Robustness and Generalization

Huan Xu
The University of Texas at Austin
huan.xu@mail.utexas.edu

Shie Mannor
Technion, Israel Institute of Technology
shie@ee.technion.ac.il

Abstract

We derive generalization bounds for learning algorithms based on their robustness: the property that if a testing sample is "similar" to a training sample, then the testing error is close to the training error. This provides a novel approach, different from the complexity or stability arguments, to study generalization of learning algorithms. We further show that a weak notion of robustness is both sufficient and necessary for generalizability, which implies that robustness is a fundamental property for learning algorithms to work.

1 Introduction

The key issue in the task of learning from a set of observed samples is the estimation of the *risk* (i.e., generalization error) of learning algorithms. Typically, since the learned hypothesis *depends on* the training data, its empirical measurement (i.e., training error) provides an optimistically biased estimation, especially when the number of training samples is small. Several approaches have been proposed to bound the deviation of the risk from its empirical measurement, among which methods based on uniform convergence and stability are most widely used.

Uniform convergence of empirical quantities to their mean (e.g., Vapnik and Chervonenkis 1974; 1991) provides ways to bound the gap between the expected risk and the empirical risk by the complexity of the hypothesis set. Examples to complexity measures are the Vapnik-Chervonenkis (VC) dimension (e.g., Vapnik & Chervonenkis, 1991; Evgeniou et al., 2000), the fat-shattering dimension (e.g., Alon et al., 1997; Bartlett, 1998), and the Rademacher complexity (Bartlett & Mendelson, 2002; Bartlett et al., 2005). Another well-known approach is based on *stability*. An algorithm is stable if its output remains "similar" for different sets of training samples that are identical up to removal or change of a single sample. The first results that relate stability to generalizability track back to Devroye and Wagner (1979a; 1979b). Later, McDiarmid's concentration inequalities (McDiarmid, 1989) facilitated new bounds on generalization error (e.g., Bousquet & Elisseeff, 2002; Poggio et al., 2004; Mukherjee et al., 2006).

In this paper we explore a different approach which we term *algorithmic robustness*. Briefly speaking, an algorithm is robust if its solution has the following property: it achieves "similar" performance on a testing sample and a training sample that are "close". This notion of robustness is rooted in *robust optimization* (Ben-Tal & Nemirovski, 1998; Ben-Tal & Nemirovski, 1999; Bertsimas & Sim, 2004) where a decision maker aims to find a solution x that minimizes a (parameterized) cost function $f(x, \xi)$ with the knowledge that the unknown true parameter ξ may deviate from the observed parameter $\hat{\xi}$. Hence, instead of solving $\min_x f(x, \hat{\xi})$ one solves $\min_x [\max_{\tilde{\xi} \in \Delta} f(x, \tilde{\xi})]$, where Δ includes all possible realizations of ξ. Robust optimization was introduced in machine learning tasks to handle exogenous noise (e.g., Bhattacharyya et al., 2004; Shivaswamy et al., 2006; Globerson & Roweis, 2006), i.e., the learning algorithm only has access to inaccurate observation of training samples. Later on, Xu et al. (2010; 2009) showed that both Support Vector Machine(SVM) and Lasso have robust optimization interpretation, i.e., they can be reformulated as

$$\min_{h \in \mathcal{H}} \max_{(\delta_1, \cdots, \delta_n) \in \Delta} \sum_{i=1}^{n} l(h, z_i + \delta_i),$$

for some Δ. Here z_i are the observed training samples and $l(\cdot, \cdot)$ is the loss function (hinge-loss for SVM, and squared loss for Lasso), which means that SVM and Lasso essentially minimize the

empirical error under the worst possible perturbation. Indeed, as Xu et al. (2010; 2009) showed, this reformulation leads to requiring that the loss of a sample "close" to z_i is small, which further implies statistical consistency of these two algorithms. In this paper we adopt this approach and study the (finite sample) generalization ability of learning algorithms by investigating the loss of learned hypotheses on samples that slightly deviate from training samples.

Of special interest is that robustness is more than just another way to establish generalization bounds. Indeed, we show that a weaker notion of robustness is a *necessary and sufficient* condition of (asymptotic) generalizability of (general) learning algorithms. While it is known having a finite VC-dimension (Vapnik & Chervonenkis, 1991) or equivalently being CVEEE$_{loo}$ stable (Mukherjee et al., 2006) is necessary and sufficient for the Empirical Risk Minimization (ERM) to generalize, much less is known in the general case. Recently, Shalev-Shwartz et al. (2009) proposed a weaker notion of stability that is necessary and sufficient for a learning algorithm to be consistent and generalizing, provided that the problem itself is *learnable*. However, learnability requires that the *convergence rate is uniform* with respect to all distributions, and is hence a fairly strong assumption. In particular, the standard supervised learning setup where the hypothesis set is the set of measurable functions is *not* learnable since no algorithm can achieve a uniform convergence rate (cf. Devroye et al., 1996). Indeed, as Shalev-Shwartz et al. (2009) stated, for supervised learning problems learnability is equivalent to the generalizability of ERM, and hence reduce to the aforementioned results on ERM algorithms.

In particular, our main contributions are the following:

1. We propose a notion of algorithmic robustness. Algorithmic robustness is a desired property for a learning algorithm since it implies a lack of sensitivity to (small) disturbances in the training data.

2. Based on the notion of algorithmic robustness, we derive generalization bounds for IID samples.

3. To illustrate the applicability of the notion of algorithmic robustness, we provide some examples of robust algorithms, including SVM, Lasso, feed-forward neural networks and PCA.

4. We propose a weaker notion of robustness and show that it is both necessary and sufficient for a learning algorithm to generalize. This implies that robustness is an essential property needed for a learning algorithm to work.

Note that while stability and robustness are similar on an intuitive level, there is a difference between the two: stability requires that identical training sets with a single sample removed lead to similar prediction rules, whereas robustness requires that a prediction rule has comparable performance if tested on a sample close to a training sample. Simply put, stability compares two prediction rules, whereas robustness investigates one prediction rule.

This paper is organized as follows. We define the notion of robustness in Section 2, and prove generalization bounds for robust algorithms in Section 3. In Section 4 we propose a relaxed notion of robustness, which is termed as pseudo-robustness, and provide corresponding generalization bounds. Examples of learning algorithms that are robust or pseudo-robust are provided in Section 5. Finally, we show that robustness is necessary and sufficient for generalizability in Section 6. Due to space constraints, some of the proofs are deferred to the full version (Xu & Mannor, 2010).

1.1 Preliminaries

We consider the following general learning model: a set of training samples are given, and the goal is to pick a hypothesis from a hypothesis set. Unless otherwise mentioned, throughout this paper the size of training set is fixed as n. Therefore, we drop the dependence of parameters on the number of training samples, while it should be understood that parameters may vary with the number of training samples. We use $\mathcal{Z}$ and $\mathcal{H}$ to denote the set from which each sample is drawn, and the hypothesis set, respectively. Throughout the paper we use $\mathbf{s}$ to denote the training sample set consists of n training samples $(s_1, \cdots, s_n)$. A learning algorithm $\mathcal{A}$ is thus a mapping from $\mathcal{Z}^n$ to $\mathcal{H}$. We use $\mathcal{A}_{\mathbf{s}}$ to represent the hypothesis learned (given training set $\mathbf{s}$). For each hypothesis $h \in \mathcal{H}$ and a point $z \in \mathcal{Z}$, there is an associated loss $l(h, z)$. We ignore the issue of measurability and further assume that $l(h, z)$ is non-negative and upper-bounded uniformly by a scalar M.

