# Switching between Hidden Markov Models using Fixed Share

**Wouter M. Koolen** and **Tim van Erven**
Centrum Wiskunde & Informatica (CWI)
P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands
{Wouter.Koolen, Tim.van.Erven}@cwi.nl

## Abstract

In prediction with expert advice the goal is to design online prediction algorithms that achieve small regret (additional loss on the whole data) compared to a reference scheme. In the simplest such scheme one compares to the loss of the best expert in hindsight. A more ambitious goal is to split the data into segments and compare to the best expert on each segment. This is appropriate if the nature of the data changes between segments. The standard fixed-share algorithm is fast and achieves small regret compared to this scheme.

Fixed share treats the experts as black boxes: there are no assumptions about how they generate their predictions. But if the experts are learning, the following question arises: should the experts learn from all data or only from data in their own segment? The original algorithm naturally addresses the first case. Here we consider the second option, which is more appropriate exactly when the nature of the data changes between segments. In general extending fixed share to this second case will slow it down by a factor of $T$ on $T$ outcomes. We show, however, that no such slowdown is necessary if the experts are hidden Markov models.

## 1 Introduction

In *prediction with expert advice* [Cesa-Bianchi and Lugosi, 2006] a sequence of outcomes $x_1, x_2, \ldots$ needs to be predicted, one outcome at a time. Thus, prediction proceeds in rounds: in each round we first consult a set of experts, who give us their predictions. (We use the word *expert* for any source of predictions that is available to us as input.) Then we make our own prediction and incur some loss based on the discrepancy between our prediction and the actual outcome. Predictions may for example be in the form of a probability distribution on outcomes. Loss may be logarithmic loss, i.e. the negative logarithm of the probability assigned to the outcome that actually occurs. The goal is to minimise our *regret*, which is the difference between our own cumulative loss on the whole data and the cumulative loss of a *reference scheme*, which typically involves tuned parameter settings unknown to us when we make our predictions. For the reference scheme there are several options; we may, for example, compare ourselves to the cumulative loss of the best expert in hindsight (after observing the data). A more ambitious scheme, called *tracking the best expert*, is addressed by the fixed-share algorithm of Herbster and Warmuth [1998].

### 1.1 Tracking the Best Expert

In tracking the best expert (TBE), the goal is to achieve small regret compared to the following reference scheme:

(a) Split the data into segments.
(b) Select an expert for each segment.
(c) Sum the loss of the selected experts on their segments.

This reference scheme is appropriate if the nature of the data changes between segments. It is harder than comparing to the single best expert in hindsight, because now there are more unknowns: both the segmentation (step a) and the reference experts (step b) are unknown when we make our predictions. In particular the reference experts may be the best experts in hindsight for their assigned segments.

In 1995 Herbster and Warmuth introduced an efficient algorithm called *fixed share* (FS) and showed that it achieves small regret (see Theorem 1 below) compared to the TBE reference scheme of Herbster and Warmuth [1998]. Given the predictions of the experts, the algorithm's running time is linear in the number of outcomes and linear in the number of experts. Problem solved. Or is it?

## 1.2 Learning Experts

In this paper we take another look at the TBE reference scheme for *learning experts* and ask: if an expert is selected for some segment, then should the expert learn from all data or only from the data in that segment?

We may assume that the experts do not know the segmentation chosen in step a of the reference scheme. (Otherwise, why wouldn't we just ask them?) Hence if we treat the experts as black boxes and only ask for their prediction at each time step as in [Herbster and Warmuth, 1998], it is natural that they learn from all data. We call this the *standard* interpretation of the TBE reference scheme (S-TBE).

However, as the following example will illustrate, it may be beneficial if experts learn only from the segment for which they are selected, because they may get confused by data in other segments that follow a different pattern. We call this the *local learners* interpretation of tracking the best expert (LL-TBE). As a slight complication, it will turn out that in LL-TBE we have a further choice: whether to tell a learning expert the timing of its segment or not, which generally makes a difference. When segment timing is preserved, we call the resulting reference scheme *sleeping LL-TBE*; when segment timing is *not* preserved we call the reference scheme *freezing LL-TBE*. The next example demonstrates that S-TBE and the two variants of LL-TBE are really different reference schemes.

**Example: Drifting Mean** In applications one would usually build up complicated prediction strategies from simpler ones in a hierarchical fashion. For example, let us first define simple static experts, parametrised by $\mu \in \mathbb{R}$, which predict according to a standard normal distribution with mean $\mu$ in each round. Now define a learning expert $\mathsf{DM}[\theta]$ that has a stochastic model for the (unobservable) drift of $\mu$ over time. This *drifting mean* learning expert predicts according to a hidden Markov model in which the hidden state at time $t$ is $\mu_t$ and the production probability of an outcome given $\mu_t$ is determined by the simple expert with parameter $\mu_t$. Initially, $\mu_1 = 0$ with probability one. Then $\mu_{t+1} = \mu_t + 1$ with probability $\theta$ and $\mu_{t+1} = \mu_t$ with probability $1 - \theta$ for some fixed parameter $\theta$. (See Figure 1.)

The expert $\mathsf{DM}[\theta]$ may be said to be learning, because its posterior distribution of $\mu_t$ given outcomes $x_1, \ldots, x_{t-1}$ indicates how much credibility the expert assigns to each value of $\mu_t$: high weight on, say, $\mu_t = 3$

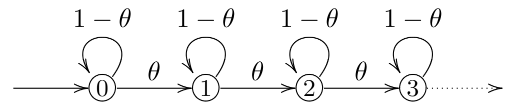
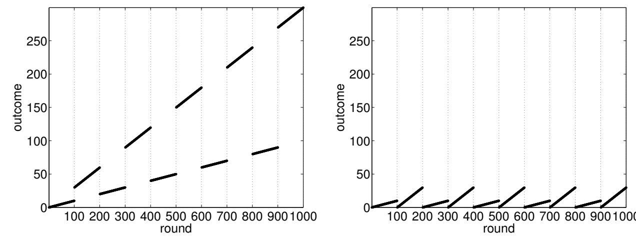
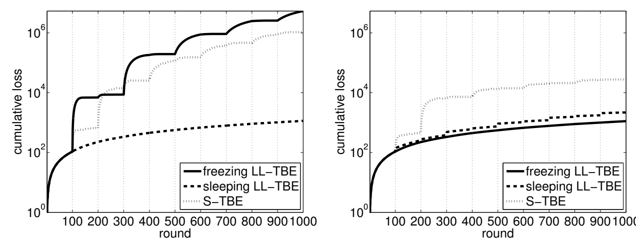



