, the pair of densities that are picked for comparison simply correspond to the smallest and the largest bin-widths remaining to be considered. In other words, if  $s_i$  and  $s_j$  are respectively the minimum and the maximum width remaining,  $f_{n-m,s_i}$  is compared against  $f_{n-m,s_j}$ .

Now let  $T_{ij}$  denote, as usual, the test function for  $f_{n-m,s_i}, f_{n-m,s_j}$ . Now we analyze the time needed to compute  $f_{n-m,s_i} \cdot T_{ij}$  and  $h \cdot T_{ij}$ . We first preprocess  $x_1, \dots, x_{n-m}$  by sorting them ( $O(n \log n)$  time). For any  $x$  the value of  $T_{ij}(x)$  can be computed in time  $O(\log n)$  (using binary search on  $x_1, \dots, x_{n-m}$ ) and hence  $h \cdot T_{ij}$  can be computed in  $O(m \log n)$  time. We can compute  $f_{n-m,s_i} \cdot T_{ij}$  in  $O(n)$  time (using one pass over the array  $x_1, \dots, x_{n-m}$ ).

Hence the efficient minimum loss-weight estimate requires only  $O(N(n+m \log n) + n \log n)$  computations in total. In contrast, modified minimum distance requires  $O(N^2(n+m \log n) + n \log n)$  and minimum distance requires  $O(N^3(n+m \log n) + n \log n)$ , making efficient minimum loss-weight the fastest of the three.

## 4 Randomized algorithm and mixtures

In this section we explore the following question: can constant 3 be improved if we allow randomized algorithms? Let  $f$  be the output of a randomized algorithm ( $f$  is a random variable with values in  $\mathcal{F}$ ). We would like to bound the expected error  $\mathbb{E}[\|f - g\|_1]$ , where the expectation is taken only with respect to coin tosses made by the algorithm (and *not* with respect to the distribution of the samples).

If instead of randomization we consider algorithms which output mixtures of densities in  $\mathcal{F}$  we obtain a

related problem. Indeed, let  $\alpha$  be the distribution on  $\mathcal{F}$  produced by a randomized algorithm, and let  $r = \sum_{s \in \mathcal{F}} \alpha_s s$  be the corresponding mixture. Then, by triangle inequality, we have

$$\|r - g\|_1 \leq \mathbb{E}[\|f - g\|_1].$$

Hence the model in which the output is allowed to be a mixture of densities in  $\mathcal{F}$  is “easier” than the model in which the density selection algorithm is randomized.

We consider here only the special case in which  $\mathcal{F}$  has only two densities  $f_1, f_2$ , and give an randomized algorithm with a better guarantee than is possible for deterministic algorithms. Later, in Section 5, we give a matching lower bound in the mixture model.

To simplify the exposition we will, without loss of generality, assume that  $\|f_1 - f_2\|_1 > 0$ . Thus for any  $h$  we have  $(f_1 - h) \cdot T_{12} + (h - f_2) \cdot T_{12} = \|f_1 - f_2\|_1 > 0$ .

### Algorithm 5 - RANDOMIZED ESTIMATE.

Let

$$r = \frac{|(f_1 - h) \cdot T_{12}|}{|(f_2 - h) \cdot T_{12}|}.$$

With probability  $1/(r+1)$  output  $f_1$ , otherwise output  $f_2$ .

(By convention, if  $|(f_2 - h) \cdot T_{12}| = 0$  then we take  $r = \infty$  and output  $f_2$  with probability 1).

**Theorem 22** *Let  $\mathcal{F} = \{f_1, f_2\}$ . Let  $f \in \mathcal{F}$  be the density output by Algorithm 5. Then*

$$\mathbb{E}[\|f - g\|_1] \leq 2 d_1(g, \mathcal{F}) + \Delta,$$

where the expectation is taken only with respect to the coin tosses made by the algorithm.

**Proof :**

Without loss of generality assume that

$$f_2 = \operatorname{argmin}_{f \in \mathcal{F}} \|f - g\|_1.$$

First we bound the error of  $f_1$  and later use it to bound the error of  $f$ . We have, by triangle inequality,

$$\|f_1 - g\|_1 \leq \|f_1 - f_2\|_1 + \|f_2 - g\|_1.$$

We can bound  $\|f_1 - f_2\|_1$  as follows

$$\begin{aligned} \|f_1 - f_2\|_1 &= (f_1 - f_2) \cdot T_{12} \\ &\leq |(f_1 - h) \cdot T_{12}| + |(f_2 - h) \cdot T_{12}| \\ &= (r+1)|(f_2 - h) \cdot T_{12}| \\ &\leq (r+1)|(f_2 - g) \cdot T_{12}| + (r+1)|(g - h) \cdot T_{12}|. \end{aligned}$$

Thus,

$$\|f_1 - g\|_1 \leq (r+2)\|f_2 - g\|_1 + (r+1)\Delta. \quad (33)$$

Hence

$$\begin{aligned} \mathbb{E}[\|f - g\|_1] &= \frac{1}{r+1}\|f_1 - g\|_1 + \frac{r}{r+1}\|f_2 - g\|_1 \\ &\leq 2\|f_2 - g\|_1 + \Delta \end{aligned}$$

where in the last inequality we used (33).  $\blacksquare$

## 5 Lower bound examples

In this section we construct an example showing that deterministic density selection algorithms based on test-functions cannot improve on the constant 3, that is, Theorems 2, 3, 4, 8 are tight. For algorithms that output mixtures (and hence randomized algorithms) the example yields a lower bound of 2, matching the constant in Theorem 22.

