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 Ξ be a finite set of experts. Each expert $\xi \in \Xi$ issues a distribution $P_\xi(\boldsymbol{x}_{n+1}|x^n)$ on the next outcome $\boldsymbol{x}_{n+1}$ given the previous observations $x^n := x_1, \ldots, x_n$. 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: $P_\xi(x^n) = P_\xi(x_1) \cdot P_\xi(x_2|x_1) \cdot \ldots \cdot P_\xi(x_n|x^{n-1})$.

A standard Bayesian approach to combine the expert predictions is to define a prior w on the experts Ξ 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 $P(\boldsymbol{x}_{n+1}|x^n) = P(x^n, \boldsymbol{x}_{n+1})/P(x^n)$, which defines a prediction strategy that combines those of the individual experts, or (c) the *posterior distribution* on the experts $P(\boldsymbol{\xi}|x^n) = P_{\boldsymbol{\xi}}(x^n)w(\boldsymbol{\xi})/P(x^n)$, 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 $P_{\hat{\xi}}(x^n) \geq P(x^n) \geq P_{\hat{\xi}}(x^n)w(\hat{\xi})$, 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

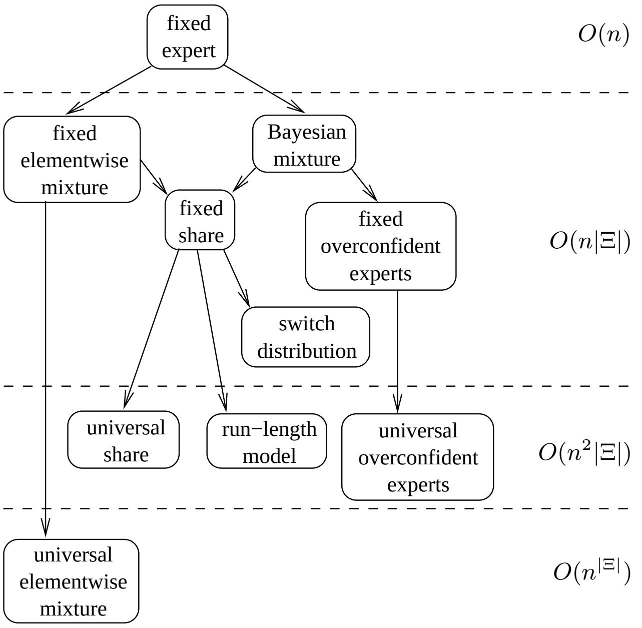
$\xi_1, \xi_2, \ldots$ is the probability that we rely on expert ξ_1's prediction of the first outcome and expert ξ_2's prediction of the second outcome, etc. This allows for the expression of more sophisticated models for the combination of expert predictions. For example, the nature of the data generating 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 $P(\boldsymbol{\xi_{n+1}}|x^n)$.

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 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.

Figure 1 Expert sequence priors: generalisation relationships and run time



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 $q^n = \langle q_1, \ldots, q_n \rangle$. We also let q^ω range over Q^ω — the set of infinite sequences over Q — and write $q^\omega = \langle q_1, \ldots \rangle$. 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^λ is a sequence, and $\kappa \leq \lambda$, then we denote by q^κ the prefix of q^λ of length κ.

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 $\triangle(\mathcal{X})$ denote the set of all probability mass functions on $\mathcal{X}$. A *(prequential) $\mathcal{X}$-forecasting system* (PFS) is a function $P : \mathcal{X}^* \to \triangle(\mathcal{X})$ 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 $P : \mathcal{X}^* \to [0,1]$ that satisfies $P(\epsilon) = 1$, where ϵ is the empty sequence, and for all $n \geq 0$, all $x^n \in \mathcal{X}^n$ we have $\sum_{x \in \mathcal{X}} P(x_1, \ldots, x_n, x) = P(x^n)$. 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 Ξ, and that each expert $\xi \in \Xi$ predicts using a forecasting system P_ξ.

We are interested in switching between different forecasting systems at different sample sizes. For a sequence of experts with prefix ξ^n, the combined forecast, where expert ξ_i predicts the ith 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 π 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:

$$P\Big(\langle \xi_1, x_1 \rangle, \ldots, \langle \xi_n, x_n \rangle\Big) := \pi(\xi^n) P_{\xi^n}(x^n). \qquad (1)$$

We adopt shorthand notation for events: we write $P(S)$, where S is a subsequence of ξ^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:

$$P(x^n) = \sum_{\xi^n \in \Xi^n} P(\xi^n, x^n). \qquad (2)$$

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 Θ. Then, after observing outcomes x^n, inference is based on the posterior $P(\boldsymbol{\theta}|x^n)$ 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 $P(\boldsymbol{\xi}^\omega|x^n)$. However, since the predictive distribution of $\boldsymbol{x}_{n+1}$ only depends on ξ_{n+1} (and x^n) we always marginalise as follows:

$$P(\xi_{n+1}|x^n) = \frac{\sum_{\xi^n} P(\xi^n, x^n) \cdot \pi(\xi_{n+1}|\xi^n)}{\sum_{\xi^n} P(\xi^n, x^n)}. \qquad (3)$$