Figure 1: State Transitions for Learning Expert $\mathsf{DM}[\theta]$, which learns a drifting mean




(a) Suitable Sleeping LL-TBE Data (b) Suitable Freezing LL-TBE Data




(c) Cumul. Loss on Data (a) (d) Cumul. Loss on Data (b)

Figure 2: The Difference Between S-TBE and the Two LL-TBE Reference Schemes. Note the logarithmic scale of the y-axis in (c) and (d)!

indicates that $\mathsf{DM}[\theta]$ considers it likely for $\mu_t = 3$ to give the best prediction for $x_t$.

Figures 2a and 2b plot two artificial data sets. For Figure 2a sleeping LL-TBE is appropriate, for Figure 2b freezing LL-TBE is more suitable. The data consist of 10 segments of 100 outcomes. In each segment the outcomes are increasing deterministically at a rate of either 0.1 or 0.3 per outcome. Note that for the freezing data all segments start from 0, whereas for sleeping any segment looks like the proces that generated it started at 0 at time 1, but went unobserved for a while.

Figures 2c and 2d show the cumulative log(arithmic) loss for all three TBE reference schemes. Note that the difference between the schemes is so large that their losses had to be plotted on a logarithmic scale. In each case we consider two experts: $\mathsf{DM}[0.1]$ and $\mathsf{DM}[0.3]$ and use the expert $\mathsf{DM}[\theta]$ for any segment with rate $\theta$. The difference between the three schemes lies in which data is used by $\mathsf{DM}[\theta]$ to learn from. In the S-TBE scheme $\mathsf{DM}[\theta]$ is shown all the data, even those outside the segment it has to predict. In the two LL-TBE schemes, on the other hand, a fresh copy of $\mathsf{DM}[\theta]$ only sees the data in the segment for which it is selected: for freezing LL-TBE, $\mathsf{DM}[\theta]$ predicts as if the current segment is the only data; for sleeping LL-TBE, $\mathsf{DM}[\theta]$ knows the timing of the segment it is predicting, and treats all samples preceding that segment as unobserved. Thus in sleeping LL-TBE the original timing of the segments is preserved, while in freezing LL-TBE it is lost.

We see that for the sleeping data the sleeping LL-TBE reference scheme has much smaller loss than the other two schemes. And for the freezing data the freezing LL-TBE scheme has the smallest loss by far. (Mind the logarithmic scale of the y-axis, which puts the loss of sleeping LL-TBE deceptively close to the loss of freezing LL-TBE in Figure 2d: a constant offset indicates a fixed multiplicative overhead.) In both cases the reason for the large differences between the reference schemes is that $\mathsf{DM}[\theta]$ gets confused if it learns from the wrong data.

### 1.3 Expert Hidden Markov Models

The learning expert $\mathsf{DM}[\theta]$ in the example above is a hidden Markov model in which the production probabilities (of outcomes given the state) depend on lower-level base experts. In general such prediction strategies are called *expert hidden Markov models* (EHMMs). The use of EHMMs is not restricted to describing learning experts. For example, many algorithms for prediction with expert advice, including $\mathsf{FS}$ itself, can be represented as EHMMs (see Koolen and De Rooij [2008a] and its references, and Monteleoni and Jaakkola [2003]). In addition any ordinary HMM is trivially an EHMM: just introduce lower-level base experts for its production probabilities. Not every algorithm can be represented as an EHMM, however. The follow-the-perturbed-leader algorithm by Hannan [1957] and *variable share* by Herbster and Warmuth [1998], for instance, are exceptions.

### 1.4 Fixed Share for Learning Experts

**LL-TBE Requires More Information** The example above shows that there is a large difference between S-TBE and the sleeping or freezing LL-TBE reference schemes. One may therefore wonder whether there exists an algorithm that achieves small regret compared to LL-TBE. Unfortunately, no algorithm will be able to do the job without additional knowledge about the learning experts. To see this, note that the reference scheme may split the data into segments in any way it sees fit. But black-box experts are not telling us what their predictions would be for any possible segmentation; they only give us a single prediction each round. Therefore, even if we knew the segmentation and the selected expert for each segment, we still would have insufficient information to achieve the reference scheme. The only way to address this problem is to get more information about the learning experts. This information should have an efficient representation and should somehow tell us what the learning experts would predict for any possible segmentation.

**Copying Experts is Less Efficient** The straightforward approach would be to introduce a fresh copy of each expert for each possible start of a new segment and run the original fixed-share algorithm on the resulting enriched set of experts. But then the number of experts would grow linearly with the number of rounds, and consequently the total running time would go up from linear to quadratic in the number of outcomes. As this makes the difference between an online algorithm that can run forever and an algorithm that effectively comes to a stop after, say, $10^5$ outcomes, it is worth seeing whether such an increase in running time is really unavoidable.

**EHMMs: the Efficient Special Case** As we will show, it turns out there is a special class of learning experts for which no increase in running time is necessary. These are the learning experts that can be described in EHMM form. Although this excludes learning experts that for example implement follow-the-perturbed-leader, the class of EHMMs is still rich enough to be of interest, if only because it includes all ordinary HMMs. In the interpretation of the two LL-TBE reference schemes for learning experts in EHMM form, we do need to be careful if the base experts in the EHMMs are learning themselves: because we make no assumptions about the base experts, they always learn from all the data.

