Learning the Switching Rate by Discretising Bernoulli Sources Online

Steven de Rooij
steven@statslab.cam.ac.uk
Centre for Mathematical Sciences
Wilberforce Road
Cambridge CB3 0WB, United Kingdom

Tim van Erven
Tim.van.Erven@cwi.nl
Centrum Wiskunde & Informatica (CWI)
Science Park 123, P.O. Box 94079
1090 GB Amsterdam, The Netherlands

Abstract

The expert tracking algorithm `Fixed-Share` depends on a parameter α, called the *switching rate*. The switching rate can be learned online with regret $\frac{1}{2}\log T + O(1)$ bits. The current fastest method to achieve this is based on optimal discretisation of the Bernoulli distributions into $O(\sqrt{T})$ bins and runs in $O(T\sqrt{T})$ time. However, the exact locations of these bins have to be determined algorithmically, and the final number of outcomes T must be known in advance.

This paper introduces a new discretisation scheme with the same regret bound for known T, that specifies the number and positions of the discretisation points explicitly. The scheme is especially useful, however, when T is not known in advance: a new fully online algorithm is presented, which runs in $O(T\sqrt{T}\log T)$ time and achieves a regret of $\frac{1}{2}\log 3 \log T + O(\log\log T)$ bits.

1 Introduction

We will attempt to sequentially predict the outcomes X_1, X_2, ... from an unknown process, where each outcome takes values in a countable set $\mathcal{X}$. At each time $t \in \mathbb{Z}^+ = \{1, 2, \ldots\}$ we have to issue a probability distribution $P(X_t \mid x^{t-1})$ on $\mathcal{X}$, which is allowed to depend on past observations $x^{t-1} = x_1, \ldots, x_{t-1}$. Then x_t is revealed and we suffer *logarithmic loss* $-\ln P(X_t = x_t \mid x^{t-1})$. (For simplicity we consider only logarithmic loss, but results for other loss functions can be obtained using methods described in e.g. [Vovk, 1999].) Suppose our understanding of the process is very limited, but luckily we do have access to n *experts*. Each expert $\xi \in \Xi = \{1, \ldots, n\}$ provides us with her prediction $P_\xi(X_t \mid x^{t-1})$, on which we may base our own forecast $P(X_t \mid x^{t-1})$. We make no assumptions about the nature of the experts, so one may think of human experts, but also of computer algorithms. This is the problem of *prediction with expert advice* (for log loss) [Cesa-Bianchi and Lugosi, 2006].

For any T, one may view the predictions $P(X_t \mid X^{t-1})$ as conditionals of the joint distribution $P(X^T) = \prod_{t=1}^T P(X_t \mid X^{t-1})$. (We regard the empty sequence x^0 as a certain event, which occurs with probability one.) In its most basic setup the goal of prediction with expert advice is to minimise the excess loss compared to the best expert on any sequence of outcomes x^T: $-\ln P(x^T) - \min_\xi[-\ln P_\xi(x^T)]$. This is called the *regret* on x^T. A more ambitious goal is to compare to the performance that can be obtained by optimally dividing the data into m segments and, within each segment, using the best expert for that segment. This is prudent in case the experts themselves may improve (study hard) or deteriorate (take to drinking), but also when their performance depends on the predictive context (some experts may be good during spring, others during winter). In this case, if the optimal segments start at times $t_1, \ldots, t_m$ for a given sequence x^T, the goal is to minimise

$$-\ln P(x^T) - \sum_{i=1}^{m} \min_{\xi} -\ln P_\xi(x_{t_i}^{t_{i+1}-1} \mid x^{t_i-1}), \quad (1)$$

where $x_a^b = x_a, \ldots, x_b$, and $t_{m+1} = T + 1$. This is the approach taken by Herbster and Warmuth [1998]; see also [Vovk, 1999, Cesa-Bianchi and Lugosi, 2006]. Let $H(p) = -p\ln p - (1-p)\ln(1-p)$ and $D(p\|q) = p\ln p/q + (1-p)\ln(1-p)/(1-q)$ denote binary entropy and Kullback-Leibler divergence, respectively; we use ln to denote the natural logarithm and log for base two. The regret of Herbster and Warmuth's `Fixed-Share` algorithm is bounded from above by $(T-1)(H(\alpha^*) + D(\alpha^*\|\alpha)) + (m-1)\ln(n-1) + \ln n$

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5.

nats, where α is the *switching rate*, a parameter of the algorithm that can be interpreted as the probability of switching between experts; the best regret bound is obtained when α equals $\alpha^* := (m-1)/(T-1)$.