**Lemma 23** *For every  $\varepsilon' > 0$  there exist distributions  $f_1, f_2$ , and  $g = h$  such that*

$$\|f_1 - g\|_1 \geq (3 - \varepsilon')\|f_2 - g\|_1,$$

and  $f_1 \cdot T_{12} = -f_2 \cdot T_{12}$  and  $h \cdot T_{12} = 0$ .

Before we prove Lemma 23 let us see how it is applied. Consider the behavior of the algorithm on empirical distribution  $h$  for  $\mathcal{F} = \{f_1, f_2\}$  and  $\mathcal{F}' = \{f'_1, f'_2\}$ , where  $f'_1 = f_2$  and  $f'_2 = f_1$ . Note that  $T'_{12} = T_{21} = -T_{12}$  and hence

$$f'_1 \cdot T'_{12} = -f'_2 \cdot T'_{12} = f_1 \cdot T_{12} = -f_2 \cdot T_{12}.$$

Moreover, we have  $h \cdot T_{12} = h \cdot T'_{12} = 0$ . Note that all the test-functions have the same value for  $\mathcal{F}$  and  $\mathcal{F}'$ . Hence a test-function based algorithm either outputs  $f_1$  and  $f'_1$ , or it outputs  $f_2$  and  $f'_2 = f_1$ . In both cases it outputs  $f_1$  for one of the inputs and hence we obtain the following consequence.

**Corollary 24** *For any  $\varepsilon > 0$  and any deterministic test-function based algorithm there exist an input  $\mathcal{F}$  and  $h = g$  such that the output  $f_1$  of the algorithm satisfies  $\|f_1 - g\|_1 \geq (3 - \varepsilon)d_1(g, \mathcal{F})$ .*

**Proof of Lemma 23:**

Consider the following probability space consisting of 4 atomic events  $A_1, A_2, A_3, A_4$ :

	$A_1$	$A_2$	$A_3$	$A_4$
$f_1$	0	$1/4 + \varepsilon$	$1/2$	$1/4 - \varepsilon$
$f_2$	$1/2 + \varepsilon$	$1/4 - \varepsilon$	0	$1/4$
$g = h$	$1/2$	$1/2$	0	0
$T_{12}$	-1	1	1	-1

Note that we have  $f_1 \cdot T_{12} = -f_2 \cdot T_{12} = \frac{1}{2} + 2\varepsilon$ , and  $\|f_1 - g\|_1 = \frac{3}{2} - 2\varepsilon$ ,  $\|f_2 - g\|_1 = \frac{1}{2} + 2\varepsilon$ . The ratio  $\|f_1 - g\|_1/\|f_2 - g\|_1$  gets arbitrarily close to 3 as  $\varepsilon$  goes to zero. ■

Consider  $f_1$  and  $f_2$  from the proof of Lemma 23. Let  $f = \alpha f_1 + (1 - \alpha)f_2$  where  $\alpha \geq 1/2$ . For  $0 < \varepsilon < 1/4$  we have  $\|f - g\|_1 = 1/2 + \alpha - 2\varepsilon\alpha \geq 1 - 2\varepsilon$ . By symmetry, for one of  $\mathcal{F} = \{f_1, f_2\}$  and  $\mathcal{F}' = \{f'_1, f'_2\}$  (with  $f'_1 = f_2$  and  $f'_2 = f_1$ ), the algorithm outputs  $\alpha f_1 + (1 - \alpha)f_2$  with  $\alpha \geq 1/2$ , and hence we obtain the following.

**Corollary 25** *For any  $\varepsilon > 0$  and any deterministic test-function based algorithm which outputs a mixture there exist an input  $\mathcal{F}$  and  $h = g$  such that the output  $f$  of the algorithm satisfies  $\|f - g\|_1 \geq (2 - \varepsilon)d_1(g, \mathcal{F})$ .*

Thus for two distributions the correct constant is 2 for randomized algorithms using test-functions. For larger families of distributions we do not know what the value of the constant is (we only know that it is from the interval  $[2, 3]$ ).

**Question 26** *What is the correct constant for deterministic test-function based algorithm which output a mixture? What is the correct constant for randomized test-function based algorithms?*

Next we construct an example showing that 9 is the right constant for Algorithm 1.