**Main Result: Achieving LL-TBE Efficiently** We present two new algorithms: $\mathsf{FS}^{\mathsf{sl}}$ for sleeping LL-TBE and $\mathsf{FS}^{\mathsf{fr}}$ for freezing LL-TBE, which both generalise $\mathsf{FS}$. We show that these algorithms achieve the same regret bound compared to their respective LL-TBE reference schemes as $\mathsf{FS}$ achieves compared to the S-TBE reference scheme. In addition, $\mathsf{FS}^{\mathsf{sl}}$ runs equally fast as the original fixed-share algorithm; for $\mathsf{FS}^{\mathsf{fr}}$ no slowdown occurs either if the EHMMs for the learning experts have a finite number of hidden states, otherwise it is typically still faster than just copying the experts.

Like fixed share, our new algorithms can be represented as EHMMs. In fact, we will build up both algorithms by describing how to combine the EHMMs for the learning experts, which the algorithms get as inputs, into a single larger EHMM. Apart from introducing the LL-TBE reference scheme, this construction is our main result: regret bounds follow from the EHMM representations using methods described in [Koolen and De Rooij, 2008a], and the algorithms are simply instances of the forward algorithm for EHMMs.

### 1.5 Overview

We start by formally introducing prediction with expert advice in the next section. Then §3 reviews EHMMs, including the representation of FS as an EHMM. It is shown how the standard regret bound for FS by Herbster and Warmuth [1998] can be proved using this representation. In §4 we formally define the freezing and sleeping LL-TBE reference schemes and present our new algorithms. Then we prove their regret bounds and state their running times.

## 2 Preliminaries: Prediction With Expert Advice

In this section we formally introduce the online learning setting of prediction with expert advice. In this setting prediction proceeds in rounds. In each round $t$, we first receive advice from each expert $e \in \mathcal{E}$ in the form of an action $a_t^e \in \mathcal{A}$. Then we distill our own action $a_t \in \mathcal{A}$ from the expert advice. Finally, the actual outcome $x_t \in \mathcal{X}$ is observed, and everybody suffers loss as specified by a fixed loss function $\ell \colon \mathcal{A} \times \mathcal{X} \to [0, \infty]$. Thus, the performance of a sequence of actions $a_{1:T} = a_1, \ldots, a_T$ on data $x_{1:T} = x_1, \ldots, x_T$ is measured by the cumulative loss $\ell(a_{1:T}, x_{1:T}) = \sum_{t=1}^{T} \ell(a_t, x_t)$.

We present our results for *log(arithmic) loss* only, which allows us to draw on familiar concepts from probability theory, like e.g. conditional probabilities and hidden Markov models. Their generalisation to arbitrary *mixable losses* is straight-forward using the methods of Vovk [1999].

**Log Loss** For log loss the actions $\mathcal{A}$ are probability mass (or density) functions on $\mathcal{X}$ and $\ell(\mathrm{p}, x) = -\log \mathrm{p}(x)$ for any $\mathrm{p} \in \mathcal{A}$, where log denotes the natural logarithm. Notice that minimising log loss is equivalent to maximising the predicted probability of outcome $x$. We write $\mathrm{p}_t^e$ for the prediction of expert $e$ at time $t$ and denote the predictions for all experts jointly by $\mathrm{p}_t^{\mathcal{E}}$. Another important property of the log loss is the *chain rule*: interpreting any prediction $\mathrm{p}_t(x_t)$ as the conditional probability $P(x_t | x_{<t})$ of outcome $x_t$ given all past outcomes $x_{<t} = x_1, \ldots, x_{t-1}$, we see that the

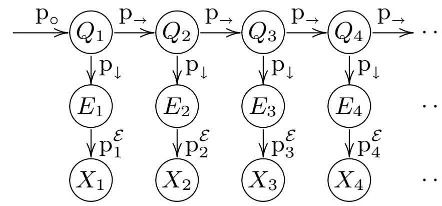


Figure 3: Bayesian Network Specification of an EHMM

cumulative log loss of a sequence of predictions

$$\sum_{t=1}^{T} -\log \mathrm{p}_t(x_t) \;=\; -\log \prod_{t=1}^{T} P(x_t | x_{<t}) \;=\; -\log P(x_{1:T}) \tag{1}$$

equals the negative logarithm of the joint $P$-probability of all data $x_{1:T}$. Thus any lower bound on $P(x_{1:T})$ directly implies an upper bound on the cumulative loss of predictions $\mathrm{p}_1, \ldots, \mathrm{p}_T$ on data $x_{1:T}$.

**Segments** For $m \leq n$, we abbreviate the *segment* $\{m, \ldots, n\}$ to $m{:}n$. For any sequence $y_1, y_2, \ldots$ and any segment $\mathcal{C} = m{:}n$ we write $y_{\mathcal{C}}$ for the subsequence $y_m, \ldots, y_n$. For example, $x_{m:n} = x_m, \ldots, x_n$ and $\mathrm{p}_{1:T}^{\mathcal{E}} = \mathrm{p}_1^{\mathcal{E}}, \ldots, \mathrm{p}_T^{\mathcal{E}}$. If all segments in a family $\mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots\}$ are pairwise disjoint and together cover $1{:}T$, then we call $\mathbb{C}$ a *segmentation* of $1{:}T$. We denote by $\langle e_{\mathcal{C}} \in \mathcal{E} \rangle_{\mathcal{C} \in \mathbb{C}}$ the labelling that assigns expert $e_{\mathcal{C}}$ to segment $\mathcal{C}$.

## 3 Expert Hidden Markov Models

EHMMs were introduced by Koolen and De Rooij [2008a] as a graphical and computational language to specify strategies for prediction with expert advice. EHMM diagrams directly represent the internal structure of the prediction strategy, facilitating the derivation of loss bounds. Moreover, there is a standard algorithm for sequential prediction, the *forward algorithm*, which greatly simplifies derivation of running time bounds.

In this paper, we use EHMMs in two ways. On the input side, we use them to represent the learning experts whose predictions we want to combine. On the output side, we specify our own prediction strategies based on expert advice as EHMMs.

An *EHMM* $\mathfrak{H}$ is a probability distribution that is constructed according to the Bayesian network in Figure 3. It is used to sequentially predict outcomes $X_1$, $X_2$, …, which take values in outcome space $\mathcal{X}$, using advice from a set of experts $\mathcal{E}$. At each time $t$, the distribution of $X_t$ depends on a hidden state $Q_t$, which determines mixing weights for the experts' predictions.