One clear advantage of `Fixed-Share` is its computational efficiency: its running time, which is $n \cdot O(T)$, is as low as that of the standard Bayesian mixture. The one real disadvantage is having to specify the switching rate. It is this problem that we address in this paper. Our contribution should be placed in the context of three earlier approaches to avoid a priori specification of the switching rate:

Decreasing Switching Rate One option is to let the switching rate *decrease with time* as $1/t$ [Van Erven et al., 2008, Koolen and de Rooij, 2008]. For this approach, the regret compared to the best segmentation in m parts is within $\ln T + O(m \log m)$ nats from the bound for `Fixed-Share` with optimally tuned α. This is fine if the number of switches in the sequence is not too large (say, $m = O(\log T)$), but if switches can occur more frequently, it may not be the best choice.

Bayes with Undiscretised Switching Rate A second option is to use a Bayesian mixture over α. Such an algorithm was described very early in the source coding literature [Volf and Willems, 1998]. This algorithm, called the *Switching Method* (not to be confused with the Switch Distribution!), achieves a regret bounded by $\frac{1}{2} \ln T + O(1)$ nats compared to the best `Fixed-Share` parameter. Note that this bound does not depend on the number of switches. The drawback of this approach is that its running time is $n \cdot O(T^2)$, which is significantly slower than the previous algorithms and may be prohibitive in some applications.

Bayes with Discretised Switching Rate A third approach to get rid of α also uses a Bayesian mixture, but rather than putting a prior on the whole range $[0, 1]$ of possible values of α, a prior is defined on a *discretised* set of parameters $\alpha_1, \alpha_2, \ldots, \alpha_j$. Monteleoni and Jaakkola [2003] argue that $O(\sqrt{T})$ levels of discretisation suffice to achieve a regret with respect to `Fixed-Share` of at most $\frac{1}{2} \ln T + O(1)$ nats, like the Switching Method. Their algorithm `Learn-`α has running time $n \cdot O(T\sqrt{T})$ however, a significant improvement over the Switching Method. However, while `Learn-`α does not require a priori knowledge of α, unlike the other approaches it does require a priori knowledge of the final number of outcomes T. The algorithm is therefore almost, but not completely, online. In Section 3.3 we discuss why the so-called doubling trick is not the best way to eliminate this dependence.

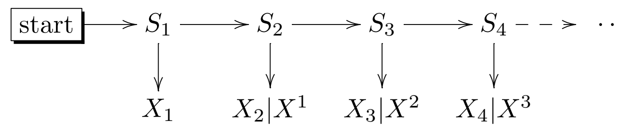

Figure 1: Bayesian network for an expert algorithm

Refine-Online In this paper we take the `Learn-`α algorithm as a starting point to develop a fourth, fully online algorithm called `Refine-Online`. It has running time $n \cdot O(T\sqrt{T} \log T)$, which makes it only slightly slower than `Learn-`α. Its regret is bounded by $\frac{1}{2} \log 3 \ln T + \log 3 \ln \ln(T+1) + O(1)$, which is worse than the bounds in the two Bayesian approaches, but would still seem an acceptable price to pay to get a fast algorithm that is completely online.

Outline In Section 2 we show how probabilistic algorithms for prediction with expert advice can be described using Hidden Markov models (HMMs), and we give basic tools to prove loss bounds for such algorithms. We then state our main results. Section 3 exhibits a new, very simple discretisation scheme that grants full control over the exact number and placement of discretisation points, in contrast to the discretisation used by `Learn-`α, which can only be determined algorithmically. Moreover, we show how this discretisation can be refined online, so that the final number of outcomes T does not have to be known.