In the special case of supervised learning, the sample space can be decomposed as $\mathcal{Z} = \mathcal{Y} \times \mathcal{X}$, and the goal is to learn a mapping from $\mathcal{X}$ to $\mathcal{Y}$, i.e., to predict the y-component given x-component. We hence use $\mathcal{A}_{\mathbf{s}}(x)$ to represent the prediction of $x \in \mathcal{X}$ if trained on $\mathbf{s}$. We call $\mathcal{X}$ the input space and $\mathcal{Y}$ the output space. The output space can either be $\mathcal{Y} = \{-1, +1\}$ for a classification problem, or $\mathcal{Y} = \mathbb{R}$ for a regression problem. We use $_{|x}$ and $_{|y}$ to denote the x-component and y-component

of a point. For example, $s_{i|x}$ is the x-component of s_i. To simplify notations, for a scaler c, we use $[c]^+$ to represent its non-negative part, i.e., $[c]^+ \triangleq \max(0, c)$.

We recall the following standard notion of covering number from van der Vaart and Wellner (2000).

Definition 1 (cf. van der Vaart & Wellner, 2000) *For a metric space S, ρ and $T \subset S$ we say that $\hat{T} \subset S$ is an* ϵ-cover *of T, if $\forall t \in T$, $\exists \hat{t} \in \hat{T}$ such that $\rho(t, \hat{t}) \leq \epsilon$. The* ϵ-covering number *of T is*

$$\mathcal{N}(\epsilon, T, \rho) = \min\{|\hat{T}| : \hat{T} \text{ is an } \epsilon - \text{cover of } T\}.$$

2 Robustness of Learning Algorithms

Before providing a precise definition of what we mean by "robustness" of an algorithm, we provide some motivating examples which share a common property: if a testing sample is close to a training sample, then the testing error is also close, a property we will later formalize as "robustness".

We first consider large-margin classifiers: Let the loss function be $l(A_{\mathbf{s}}, z) = \mathbf{1}(A_{\mathbf{s}}(z_{|x}) \neq z_{|y})$. Fix $\gamma > 0$. An algorithm $\mathcal{A}_{\mathbf{s}}$ has a margin γ if for $j = 1, \cdots, n$

$$\mathcal{A}_{\mathbf{s}}(x) = \mathcal{A}_{\mathbf{s}}(s_{j|x}); \quad \forall x : \|x - s_{j|x}\|_2 < \gamma.$$

That is, any training sample is at least γ away from the classification boundary.

Example 1 *Fix $\gamma > 0$ and put $K = 2\mathcal{N}(\gamma/2, \mathcal{X}, \|\cdot\|_2)$. If $\mathcal{A}_{\mathbf{s}}$ has a margin γ, then $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted by $\{C_i\}_{i=1}^K$, such that if s_j and $z \in \mathcal{Z}$ belong to a same C_i, then $|l(\mathcal{A}_{\mathbf{s}}, s_j) - l(\mathcal{A}_{\mathbf{s}}, z)| = 0$.*

Proof: By the definition of covering number, we can partition $\mathcal{X}$ into $\mathcal{N}(\gamma/2, \mathcal{X}, \|\cdot\|_2)$ subsets (denoted $\hat{X}_i$) such that each subset has a diameter less or equal to γ. Further, $\mathcal{Y}$ can be partitioned to $\{-1\}$ and $\{+1\}$. Thus, we can partition $\mathcal{Z}$ into $2\mathcal{N}(\gamma/2, \mathcal{X}, \|\cdot\|_2)$ subsets such that if z_1, z_2 belong to a same subset, then $y_{1|y} = y_{2|y}$ and $\|x_{1|y} - x_{2|y}\| \leq \gamma$. By the definition of the margin, this guarantees that if s_j and $z \in \mathcal{Z}$ belong to a same C_i, then $|l(\mathcal{A}_{\mathbf{s}}, s_j) - l(\mathcal{A}_{\mathbf{s}}, z)| = 0$. ■

The next example is a linear regression algorithm. Let the loss function be $l(A_{\mathbf{s}}, z) = |z_{|y} - A_{\mathbf{s}}(z_{|x})|$, and let $\mathcal{X}$ be a bounded subset of $\mathbb{R}^m$ and fix $c > 0$. The norm-constrained linear regression algorithm is

$$\mathcal{A}_{\mathbf{s}} = \min_{w \in \mathbb{R}^m : \|w\|_2 \leq c} \sum_{i=1}^{n} |s_{i|y} - w^\top s_{i|x}|, \tag{1}$$

i.e., minimizing the empirical error among all linear classifiers whose norm is bounded.

Example 2 *Fix $\epsilon > 0$ and let $K = \mathcal{N}(\epsilon/2, \mathcal{X}, \|\cdot\|_2) \times \mathcal{N}(\epsilon/2, \mathcal{Y}, |\cdot|)$. Consider the algorithm as in (1). The set $\mathcal{Z}$ can be partitioned into K disjoint sets, such that if s_j and $z \in \mathcal{Z}$ belong to a same C_i, then*

$$|l(\mathcal{A}_{\mathbf{s}}, s_j) - l(\mathcal{A}_{\mathbf{s}}, z)| \leq (c + 1)\epsilon.$$

Proof: Similarly to the previous example, we can partition $\mathcal{Z}$ to $\mathcal{N}(\epsilon/2, \mathcal{X}, \|\cdot\|_2) \times \mathcal{N}(\epsilon/2, \mathcal{Y}, |\cdot|)$ subsets, such that if z_1, z_2 belong to a same C_i, then $\|z_{1|x} - z_{2|x}\|_2 \leq \epsilon$, and $|z_{1|y} - z_{2|y}| \leq \epsilon$. Since $\|w\|_2 \leq c$, we have

$$\begin{aligned} |l(w, z_1) - l(w, z_2)| &= \big||z_{1|y} - w^\top z_{1|x}| - |z_{2|y} - w^\top z_{2|x}|\big| \\ &\leq \big|(z_{1|y} - w^\top z_{1|x}) - (z_{2|y} - w^\top z_{2|x})\big| \\ &\leq |z_{1|y} - z_{2|y}| + \|w\|_2 \|z_{1|x} - z_{2|x}\|_2 \\ &\leq (1 + c)\epsilon, \end{aligned}$$

whenever z_1, z_2 belong to a same C_i. ■

The two motivating examples both share a property: we can partition the sample set into finite subsets, such that if a new sample falls into the same subset as a training sample, then the loss of the former is close to the loss of the latter. We call an algorithm having this property "robust."

Definition 2 *Algorithm $\mathcal{A}$ is $(K, \epsilon(\mathbf{s}))$ robust if $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted by $\{C_i\}_{i=1}^K$, such that $\forall s \in \mathbf{s}$,*

$$s, z \in C_i, \quad \Longrightarrow \quad |l(\mathcal{A}_{\mathbf{s}}, s) - l(\mathcal{A}_{\mathbf{s}}, z)| \leq \epsilon(\mathbf{s}). \tag{2}$$

In the definition, both K and the partition sets $\{C_i\}_{i=1}^K$ do not depend on the training set $\mathbf{s}$. Note that the definition of robustness requires that (2) holds *for every* training sample. Indeed, we can relax the definition, so that the condition needs only hold for a subset of training samples. We call an algorithm having this property "pseudo robust." See Section 4 for details.

3 Generalization Properties of Robust Algorithms

In this section we investigate generalization of robust algorithms. In particular, in the following subsections we derive PAC bounds for robust algorithms under two different conditions: (1) The ubiquitous learning setup where the samples are i.i.d. and the goal of learning is to minimize expected loss. (2) The learning goal is to minimize quantile loss. Indeed, the fact that we can provide results in (2) indicates the fundamental nature of robustness as a property of learning algorithms.

3.1 IID samples and expected loss

In this section, we consider the standard learning setup, i.e., the sample set $\mathbf{s}$ consists of n i.i.d. samples generated by an unknown distribution μ, and the goal of learning is to minimize expected test loss. Let $\hat{l}(\cdot)$ and $l_{\text{emp}}(\cdot)$ denote the expected error and the training error, i.e.,

$$\hat{l}(\mathcal{A}_{\mathbf{s}}) \triangleq \mathbb{E}_{z\sim\mu} l(\mathcal{A}_{\mathbf{s}}, z); \quad l_{\text{emp}}(\mathcal{A}_{\mathbf{s}}) \triangleq \frac{1}{n}\sum_{s_i\in\mathbf{s}} l(\mathcal{A}_{\mathbf{s}}, s_i).$$

Recall that the loss function $l(\cdot,\cdot)$ is upper bounded by M.