At each moment in time we predict the data using the posterior, which is a mixture over our experts' predictions. Ideally, the ES-prior π 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 Ξ be a set of experts, and let P_ξ 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 Ξ, 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_{bayes} is given by

$$P_{\text{bayes}}(x^n) = \sum_{\xi \in \Xi} P_\xi(x^n) w(\xi). \qquad (4)$$

Recall that $P_\xi(x^n)$ means $\prod_{i=1}^n P_\xi(x_i|x^i)$. 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-ξ sequence for each $\xi \in \Xi$:

$$\pi_{\text{bayes}}(\xi^n) = \begin{cases} w(k) & \text{if } \xi_i = k \text{ for all } i = 1, \ldots, n, \\ 0 & \text{o.w.} \end{cases}$$

We will use the adjective "Bayesian" generously throughout this paper, but when we write *the standard Bayesian ES-prior* this always refers to π_{bayes}. ◇

Example 2.1.2 (Elementwise Mixtures)**.** The *elementwise mixture*[1] is formed from some mixture weights $\alpha \in \triangle(\Xi)$ 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

[1] 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:

$$\begin{aligned}P_{\text{mix},\alpha}(x^n) &= \prod_{i=1}^{n}\sum_{\xi\in\Xi} P_\xi(x_i|x^{i-1})\alpha(\xi)\\ &= \sum_{\xi^n\in\Xi^n}\prod_{i=1}^{n} P_{\xi_i}(x_i|x^{i-1})\alpha(\xi_i) \qquad (5a)\\ &= \sum_{\xi^n} P_{\xi^n}(x^n)\pi_{\text{mix},\alpha}(\xi^n),\end{aligned}$$

which is the ES-joint based on the prior

$$\pi_{\text{mix},\alpha}(\xi^n) := \prod_{i=1}^{n}\alpha(\xi_i). \qquad (5b)$$

Thus, the ES-prior for elementwise mixtures is just the product distribution of α. ◇

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 Ξ. 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

$$\mathbb{A} = \left\langle Q, Q_\text{p}, \mathrm{P}_\circ, \mathrm{P}, \left\langle P_q\right\rangle_{q\in Q_\text{p}}\right\rangle$$

a *hidden Markov model* on $\mathcal{O}$ if Q is a countable set, $Q_\text{p}\subseteq Q$, $\mathrm{P}_\circ\in\triangle(Q)$, $\mathrm{P} : Q\to\triangle(Q)$ and P_q is an $\mathcal{O}$-forecasting system for each $q\in Q_\text{p}$.

Terminology and Notation We call elements of Q *states*. We call the states in Q_p *productive* and the other states *silent*. We call $\mathrm{P}_\circ$ the *initial distribution*, let I denote its support (i.e. $I := \left\{q\in Q \mid \mathrm{P}_\circ(q) > 0\right\}$) and call I the set of *initial states*. We call P the *stochastic transition function*. We let S_q denote the support of $\mathrm{P}(q)$, and call $q'\in S_q$ a *direct successor* of q. We abbreviate $\mathrm{P}(q)(q')$ to $\mathrm{P}(q\to q')$. A finite or infinite sequence of states $q^\lambda\in Q^{\leq\omega}$ is called a *branch* through $\mathbb{A}$. A branch q^λ 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^λ, we denote by q_p^λ its subsequence of productive states. We denote the elements of q_p^λ by q_1^p, q_2^p etc. We call an HMM *continuous* if q_p^ω is infinite for each infinite run q^ω.

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 $\lambda\geq 1$

$$\pi_\mathbb{A}(q^\lambda) := \mathrm{P}_\circ(q_1)\prod_{i=1}^{\lambda-1}\mathrm{P}(q_i\to q_{i+1}).$$

Then via the PFSs, $\mathbb{A}$ 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

$$P_\mathbb{A}(o^n, q^\lambda) := \pi_\mathbb{A}(q^\lambda)\prod_{i=1}^{n}P_{q_i^\text{p}}(o_i|o^{i-1}).$$

The value of $P_\mathbb{A}$ at arguments o^n, q^λ 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

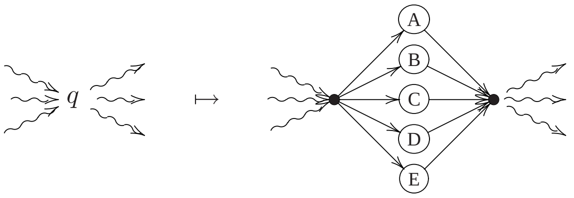
In applications HMMs are often used to model data. This is often useful if there are local correlations between outcomes. A graphical model depicting this approach is displayed in Figure 2a.