**Lemma 27** *For every  $\varepsilon' > 0$  there exist probability distributions  $f_1, f_2, f_3 = f'_3$  and  $g$  such that*

$$\|f_1 - g\|_1 \geq (9 - \varepsilon')\|f_2 - g\|_1,$$

yet the Algorithm 1, for  $\mathcal{F} = \{f_1, f_2, f_3, f'_3\}$ , even when given the true distribution (that is,  $h = g$ ) outputs  $f_1$ .

**Proof :**

Consider the following probability space with 6 events  $A_1, \dots, A_6$  and  $f_1, f_2$  and  $g$  with the probabilities given by the following two tables:

	$A_1$	$A_2$	$A_3$
$g = h$	$2/3 - 21\varepsilon$	$1/9 - 2\varepsilon$	$9\varepsilon$
$f_1$	0	$18\varepsilon$	$2/3 - 12\varepsilon$
$f_2$	$2/3 - 30\varepsilon$	0	0
$f_3$	$2/3 - 21\varepsilon$	$9\varepsilon$	$9\varepsilon$
$T_{12}$	-1	1	1
$T_{13}$	-1	1	1
$T_{23}$	-1	-1	-1

	$A_4$	$A_5$	$A_6$
$g = h$	0	$2/9 + 14\varepsilon$	0
$f_1$	$2/9 - 13\varepsilon$	$9\varepsilon$	$1/9 - 2\varepsilon$
$f_2$	0	$2/9 + 14\varepsilon$	$1/9 + 16\varepsilon$
$f_3$	$2/9 - 4\varepsilon$	0	$1/9 + 7\varepsilon$
$T_{12}$	1	-1	-1
$T_{13}$	-1	1	-1
$T_{23}$	-1	1	1

Note that we have

$$\begin{aligned} f_1 \cdot T_{12} &= 7/9 - 14\varepsilon, & h \cdot T_{12} &= -7/9 + 14\varepsilon, \\ f_2 \cdot T_{12} &= -1, & f_1 \cdot T_{13} &= 1/3 + 30\varepsilon, \\ h \cdot T_{13} &= -1/3 + 42\varepsilon, & f_3 \cdot T_{13} &= -1 + 36\varepsilon, \\ f_2 \cdot T_{23} &= -1/3 + 60\varepsilon, & h \cdot T_{23} &= -5/9 + 28\varepsilon, \\ f_3 \cdot T_{23} &= -7/9 + 14\varepsilon. \end{aligned}$$

Hence  $f_1$  wins over  $f_3$ ,  $f_3$  wins over  $f_2$ , and  $f_2$  wins over  $f_1$ . Since  $f_3 = f'_3$  we have that  $f_1$  is the tournament winner. Finally, we have  $\|f_1 - g\|_1 = 2 - 72\varepsilon$  and  $\|f_2 - g\|_1 = 2/9 + 32\varepsilon$ . As  $\varepsilon \rightarrow 0$  the ratio  $\|f_1 - g\|_1/\|f_2 - g\|_1$  gets arbitrarily close to 9. ■

## References

- [DG85] Luc Devroye and László Györfi. *Nonparametric density estimation: the  $L_1$  view*. Wiley series in probability and mathematical statistics. John Wiley & Sons, New York, 1985.
- [DGL02] Luc Devroye, László Györfi, and Gábor Lugosi. A note on robust hypothesis testing. *IEEE Transactions on Information Theory*, 48(7):2111–2114, 2002.
- [DL96] Luc Devroye and Gábor Lugosi. A universally acceptable smoothing factor for kernel density estimates. *Ann. Statist.*, 24(6):2499–2512, 1996.
- [DL97] Luc Devroye and Gábor Lugosi. Nonasymptotic universal smoothing factors, kernel complexity and Yatracos classes. *Ann. Statist.*, 25(6):2626–2637, 1997.
- [DL01] Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation*. Springer Series in Statistics. Springer-Verlag, New York, 2001.
- [DL04] Luc Devroye and Gábor Lugosi. Bin width selection in multivariate histograms by the combinatorial method. *Test*, 13:1–17, 2004.
- [Ind06] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM*, 53(3):307–323, 2006.
- [LHC07] Ping Li, Trevor J. Hastie, and Kenneth W. Church. Nonlinear estimators and tail bounds for dimension reduction. *Journal of Machine Learning Research*, 8:2497–2532, 2007.
- [MŠ08] Satyaki Mahalanabis and Daniel Štefankovič. Approximating  $l_1$ -distances between mixture distributions using random projections. *arXiv.org*, <http://arxiv.org/abs/0804.1170>, April 2008.
- [Sch47] Henry Scheffé. A useful convergence theorem for probability distributions. *Ann. Math. Statistics*, 18:434–438, 1947.
- [VČ71] Vladimir N. Vapnik and Alexey J. Červonenkis. The uniform convergence of frequencies of the appearance of events to their probabilities. *Teor. Verojatnost. i Primenen.*, 16:264–279, 1971.
- [Wol57] Jacob Wolfowitz. The minimum distance method. *The Annals of Mathematical Statistics*, 28:75–88, 1957.
- [Yat85] Yannis G. Yatracos. Rates of convergence of minimum distance estimators and Kolmogorov’s entropy. *Ann. Statist.*, 13(2):768–774, 1985.