Formally, the *production function* $\mathrm{p}_\downarrow$ determines the interpretation of a state: it maps any state $q_t \in \mathcal{Q}$ to a distribution $\mathrm{p}_\downarrow^{q_t}$ on the identity $E_t$ of the expert that should be used to predict $X_t$. Then given $E_t = e$, the distribution of $X_t$ is expert $e$'s prediction $\mathrm{p}_t^e$. It remains to define the distribution of the hidden states. The starting state $Q_1$ has *initial distribution* $\mathrm{p}_\circ$, and the state evolves according to the *transition function* $\mathrm{p}_\rightarrow$, which maps any state $q_t$ to a distribution $\mathrm{p}_\rightarrow^{q_t}$ on its successor states.

An EHMM $\mathfrak{H}$ defines a prediction strategy as follows: after observing $x_{<t}$, predict the next outcome $X_t$ using the marginal $\mathfrak{H}(X_t|x_{<t})$, which is a *mixture* of the experts' predictions $\mathrm{p}_t^{\mathcal{E}}$.

We present four example EHMMs. The first three examples are suitable as input learning experts, which might be combined in the sleeping or freezing LL-TBE reference scheme. The fourth example represents $\mathsf{FS}$ as an EHMM, which will later be helpful when we compare it to our new generalisations.

**Example 3.1** (Figure 1: Expert that Learns a Drifting Mean)**.** Here we formally define the EHMM $\mathsf{DM}[\theta]$ from the example in the introduction. Recall that the base experts predict according to standard normal distributions with fixed mean $\mu$, which only takes integer values. Thus

$$\mathrm{p}_t^\mu(x) \coloneqq \frac{1}{\sqrt{2\pi}} e^{-(x-\mu)^2/2}$$

for all $\mu \in \mathcal{E} \coloneqq \mathbb{N} = \{0, 1, 2, \ldots\}$. In this EHMM it is sufficient to have a one-to-one correspondence between hidden states and experts, such that $Q_t = E_t$. This is expressed by $\mathcal{Q} \coloneqq \mathcal{E}$ and $\mathrm{p}_\downarrow \coloneqq \mathbf{I}$, where $\mathbf{I}$ denotes the identity operator. The definition of $\mathsf{DM}[\theta]$ is completed by letting the initial distribution $\mathrm{p}_\circ$ be a point-mass on $\mu = 0$, and defining the transition function $\mathrm{p}_\rightarrow$ as in Figure 1: for any two states $\mu, \mu' \in \mathcal{Q}$

$$\mathrm{p}_\rightarrow^\mu(\mu') \coloneqq \begin{cases} \theta & \text{if } \mu' = \mu + 1, \\ 1-\theta & \text{if } \mu' = \mu, \\ 0 & \text{otherwise.} \end{cases} \qquad \diamond$$

**Example 3.2** (Bayes on base experts)**.** Consider the Bayesian mixture (also known as the exponentially weighted average predictor) of base experts $\mathcal{E}$ with prior $w$. We identify this prediction strategy with the following EHMM $\mathsf{B}[w]$, which makes the same predictions. As in the previous example, let $\mathcal{Q} \coloneqq \mathcal{E}$ and $\mathrm{p}_\downarrow \coloneqq \mathbf{I}$, so that $Q_t = E_t$. This time, however, let $\mathrm{p}_\circ \coloneqq w$ and $\mathrm{p}_\rightarrow \coloneqq \mathbf{I}$. Despite its deceptive simplicity, this EHMM *learns*: its marginal distribution of $X_{t+1}$ given previous outcomes $x_{1:t}$ is a mixture of the base expert's predictions according to the Bayesian posterior. $\diamond$

**Example 3.3** (Bayes on EHMMs)**.** Let $\mathcal{H} = \{\mathfrak{H}^1, \ldots, \mathfrak{H}^n\}$ be EHMMs with base experts $\mathcal{E}^1, \ldots, \mathcal{E}^n$, and let $w$ be a prior on $\mathcal{H}$. Then, instead of treating $\mathfrak{H}^1, \ldots, \mathfrak{H}^n$ as black box predictors as in the previous example, their Bayesian mixture can also be expressed as a single EHMM $\mathsf{B}[w, \mathcal{H}]$ on the union of their base experts $\mathcal{E} \coloneqq \bigcup_{i=1}^n \mathcal{E}^i$: assume without loss of generality that $\mathfrak{H}^1, \ldots, \mathfrak{H}^n$ have disjoint state spaces $\mathcal{Q}^1, \ldots, \mathcal{Q}^n$ and let $\mathcal{Q} \coloneqq \bigcup_{i=1}^n \mathcal{Q}^i$. For any state $q \in \mathcal{Q}^i$, let $\mathrm{p}_\downarrow^q$ equal $\mathrm{p}_\downarrow^{q,i}$, where $\mathrm{p}_\downarrow^i$ is the production function of $\mathfrak{H}^i$, so that all states keep their original interpretation. In addition let $\mathrm{p}_\circ(q) \coloneqq w(i)\,\mathrm{p}_\circ^i(q)$, where $\mathrm{p}_\circ^i$ denotes the initial distribution of $\mathfrak{H}^i$. Finally, let $\mathrm{p}_\rightarrow^q(q')$ equal $\mathrm{p}_\rightarrow^{q,i}(q')$, the transition probability from $q$ to $q'$ for $\mathfrak{H}^i$ if $q, q' \in \mathcal{Q}^i$ and let $\mathrm{p}_\rightarrow^q(q') \coloneqq 0$ otherwise. Again, this EHMM *learns* which of the EHMMs in $\mathcal{H}$ is the best predictor. $\diamond$