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 ξ^{n-1}. In other words, we can only use an HMM on Ξ 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 $q \in Q_{\mathrm{p}}$ by the following gadget:

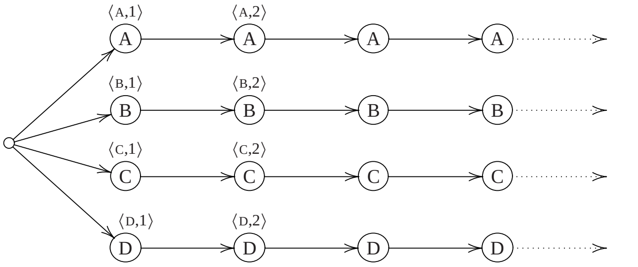


In the left diagram, the state q has distribution P_q on outcomes $\mathcal{O} = \{\text{A}, \dots, \text{E}\}$. In the right diagram, the leftmost silent state has transition probability $P_q(o)$ to a state that deterministically outputs outcome o. We often make the functional relationship explicit and by calling $\langle Q, Q_{\mathrm{p}}, \mathrm{P}_\circ, \mathrm{P}, \Lambda \rangle$ a *deterministic HMM* on $\mathcal{O}$ if $\Lambda : Q_{\mathrm{p}} \to \mathcal{O}$. 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. ●, for a silent state, and an ellipse labelled $\Lambda(q)$, e.g. Ⓓ, for a productive state. We draw an arrow from N_q to $N_{q'}$ if q' is a direct successor of q. We often reify the initial distribution $\mathrm{P}_\circ$ by including a virtual node, drawn as an open circle, e.g. ○, with an outgoing arrow to N_q for each initial state $q \in I$. The transition probability $P(q \to q')$ is not displayed in the graph.

Figure 3 Standard Bayesian mixture.



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 π_{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 Ξ that generates the prior π_{bayes} is given by $\mathbb{A}_{\text{bayes}} = \langle Q, Q_{\mathrm{p}}, \mathrm{P}_\circ, \mathrm{P}, \Lambda \rangle$, where

$$Q, Q_{\mathrm{p}} = \Xi \times \mathbb{Z}_+ \quad \Lambda(\xi, n) = \xi \quad \mathrm{P}_\circ(\xi, 1) = w(\xi) \tag{6a}$$

$$\mathrm{P}\left(\langle \xi, n \rangle \to \langle \xi, n+1 \rangle\right) = 1 \tag{6b}$$

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

$$-\log P_{\mathbb{A}_{\text{bayes}}}(x^n) + \log P_\xi(x^n) \leq -\log w(\xi). \tag{7}$$

In particular this bound holds for $\hat{\xi} = \operatorname{argmax}_\xi P_\xi(x^n)$, 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. ◇

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 α. Formally, $\mathbb{A}_{\text{mix},\alpha}$ is given using $Q = Q_{\mathrm{s}} \cup Q_{\mathrm{p}}$ by

$$\begin{aligned} Q_{\mathrm{s}} &= \{\mathsf{p}\} \times \mathbb{N} \quad & \mathrm{P}_\circ(\mathsf{p}, 0) &= 1 \\ Q_{\mathrm{p}} &= \Xi \times \mathbb{Z}_+ \quad & \Lambda(\xi, n) &= \xi \end{aligned} \tag{8a}$$

$$\mathrm{P}\begin{pmatrix} \langle \mathsf{p}, n \rangle \to \langle \xi, n+1 \rangle \\ \langle \xi, n \rangle \to \langle \mathsf{p}, n \rangle \end{pmatrix} = \begin{pmatrix} \alpha(\xi) \\ 1 \end{pmatrix} \tag{8b}$$

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 $O(n|\Xi|)$. ◇

Figure 2 HMMs. $\boldsymbol{q}_i^{\mathrm{p}}$, $\boldsymbol{\xi}_i$ and $\boldsymbol{x}_i$ are the i^{th} productive state, expert and observation.

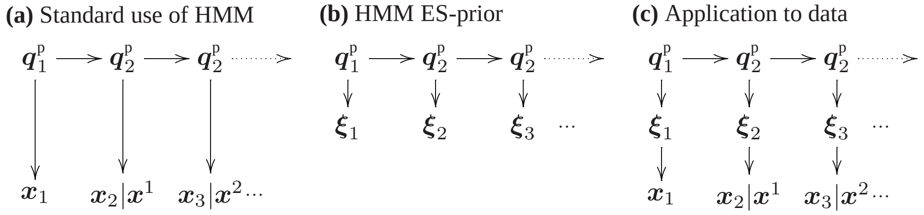
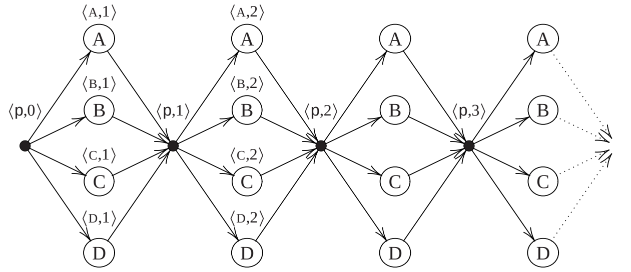

(a) Standard use of HMM **(b)** HMM ES-prior **(c)** Application to data
$\boldsymbol{q}_1^{\mathrm{p}}$
$\boldsymbol{q}_2^{\mathrm{p}}$
$\boldsymbol{q}_2^{\mathrm{p}}$
$\boldsymbol{\xi}_1$
$\boldsymbol{\xi}_2$
$\boldsymbol{\xi}_3$
···
$\boldsymbol{x}_1$
$\boldsymbol{x}_2|\boldsymbol{x}^1$
$\boldsymbol{x}_3|\boldsymbol{x}^2$ ···