---

# The Learning Power of Evolution

---

**Vitaly Feldman**

IBM Almaden Research Center  
650 Harry rd.  
San Jose, CA 95120  
vitaly@post.harvard.edu

**Leslie G. Valiant\***

Harvard University  
33 Oxford st.  
Cambridge, MA 02138  
valiant@seas.harvard.edu

It has been widely recognized that learning and evolution have the commonality of involving adaptive processes that once started do not need a programmer or designer. It is tempting to seek some mystical extra power in evolution, beyond that of learning, simply because of the apparently spectacular consequences of evolution that we see around us. However, such approaches have not succeeded to date.

In response to this situation one of the authors made the, apparently radical, suggestion that evolution is nothing other than a *constrained* form of computational learning. In [Val08] a notion of evolvability was defined in a similar spirit to the definition of learnability. The goal of the definition is to offer a rigorous basis for the analysis of evolution and for distinguishing between *efficient evolution* and evolution that is only realized in some exponentially far limit.

Before summarizing this framework we describe the following motivating concrete instance. Consider the 20,000 or so genes in the human genome. For each such gene the condition under which the protein corresponding to it is expressed, in terms of all the other proteins, is encoded in its regulatory region. In other words each of the 20,000 or so proteins is controlled by a function  $f$  of the other 20,000 or so proteins. The issue here is that if the function  $f$  is restricted to too small a class then it will not be expressive enough to perform the complex functions of biology. On the other hand, if the function is an arbitrary function, or from a too extensive a class, then no evolutionary algorithm will exist to maintain the viability of this genetic network of functions as environmental conditions change. The goal of this evolvability theory is, among other things, to understand how broad and expressive these functions can be allowed to be while still permitting their efficient evolution.

The following is an abbreviated summary of the basic definitional framework. Let  $X = \{0, 1\}^n$  be an  $n$ -dimensional space of experiences or *examples* (e.g. in the above instance the expression levels of the proteins), a set  $C$  of *functions* (e.g. the functions by which the expression level of each protein is determined in terms of the expression levels of the others), and a set  $R$  of *representations* of functions (e.g. the DNA strings of the genes). Also we define an *ideal function*  $f$ , which would define for each vector  $x \in X$  the *best value* from the viewpoint of the evolving organism. In the current instance, for each combination of expression levels

of the other proteins it would define the ideal expression level of the protein at hand. For simplicity here we discuss only Boolean functions with values in  $\{-1, 1\}$ . We define a distribution  $D$  over  $X$  that defines the relative probabilities of the various possible vectors  $x \in X$  that can occur. We define the *performance* of a representation  $r$  to be the correlation of  $r$  with the ideal function  $f$  taken over all points in  $X$  weighted according to  $D$ . Formally, we denote  $\text{Perf}_f(r, D) = \mathbf{E}_D[r(x) \cdot f(x)]$ . In addition, since the exact performance cannot be efficiently computed in many cases without exponential resources, we define the *empirical performance*  $\text{Perf}_f(r, D, s)$  of  $r$  on samples of size  $s$ . It is a random variable that equals  $\frac{1}{s} \sum_{i \leq s} (r(z_i) \cdot f(z_i))$  for  $z_1, z_2, \dots, z_s \in X$  chosen randomly and independently according to  $D$ . A representation  $r$  is good if it is similar to the ideal  $f$ , or  $\text{Perf}_f(r, D) \geq 1 - \epsilon$  for some small  $\epsilon > 0$ . An *evolutionary algorithm* is defined by a quadruple  $A = (R, \text{Neigh}, \mu, t)$  where:

- $R$  is a set of representations of functions over  $X$ ;
- $\text{Neigh}(r, \epsilon)$  is a function that for  $r \in R$ , equals the *neighborhood* of  $r$ , that is, the set of representations into which  $r$  randomly “mutates”. For all  $r$  and  $\epsilon$ ,  $r \in \text{Neigh}(r, \epsilon)$  and  $|\text{Neigh}(r, \epsilon)| \leq p_A(n, 1/\epsilon)$  for a fixed polynomial  $p_A$ .
- $\mu(r, r_1, \epsilon)$  is a function that for  $r \in R$  and  $r_1 \in \text{Neigh}(r, \epsilon)$ , gives the probability that  $r$  “mutates” into  $r_1$ ;
- $t(\epsilon)$  is the function that equals the *tolerance* of  $A$ . The tolerance determines the difference in performance that a “mutation” has to exhibit to be considered beneficial (or deleterious). The tolerance is bounded from below by a polynomial in  $1/n$  and  $\epsilon$ .