2 Expert Algorithms as HMMs

Many algorithms for prediction with expert advice can be described as a hidden Markov model (HMM) $\mathbb{P}$, where the *hidden state* S_t at any time t identifies an expert ξ_t to predict outcome X_t [Koolen and de Rooij, 2008]. Figure 1 depicts the corresponding Bayesian network, where we write $X_t|X^{t-1}$ to indicate that the expert may base her prediction of X_t on all previous outcomes X^{t-1}. Each S_t takes values in a set of *hidden states* $\mathcal{S} = \{\langle \xi, t, \ldots \rangle \mid \xi \in \Xi, t \in \mathbb{Z}^+\}$, where t denotes a time index and states with the wrong time index get probability zero: $\mathbb{P}(S_t = \langle \xi, t', \ldots \rangle) = 0$ if $t' \neq t$. Depending on the specifics of the algorithm the hidden states can contain more information, represented here by dots. Given a state $\langle \xi, t, \ldots \rangle \in \mathcal{S}$ and previous outcomes x^{t-1} the probability of X_t is determined by the prediction of expert ξ:

$$\mathbb{P}(X_t \mid \langle \xi, t, \ldots \rangle, x^{t-1}) = P_\xi(X_t \mid x^{t-1}).$$

The advantage of casting these algorithms as HMMs is that the standard algorithms for HMMs can be applied. Specifically, the *forward algorithm* can compute the predictions $\mathbb{P}(X_1)$, …, $\mathbb{P}(X_T \mid x^{T-1})$ in time proportional to the number of transitions in the HMM [Koolen and de Rooij, 2008].

Bayes We first consider the standard Bayesian prediction strategy that puts a prior w on experts Ξ. This corresponds to the HMM $\mathbb{H}$ with hidden states $\{\langle \xi, t\rangle \mid \xi \in \Xi, t \in \mathbb{Z}^+\}$. Initially all experts get probability according to the prior, $\mathbb{H}(\langle \xi_1, 1\rangle) = w(\xi_1)$, but afterwards no more switches between experts are allowed: $\mathbb{H}(\langle \xi_{t+1}, t+1\rangle \mid \langle \xi_t, t\rangle)$ is 1 if $\xi_{t+1} = \xi_t$, and 0 otherwise.

Fixed-Share There is also an HMM $\mathbb{F}_\alpha$ that corresponds to the `Fixed-Share` algorithm [Koolen and de Rooij, 2008]. As in [Herbster and Warmuth, 1998], all experts are initially given equal weight, $\mathbb{F}_\alpha(\langle \xi_1, 1\rangle) = 1/n$, which gives the best worst-case bound. After each outcome, $\mathbb{F}_\alpha$ allows *switches* between experts to occur with probability $\alpha \in [0,1]$, which is called the *switching rate*:

$$\mathbb{F}_\alpha(\langle \xi_{t+1}, t+1\rangle \mid \langle \xi_t, t\rangle) = \begin{cases} 1-\alpha & \text{if } \xi_{t+1} = \xi_t, \\ \alpha/(n-1) & \text{otherwise.} \end{cases}$$

Note that $\mathbb{F}_0 = \mathbb{H}$ (using a uniform prior w). Naive application of the forward algorithm to $\mathbb{F}_\alpha$ gives $O(n^2)$ transitions per time step, adding up to a total running time of $n^2 \cdot O(T)$. This is reduced to $O(n)$ transitions by introducing an intermediate *pool* state that first collects all probability mass for switches between experts and then redistributes it (see [Koolen and de Rooij, 2008] for details). The running time then becomes $n \cdot O(T)$ as in [Herbster and Warmuth, 1998].

2.1 Tracking HMMs and Bernoulli HMMs

The `Fixed-Share` algorithm has a fixed switching rate α. This may be generalised to a *tracking HMM* $\mathbb{S}$ with hidden states $\{\langle \xi, t, \alpha\rangle \mid \xi \in \Xi, t \in \mathbb{Z}^+, \alpha \in \mathcal{A}_t\}$. The initial states have weights given by $\mathbb{S}(\langle \xi_1, 1, \alpha_1\rangle) = \mathbb{B}(\langle \alpha_1, 1\rangle) \cdot \frac{1}{n}$, and the transition probabilities are