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 Ξ^ω with a PFS P_ξ 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_ξ 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 $\mathbb{A} = \left\langle Q, Q_{\mathrm{p}}, \mathrm{P}_\circ, \mathrm{P}, \Lambda \right\rangle$ on Ξ. 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 $\mathbb{X} := \left\langle Q, Q_{\mathrm{p}}, \mathrm{P}_\circ, \mathrm{P}, \left\langle P_{\Lambda(q)} \right\rangle_{q \in Q_{\mathrm{p}}} \right\rangle$ 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 $\triangle(\Xi)$. Using (5) we obtain the following marginal distribution:

$$\begin{aligned} P_{\text{umix}}(x^n) &= \int_{\triangle(\Xi)} P_{\text{mix},\alpha}(x^n) w(\alpha) \, \mathrm{d}\alpha \\ &= \int_{\triangle(\Xi)} \sum_{\xi^n} P_{\xi^n}(x^n) \pi_{\text{mix},\alpha}(\xi^n) w(\alpha) \, \mathrm{d}\alpha \\ &= \sum_{\xi^n} P_{\xi^n}(x^n) \pi_{\text{umix}}(\xi^n), \quad \text{where} \\ \pi_{\text{umix}}(\xi^n) &= \int_{\triangle(\Xi)} \pi_{\text{mix},\alpha}(\xi^n) w(\alpha) \, \mathrm{d}\alpha. \end{aligned} \tag{9}$$

Thus P_{umix} is the ES-joint with ES-prior π_{umix}. This applies more generally: parameters α 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 α.

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 π, ρ be ES-priors s.t. ρ is zero whenever π 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_ρ is zero whenever P_π 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_{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. α. 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

$$\begin{aligned}-\log P_{\text{umix}}(x^n) + \log P_{\text{mix},\hat\alpha(x^n)}(x^n) \;\leq\;&\\ \max_{\xi^n}\Big(-\log\pi_{\text{umix}}(\xi^n) + \log\pi_{\text{mix},\hat\alpha(x^n)}(\xi^n)\Big)&. \qquad (10)\end{aligned}$$

We now bound the right hand side. Let $\hat\alpha(\xi^n)$ maximise $\pi_{\text{mix},\alpha}(\xi^n)$ w.r.t. α. Then for all x^n and ξ^n

$$\pi_{\text{mix},\hat\alpha(x^n)}(\xi^n) \;\leq\; \pi_{\text{mix},\hat\alpha(\xi^n)}(\xi^n). \qquad (11)$$

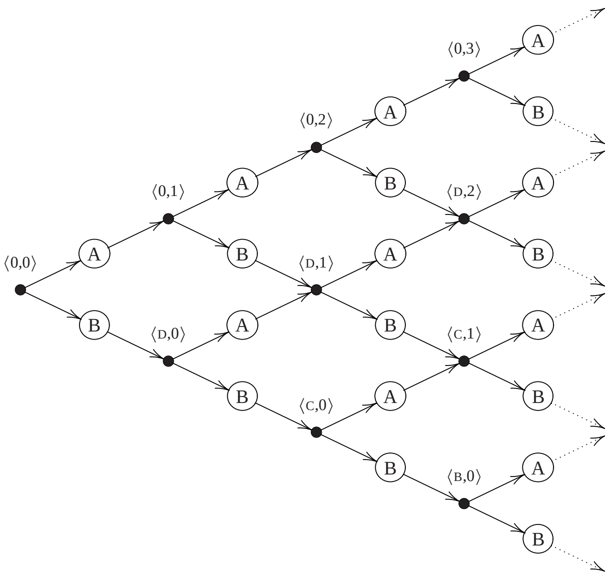
For the $\left(\frac{1}{2},\dots,\frac{1}{2}\right)$-Dirichlet prior, for all ξ^n

$$-\log\pi_{\text{umix}}(\xi^n) + \log\pi_{\text{mix},\hat\alpha(\xi^n)}(\xi^n) \;\leq\; \frac{|\Xi|-1}{2}\log\frac{n}{\pi} + c$$

for some fixed constant c (see e.g. [13]) Combination with (11) and (10) completes the proof. □

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_{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 π_{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 $\left(\frac{1}{2},\dots,\frac{1}{2}\right)$-Dirichlet prior is given using $Q = Q_{\text{s}} \cup Q_{\text{p}}$ by

$$Q_{\text{s}} = \mathbb{N}^\Xi \quad Q_{\text{p}} = \mathbb{N}^\Xi \times \Xi \quad \mathrm{P}_\circ(\mathbf{0}) = 1 \quad \Lambda(\vec n,\xi) = \xi$$

$$\mathrm{P}\begin{pmatrix} \langle\vec n\rangle \to \langle\vec n,\xi\rangle \\ \langle\vec n,\xi\rangle \to \langle\vec n + \mathbf{1}_\xi\rangle \end{pmatrix} = \begin{pmatrix} \frac{1/2+n_\xi}{|\Xi|/2+\sum_\xi n_\xi} \\ 1 \end{pmatrix} \qquad (12)$$