Theorem 3 *If* $\mathbf{s}$ *consists of* n *i.i.d. samples, and* $\mathcal{A}$ *is* $(K, \epsilon(\mathbf{s}))$*-robust, then for any* $\delta > 0$*, with probability at least* $1-\delta$*,*

$$\left|\hat{l}(\mathcal{A}_{\mathbf{s}}) - l_{\text{emp}}(\mathcal{A}_{\mathbf{s}})\right| \le \epsilon(s) + M\sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}.$$

Proof: Let N_i be the set of index of points of $\mathbf{s}$ that fall into C_i. Note that $(|N_1|, \cdots, |N_K|)$ is an IID multinomial random variable with parameters n and $(\mu(C_1), \cdots, \mu(C_K))$. The following holds by the Breteganolle-Huber-Carol inequality (cf Proposition A6.6 of(van der Vaart & Wellner, 2000)):

$$\Pr\left\{\sum_{i=1}^{K}\left|\frac{|N_i|}{n} - \mu(C_i)\right| \ge \lambda\right\} \le 2^K \exp(\frac{-n\lambda^2}{2}).$$

Hence, the following holds with probability at least $1-\delta$,

$$\sum_{i=1}^{K}\left|\frac{|N_i|}{n} - \mu(C_i)\right| \le \sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}. \tag{3}$$

We have

$$\begin{aligned}
&\left|\hat{l}(\mathcal{A}_{\mathbf{s}}) - l_{\text{emp}}(\mathcal{A}_{\mathbf{s}})\right| \\
=&\left|\sum_{i=1}^{K}\mathbb{E}\big(l(\mathcal{A}_{\mathbf{s}}, z)|z\in C_i\big)\mu(C_i) - \frac{1}{n}\sum_{i=1}^{n} l(\mathcal{A}_{\mathbf{s}}, s_i)\right| \\
\overset{(a)}{\le}&\left|\sum_{i=1}^{K}\mathbb{E}\big(l(\mathcal{A}_{\mathbf{s}}, z)|z\in C_i\big)\frac{|N_i|}{n} - \frac{1}{n}\sum_{i=1}^{n} l(\mathcal{A}_{\mathbf{s}}, s_i)\right| \\
&\quad + \left|\sum_{i=1}^{K}\mathbb{E}\big(l(\mathcal{A}_{\mathbf{s}}, z)|z\in C_i\big)\mu(C_i) - \sum_{i=1}^{K}\mathbb{E}\big(l(\mathcal{A}_{\mathbf{s}}, z)|z\in C_i\big)\frac{|N_i|}{n}\right| \\
\overset{(b)}{\le}&\left|\frac{1}{n}\sum_{i=1}^{K}\sum_{j\in N_i}\max_{z_2\in C_i}|l(\mathcal{A}_{\mathbf{s}}, s_j) - l(\mathcal{A}_{\mathbf{s}}, z_2)|\right| + \left|\max_{z\in\mathcal{Z}}|l(\mathcal{A}_{\mathbf{s},z})|\sum_{i=1}^{K}\left|\frac{|N_i|}{n} - \mu(C_i)\right|\right| \\
\overset{(c)}{\le}&\epsilon(s) + M\sum_{i=1}^{K}\left|\frac{|N_i|}{n} - \mu(C_i)\right|,
\end{aligned} \tag{4}$$

where (a), (b), and (c) are due to the triangle inequality, the definition of N_i, and the definition of $\epsilon(\mathbf{s})$ and M, respectively. The right-hand-side of (4) is upper-bounded by $\epsilon(s) + M\sqrt{\frac{2K\ln 2+2\ln(1/\delta)}{n}}$ with probability at least $1-\delta$ due to (3). The theorem follows. ■

Theorem 3 requires that we fix a K *a priori*. However, it is often worthwhile to consider adaptive K. For example, in the large-margin classification case, typically the margin is known only after $\mathbf{s}$ is realized. That is, the value of K depends on $\mathbf{s}$. Because of this dependency, we needs a generalization bound that holds uniformly for all K.

Corollary 4 *If* $\mathbf{s}$ *consists of* n *i.i.d. samples, and* $\mathcal{A}$ *is* $(K, \epsilon_K(\mathbf{s}))$ *robust for all* $K \geq 1$*, then for any* $\delta > 0$*, with probability at least* $1 - \delta$*,*

$$\left|\hat{l}(\mathcal{A}_{\mathbf{s}}) - l_{\text{emp}}(\mathcal{A}_{\mathbf{s}})\right| \leq \inf_{K \geq 1} \left[\epsilon_K(s) + M\sqrt{\frac{2K\ln 2 + 2\ln\frac{K(K+1)}{\delta}}{n}}\right].$$

Proof: Let

$$E(K) \triangleq \left\{\left|\hat{l}(\mathcal{A}_{\mathbf{s}}) - l_{\text{emp}}(\mathcal{A}_{\mathbf{s}})\right| > \epsilon_K(s) + M\sqrt{\frac{2K\ln 2 + 2\ln\frac{K(K+1)}{\delta}}{n}}\right\}.$$

From Theorem 3 we have $\Pr(E(K)) \leq \delta/(K(K+1)) = \delta/K - \delta/(K+1)$. By the union bound we have

$$\Pr\left\{\bigcup_{K\geq 1} E(K)\right\} \leq \sum_{K\geq 1} \Pr\left(E(K)\right) \leq \sum_{K\geq 1}\left[\frac{\delta}{K} - \frac{\delta}{K+1}\right] = \delta,$$

and the corollary follows. ■

If $\epsilon(s)$ does not depend on $\mathbf{s}$, we can sharpen the bound given in Corollary 4.

Corollary 5 *If* $\mathbf{s}$ *consists of* n *i.i.d. samples, and* $\mathcal{A}$ *is* (K, ϵ_K) *robust for all* $K \geq 1$*, then for any* $\delta > 0$*, with probability at least* $1 - \delta$*,*

$$\left|\hat{l}(\mathcal{A}_{\mathbf{s}}) - l_{\text{emp}}(\mathcal{A}_{\mathbf{s}})\right| \leq \inf_{K \geq 1} \left[\epsilon_K + M\sqrt{\frac{2K\ln 2 + 2\ln\frac{1}{\delta}}{n}}\right].$$

Proof: Take K^* that minimizes the right hand side, and note that it does not depend on $\mathbf{s}$. Therefore, plugging K^* into Theorem 3 establishes the corollary. ■

3.2 Quantile Loss

So far we considered the standard expected loss setup. In this section we consider some less extensively investigated loss functions, namely quantile value and truncated expectation (see the following for precise definitions). These loss functions are of interest because they are less sensitive to the presence of outliers than the standard average loss (Huber, 1981).

Definition 6 *For a non-negative random variable* X*, the* β-quantile value *is*

$$\mathbb{Q}^{\beta}(X) \triangleq \inf\left\{c \in \mathbb{R} : \Pr\big(X \leq c\big) \geq \beta\right\}.$$

The β-truncated mean *is*

$$\mathbb{T}^{\beta}(X) \triangleq \begin{cases} \mathbb{E}\left[X \cdot \mathbf{1}(X < \mathbb{Q}^{\beta}(X))\right] & \text{if } \Pr\left[X = \mathbb{Q}^{\beta}(X)\right] = 0; \\ \mathbb{E}\left[X \cdot \mathbf{1}(X < \mathbb{Q}^{\beta}(X))\right] + \frac{\beta - \Pr\left[X < \mathbb{Q}^{\beta}(X)\right]}{\Pr\left[X = \mathbb{Q}^{\beta}(X)\right]}\mathbb{Q}^{\beta}(X) & \text{otherwise.} \end{cases}$$

In words, the $\beta-$quantile loss is the smallest value that is larger or equal to X with probability at least β. The β-truncated mean is the contribution to the expectation of the leftmost β fraction of the distribution. For example, suppose X is supported on $\{c_1, \cdots, c_{10}\}$ $(c_1 < c_2 < \cdots < c_{10})$ and the probability of taking each value equals 0.1. Then the 0.63-quantile loss of X is c_7, and the 0.63-truncated mean of X equals $0.1(\sum_{i=1}^{6} c_i + 0.3c_7)$.