**Example 3.4** (Fixed share)**.** The fixed-share algorithm take a parameter $\alpha$, called the *switching rate*. Fixed share with prior distribution $w$ on experts $\mathcal{E}$ and switching rate $\alpha$ can be represented as an EHMM $\mathsf{FS}[\alpha, w]$ as follows. As in the Bayesian mixture on base experts, let $\mathcal{Q} \coloneqq \mathcal{E}$ and $\mathrm{p}_\downarrow \coloneqq \mathbf{I}$, so that $Q_t = E_t$, and let $\mathrm{p}_\circ \coloneqq w$. Instead of the identity operator, however, use the transition function

$$\mathrm{p}_\rightarrow \coloneqq (1-\alpha)\mathbf{I} + \alpha w \mathbf{1}^{\mathrm{T}},$$

where $\mathbf{1}^{\mathrm{T}}$ denotes the operator that sums the probability masses of all the hidden states. This transition function may be interpreted as follows: behave like the Bayesian mixture with probability $1-\alpha$, but with probability $\alpha$ take all the probability mass and redistribute it according to the prior $w$. Observe that for any probability distribution $\lambda$ on states $\mathcal{Q}$, we can compute $\mathrm{p}_\rightarrow \lambda = (1-\alpha)\lambda + \alpha w$ in constant time per state. We also note that in [Herbster and Warmuth, 1998] the prior $w$ is always taken to be the uniform distribution, which gives the best worst-case regret bound. $\diamond$

### 3.1 Standard Fixed Share Loss Bound

To demonstrate the graphical derivation of loss bounds for EHMMs we now prove a regret bound for $\mathsf{FS}$ using its representation as an EHMM. The general technique is to give lower bounds on the transition function and the initial distribution. For simplicity the bound we show is slightly weaker than the standard regret bound [Herbster and Warmuth, 1998, Corollary 1]. (One could get the exact same bound by taking into account the remark in footnote 3 of [Koolen and De Rooij, 2008a], but this unnecessarily complicates the proof.)

**Theorem 1.** *Fix a prior $w$ on experts $\mathcal{E}$ and a switching rate $\alpha$. Then for any data $x_{1:T}$, expert predictions*

$\mathrm{p}^{\mathcal{E}}_{1:T}$, *reference segmentation* $\mathbb{C}$ *and assignment of experts to segments* $\langle e_{\mathcal{C}} \in \mathcal{E}\rangle_{\mathcal{C}\in\mathbb{C}}$

$$\ell\big(\mathsf{FS}[\alpha,w],x_{1:T}\big) \;\le\; \underbrace{\sum_{\mathcal{C}\in\mathbb{C}} \ell(e_{\mathcal{C}},x_{\mathcal{C}})}_{\text{S-TBE ref. scheme}} + \underbrace{(T-1)\,\mathrm{H}(\alpha^*,\alpha)}_{\text{Switching}} + \underbrace{\sum_{\mathcal{C}\in\mathbb{C}} -\log w(e_{\mathcal{C}})}_{\text{Expert selection}},$$

*where* $\mathrm{H}(\alpha,\beta) = -\alpha\log\beta - (1-\alpha)\log(1-\beta)$ *and* $\alpha^* = \frac{|\mathbb{C}|-1}{T-1}$.

Note that if $w$ is the uniform distribution then $-\log w(e_{\mathcal{C}}) = \log|\mathcal{E}|$ for all $e_{\mathcal{C}}$. Then the difference with the standard bound in [Herbster and Warmuth, 1998] is $(|\mathbb{C}|-1)(\log|\mathcal{E}| - \log(|\mathcal{E}|-1))$, which is negligible.

*Proof.* Recall that $\mathsf{FS} \equiv \mathsf{FS}[\alpha,w]$ has transition function $\mathrm{p}_{\rightarrow} = (1-\alpha)\mathbf{I} + \alpha w \mathbf{1}^{\mathrm{T}}$. Therefore for any reference segmentation $\mathbb{C}$ the joint probability $\mathsf{FS}(x_{1:T})$ of any data sequence $x_{1:T}$ can be bounded from below by replacing transitions in $\mathsf{FS}$ *between* segments by $\alpha w \mathbf{1}^{\mathrm{T}}$, and those *within* the same segment by $(1-\alpha)\mathbf{I}$. The EHMM then degenerates into a sequence of independent Bayesian mixture EHMMs $\mathsf{B}[w]$ (see Example 3.2), one for each segment. Therefore

$$\mathsf{FS}(x_{1:T}) \ge \alpha^{|\mathbb{C}|-1}(1-\alpha)^{T-|\mathbb{C}|}\prod_{\mathcal{C}\in\mathbb{C}} \mathsf{B}[w](x_{\mathcal{C}}).$$

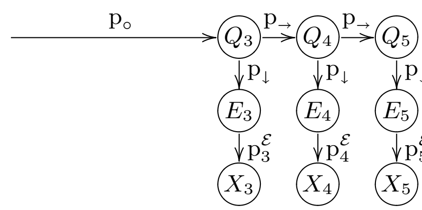
Similarly we can lower-bound the initial distribution of $\mathsf{B}[w]$ by a function that assigns weight $w(e_{\mathcal{C}})$ to the expert $e_{\mathcal{C}}$ selected for $\mathcal{C}$ in the reference segmentation and is 0 otherwise. It follows that $\mathsf{B}[w](x_{\mathcal{C}}) = \sum_e w(e)\mathrm{p}^e_{\mathcal{C}}(x_{\mathcal{C}}) \ge w(e_{\mathcal{C}})\mathrm{p}^{e_{\mathcal{C}}}_{\mathcal{C}}(x_{\mathcal{C}})$, where $\mathrm{p}^e_{\mathcal{C}}(x_{\mathcal{C}})$ denotes the joint probability of outcomes $x_{\mathcal{C}}$ according to the predictions of expert $e$. Hence by (1) we can conclude that

$$\begin{aligned}
&\ell\big(\mathsf{FS},x_{1:T}\big) = -\log \mathsf{FS}(x_{1:T})\\
&\le -\log\alpha^{|\mathbb{C}|-1}(1-\alpha)^{T-|\mathbb{C}|} + \sum_{\mathcal{C}\in\mathbb{C}} -\log \mathrm{p}^{e_{\mathcal{C}}}_{\mathcal{C}}(x_{\mathcal{C}}) - \log w(e_{\mathcal{C}})\\
&= (T-1)\,\mathrm{H}(\alpha^*,\alpha) + \sum_{\mathcal{C}\in\mathbb{C}} \ell(e_{\mathcal{C}},x_{\mathcal{C}}) + \sum_{\mathcal{C}\in\mathbb{C}} -\log w(e_{\mathcal{C}}),
\end{aligned}$$

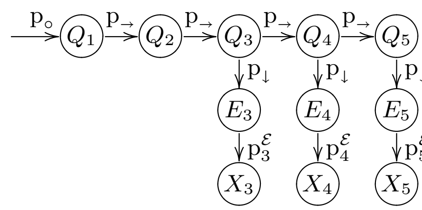
which completes the proof. □

## 4 Fixed Share for Learning Experts

In this section we define the freezing and sleeping LL-TBE reference schemes for learning experts. Then, for each scheme, we provide our prediction strategy $\mathsf{FS}^{\mathsf{fr}}$ and $\mathsf{FS}^{\mathsf{sl}}$ and we prove that it achieves as small regret as $\mathsf{FS}$.