Functions  $\text{Neigh}$ ,  $\mu$ , and  $t$  all need to be computable by a randomized algorithm in time polynomial in  $n$  and  $1/\epsilon$ . The interpretation here is that for each genome the number of variants, determined by  $\text{Neigh}$ , that can be searched effectively is not unlimited, because the population at any time is not unlimited, but is polynomial bounded. But a significant number of experiences with each variant must be available so that differences in performance can be detected reliably.

We now describe the basic step of such an evolutionary algorithm, designed to model a step of evolution. For a function  $f$ , distribution  $D$ , evolutionary algorithm  $A =$

---

\*Supported by grants from the National Science Foundation NSF-CCF-04-32037 and NSF-CCF-04-27129.

$(R, \text{Neigh}, \mu, t)$ , a representation  $r \in R$ , accuracy  $\epsilon$ , and sample size  $s$ , the *mutator*  $\text{Mu}(f, D, A, r, \epsilon, s)$  is a random variable that takes a value  $r_1$  determined as follows. For each  $r' \in \text{Neigh}(r, \epsilon)$ , it first computes an empirical value of  $v(r') = \text{Perf}_f(r', D, s)$ . Let

$$\text{Bene} = \{r' \mid v(r') \geq v(r) + t(\epsilon)\}$$

and

$$\text{Neut} = \{r' \mid |v(r') - v(r)| < t(\epsilon)\}.$$

Then

- (i) if  $\text{Bene} \neq \emptyset$  then output  $r_1 \in \text{Bene}$  with probability

$$\mu(r, r_1, \epsilon) / \sum_{r' \in \text{Bene}} \mu(r, r', \epsilon);$$

- (ii) if  $\text{Bene} = \emptyset$  then output  $r_1 \in \text{Neut}$  with probability

$$\mu(r, r_1, \epsilon) / \sum_{r' \in \text{Neut}} \mu(r, r', \epsilon).$$

In this definition a distinction between beneficial and neutral mutations is made as revealed by a set of  $s$  experiments. If some beneficial mutations are available, one is chosen according to their relative probabilities assigned by  $\mu$ . If none is available then one of the neutral mutations is chosen according to their relative probabilities assigned by  $\mu$ . Since in our definition we insist that for all  $r$  and  $\epsilon$ ,  $r \in \text{Neigh}(r, \epsilon)$ ,  $r$  will always be empirically neutral, and hence  $\text{Neut}$  nonempty.

Finally we say that a class of functions  $C$  is *evolvable* over distribution  $D$  if there is an *evolutionary algorithm*  $A = (R, \text{Neigh}, \mu, t)$  that for any starting representation  $r_0 \in R$  and any ideal function  $f \in C$  will converge *efficiently* to a representation  $r$  whose performance is close to the performance of  $f$ . Formally, there exist polynomials  $s(n, 1/\epsilon)$  and  $g(n, 1/\epsilon)$  such that for every  $f \in C$ , every  $\epsilon > 0$ , and every  $r_0 \in R$ , with probability at least  $1 - \epsilon$ , a sequence  $r_0, r_1, r_2, \dots$ , where

$$r_i = \text{Mu}(f, D, A, r_{i-1}, \epsilon, s(n, 1/\epsilon))$$

will have  $\text{Perf}_f(r_{g(n, 1/\epsilon)}, D) > 1 - \epsilon$ .

The polynomial  $g(n, 1/\epsilon)$  upper bounds the number of generations needed for the evolution process. A concept class  $C$  is *evolvable* if it is *evolvable* over all distributions by a single evolutionary mechanism. We emphasize this by saying *distribution-independently* evolvable.

As in other computational models, such as Turing Machines, the question of how robust the model is under reasonable variations is an important one. Some results along these lines are known. These include the equivalence of evolvability with fixed tolerance  $t$  to evolvability with tolerance that might depend on  $r$  (see [Val08] for the definitions).

Initial results [Val08] say that monotone conjunctions are evolvable over the uniform distribution and that the evolvable is a subclass of the class that is learnable by Statistical Queries (SQ), defined earlier by Kearns [Kea98], which is known to be a proper subclass of the PAC learnable. Michael gives an algorithm for evolving decision trees over the uniform distribution that is based on a slightly different notion

of performance [Mic07]. Further, Feldman shows that evolvability is equivalent to learning by a natural restriction of statistical queries [Fel08], referred to as *correlational statistical queries*. A correlational statistical query (or CSQ) is a query for the correlation of a given function  $g$  with the unknown target function  $f$ . The correlation is measured relative to the distribution  $D$  over the domain of the learning problem and equals  $\mathbf{E}_{x \sim D}[f(x)g(x)]$ . To such a query a CSQ oracle returns an estimate of  $\mathbf{E}_{x \sim D}[f(x)g(x)]$  within certain tolerance. For comparison, the general SQ model allows queries that provide estimates of  $\mathbf{E}_{x \sim D}[\psi(x, f(x))]$  for any function on labeled examples  $\psi : \{0, 1\}^n \times \{-1, 1\} \rightarrow \{-1, 1\}$ . This equivalence implies that every concept class known to be SQ learnable is evolvable when the distribution over the domain is fixed. In addition, it was shown that decision lists are not evolvable distribution-independently [Fel08], and hence that the evolvable is a proper subclass of SQ.