Given $h \in \mathcal{H}$, $\beta \in (0, 1)$, and a probability measure μ on $\mathcal{Z}$, let

$$\mathcal{Q}(h, \beta, \mu) \triangleq \mathbb{Q}^{\beta}(l(h, z)); \quad \text{where: } z \sim \mu;$$

and

$$\mathcal{T}(h, \beta, \mu) \triangleq \mathbb{T}^{\beta}(l(h, z)); \quad \text{where: } z \sim \mu;$$

i.e., the β-quantile value and β-truncated mean of the (random) testing error of hypothesis h if the testing sample follows distribution μ. We have the following theorem that is a special case of Theorem 10, hence we omit the proof.

Theorem 7 (Quantile Value & Truncated Mean) *Suppose* $\mathbf{s}$ *are* n *i.i.d. samples drawn according to* μ*, and denote the empirical distribution of* $\mathbf{s}$ *by* μ_{emp}*. Let* $\lambda_0 = \sqrt{\frac{2K\ln 2+2\ln(1/\delta)}{n}}$*. If* $0 \le \beta - \lambda_0 \le \beta + \lambda_0 \le 1$ *and* $\mathcal{A}$ *is* $(K, \epsilon(\mathbf{s}))$ *robust, then with probability at least* $1-\delta$*, the followings hold*

$$(I) \quad \mathcal{Q}\left(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \mu_{\text{emp}}\right) - \epsilon(\mathbf{s}) \le \mathcal{Q}\left(\mathcal{A}_{\mathbf{s}}, \beta, \mu\right) \le \mathcal{Q}\left(\mathcal{A}_{\mathbf{s}}, \beta + \lambda_0, \mu_{\text{emp}}\right) + \epsilon(\mathbf{s});$$
$$(II) \quad \mathcal{T}\left(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \mu_{\text{emp}}\right) - \epsilon(\mathbf{s}) \le \mathcal{T}\left(\mathcal{A}_{\mathbf{s}}, \beta, \mu\right) \le \mathcal{T}\left(\mathcal{A}_{\mathbf{s}}, \beta + \lambda_0, \mu_{\text{emp}}\right) + \epsilon(\mathbf{s}).$$

In words, Theorem 7 essentially means that with high probability, the β-quantile value/truncated mean of the testing error (recall that the testing error is a random variable) is (approximately) bounded by the $(\beta \pm \lambda_0)$-quantile value/truncated mean of the empirical error, thus providing a way to estimate the quantile value/truncated expectation of the testing error based on empirical observations.

4 Pseudo Robustness

In this section we propose a relaxed definition of robustness that accounts for the case where Equation (2) holds for most of training samples, as opposed to Definition 2 where Equation (2) holds for all training samples. Recall that the size of training set is fixed as n.

Definition 8 *Algorithm* $\mathcal{A}$ *is* $(K, \epsilon(\mathbf{s}), \hat{n}(\mathbf{s}))$ pseudo robust *if* $\mathcal{Z}$ *can be partitioned into* K *disjoint sets, denoted as* $\{C_i\}_{i=1}^K$*, and there exists a subset of training samples* $\hat{\mathbf{s}}$ *with* $|\hat{\mathbf{s}}| = \hat{n}(\mathbf{s})$ *such that* $\forall s \in \hat{\mathbf{s}}$*,*

$$s, z \in C_i, \quad \Longrightarrow \quad |l(\mathcal{A}_{\mathbf{s}}, s) - l(\mathcal{A}_{\mathbf{s}}, z)| \le \epsilon(\mathbf{s}).$$

Observe that $(K, \epsilon(\mathbf{s}))$-robust is equivalent to $(K, \epsilon(\mathbf{s}), n)$ pseudo robust.

Theorem 9 *If* $\mathbf{s}$ *consists of* n *i.i.d. samples, and* $\mathcal{A}$ *is* $(K, \epsilon(\mathbf{s}), \hat{n}(\mathbf{s}))$ *pseudo robust, then for any* $\delta > 0$*, with probability at least* $1-\delta$*,*

$$\left|\hat{l}(\mathcal{A}_{\mathbf{s}}) - l_{\text{emp}}(\mathcal{A}_{\mathbf{s}})\right| \le \frac{\hat{n}(\mathbf{s})}{n}\epsilon(s) + M\left(\frac{n-\hat{n}(\mathbf{s})}{n} + \sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}\right).$$

Proof: Let N_i and $\hat{N}_i$ be the set of indices of points of $\mathbf{s}$ and $\hat{\mathbf{s}}$ that fall into the C_i, respectively. Similarly to the proof of Theorem 3, we note that $(|N_1|, \cdots, |N_K|)$ is an IID multinomial random variable with parameters n and $(\mu(C_1), \cdots, \mu(C_K))$. And hence due to Breteganolle-Huber-Carol Inequality, the following holds with probability at least $1-\delta$,

$$\sum_{i=1}^K \left|\frac{|N_i|}{n} - \mu(C_i)\right| \le \sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}. \tag{5}$$

Furthermore, we have

$$\begin{aligned}
&\left|\hat{l}(\mathcal{A}_{\mathbf{s}}) - l_{\text{emp}}(\mathcal{A}_{\mathbf{s}})\right| \\
=&\left|\sum_{i=1}^K \mathbb{E}\big(l(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\mu(C_i) - \frac{1}{n}\sum_{i=1}^n l(\mathcal{A}_{\mathbf{s}}, s_i)\right| \\
\le&\left|\sum_{i=1}^K \mathbb{E}\big(l(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\frac{|N_i|}{n} - \frac{1}{n}\sum_{i=1}^n l(\mathcal{A}_{\mathbf{s}}, s_i)\right| \\
&\quad + \left|\sum_{i=1}^K \mathbb{E}\big(l(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\mu(C_i) - \sum_{i=1}^K \mathbb{E}\big(l(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big)\frac{|N_i|}{n}\right| \\
\le&\left|\frac{1}{n}\sum_{i=1}^K \left[|N_i| \times \mathbb{E}\big(l(\mathcal{A}_{\mathbf{s}}, z)|z \in C_i\big) - \sum_{j\in\hat{N}_i} l(\mathcal{A}_{\mathbf{s}}, s_j) - \sum_{j\in N_i, j\notin\hat{N}_i} l(\mathcal{A}_{\mathbf{s}}, s_j)\right]\right| \\
&\quad + \left|\max_{z\in\mathcal{Z}} |l(\mathcal{A}_{\mathbf{s},z})| \sum_{i=1}^K \left|\frac{|N_i|}{n} - \mu(C_i)\right|\right|.
\end{aligned}$$

Note that due to the triangle inequality as well as the assumption that the loss is non-negative and upper bounded by M, the right-hand side can be upper bounded by

$$\begin{aligned}
&\left|\frac{1}{n}\sum_{i=1}^{K}\sum_{j\in\hat{N}_i}\max_{z_2\in C_i}|l(\mathcal{A}_{\mathbf{s}},s_j)-l(\mathcal{A}_{\mathbf{s}},z_2)|\right|+\left|\frac{1}{n}\sum_{i=1}^{K}\sum_{j\in N_i,j\notin\hat{N}_i}\max_{z_2\in C_i}|l(\mathcal{A}_{\mathbf{s}},s_j)-l(\mathcal{A}_{\mathbf{s}},z_2)|\right| \\
&\quad+M\sum_{i=1}^{K}\left|\frac{|N_i|}{n}-\mu(C_i)\right| \\
\leq&\frac{\hat{n}(\mathbf{s})}{n}\epsilon(s)+\frac{n-\hat{n}(\mathbf{s})}{n}M+M\sum_{i=1}^{K}\left|\frac{|N_i|}{n}-\mu(C_i)\right|.
\end{aligned}$$

where the inequality holds due to definition of N_i and $\hat{N}_i$. The theorem follows by applying (5). ■

Similarly, Theorem 7 can be generalized to the pseudo robust case. See the full version (Xu & Mannor, 2010) for the proof.