(a) Freezing: EHMM $\mathfrak{H}^{\mathsf{fr}}_{3:5}$



(b) Sleeping: EHMM $\mathfrak{H}^{\mathsf{sl}}_{3:5}$

Figure 4: Freezing and Sleeping EHMM $\mathfrak{H}$ on Example Segment $x_{3:5}$

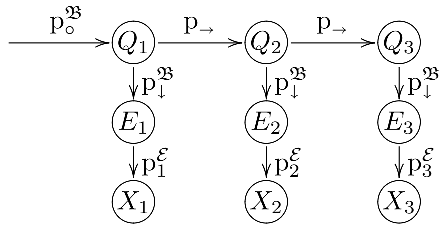
### 4.1 LL-TBE and the Loss of an EHMM on a Segment

In order to state the loss of the freezing and sleeping LL-TBE reference schemes, we first define the loss of a single learning expert on a single segment. Then we define the loss of a whole segmentation.

Let $\mathfrak{H}$ be the EHMM for a learning expert with arbitrary base experts $\mathcal{E}$. Then the freezing and sleeping probability distributions $\mathfrak{H}^{\mathsf{fr}}_{i:j}$ and $\mathfrak{H}^{\mathsf{sl}}_{i:j}$ on segment $x_{i:j}$ are specified by the Bayesian networks of Figure 4. For freezing, the state at time $i$ is simply initialised according to $\mathfrak{H}$'s initial distribution $\mathrm{p}_{\circ}$. For sleeping, we forward the initial distribution to time $i$ by repeatedly applying the transition function $\mathrm{p}_{\rightarrow}$. Thus, the cumulative freezing and sleeping losses of $\mathfrak{H}$ on segment $x_{i:j}$ are given by $\ell(\mathfrak{H}^{\mathsf{fr}}_{i:j}, x_{i:j}) := -\log\mathfrak{H}^{\mathsf{fr}}_{i:j}(x_{i:j})$ and $\ell(\mathfrak{H}^{\mathsf{sl}}_{i:j}, x_{i:j}) := -\log\mathfrak{H}^{\mathsf{sl}}_{i:j}(x_{i:j})$. Note that we treat the base experts $\mathcal{E}$ as black boxes, so they may learn from the whole data.

**Definition 1** (LL-TBE reference loss)**.** Fix data $x_{1:T}$ and a set of EHMMs $\mathcal{H}$. Let $\mathbb{C}$ be a segmentation of $1{:}T$ and let $\langle \mathfrak{H}_{\mathcal{C}} \in \mathcal{H}\rangle_{\mathcal{C}\in\mathbb{C}}$ be an assignment of experts to segments. Then the losses of the freezing and sleeping LL-TBE reference schemes are $\sum_{\mathcal{C}\in\mathbb{C}} \ell(\mathfrak{H}^{\mathsf{fr}}_{\mathcal{C}}, x_{\mathcal{C}})$ and $\sum_{\mathcal{C}\in\mathbb{C}} \ell(\mathfrak{H}^{\mathsf{sl}}_{\mathcal{C}}, x_{\mathcal{C}})$.

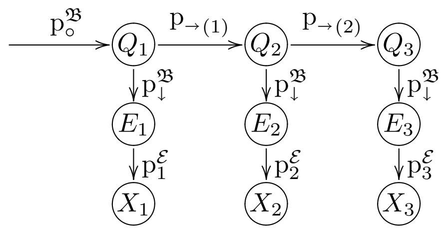
Note that selecting a learning expert on consecutive segments differs from selecting that expert on their union, since experts are reset between segments.



$$\mathrm{p}_{\rightarrow} \;\coloneqq\; (1-\alpha)\,\mathrm{p}^{\mathfrak{B}}_{\rightarrow} + \alpha\,\mathrm{p}^{\mathfrak{B}}_{\circ}\,\mathbf{1}^{\mathrm{T}}$$

Any switch reverts to $\mathrm{p}^{\mathfrak{B}}_{\circ}$, the initial distribution of $\mathfrak{B}$.

(a) EHMM $\mathsf{FS}^{\mathsf{fr}}[\alpha, \mathfrak{B}]$



$$\mathrm{p}_{\rightarrow(t)} \;\coloneqq\; (1-\alpha)\,\mathrm{p}^{\mathfrak{B}}_{\rightarrow} + \alpha(\mathrm{p}^{\mathfrak{B}}_{\rightarrow})^{t}\,\mathrm{p}^{\mathfrak{B}}_{\circ}\,\mathbf{1}^{\mathrm{T}}$$

The switch between time $t$ and $t+1$ reverts to $(\mathrm{p}^{\mathfrak{B}}_{\rightarrow})^{t}\,\mathrm{p}^{\mathfrak{B}}_{\circ}$, the $t^{\text{th}}$ evolution of the initial distribution of $\mathfrak{B}$.