## Open Problems

The main open problem in this direction is characterizing the power of distribution-independent evolvability. In particular, it is unknown whether conjunctions and low-weight linear threshold functions are evolvable distribution-independently. It is easy to see that both classes are weakly evolvable distribution-independently [Fel08] and hence a possible approach is to design an evolutionary variant of boosting (which may be of independent interest). Known boosting techniques rely heavily on information that is not available to an evolutionary algorithm. Evolutionary algorithms for these basic concept classes that use simple mutation mechanisms and converge fast over wide classes of natural distributions would be of particular interest.

More generally, we think that it is important to seek new learning techniques that rely on evolutionary mechanisms of adaptation. Such techniques might shed new light on evolution as it has occurred on Earth and could also find applications outside of the biological context. Identifying such potential applications is another interesting avenue of research.

## References

- [Fel08] V. Feldman. Evolvability from learning algorithms. Manuscript. To appear in *Proceedings of STOC*, 2008.
- [Kea98] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.
- [Mic07] L. Michael. Evolving decision lists. Manuscript, 2007.
- [Val08] L. G. Valiant. Evolvability. To appear in *Journal of the ACM*, 2008.

---

# A Query Algorithm for Agnostically Learning DNF?

---

Parikshit Gopalan\* and Adam T. Kalai† and Adam R. Klivans‡

Let  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$  be an *arbitrary* Boolean function and let  $\mathcal{C}$  be a concept class where each concept has size at most  $t$ . Define

$$\text{opt} = \min_{c \in \mathcal{C}} \Pr_{x \in \{-1, 1\}^n} [c(x) \neq f(x)]$$

where  $x$  is chosen uniformly at random from  $\{-1, 1\}^n$ . We say that  $\mathcal{C}$  is *agnostically learnable with queries with respect to the uniform distribution* if there exists an algorithm that—given black box access to any  $f$ —runs in time  $\text{poly}(n, t, \epsilon^{-1})$  and outputs a hypothesis  $h$  such that

$$\Pr_{x \in \{-1, 1\}^n} [h(x) \neq f(x)] \leq \text{opt} + \epsilon.$$

The algorithm may be randomized, in which case it must output such an  $h$  with high probability.

The main question is as follows: are polynomial-size DNF formulas agnostically learnable with queries with respect to the uniform distribution? A related question is, are halfspaces agnostically learnable with queries with respect to the uniform distribution?

**Motivation:** One of the most celebrated results in computational learning theory is Jackson’s query algorithm for PAC learning DNF formulas with respect to the uniform distribution [3]. A natural question is whether DNF formulas can be learned (even with queries and with respect to the uniform distribution) in a highly noisy setting, i.e., the well-known agnostic framework of learning [5].

Additionally, it is straightforward to see that an agnostic learning algorithm for DNF formulas would give algorithms for weakly learning polynomial-size depth-3 circuits with respect to the uniform distribution in the standard PAC learning model.

Halfspaces are another simple and important concept class of functions still not known to be agnostically learnable with respect to the uniform distribution, even if the learner can make queries (although some relevant work exists for the uniform distribution that we mention below).

**Current status:** Very recently, Gopalan et al. [2] have shown that the weaker concept class of *decision trees* are

agnostically learnable with queries with respect to the uniform distribution. Their algorithm implicitly solves a high-dimensional convex program using the well-known Kushilevitz/Mansour [6] algorithm for finding large Fourier coefficients as a subroutine.

Applying a result due to Mansour on the sparsity of DNF formulas [7], the Gopalan et al. query algorithm will agnostically learn DNF formulas with respect to the uniform distribution in time  $n^{O(\log(1/\epsilon) \log \log n)}$ . If the Friedgut-Kalai “Entropy/Influence” conjecture [1] is true (or better bounds are proved on the sparsity of DNF formulas) then the running time can be improved even further (see also Gil Kalai’s post on Terry Tao’s weblog [8]).

Gopalan et al. do show, however, that their algorithm will *not* agnostically learn DNF formulas in polynomial time in all the relevant parameters.

Given Jackson’s algorithm and Gopalan et al.’s recent work for agnostically learning decision trees, we feel that the case of DNF formulas is particularly compelling.

Regarding halfspaces, Kalai et al. [4] showed how to agnostically learn halfspaces with respect to the uniform distribution *without queries* in time  $n^{O(1/\epsilon^4)}$ . Further, they showed that any algorithm running in time  $n^{O(1/\epsilon^{2-\gamma})}$  for any  $\gamma > 0$  would give the fastest known algorithm for the notoriously difficult “learning parity with noise problem.” As such, we do not think that much further progress will be made on this problem unless the learner is allowed to make queries.