Theorem 10 (Quantile Value & Truncated Expectation) *Suppose* $\mathbf{s}$ *has* n *samples drawn i.i.d. according to* μ*, and denote the empirical distribution of* $\mathbf{s}$ *as* μ_{emp}*. Let* $\lambda_0=\sqrt{\frac{2K\ln 2+2\ln(1/\delta)}{n}}$*. Suppose* $0\leq\beta-\lambda_0-(n-\hat{n})/n\leq\beta+\lambda_0+(n-\hat{n})/n\leq 1$ *and* $\mathcal{A}$ *is* $(K,\epsilon(\mathbf{s}),\hat{n}(\mathbf{s}))$ *pseudo robust. Then with probability at least* $1-\delta$*, the followings hold*

$$(I)\quad \mathcal{Q}\left(\mathcal{A}_{\mathbf{s}},\beta-\lambda_0-\frac{n-\hat{n}(\mathbf{s})}{n},\mu_{\text{emp}}\right)-\epsilon(\mathbf{s})\leq\mathcal{Q}\left(\mathcal{A}_{\mathbf{s}},\beta,\mu\right)\leq\mathcal{Q}\left(\mathcal{A}_{\mathbf{s}},\beta+\lambda_0+\frac{n-\hat{n}(\mathbf{s})}{n},\mu_{\text{emp}}\right)+\epsilon(\mathbf{s});$$

$$(II)\quad \mathcal{T}\left(\mathcal{A}_{\mathbf{s}},\beta-\lambda_0-\frac{n-\hat{n}(\mathbf{s})}{n},\mu_{\text{emp}}\right)-\epsilon(\mathbf{s})\leq\mathcal{T}\left(\mathcal{A}_{\mathbf{s}},\beta,\mu\right)\leq\mathcal{T}\left(\mathcal{A}_{\mathbf{s}},\beta+\lambda_0+\frac{n-\hat{n}(\mathbf{s})}{n},\mu_{\text{emp}}\right)+\epsilon(\mathbf{s}).$$

5 Examples of Robust Algorithms

In this section we provide some examples of robust algorithms. The proofs of the examples can be found in the full version (Xu & Mannor, 2010). Our first example is Majority Voting (MV) classification (cf Section 6.3 of Devroye et al., 1996) that partitions the input space $\mathcal{X}$ and labels each partition set according to a majority vote of the training samples belonging to it.

Example 3 (Majority Voting) *Let* $\mathcal{Y}=\{-1,+1\}$*. Partition* $\mathcal{X}$ *to* $\mathcal{C}_1,\cdots,\mathcal{C}_K$*, and use* $\mathcal{C}(x)$ *to denote the set to which* x *belongs. A new sample* $x_a\in\mathcal{X}$ *is labeled by*

$$\mathcal{A}_{\mathbf{s}}(x_a)\triangleq\begin{cases}1, & \text{if } \sum_{s_i\in\mathcal{C}(x_a)}\mathbf{1}(s_{i|y}=1)\geq\sum_{s_i\in\mathcal{C}(x_a)}\mathbf{1}(s_{i|y}=-1);\\ -1, & \text{otherwise.}\end{cases}$$

If the loss function is $l(\mathcal{A}_s,z)=f(z_{|y},\mathcal{A}_{\mathbf{s}}(z_{|x}))$ *for some function* f*, then MV is* $(2K,0)$ *robust.*

MV algorithm has a natural partition of the sample space that makes it robust. Another class of robust algorithms are those that have approximately the same testing loss for testing samples that are close (in the sense of geometric distance) to each other, since we can partition the sample space with norm balls. The next theorem states that an algorithm is robust if two samples being close implies that they have similar testing error.

Theorem 11 *Fix* $\gamma>0$ *and metric* ρ *of* $\mathcal{Z}$*. Suppose* $\mathcal{A}$ *satisfies*

$$|l(\mathcal{A}_{\mathbf{s}},z_1)-l(\mathcal{A}_{\mathbf{s}},z_2)|\leq\epsilon(\mathbf{s}),\quad\forall z_1,z_2:z_1\in\mathbf{s},\,\rho(z_1,z_2)\leq\gamma,$$

and $\mathcal{N}(\gamma/2,\mathcal{Z},\rho)<\infty$*. Then* $\mathcal{A}$ *is* $\big(\mathcal{N}(\gamma/2,\mathcal{Z},\rho),\,\epsilon(\mathbf{s})\big)$*-robust.*

Proof: Let $\{c_1,\cdots,c_{\mathcal{N}(\gamma/2,\mathcal{Z},\rho)}\}$ be a $\gamma/2$-cover of $\mathcal{Z}$. whose existence is guaranteed by the definition of covering number. Let $\hat{C}_i=\{z\in\mathcal{Z}|\rho(z,c_i)\leq\gamma/2\}$, and $C_i=\hat{C}_i\bigcap\big(\bigcup_{j=1}^{i-1}\hat{C}_j\big)^c$. Thus, $C_1,\cdots,C_{\mathcal{N}(\gamma/2,\mathcal{Z},\rho)}$ is a partition of $\mathcal{Z}$, and satisfies

$$z_1,z_2\in C_i\quad\Longrightarrow\quad\rho(z_1,z_2)\leq\rho(z_1,c_i)+\rho(z_2,c_i)\leq\gamma.$$

Therefore,

$$|l(\mathcal{A}_{\mathbf{s}},z_1)-l(\mathcal{A}_{\mathbf{s}},z_2)|\leq\epsilon(\mathbf{s}),\quad\forall z_1,z_2:z_1\in\mathbf{s},\,\rho(z_1,z_2)\leq\gamma,$$

implies

$$z_1 \in \mathbf{s}\ z_1, z_2 \in C_i \quad \Longrightarrow |l(\mathcal{A}_\mathbf{s}, z_1) - l(\mathcal{A}_\mathbf{s}, z_2)| \leq \epsilon(\mathbf{s}),$$

and the theorem follows. ■

Theorem 11 immediately leads to the next example: if the testing error given the output of an algorithm is Lipschitz continuous, then the algorithm is robust.

Example 4 (Lipschitz continuous functions) *If $\mathcal{Z}$ is compact w.r.t. metric ρ, $l(\mathcal{A}_\mathbf{s}, \cdot)$ is Lipschitz continuous with Lipschitz constant $c(\mathbf{s})$, i.e.,*

$$|l(\mathcal{A}_\mathbf{s}, z_1) - l(\mathcal{A}_\mathbf{s}, z_2)| \leq c(\mathbf{s})\rho(z_1, z_2), \quad \forall z_1, z_2 \in \mathcal{Z},$$

then $\mathcal{A}$ is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \rho),\, c(\mathbf{s})\gamma\big)$-robust for all $\gamma > 0$.

Theorem 11 also implies that SVM, Lasso, feed-forward neural network and PCA are robust, as stated in Example 5 to Example 8. The proofs are deferred to Appendix.