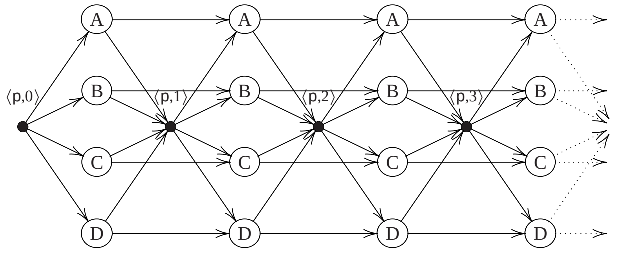
We write $\mathbb{N}^\Xi$ for the set of assignments of counts to experts; $\mathbf{0}$ for the all zero assignment, and $\mathbf{1}_\xi$ marks one count for expert ξ. We show the diagram of $\mathbb{A}_{\text{umix}}$ for the practical limit of two experts in Figure 5. In this case, the forward algorithm 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 α are possible. In particular, when all mass is placed on a single value of α, 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 Ξ.[2] 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 $Q = Q_{\text{s}} \cup Q_{\text{p}}$:

$$\begin{aligned} Q_{\text{s}} &= \{\mathsf{p}\} \times \mathbb{N} & \mathrm{P}_\circ(\mathsf{p}, 0) &= 1 \\ Q_{\text{p}} &= \Xi \times \mathbb{Z}_+ & \Lambda(\xi, n) &= \xi \end{aligned} \tag{13a}$$

$$\mathrm{P}\begin{pmatrix} \langle \mathsf{p}, n\rangle \to \langle \xi, n+1\rangle \\ \langle \xi, n\rangle \to \langle \mathsf{p}, n\rangle \\ \langle \xi, n\rangle \to \langle \xi, n+1\rangle \end{pmatrix} = \begin{pmatrix} w(\xi) \\ \alpha \\ 1-\alpha \end{pmatrix} \tag{13b}$$

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.[3]

Let $\hat{L}$ denote the loss achieved by the best partition, with switching rate $\alpha^* := m/(n-1)$. Let $L_{\text{fs},\alpha}$ denote the loss of fixed share with uniform w and parameter α. Herbster and Warmuth prove[4]

$$L_{\text{fs},\alpha} - \hat{L} \;\le\; (n-1)H(\alpha^*, \alpha) + (m-1)\log(|\Xi|-1) + \log|\Xi|\,,$$

which we for brevity loosen slightly to

$$L_{\text{fs},\alpha} - \hat{L} \;\le\; nH(\alpha^*, \alpha) + m\log|\Xi|\,. \tag{14}$$

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 value of α 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:

$$P_{\text{us}}(x^n) = \int P_{\text{fs},\alpha}(x^n) w(\alpha)\,\mathrm{d}\alpha.$$

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 $L_{\text{fs},\alpha}$ be as before, and let $L_{\text{us}} = -\log P_{\text{us}}(x^n)$ denote the loss of universal share with Jeffreys' prior $w(\alpha) = \alpha^{-1/2}(1-\alpha)^{-1/2}/\pi$. Then

$$L_{\text{us}} - \min_\alpha L_{\text{fs},\alpha} \le 1 + \tfrac{1}{2}\log n. \tag{15}$$

Thus P_{us} is universal for the model class $\left\{P_{\text{fs},\alpha} \mid \alpha \in [0,1]\right\}$ 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 $\left(\frac{1}{2}, \frac{1}{2}\right)$-Dirichlet prior on the switching rate α is displayed in Figure 7. It is formally specified (using $Q = Q_{\text{s}} \cup Q_{\text{p}}$) by:

$$\begin{aligned} Q_{\text{s}} &= \{\mathsf{p}, \mathsf{q}\} \times \left\{\langle m, n\rangle \in \mathbb{N}^2 \mid m \le n\right\} & \Lambda(\xi, m, n) &= \xi \\ Q_{\text{p}} &= \quad\;\; \Xi \quad\;\; \times \left\{\langle m, n\rangle \in \mathbb{N}^2 \mid m < n\right\} & \mathrm{P}_\circ(\mathsf{p}, 0, 0) &= 1 \end{aligned}$$

$$\mathrm{P}\begin{pmatrix} \langle \mathsf{p}, m, n\rangle \to \langle \xi, m, n+1\rangle \\ \langle \mathsf{q}, m, n\rangle \to \langle \mathsf{p}, m+1, n\rangle \\ \langle \xi, m, n\rangle \to \langle \mathsf{q}, m, n\rangle \\ \langle \xi, m, n\rangle \to \langle \xi, m, n+1\rangle \end{pmatrix} = \begin{pmatrix} w(\xi) \\ 1 \\ (m+\frac{1}{2})\big/\,n \\ (n-m-\frac{1}{2})\big/\,n \end{pmatrix} \tag{16}$$

Each productive state $\langle \xi, n, m\rangle$ represents the fact that at sample size n expert ξ 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