## References

- [1] E. Friedgut and G. Kalai. Every monotone graph property has a sharp threshold. *Proceedings of the American Mathematical Society*, 124:2993–3002, 1996.
- [2] P. Gopalan, A. Kalai, and A. Klivans. Agnostically learning decision trees. In *Proceedings of the 40<sup>th</sup> ACM Symp. on Theory of Computing*, 2008.
- [3] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55:414–440, 1997.
- [4] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46<sup>th</sup> IEEE Symp. on Foundations of Computer Science*, 2005.
- [5] M. Kearns, R. Schapire, and L. Sellie. Toward Efficient

---

\*University of Washington; parik@cs.washington.edu

†Georgia Tech.; adamology@gmail.com

‡University of Texas at Austin; klivans@cs.utexas.edu

- Agnostic Learning. *Machine Learning*, 17(2/3):115–141, 1994.
- [6] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM J. on Computing*, 22(6):1331–1348, 1993.
- [7] Y. Mansour. An  $o(n^{\log \log n})$  learning algorithm for DNF under the uniform distribution. *Journal of Computer and System Sciences*, 50:543–550, 1995.
- [8] T. Tao. The entropy/influence conjecture (blog post by gil kalai). Available at <http://terrytao.wordpress.com/2007/08/16/gil-kalai-the-entropyinfluence-conjecture/>.

---

# Learning Rotations

---

Adam M. Smith and Manfred K. Warmuth  
University of California - Santa Cruz  
{amsmith,manfred}@cs.ucsc.edu

## Abstract

Many different matrix classes have been tackled recently using online learning techniques, but at least one major class has been left out: rotations. We pose the online learning of rotations as an open problem and discuss the importance of this problem.

## 1 Problem Statement

### Online Rotation Problem:

Given a stream of instances  $\mathbf{x}_t$ , which are unit vectors in  $\mathbb{R}^n$ , predict  $\hat{\mathbf{y}}_t = R_t \mathbf{x}_t$ , a rotated version of  $\mathbf{x}_t$ . Receive the true result vector  $\mathbf{y}_t$  (the result of some unknown rotation) and incur a loss  $L_t(R_t) = |R_t \mathbf{x}_t - \mathbf{y}_t|^2$ . Find an online algorithm with bounded regret with respect to the best rotation chosen offline.

## 2 Why study rotations?

We claim that the online rotation problem is both hard and interesting. The space of rotations (formally the  $SO(n)$  group) is a curved, compact manifold, ruling out the direct formation of linear combinations of rotations. Therefore designing updates for this parameter class is challenging.

From an application standpoint, many seemingly more general problems reduce to learning rotations. For example, using a conformal embedding (adding two special dimensions to application-level vectors), rotations naturally extend to a representation of all Euclidean transformations [WCL05]. Furthermore, learning rotations would bring us closer to representing general orthogonal transformations in the  $O(n, n)$  group. This group (with suitable embedding) provides a universal representation for *all Lie groups* including many matrix classes of interest that are currently treated individually [DHSA93]. Clearly, this is an interesting goal to pursue!

## 3 What needs to be explored?

There are several directions of inquiry that may lead to progress in this area. (1) Identify online algorithms that exploit the Lie group structure of the rotations. (2) Identify divergences that lead to suitable updates. (3) Identify alternative loss functions (other than the square loss between vectors used above) that better exploit spherical geometry. (4) Identify upper and lower regret bounds.

## 4 Related Work

The batch version of related problems have been solved in other domains. For example, a 3D Euclidean version, estimating spacecraft attitude, has been solved in the field of astronautics where it is known as Wahba's Problem [Wah65]. Also, in psychometrics, the Orthogonal Procrustes Problem of estimating the closest orthogonal matrix to a general matrix has been solved [Sch66].

Other matrix classes have been tackled successfully in the online learning model. For example, linear regression has been generalized to density matrix parameters (symmetric positive matrices of trace one) [TRW05]. Furthermore, the class of arbitrary matrices has been handled by an extension of these methods [War07]. Note that algorithms for arbitrary matrices are not immediately useful for learning rotations because they do not exploit the special structure of the rotation group and would require repeated projection and/or approximation.

For a teaser problem, consider learning rotations on the unit circle (the  $S^1$  group). What does your algorithm do when it observes a rotation that is the opposite of the best estimate?

## References

- [DHSA93] C. Doran, D. Hestenes, F. Sommen, and N. Van Acker. Lie groups as spin groups. *J. Math. Phys.*, 34(8):3642–3669, August 1993.
- [Sch66] P. Schonemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1), March 1966.
- [TRW05] K. Tsuda, G. Rätsch, and M. K. Warmuth. Matrix exponentiated gradient updates for on-line learning and Bregman projections. *Journal of Machine Learning Research*, 6:995–1018, June 2005.
- [Wah65] G. Wahba. Problem 65-1, a least squares estimate of satellite attitude. *SIAM Review*, 7(3), July 1965.
- [War07] Manfred K. Warmuth. Winnowing subspaces. Unpublished manuscript, February 2007.
- [WCL05] R. Wareham, J. Cameron, and J. Lasenby. Applications of conformal geometric algebra in computer vision and graphics. *6th International Workshop IWMM 2004*, pages 329–349, 2005.