Example 5 (Support Vector Machines) *Let $\mathcal{X}$ be compact. Consider the standard SVM formulation (Cortes & Vapnik, 1995; Schölkopf & Smola, 2002)*

$$\begin{aligned} \textit{Minimize:}_{\mathbf{w},d} \quad & c\|w\|_\mathcal{H}^2 + \frac{1}{n}\sum_{i=1}^{n} \xi_i \\ s.\ t. \quad & 1 - s_{i|y}[\langle w,\, \phi(s_{i|x})\rangle + d] \leq \xi_i; \\ & \xi_i \geq 0. \end{aligned}$$

Here $\phi(\cdot)$ is a feature mapping, $\|\cdot\|_\mathcal{H}$ is its RKHS kernel, and $k(\cdot,\cdot)$ is the kernel function. Let $l(\cdot,\cdot)$ be the hinge-loss, i.e., $l\big((w,d),z\big) = [1 - z_{|y}(\langle w, \phi(z_{|x})\rangle + d)]^+$, and define $f_\mathcal{H}(\gamma) \triangleq \max_{\mathbf{a},\mathbf{b}\in\mathcal{X}, \|\mathbf{a}-\mathbf{b}\|_2\leq\gamma} \big(k(\mathbf{a},\mathbf{a}) + k(\mathbf{b},\mathbf{b}) - 2k(\mathbf{a},\mathbf{b})\big)$. If $k(\cdot,\cdot)$ is continuous, then for any $\gamma > 0$, $f_\mathcal{H}(\gamma)$ is finite, and SVM is $(2\mathcal{N}(\gamma/2, \mathcal{X}, \|\cdot\|_2), \sqrt{f_\mathcal{H}(\gamma)/c})$ robust.

Example 6 (Lasso) *Let $\mathcal{Z}$ be compact and the loss function be $l(A_\mathbf{s}, z) = |z_{|y} - A_\mathbf{s}(z_{|x})|$. Lasso (Tibshirani, 1996), which is the following regression formulation:*

$$\min_w : \frac{1}{n}\sum_{i=1}^{n}(s_{i|y} - w^\top s_{i|x})^2 + c\|w\|_1, \tag{6}$$

is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \|\cdot\|_\infty),\, (Y(\mathbf{s})/c + 1)\gamma\big)$-robust for all $\gamma > 0$, where $Y(\mathbf{s}) \triangleq \frac{1}{n}\sum_{i=1}^{n} {s_{i|y}}^2$.

Example 7 (Feed-forward Neural Networks) *Let $\mathcal{Z}$ be compact and the loss function be $l(\mathcal{A}_\mathbf{s}, z) = |z_{|y} - \mathcal{A}_\mathbf{s}(z_{|x})|$. Consider the d-layer neural network (trained on $\mathbf{s}$), which is the following predicting rule given an input $x \in \mathcal{X}$*

$$\begin{aligned} & x^0 := z_{|x} \\ \forall v = 1, \cdots, d-1 : \qquad & x_i^v := \sigma\Big(\sum_{j=1}^{N_{v-1}} w_{ij}^{v-1} x_j^{v-1}\Big); \quad i = 1, \cdots, N_v; \\ & \mathcal{A}_\mathbf{s}(x) := \sigma\Big(\sum_{j=1}^{N_{d-1}} w_j^{d-1} x_j^{d-1}\Big); \end{aligned}$$

If there exists α and β such that the d-layer neural network satisfying that $|\sigma(a) - \sigma(b)| \leq \beta|a - b|$, and $\sum_{j=1}^{N_v} |w_{ij}^v| \leq \alpha$ for all v, i, then it is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \|\cdot\|_\infty),\, \alpha^d\beta^d\gamma\big)$-robust, for all $\gamma > 0$.

We remark that in Example 7, the number of hidden units in each layer has no effect on the robustness of the algorithm and consequently the bound on the testing error. This indeed agrees with Bartlett (1998), where the author showed (using a different approach based on fat-shattering dimension) that for neural networks, the weight plays a more important role than the number of hidden units.

The next example considers an unsupervised learning algorithm, namely the principal component analysis algorithm. We show that it is robust if the sample space is *bounded*. This does not contradict with the well known fact that the principal component analysis is sensitive to outliers which are far away from the origin.

Example 8 (Principal Component Analysis (PCA)) *Let $\mathcal{Z} \subset \mathbb{R}^m$ be such that $\max_{z\in\mathcal{Z}} \|z\|_2 \leq B$. If the loss function is $l((w_1, \cdots, w_d), z) = \sum_{k=1}^{d} (w_k^\top z)^2$, then finding the first d principal components, which solves the following optimization problem over d vectors $w_1, \cdots, w_d \in \mathbb{R}^m$,*

$$\begin{aligned} \textit{Maximize:} \quad & \sum_{i=1}^{n}\sum_{k=1}^{d} (w_k^\top s_i)^2 \\ \textit{Subject to:} \quad & \|w_k\|_2 = 1, \quad k = 1, \cdots, d; \\ & w_i^\top w_j = 0, \quad i \neq j. \end{aligned}$$

is $(\mathcal{N}(\gamma/2, \mathcal{Z}, \|\cdot\|_2), 2d\gamma B)$-robust.

The last example is large-margin classification, which is a generalization of Example 1. We need the following standard definition (e.g., Bartlett, 1998) of the distance of a point to a classification rule.

Definition 12 *Fix a metric ρ of $\mathcal{X}$. Given a classification rule Δ and $x \in \mathcal{X}$, the* distance *of x to Δ is*

$$\mathcal{D}(x, \Delta) \triangleq \inf\{c \geq 0 | \exists x' \in \mathcal{X} : \rho(x, x') \leq c, \Delta(x) \neq \Delta(x')\}.$$

A large margin classifier is a classification rule such that most of the training samples are "far away" from the classification boundary.

Example 9 (Large-margin classifier) *If there exist γ and $\hat{n}$ such that*

$$\sum_{i=1}^{n} \mathbf{1}\big(\mathcal{D}(s_{i|x}, \mathcal{A}_s) > \gamma\big) \geq \hat{n},$$

then algorithm $\mathcal{A}$ is $(2\mathcal{N}(\gamma/2, \mathcal{X}, \rho), 0, \hat{n})$ pseudo robust, provided that $\mathcal{N}(\gamma/2, \mathcal{X}, \rho) < \infty$.

Proof: Set $\hat{\mathbf{s}}$ as

$$\hat{\mathbf{s}} \triangleq \{s_i \in \mathbf{s} | \mathcal{D}(s_i, \mathcal{A}_{\mathbf{s}}) > \gamma\}.$$

And let $c_1, \cdots, c_{\mathcal{N}(\gamma/2, \mathcal{X}, \rho)}$ be a $\gamma/2$ cover of $\mathcal{X}$. Thus, we can partition $\mathcal{Z}$ to $2\mathcal{N}(\gamma/2, \mathcal{X}, \rho)$ subsets $\{C_i\}$, such that if

$$z_1, z_2 \in C_i; \quad \Longrightarrow \quad y_1 = y_2; \ \& \, \rho(x_1, x_2) \leq \gamma.$$

This implies that:

$$z_1 \in \hat{\mathbf{s}}, \, z_1, z_2 \in C_i; \quad \Longrightarrow \quad y_1 = y_2; \ \mathcal{A}_{\mathbf{s}}(x_1) = \mathcal{A}_{\mathbf{s}}(x_2); \quad \Longrightarrow \quad l(\mathcal{A}_{\mathbf{s}}, z_1) = l(\mathcal{A}_{\mathbf{s}}, z_2).$$

By definition, $\mathcal{A}$ is $(2\mathcal{N}(\gamma/2, \mathcal{X}, \rho), 0, \hat{n})$ pseudo robust. ■

Note that by taking ρ to be the Euclidean norm, and letting $\hat{n} = n$, we recover Example 1.

6 Necessity of Robustness

Thus far we have considered finite sample generalization bounds of robust algorithms. We now turn to asymptotic analysis, i.e., we are given an increasing set of training samples $\mathbf{s} = (s_1, s_2, \cdots)$ and are tested on an increasing set of testing samples $\mathbf{t} = (t_1, t_2, \cdots)$. We use $\mathbf{s}(n)$ and $\mathbf{t}(n)$ to denote the first n elements of training samples and testing samples respectively. For succinctness, we let $\mathcal{L}(\cdot, \cdot)$ to be the average loss given a set of samples, i.e., for $h \in \mathcal{H}$,

$$\mathcal{L}(h, \mathbf{t}(n)) \equiv \frac{1}{n}\sum_{i=1}^{n} l(h, t_i).$$

We show in this section that robustness is an essential property of successful learning. In particular, a (weaker) notion of robustness characterizes generalizability, i.e., a learning algorithm generalizes if and only if it is weakly robust. To make this precise, we define the notion of generalizability and weak robustness first.