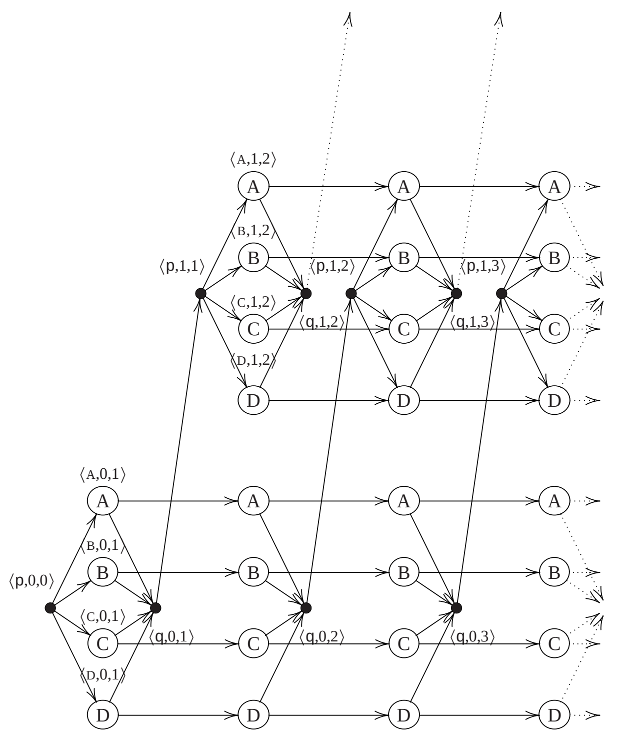
So far we have considered two models for switching between experts: fixed share and its generalisation, universal share.

[2] This is actually a slight generalisation: the original algorithm uses a uniform $w(\xi) = 1/|\Xi|$.

[3] 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 α can be simulated by reflexive switching with fixed rate $\beta = \frac{\alpha|\Xi|}{|\Xi|-1}$ (provided $\beta \le 1$). For non-uniform w, the rate becomes expert-dependent.

[4] This bound can be obtained for the fixed share HMM using the previous footnote.

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.[5] 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 σ^ω and τ^ω 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 $Q = Q_\text{s} \cup Q_\text{p}$:

$$Q_\text{s} = \left\{\mathsf{p}, \mathsf{p}_\mathsf{s}, \mathsf{p}_\mathsf{u}\right\} \times \mathbb{N} \qquad \mathrm{P}_\circ(\mathsf{p}, 0) = 1 \qquad \Lambda(\mathsf{s}, \xi, n) = \xi$$
$$Q_\text{p} = \{\mathsf{s}, \mathsf{u}\} \times \Xi \times \mathbb{Z}_+ \qquad\qquad \Lambda(\mathsf{u}, \xi, n) = \xi$$

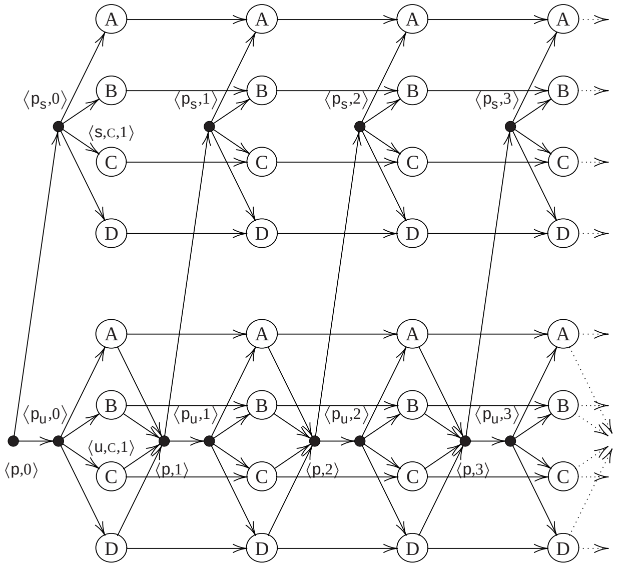
$$\mathrm{P}\begin{pmatrix} \langle \mathsf{p}, n\rangle \to \left\langle \mathsf{p}_\mathsf{u}, n\right\rangle \\ \langle \mathsf{p}, n\rangle \to \left\langle \mathsf{p}_\mathsf{s}, n\right\rangle \\ \left\langle \mathsf{p}_\mathsf{u}, n\right\rangle \to \langle \mathsf{u}, \xi, n+1\rangle \\ \left\langle \mathsf{p}_\mathsf{s}, n\right\rangle \to \langle \mathsf{s}, \xi, n+1\rangle \\ \langle \mathsf{s}, \xi, n\rangle \to \langle \mathsf{s}, \xi, n+1\rangle \\ \langle \mathsf{u}, \xi, n\rangle \to \langle \mathsf{u}, \xi, n+1\rangle \\ \langle \mathsf{u}, \xi, n\rangle \to \langle \mathsf{p}, n\rangle \end{pmatrix} = \begin{pmatrix} \tau_n(0) \\ \tau_n(1) \\ w(\xi) \\ w(\xi) \\ 1 \\ \sigma_n(0) \\ \sigma_n(1) \end{pmatrix}$$