(b) EHMM $\mathsf{FS}^{\mathsf{sl}}[\alpha, \mathfrak{B}]$

Figure 5: EHMMs for Tracking the EHMM $\mathfrak{B}$ with Switching Rate $\alpha$

### 4.2 Main Result: Construction of the Freezing and Sleeping EHMMs

We now present the construction of EHMMs for the freezing and sleeping algorithms $\mathsf{FS}^{\mathsf{fr}}$ and $\mathsf{FS}^{\mathsf{sl}}$. Let $\mathcal{H}$ be a set of learning experts, each expert $\mathfrak{H} \in \mathcal{H}$ presented as an EHMM on basic experts $\mathcal{E}$. Let $w$ be a prior on $\mathcal{H}$, and let $\alpha$ be a switching rate. We proceed in two steps. First construct the Bayesian EHMM $\mathfrak{B} = \mathsf{B}[w, \mathcal{H}]$ as in Example 3.3. Recall that $\mathfrak{B}$ learns which of the EHMMs in $\mathcal{H}$ predicts best. Second, construct the freezing EHMM $\mathsf{FS}^{\mathsf{fr}}[\alpha, \mathfrak{B}]$ or the sleeping EHMM[1] $\mathsf{FS}^{\mathsf{sl}}[\alpha, \mathfrak{B}]$ as shown in Figure 5. Note how, on a switch, both EHMMs reset the entire state of $\mathfrak{B}$, which includes the states of experts in $\mathcal{H}$. In contrast, $\mathsf{FS}$ only resets its weighting on $\mathcal{H}$, but does not touch the internal state of the experts in $\mathcal{H}$.

[1]Strictly speaking, the Bayesian network in Figure 5b is not an EHMM, since the transition function depends on the time. Nevertheless, this time-dependency can be removed without any computational overhead using a process called *unfolding*, see [Koolen and De Rooij, 2008b].

### 4.3 Prediction Algorithms

To sequentially predict data using our prediction strategies $\mathsf{FS}^{\mathsf{fr}}$ and $\mathsf{FS}^{\mathsf{sl}}$, one needs to run the forward algorithm on their respective EHMMs. An explicit rendering of this process is included in Algorithm 1.

Construct $\mathfrak{B} = \mathsf{B}[w, \mathcal{H}]$ with $\mathcal{Q}$, $\mathrm{p}_{\circ}$,$\mathrm{p}_{\downarrow}$ and $\mathrm{p}_{\rightarrow}$ as in Example 3.3.
Initialisation: $\lambda \leftarrow \mathrm{p}_{\circ}$
**for** $t = 1, \ldots$ **do** ▷ Invariant: $\lambda(q) = \mathsf{FS}^{\mathsf{v}}[\alpha, \mathfrak{B}](Q_t = q | x_{<t})$
  Receive expert advice $\mathrm{p}^{\mathcal{E}}_{t}$.
  Predict $X_t$ using
$$\lambda(X_t) = \sum_{e \in \mathcal{E}, q \in \mathcal{Q}} \lambda(q)\,\mathrm{p}^{q}_{\rightarrow}(e)\mathrm{p}^{e}_{t}(X_t).$$
  Observe $X_t = x_t$. Suffer loss $\ell\big(\lambda(X_t), x_t\big)$.
  Loss update: $\lambda(q) \leftarrow \lambda(q, x_t)/\lambda(x_t)$, where
$$\lambda(q, x_t) = \sum_{e \in \mathcal{E}} \lambda(q)\,\mathrm{p}^{q}_{\rightarrow}(e)\mathrm{p}^{e}_{t}(x_t).$$
  State evolution:
$$\lambda \leftarrow \begin{cases} (1-\alpha)\,\mathrm{p}_{\rightarrow}\,\lambda + \alpha\,\mathrm{p}_{\circ} & \text{(Freezing)} \\ (1-\alpha)\,\mathrm{p}_{\rightarrow}\,\lambda + \alpha(\mathrm{p}_{\rightarrow})^{t}\,\mathrm{p}_{\circ} & \text{(Sleeping)} \end{cases}$$
**end for**

Algorithm 1: Explicit Forward Algorithm on $\mathsf{FS}^{\mathsf{v}}$ for both Freezing and Sleeping ($\mathsf{v} \in \{\mathsf{fr}, \mathsf{sl}\}$)

At any time $t$, the algorithm for $\mathsf{FS}^{\mathsf{sl}}$ only maintains non-zero weights on hidden states of the input learning experts that are reachable in *exactly* $t$ steps from the starting states, just like the original $\mathsf{FS}$ algorithm. It therefore has the same running time. The algorithm for $\mathsf{FS}^{\mathsf{fr}}$, however, has to keep track of all states reachable in *at most* $t$ steps. Consequently, in the worst case (over input EHMMs) it may be as slow as restarting expert copies (see §1.4). But if the input EHMMs have a finite number of hidden states, then its running time is of the same order as that of $\mathsf{FS}$. And if the states (of the input EHMMs) that are reachable in exactly $t$ steps are the same ones as the states reachable in at most $t$ steps, which holds e.g. for the drifting-mean expert $\mathsf{DM}[\theta]$ from the introduction, then we also recover the efficiency of $\mathsf{FS}$.

### 4.4 Loss Bound

Theorem 1 bounds the regret of $\mathsf{FS}$ compared to the S-TBE reference scheme by a "switching" and an "expert selection" term. We bound the regret of $\mathsf{FS}^{\mathsf{fr}}$ and $\mathsf{FS}^{\mathsf{sl}}$ compared to their LL-TBE reference scheme by the same two terms.

**Theorem 2.** *Fix a set of EHMMs $\mathcal{H}$ on basic experts $\mathcal{E}$, a prior $w$ on $\mathcal{H}$, a switching rate $\alpha$ and $\mathsf{v} \in \{\mathsf{fr}, \mathsf{sl}\}$. Let $\mathfrak{B} = \mathsf{B}[w, \mathcal{H}]$. Then for any data $x_{1:T}$, expert predictions $\mathrm{p}^{\mathcal{E}}_{1:T}$, reference segmentation $\mathbb{C}$ and assign-*

*ment of experts to segments* $\langle \mathfrak{H}_\mathcal{C} \in \mathcal{H} \rangle_{\mathcal{C} \in \mathbb{C}}$

$$\ell\left(\mathsf{FS}^{\mathsf{v}}[\alpha, \mathfrak{B}], x_{1:T}\right) \leq \underbrace{\sum_{\mathcal{C} \in \mathbb{C}} \ell(\mathfrak{H}^{\mathsf{v}}_\mathcal{C}, x_\mathcal{C})}_{\text{LL-TBE ref. scheme}} + \underbrace{(T-1)\,\mathrm{H}(\alpha^*, \alpha)}_{\text{Switching}} + \underbrace{\sum_{\mathcal{C} \in \mathbb{C}} -\log w(\mathfrak{H}_\mathcal{C})}_{\text{Expert selection}},$$