Definition 13 *1. A learning algorithm $\mathcal{A}$* generalizes w.r.t. $\mathbf{s}$ *if*

$$\limsup_n \Big\{\mathbb{E}_t\Big(l(\mathcal{A}_{\mathbf{s}(n)}, t)\Big) - \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n))\Big\} \leq 0.$$

2. A learning algorithm $\mathcal{A}$ generalize w.p. 1 *if it generalize w.r.t. almost every* $\mathbf{s}$.

We remark that the proposed notion of generalizability differs slightly from the standard one in the sense that the latter requires that the empirical risk and the expected risk converges in mean, while the proposed notion requires convergence w.p.1. It is straightforward that the proposed notion implies the standard one.

Definition 14 *1. A learning algorithm* $\mathcal{A}$ *is* weakly robust w.r.t $\mathbf{s}$ *if there exists a sequence of* $\{\mathcal{D}_n \subseteq \mathcal{Z}^n\}$ *such that* $\Pr(\mathbf{t}(n) \in \mathcal{D}_n) \to 1$, *and*

$$\limsup_n \left\{ \max_{\hat{\mathbf{s}}(n) \in \mathcal{D}_n} \left[\mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \hat{\mathbf{s}}(n)) - \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) \right] \right\} \leq 0.$$

2. A learning algorithm $\mathcal{A}$ *is* a.s. weakly robust *if it is robust w.r.t. almost every* $\mathbf{s}$.

We briefly comment on the definition of weak robustness. Recall that the definition of robustness requires that the sample space can be partitioned into disjoint subsets such that if a testing sample belongs to the same partitioning set of a training sample, then they have similar loss. Weak robustness generalizes such notion by considering the average loss of testing samples and training samples. That is, if for a large (in the probabilistic sense) subset of $\mathcal{Z}^n$, the testing error is close to the training error, then the algorithm is weakly robust. It is easy to see, by Breteganolle-Huber-Carol lemma, that if for any fixed $\epsilon > 0$ there exists K such that $\mathcal{A}$ is (K, ϵ) robust, then $\mathcal{A}$ is weakly robust.

We now establish the main result of this section: weak robustness and generalizability are equivalent.

Theorem 15 *An algorithm* $\mathcal{A}$ *generalizes w.r.t.* $\mathbf{s}$ *if and only if it is weakly robust w.r.t.* $\mathbf{s}$.

Proof: We prove the sufficiency of weak robustness first. When $\mathcal{A}$ is weakly robust w.r.t. $\mathbf{s}$, by definition there exists $\{D_n\}$ such that for any $\delta, \epsilon > 0$, there exists $N(\delta, \epsilon)$ such that for all $n > N(\delta, \epsilon)$, $\Pr(\mathbf{t}(n) \in D_n) > 1 - \delta$, and

$$\sup_{\hat{\mathbf{s}}(n) \in D_n} \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \hat{\mathbf{s}}(n)) - \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) < \epsilon. \tag{7}$$

Therefore, the following holds for any $n > N(\delta, \epsilon)$,

$$\begin{aligned}
&\mathbb{E}_t\Big(l(\mathcal{A}_{\mathbf{s}(n)}, t)\Big) - \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) \\
=&\mathbb{E}_{\mathbf{t}(n)}\Big(\mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{t}(n))\Big) - \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) \\
=&\Pr(\mathbf{t}(n) \notin D_n)\mathbb{E}\Big(\mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{t}(n))|\mathbf{t}(n) \notin D_n\Big) + \Pr(\mathbf{t}(n) \in D_n)\mathbb{E}\Big(\mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{t}(n))|\mathbf{t}(n) \in D_n\Big) \\
&\quad - \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) \\
\leq&\delta M + \sup_{\hat{\mathbf{s}}(n) \in D_n} \left\{\mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \hat{\mathbf{s}}(n)) - \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n))\right\} \leq \delta M + \epsilon.
\end{aligned}$$

Here, the first equality holds since $\mathbf{t}(n)$ are i.i.d., and the second equality holds by conditional expectation. The inequalities hold due to the assumption that the loss function is upper bounded by M, as well as (7).

We thus conclude that the algorithm $\mathcal{A}$ generalizes for $\mathbf{s}$, because ϵ and δ can be arbitrary.

Now we turn to the necessity of weak robustness. First, we establish the following lemma.

Lemma 16 *Given* $\mathbf{s}$, *if algorithm* $\mathcal{A}$ *is not weakly robust w.r.t.* $\mathbf{s}$, *then there exists* $\epsilon^*, \delta^* > 0$ *such that the following holds for infinitely many* n,

$$\Pr\Big(\mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{t}(n)) \geq \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) + \epsilon^*\Big) \geq \delta^*. \tag{8}$$

Proof: We prove the lemma by contradiction. Assume that such ϵ^* and δ^* do not exist. Let $\epsilon_v = \delta_v = 1/v$ for $v = 1, 2 \cdots$, then there exists a non-decreasing sequence $\{N(v)\}_{v=1}^{\infty}$ such that for all v, if $n \geq N(v)$ then $\Pr\Big(\mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{t}(n)) \geq \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) + \epsilon_v\Big) < \delta_v$. For each n, define the following set:

$$\mathcal{D}_n^v \triangleq \{\hat{\mathbf{s}}(n) | \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \hat{\mathbf{s}}(n)) - \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) < \epsilon_v\}.$$

Thus, for $n \geq N(v)$ we have

$$\Pr(\mathbf{t}(n) \in \mathcal{D}_n^v) = 1 - \Pr\Big(\mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{t}(n)) \geq \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) + \epsilon_v\Big) > 1 - \delta_v.$$

For $n \geq N(1)$, define $\mathcal{D}_n \triangleq \mathcal{D}_n^{v(n)}$, where: $v(n) \triangleq \max\big(v | N(t) \leq n;\ v \leq n\big)$. Thus for all $n \geq N(1)$ we have that $\Pr(\mathbf{t}(n) \in \mathcal{D}_n) > 1 - \delta_{v(n)}$ and $\sup_{\hat{\mathbf{s}}(n) \in \mathcal{D}_n} \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \hat{\mathbf{s}}(n)) - \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) < \epsilon_{v(n)}$. Note that $v(n) \uparrow \infty$, it follows that $\delta_{v(n)} \to 0$ and $\epsilon_{v(n)} \to 0$. Therefore, $\Pr(\mathbf{t}(n) \in \mathcal{D}_n) \to 1$, and

$$\limsup_{n \to \infty} \Big\{ \sup_{\hat{\mathbf{s}}(n) \in \mathcal{D}_n} \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \hat{\mathbf{s}}(n)) - \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) \Big\} \leq 0.$$

That is, $\mathcal{A}$ is weakly robust w.r.t. $\mathbf{s}$, which is a desired contradiction. ■

We now prove the necessity of weak robustness. Recall that $l(\cdot, \cdot)$ is uniformly bounded. Thus by Hoeffding's inequality we have that for any ϵ and δ, there exists n^* such that for any $n > n^*$, with probability at least $1 - \delta$, we have $\left| \frac{1}{n} \sum_{i=1}^n l(\mathcal{A}_{\mathbf{s}(n)}, t_i) - \mathbb{E}_t(l(\mathcal{A}_{\mathbf{s}(n)}, t)) \right| \leq \epsilon$. This implies that

$$\mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{t}(n)) - \mathbb{E}_t l(\mathcal{A}_{\mathbf{s}(n)}, t) \xrightarrow{\Pr} 0. \tag{9}$$

Since algorithm $\mathbb{A}$ is not weakly robust, Lemma 16 implies that (8) holds for infinitely many n. This, combined with Equation (9) implies that for infinitely many n,

$$\mathbb{E}_t l(\mathcal{A}_{\mathbf{s}(n)}, t) \geq \mathcal{L}(\mathcal{A}_{\mathbf{s}(n)}, \mathbf{s}(n)) + \frac{\epsilon^*}{2},$$

which means that $\mathcal{A}$ does not generalize. Thus, the necessity of weak robustness is established. ■

Theorem 15 immediately leads to the following corollary.