This HMM contains two "expert bands". Consider a productive state $\langle \mathsf{u}, \xi, n\rangle$ 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 $\langle \mathsf{u}, \xi, n+1\rangle$ and the story repeats. We say that *no switch occurs*. With probability $\sigma_n(1)$ the process continues to the silent state $\langle \mathsf{p}, n\rangle$ 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 $\left\langle \mathsf{p}_\mathsf{u}, n\right\rangle$ and then branches out to some productive state $\left\langle \mathsf{u}, \xi', n+1\right\rangle$ (possibly $\xi = \xi'$), and the story repeats. With probability $\tau_n(1)$ the process continues to $\left\langle \mathsf{p}_\mathsf{s}, n\right\rangle$ in the top band, called the *stable* band. Also here it branches out to some productive state $\left\langle \mathsf{s}, \xi', n+1\right\rangle$. But from this point onward there are no choices anymore; expert ξ' 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 θ. The presence of the stable band enables us to improve the loss bound of fixed share in the particular

[5] 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 $\sigma_n(1) = \pi_{\mathrm{T}}(\boldsymbol{Z} = n|\boldsymbol{Z} \geq n)$ an arbitrary distribution π_{T} 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 ξ^n. Let m denote the number of blocks in ξ^n, where the blocks are the maximal subsequences containing only a single expert. Let $1 = t_1 < t_2 < \cdots < t_m \leq n$ be the indices where the blocks start. Then*

$$\pi_{\mathrm{sw}}(\xi^n) \geq 2^{-m} w(\xi_1) \prod_{i=2}^{m} w(\xi_{t_i}) \pi_{\mathrm{T}}(\boldsymbol{Z} = t_i | \boldsymbol{Z} > t_{i-1}).$$

Theorem 6. *Fix data x^n. Let ξ^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 ξ^n. Let $\pi_{\mathrm{T}}(n) = 1/(n(n-1))$ and w be uniform. Then the loss overhead $-\log P_{\mathrm{sw}}(x^n) + \log P_{\xi^n}(x^n)$ of the switch distribution is bounded by*

$$m + m \log|\Xi| + \log \binom{t_m}{m} + \log(m!).$$

Proof. We have

$$\begin{aligned} -\log P_{\mathrm{sw}}(x^n) + \log P_{\xi^n}(x^n) \;&\leq\; -\log \pi_{\mathrm{sw}}(\xi^n) \\ &\leq -\log\left(2^{-m} w(\xi_1) \prod_{i=2}^{m} \pi_{\mathrm{T}}(t_i | t_i > t_{i-1}) w(\xi_{t_i})\right) \\ &= m + m\log|\Xi| - \sum_{i=2}^{m} \log \pi_{\mathrm{T}}(t_i|t_i > t_{i-1}). \end{aligned} \tag{17}$$

The prior π_{T} may be written $\pi_{\mathrm{T}}(n) = \frac{1}{n-1} - \frac{1}{n}$, so that

$$\pi_{\mathrm{T}}(t_i|t_i > t_{i-1}) = \frac{1/(t_i(t_i-1))}{\sum_{n>t_{i-1}} \left(\frac{1}{n-1} - \frac{1}{n}\right)} = \frac{t_{i-1}}{t_i(t_i-1)}.$$

If we substitute this in the last term of (17), the sum telescopes and we are left with

$$\underbrace{-\log(t_1)}_{=0} + \log(t_m) + \sum_{i=2}^{m} \log(t_i - 1). \tag{18}$$

If we fix t_m, this expression is maximised if $t_2, \ldots, t_{m-1}$ take on the values $t_m - m + 2, \ldots, t_m - 1$, 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. □

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

$$\begin{aligned} &\left(m + m\log|\Xi| + \log\binom{n}{m} + \log(m!)\right) - \\ &\qquad \left(\log\binom{n}{m} + m\log|\Xi|\right) \\ &\qquad = m + \log(m!) \;\leq\; m + m\log m. \end{aligned} \tag{19}$$

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 α 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 π_{T}. 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 π_{T}. 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 $\mathbb{S} := \big\{\langle m,n\rangle \in \mathbb{N}^2 \mid m<n\big\}$, and let π_{T} be a distribution on $\mathbb{Z}_+$. The specification of the run-length HMM is given using $Q = Q_{\mathrm{s}} \cup Q_{\mathrm{p}}$ by:

$$\begin{aligned} Q_{\mathrm{s}} &= \{\mathsf{q}\}\times\mathbb{S} \,\cup\, \{\mathsf{p}\}\times\mathbb{N} & \Lambda(\xi,m,n) &= \xi \\ Q_{\mathrm{p}} &= \Xi\times\mathbb{S} & \mathrm{P}_{\circ}(\mathsf{p},0) &= 1 \end{aligned}$$

$$\mathrm{P}\begin{pmatrix} \langle \mathsf{p},n\rangle \to \langle \xi,n,n+1\rangle \\ \langle \xi,m,n\rangle \to \langle \xi,m,n+1\rangle \\ \langle \xi,m,n\rangle \to \langle \mathsf{q},m,n\rangle \\ \langle \mathsf{q},m,n\rangle \to \langle \mathsf{p},n\rangle \end{pmatrix} = \begin{pmatrix} w(\xi) \\ \pi_{\mathrm{T}}(\boldsymbol{Z}>n|\boldsymbol{Z}\geq n) \\ \pi_{\mathrm{T}}(\boldsymbol{Z}=n|\boldsymbol{Z}\geq n) \\ 1 \end{pmatrix}$$