$$\begin{aligned} &\mathbb{S}(\langle \xi_{t+1}, t+1, \alpha_{t+1}\rangle \mid \langle \xi_t, t, \alpha_t\rangle) \\ &= \mathbb{B}(\langle \alpha_{t+1}, t+1\rangle \mid \langle \alpha_t, t\rangle) \cdot \mathbb{F}_{\alpha_t}(\langle \xi_{t+1}, t+1\rangle \mid \langle \xi_t, t\rangle), \end{aligned}$$

where $\mathbb{B}$, called a *Bernoulli HMM*, describes the evolution of α. The original `Fixed-Share` method $\mathbb{F}_\alpha$ can be recovered by using $\mathcal{A}_t = \{\alpha\}$ and $\mathbb{B} = \mathbb{B}^\alpha_{\text{fixed}}$, where

$$\mathbb{B}^\alpha_{\text{fixed}}(\langle \alpha_{t+1}, t+1\rangle \mid \langle \alpha_t, t\rangle) = \mathbb{B}^\alpha_{\text{fixed}}(\langle \alpha_{t+1}, 1\rangle) = 1.$$

We consider various other options for the Bernoulli HMM $\mathbb{B}$ as well. In general let $\mathbb{S}^b_a$ denote the tracking HMM $\mathbb{S}$ defined with respect to the Bernoulli HMM $\mathbb{B}^b_a$. Thus $\mathbb{S}^\alpha_{\text{fixed}} = \mathbb{F}_\alpha$.

It is essential now to distinguish between two levels: `Fixed-Share` and the tracking HMM $\mathbb{S}$, which aim to predict outcomes X_1, X_2, …, operate on the upper level. On the lower level there is the Bernoulli HMM $\mathbb{B}$. Although $\mathbb{B}$ is used as a building block in the construction of $\mathbb{S}$, it is convenient to also interpret $\mathbb{B}$ as an algorithm for prediction with expert advice in itself. In this view, let Y_1, Y_2, … be binary outcomes, which $\mathbb{B}$ has to predict, and let P_α denote the Bernoulli distribution with $P_\alpha(Y=1) = \alpha$, extended to sequences by taking product distributions. In a Bernoulli HMM the experts are instantiated to such Bernoulli sources, and are indexed by $\alpha \in \mathcal{A}_t$. Thus $\mathbb{B}$ has hidden states $\{\langle \alpha, t\rangle \mid \alpha \in \mathcal{A}_t, t \in \mathbb{Z}^+\}$ and $\mathbb{B}(Y_t \mid \langle \alpha, t\rangle) = P_\alpha(Y_t)$.

The total running time of the forward algorithm applied to a tracking HMM may be computed by summing up the number of transitions for each time step. This is the number of transitions of `Fixed-Share`, which is $O(n)$, times the number of transitions of the corresponding Bernoulli HMM. Thus the forward algorithm for a tracking HMM runs in $O(n)$ times the running time of the forward algorithm for its Bernoulli HMM.

All approaches to learning the switching rate that were discussed in the introduction, including the new `Refine-Online` method, can be implemented using tracking HMMs with different choices for the Bernoulli HMM $\mathbb{B}$. We will illustrate this for `Learn-`α. In Section 3.3 we do the same for $\mathbb{B}_{\text{ro}}$, which defines the `Refine-Online` algorithm. From the description of the Switching Method in [Koolen and de Rooij, 2008] it is not hard to see how it can be cast as a Bernoulli HMM as well, but for brevity we do not discuss the details here.

Example: `Learn-`α Given the final number of outcomes, T, the algorithm `Learn-`α [Monteleoni and Jaakkola, 2003] applies Bayes at a meta-level to learn the switching rate α of the `Fixed-Share` algorithm: it puts a uniform prior (which gives the best worst-case bound) on a discretised set $\mathcal{A}_T$ of switching rates, where the discretisation depends on T. It turns out that this approach corresponds exactly to a tracking HMM $\mathbb{S}_{\text{Bayes}}$. The corresponding Bernoulli HMM $\mathbb{B}_{\text{Bayes}}$ has $\mathcal{A}_t = \mathcal{A}_T$ for all t, initial weights $\mathbb{B}_{\text{Bayes}}(\langle \alpha_1, 1\rangle) = 1/|\mathcal{A}_T|$ and transition probabilities