Corollary 17 *An algorithm $\mathcal{A}$ generalizes w.p. 1 if and only if it is a.s. weakly robust.*

7 Discussion

In this paper we investigated the generalization of learning algorithm based on their robustness: the property that if a testing sample is "similar" to a training sample, then its loss is close to the training error. This provides a novel approach, different from complexity or stability arguments, in studying the performance of learning algorithms. We further showed that a weak notion of robustness characterizes generalizability, which implies that robustness is the fundamental property that makes learning algorithms work.

Before concluding the paper, we outline several directions for future research.

- *Adaptive partition:* In Definition 2 when the notion of robustness was introduced, we required that the partitioning of $\mathcal{Z}$ into K sets is *fixed*. That is, regardless of the training sample set, we partition $\mathcal{Z}$ into the same K sets. A natural and interesting question is what if such fixed partition does not exist, while instead we can only partition $\mathcal{Z}$ into K sets *adaptively*, i.e., for different training set we will have a different partitioning of $\mathcal{Z}$. Adaptive partition setup can be used to study algorithms such as k-NN. Our current proof technique does not straightforwardly extend to such a setup, and we would like to understand whether a meaningful generalization bound under this weaker notion of robustness can be obtained.

- *Mismatched datasets:* One advantage of algorithmic robustness framework is the ability to handle non-standard learning setups. For example, in Section 3.2 we derived generalization bounds for quantile loss. A problem of the same essence is the *mismatched datasets*, also called as *domain adaption*, see Ben-David et al. (2007), Mansour et al. (2009) and reference therein. Here the training samples are generated according to a distribution slightly different from that of the testing samples, e.g., the two distributions may have a small K-L divergence. We conjecture that in this case a generalization bound similar to Theorem 3 would be possible, with an extra term depending on the magnitude of the difference of the two distributions.

- *Outlier removal:* One possible reason that the training samples is generated differently from the testing sample is corruption by outliers. It is often the case that the training sample set is corrupted by some outliers. In addition, algorithms designed to be outlier resistent abound in the literature (Huber, 1981; Rousseeuw & Leroy, 1987). The robust framework may provide a novel approach in studying both the generalization ability and the outlier resistent property of these algorithms. In particular, the results reported in Section 3.2 can serve as a starting point of future research in this direction.

- *Consistency:* We addressed in this paper the relationship between robustness and generalizability. An equally important feature of learning algorithms is *consistency*: the property that a learning algorithm guarantees to recover the global optimal solution as the size of the training set increases. While it is straightforward that if an algorithm minimizes the empirical error asymptotically and also generalizes (or equivalently is *weakly robust*), then it is consistent, much less is known for a necessary condition for an algorithm to be consistent. It is certainly interesting to investigate the relationship between consistency and robustness, and in particular whether robustness is necessary for consistency, at least for algorithms that asymptotically minimize the empirical error.

- *Other robust algorithms:* The proposed robust approach considers a general learning setup. However, except for PCA, the algorithms investigated in Section 5 are in the supervised learning setting. One natural extension is to investigate other robust unsupervised and semi-supervised learning algorithms. One difficulty is that compared to supervised learning case, the analysis of unsupervised/semi-supervised learning algorithms can be challenging, due to the fact that many of them are random iterative algorithms (e.g., k-means).

Acknowledgement

We thank Constantine Caramanis and the anonymous reviewers for their insightful comments. The research of Huan Xu was partially supported by DTRA grant HDTRA1-08-0029. Shie Mannor was supported by a Horev Fellowship, by the European Union under a Marie-Curie Reintegration Grant and by the Israel Science Foundation (contract 890015).

References

Alon, N., Ben-David, S., Cesa-Bianchi, N., & Haussler, D. (1997). Scale-sensitive dimension, uniform convergence, and learnability. *Journal of the ACM*, *44*, 615–631.

Bartlett, P. L. (1998). The sample complexity of pattern classification with neural networks: The size of the weight is more important than the size of the network. *IEEE Transactions on Information Theory*, *44*, 525–536.

Bartlett, P. L., Bousquet, O., & Mendelson, S. (2005). Local Rademacher complexities. *The Annals of Statistics*, *33*, 1497–1537.

Bartlett, P. L., & Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, *3*, 463–482.

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2007). Analysis of representations for domain adaptation. *Advances in Neural Information Processing Systems 19*.

Ben-Tal, A., & Nemirovski, A. (1998). Robust convex optimization. *Mathematics of Operations Research*, *23*, 769–805.

Ben-Tal, A., & Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations Research Letters*, *25*, 1–13.

Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, *52*, 35–53.

Bhattacharyya, C., Pannagadatta, K. S., & Smola, A. J. (2004). A second order cone programming formulation for classifying missing data. *Advances in Neural Information Processing Systems (NIPS17)*. Cambridge, MA: MIT Press.

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, *2*, 499–526.

Cortes, C., & Vapnik, V. N. (1995). Support vector networks. *Machine Learning*, *20*, 1–25.

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. Springer, New York.

Devroye, L., & Wagner, T. (1979a). Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions of Information Theory*, *25*, 202–207.

Devroye, L., & Wagner, T. (1979b). Distribution-free performance bounds for potential function rules. *IEEE Transactions of Information Theory*, *25*, 601–604.

Evgeniou, T., Pontil, M., & Poggio, T. (2000). Regularization networks and support vector machines. *Advances in Large Margin Classifiers* (pp. 171–203). Cambridge, MA: MIT Press.

Globerson, A., & Roweis, S. (2006). Nightmare at test time: Robust learning by feature deletion. *Proceedings of the 23rd International Conference on Machine Learning* (pp. 353–360). New York, NY, USA: ACM Press.

Huber, P. J. (1981). *Robust statistics*. John Wiley & Sons, New York.

Mansour, Y., Mohri, M., & Rostamizadeh, A. (2009). Domain adaptation: Learning bounds and algorithms. *Proceedings of The 22nd Annual Conference on Learning Theory*.

McDiarmid, C. (1989). On the method of bounded differences. *Surveys in Combinatorics* (pp. 148–188).

Mukherjee, S., Niyogi, P., Poggio, T., & Rifkin, R. (2006). Learning theory: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, *25*, 161–193.

Poggio, T., Rifkin, R., Mukherjee, S., & Niyogi, P. (2004). General conditions for predictivity in learning theory. *Nature*, *428*, 419–422.

Rousseeuw, P. J., & Leroy, A. M. (1987). *Robust regression and outlier detection*. John Wiley & Sons, New York.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. MIT Press.

Shalev-Shwartz, S., Shamir, O., Srebro, N., & Sridharan, K. (2009). Learnability and stability in the general learning setting. *Proceedings of 22nd Annual Conference of Learning Theory*.

Shivaswamy, P. K., Bhattacharyya, C., & Smola, A. J. (2006). Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, *7*, 1283–1314.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, *58*, 267–288.

van der Vaart, A. W., & Wellner, J. A. (2000). *Weak convergence and empirical processes*. Springer-Verlag, New York.

Vapnik, V. N., & Chervonenkis, A. (1974). *Theory of pattern recognition*. Nauka, Moscow.

Vapnik, V. N., & Chervonenkis, A. (1991). The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis*, *1*, 260–284.

Xu, H., Caramanis, C., & Mannor, S. (2009). Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, *10*, 1485–1510.

Xu, H., Caramanis, C., & Mannor, S. (2010). Robust regression and Lasso. To appear in *IEEE Transactions on Information Theory*.

Xu, H., & Mannor, S. (2010). Robustness and generalization. ArXiv: 1005.2243.