5.2.2 A Loss Bound

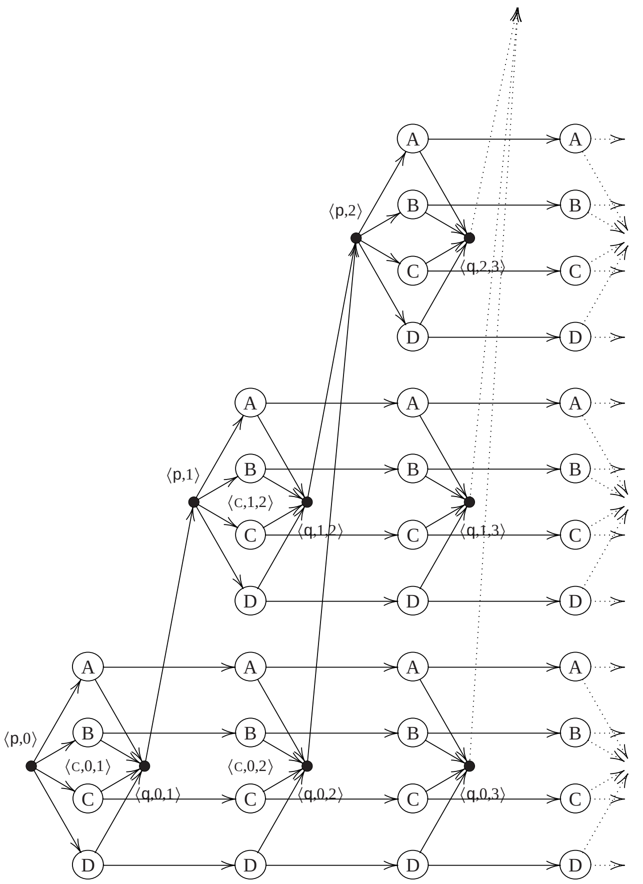
Fix an expert sequence ξ^n with m blocks. For $i=1,\ldots,m$, let δ_i and k_i denote the length and expert of block i. From the definition of the HMM above, we obtain that $\pi_{\mathrm{rl}}(\xi^n)$ equals

$$\sum_{i=1}^{m} -\log w(k_i) + \sum_{i=1}^{m-1} -\log \pi_{\mathrm{T}}(\boldsymbol{Z}=\delta_i) - \log \pi_{\mathrm{T}}(\boldsymbol{Z}\geq\delta_m).$$

Theorem 7. *Fix data x^n. Let ξ^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 π_{T} be log-convex. Then the loss overhead is bounded thus*

$$-\log P_{\mathrm{rl}}(x^n) + \log P_{\xi^n}(x^n) \;\leq\; m\left(\log|\Xi| - \log\pi_{\mathrm{T}}\left(\tfrac{n}{m}\right)\right).$$

Figure 9 The run-length model



Proof. Let δ_i denote the length of block i. We overestimate

$$\begin{aligned} &-\log P_{\mathrm{rl}}(x^n) + \log P_{\xi^n}(x^n) \;\leq\; -\log\pi_{\mathrm{rl}}(\xi^n) \\ &= m\log|\Xi| + \sum_{i=1}^{m-1} -\log\pi_{\mathrm{T}}(\boldsymbol{Z}=\delta_i) - \log\pi_{\mathrm{T}}(\boldsymbol{Z}\geq\delta_m) \\ &\leq m\log|\Xi| + \sum_{i=1}^{m} -\log\pi_{\mathrm{T}}(\delta_i). \end{aligned} \tag{20}$$

Since $-\log\pi_{\mathrm{T}}$ is concave, by Jensen's inequality we have

$$\sum_{i=1}^{m} \frac{-\log\pi_{\mathrm{T}}(\delta_i)}{m} \leq -\log\pi_{\mathrm{T}}\left(\sum_{i=1}^{m}\frac{\delta_i}{m}\right) = -\log\pi_{\mathrm{T}}\left(\frac{n}{m}\right).$$

In other words, the block lengths δ_i are all equal in the worst case. Plugging this into (20) we obtain the theorem. □

5.2.3 Finite Support

We have seen that the run-length model reduces to fixed share if the prior on switch distances π_{T} is geometric, so that it can be evaluated in linear time in that case. We also obtain a linear time algorithm when π_{T} 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 π_{T} 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 π_{T} 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 π_{T} for the run-length model that satisfies $-\log \pi_{\text{T}}(n) \leq \log n + 2\log\log(n+1) + 3$ (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 π_{T} as above) if $m = o\big((\log n)^2\big)$, and asymptotically higher if $m = \Omega\big((\log n)^2\big)$.*[6]

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. □

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 its 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.

[6] Let $f, g : \mathbb{N} \to \mathbb{N}$. We say $f = o(g)$ if $\lim_{n\to\infty} f(n)/g(n) = 0$. We say $f = \Omega(g)$ if $\exists c > 0 \exists n_0 \forall n \geq n_0 : f(n) \geq cg(n)$.