## Author Index

### A

Abernethy, Jacob ..... 263, 415, 437  
 Ailon, Nir ..... 87  
 Angluin, Dana ..... 169  
 Aspnes, James ..... 169

### B

Balcan, Maria-Florina ..... 45, 287  
 Bartlett, Peter ..... 335, 415  
 Ben-David, Shai ..... 33, 379  
 Bernstein, Andrey ..... 323  
 Blais, Eric ..... 193  
 Blum, Avrim ..... 287

### C

Campbell, Colin ..... 217  
 Caramanis, Constantine ..... 467  
 Cavallanti, Giovanni ..... 251  
 Cesa-Bianchi, Nicolo ..... 251  
 Chaudhuri, Kamalika ..... 9, 21, 391  
 Chen, Jiang ..... 169  
 Choi, Sung-Soon ..... 123

### D

Dani, Varsha ..... 335  
 Dani, Varsha ..... 355  
 De Rooij, Steven ..... 275  
 Doliwa, Thorsten ..... 157

### E

Eisenstat, David ..... 169

### F

Feldman, Vitaly ..... 147, 513  
 Fukumizu, Kenji ..... 111

### G

Gentile, Claudio ..... 251  
 Gopalan, Parikshit ..... 515  
 Greenwald, Amy ..... 239  
 Gretton, Arthur ..... 111  
 Grunwald, Peter ..... 1  
 Gyorgy, Andras ..... 447

### H

Hanneke, Steve ..... 45  
 Hanson, Robin ..... 3  
 Hatano, Kohei ..... 69  
 Hayes, Thomas ..... 335, 355  
 Hazan, Elad ..... 57, 263  
 Holte, Robert ..... 135

### I

Ishibashi, Kosuke ..... 69

### J

Jung, Kyomin ..... 123

### K

Kakade, Sham ..... 335, 355, 403  
 Kalai, Adam ..... 515  
 Kale, Satyen ..... 57  
 Kallweit, Michael ..... 157  
 Kearns, Michael ..... 99  
 Khot, Subhash ..... 81  
 Kim, Jeong Han ..... 123  
 Klein, Dan ..... 5  
 Kleinberg, Robert D. .... 425  
 Klivans, Adam R. .... 515  
 Koltchinskii, Vladimir ..... 229  
 Koolen, Wouter M. .... 275

### L

Lafferty, John ..... 455  
 Lanckriet, Gert ..... 111  
 Lange, Steffen ..... 135  
 Li, Zheng ..... 239  
 Lu, Tyler ..... 33  
 Lugosi, Gabor ..... 7, 447

### M

Mahalanabis, Satyaki ..... 503  
 Mannor, Shie ..... 467  
 McGregor, Andrew ..... 391  
 Mohri, Mehryar ..... 87

### N

Niculescu-Mizil, Alexandru ..... 425  
 Nowack, Robert ..... 491

### O

O'Donnell, Ryan ..... 193  
 Ottucsak, Gyorgy ..... 447

### P

Pal, David ..... 33  
 Ponnuswami, Ashok Kumar ..... 81

### R

Rakhlin, Alexander ..... 263, 335, 415  
 Rao, Satish ..... 9, 21  
 Reyzin, Lev ..... 169  
 Ritov, Yaacov ..... 205  
 Rubinstein, Benjamin I. P. .... 299

## Author Index

Rubinstein, J. Hyam.....299

### S

Schudy, Warren .....239  
Scott, Clayton .....491  
Sellie, Linda.....181  
Shalev-Shwartz, Shai .....311  
Shamir, Ohad .....367  
Sharma, Yogeshwer .....425  
Shimkin, Nahum .....323  
Shoelkopf, Bernhard .....111  
Simon, Hans Ulrich .....157  
Singer, Yoram.....311  
Singh, Aarti.....491  
Slivkins, Aleksandrs .....343  
Smith, Adam M. ....517  
Srebo, Nathan .....287  
Sridharan, Karthik.....403  
Sriperumbudur, Bharath.....111  
Stefankovic, Daniel.....503

### T

Takeda, Masayuki.....69  
Tewari, Ambuj .....335, 415  
Tishby, Naftali .....367

### U

Upfal, Eli .....343

### V

Valiant, Leslie G.....513  
von Luxburg,Ulrike .....379

### W

Wang, Liwei .....479  
Warmuth, Manfred K.....437, 517  
Wasserman, Larry .....455  
Wimmer, Karl .....193  
Wortman, Jennifer .....45, 99

### Y

Yellin, Joel.....437  
Ying, Yiming .....217  
Yuan, Ming .....229

### Z

Zakai, Alon .....205  
Zhou, Shuheng .....455  
Zilles, Sandra .....135  
Zinkevich, Martin .....135