*where* $\mathrm{H}(\alpha^*, \alpha)$ *and* $\alpha^* = \frac{|\mathbb{C}|-1}{T-1}$ *are as in Theorem 1.*

*Proof.* The proof proceeds like that of Theorem 1. Lower-bounding transitions between segments by $\alpha\, \mathrm{p}_\circ^{\mathfrak{B}}\, \mathbf{1}^\mathrm{T}$ (freezing) or $\alpha (\mathrm{p}_\rightarrow^{\mathfrak{B}})^t\, \mathrm{p}_\circ^{\mathfrak{B}}\, \mathbf{1}^\mathrm{T}$ (sleeping), and transitions within each segment by $(1-\alpha)\,\mathrm{p}_\rightarrow^{\mathfrak{B}}$, we get

$$\mathsf{FS}^{\mathsf{v}}[\alpha, \mathfrak{B}] \geq \alpha^{|\mathbb{C}|-1}(1-\alpha)^{T-|\mathbb{C}|} \prod_{\mathcal{C} \in \mathbb{C}} \mathfrak{B}^{\mathsf{v}}_\mathcal{C}(x_\mathcal{C}), \qquad (2)$$

where $\mathfrak{B}^{\mathsf{v}}_\mathcal{C}$ denotes the result of freezing or sleeping $\mathfrak{B}$ on segment $\mathcal{C} \in \mathbb{C}$ as in Figure 4. Observe that freezing and sleeping distribute over taking the Bayesian mixture: $\mathfrak{B}^{\mathsf{v}}_\mathcal{C} = \mathsf{B}[w, \mathcal{H}^{\mathsf{v}}_\mathcal{C}]$, where $\mathcal{H}^{\mathsf{v}}_\mathcal{C} \coloneqq \{\mathfrak{H}^{\mathsf{v}}_\mathcal{C} \mid \mathfrak{H} \in \mathcal{H}\}$. As $\mathsf{B}[w, \mathcal{H}^{\mathsf{v}}_\mathcal{C}](x_\mathcal{C}) = \sum_{\mathfrak{H}} w(\mathfrak{H}) \mathfrak{H}^{\mathsf{v}}_\mathcal{C}(x_\mathcal{C}) \geq w(\mathfrak{H}_\mathcal{C}) \mathfrak{H}^{\mathsf{v}}_\mathcal{C}(x_\mathcal{C})$, the theorem follows from (1), like in the proof of Theorem 1. □

## 5 Conclusion

We revisited the tracking the best expert reference scheme (TBE), which asks for a strategy for prediction with expert advice that suffers small additional loss compared to the best expert per segment. This goal is natural when the characteristics of the data, and hence the best expert, are different between segments.

For learning experts, the standard interpretation of experts as black boxes implies training the experts on all data. We proposed a variation, adapted to learning experts, in which experts are only trained on the segment on which they are evaluated. Our scheme is able to exploit patterns in the data *per segment*, leading to smaller loss.

Although in general extending the standard fixed-share algorithm to our setting will slow it down by a factor of $T$ on $T$ outcomes, we showed that no such slowdown is necessary if the learning experts can be represented as expert hidden Markov models (EHMMs). We proved the loss bounds one would expect based on the loss bound for the original fixed-share algorithm.

### 5.1 Discussion and Future Work

**Learning the Switching Rate** Like fixed share, our algorithms depend on a switching rate parameter $\alpha$, which has to be fixed. Instead, one may want to tune $\alpha$ automatically based on the data. For $\mathsf{FS}$ this can be done efficiently (see [De Rooij and Van Erven, 2009] and references therein). The same methods transfer directly to $\mathsf{FS}^{\mathsf{fr}}$ and $\mathsf{FS}^{\mathsf{sl}}$.

**S-TBE vs LL-TBE** We have discussed experts that learn only on their assigned segment. Perhaps surprisingly, this does *not always* increase performance. For example, we may have homogeneous data and experts that learn its global pattern at different rates. In such cases we clearly want to train each expert on all observations and, by switching at the right times, select the expert that has learned most until then. This scenario is analysed by Van Erven et al. [2008], where experts are parameter estimators for a series of statistical models of increasing complexity.

**Partitions instead of Segmentations** Rather than split the data into segments as in the TBE reference scheme, one may wish to partition it arbitrarily into cells such that observations in the same cell need not be consecutive. Like fixed share, the corresponding algorithm [Bousquet and Warmuth, 2002] can be generalised to the LL-TBE setting without increasing its running time. In this case naively introducing copies of the experts for all possible partitions is infeasible: it would slow down the algorithm by an exponential factor $2^T$ on $T$ outcomes.

## References

O. Bousquet and M. K. Warmuth. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research*, 3:363–396, 2002.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

S. de Rooij and T. van Erven. Learning the switching rate by discretising Bernoulli sources online. In *Proc. of the 12th International Conference on AI and Statistics*, volume 5 of *JMLR: W&CP*, pages 432–439, 2009.

J. Hannan. Approximation to Bayes risk in repeated play. In *Contributions to the Theory of Games*, volume 3, pages 97–139. Princeton University Press, 1957.

M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32:151–178, 1998.

W. M. Koolen and S. de Rooij. Combining expert advice efficiently. In *Proceeding of the 21st Annual Conference on Learning Theory*, pages 275–286, 2008a.

W. M. Koolen and S. de Rooij. Combining expert advice efficiently. arXiv:0802.2015, Feb. 2008b.

C. Monteleoni and T. Jaakkola. Online learning of non-stationary sequences. *Advances in Neural Information Processing Systems*, 16:1093–1100, 2003.

T. van Erven, P. D. Grünwald, and S. de Rooij. Catching up faster in Bayesian model selection and model averaging. In *Advances in Neural Information Processing Systems 20*, 2008.

V. Vovk. Derandomizing stochastic prediction strategies. *Machine Learning*, 35:247–282, 1999.