$$\mathbb{B}_{\text{Bayes}}(\langle \alpha_{t+1}, t+1\rangle \mid \langle \alpha_t, t\rangle) = \mathbf{1}_{\{\alpha_t\}}(\alpha_{t+1}), \qquad (2)$$

where $\mathbf{1}_A(z)$ denotes the *indicator function*, which is 1 if $z \in A$ and 0 otherwise. Note that $\mathbb{B}_{\text{Bayes}}$ is exactly the Bayesian HMM $\mathbb{H}$ with a uniform prior on $\mathcal{A}_T$, where the experts Ξ have been identified with Bernoulli parameters $\mathcal{A}_T$. In Section 3 we will choose $\mathcal{A}_T$ differently from [Monteleoni and Jaakkola, 2003] based on our new discretisation scheme.

2.2 Regret Bounds

The following lemma will be our main tool to show regret bounds. It bounds the likelihood ratio between any two tracking HMMs in terms of the worst-case likelihood ratio of their corresponding Bernoulli HMMs.

In other words, the lemma allows us to *lift* any uniform performance guarantees we may prove for Bernoulli HMMs to the level of tracking HMMs.

Lemma 1 (Lifting Lemma for Tracking)**.** *Suppose $\mathbb{B}_a$ and $\mathbb{B}_b$ are Bernoulli HMMs, and $\mathbb{B}_b(y^{T-1}) > 0$ for all binary sequences y^{T-1}. Then for any x^T*

$$\mathbb{S}_a(x^T) \leq \mathbb{S}_b(x^T) \max_{y^{T-1}} \frac{\mathbb{B}_a(y^{T-1})}{\mathbb{B}_b(y^{T-1})}.$$

By invoking this lemma with $\mathbb{S}_a = \mathbb{F}_{\hat{\alpha}(x^T)}$, where $\hat{\alpha}(x^T)$ is the best possible switching rate, we can obtain a bound on the regret for any tracking HMM with respect to the `Fixed-Share` algorithm with optimally tuned parameter. This is the idea behind our main results, which appear as Theorem 1 below. The proof of the lemma uses the following more general lemma.

Lemma 2. *Let P and Q be distributions on countable space $\mathcal{Z} \times \Psi$ such that for all outcomes $\langle z, \psi \rangle$ we have $P(z \mid \psi) = Q(z \mid \psi)$ and $Q(\psi) > 0$. Then, for $z \in \mathcal{Z}$,*

$$P(z) \leq Q(z) \cdot \max_{\psi \in \Psi} \frac{P(\psi)}{Q(\psi)}.$$

Proof. $P(z) = \sum_\psi P(\psi) P(z \mid \psi) \leq \max_\psi \frac{P(\psi)}{Q(\psi)} \sum_\psi Q(\psi) P(z \mid \psi) = Q(z) \max_\psi \frac{P(\psi)}{Q(\psi)}$. ☐

Proof of Lemma 1. Let $Y_t = 1 - \mathbf{1}_{\{\xi_t\}}(\xi_{t+1})$ for $t = 1, \ldots, T$ indicate whether or not a switch occurs. Now let $\mathcal{Z} = \mathcal{X}^T$ and $\Psi = \{0,1\}^{T-1}$, and notice that for any $\langle x^T, y^{T-1} \rangle \in \mathcal{Z} \times \Psi$ we have

$$\mathbb{S}_a(x^T, y^{T-1}) = \mathbb{F}(x^T \mid y^{T-1}) \mathbb{B}_a(y^{T-1})$$
$$\mathbb{S}_b(x^T, y^{T-1}) = \mathbb{F}(x^T \mid y^{T-1}) \mathbb{B}_b(y^{T-1}),$$

where $\mathbb{F}(x^T \mid y^{T-1}) \equiv \mathbb{F}_\alpha(x^T \mid y^{T-1})$ denotes a conditional probability in the `Fixed-Share` HMM that does not depend on α. Lemma 2 completes the proof. ☐

The lifting lemma is tight in the following sense. Consider two experts, whose predictions for all x^{t-1} are simply $P_1(X_t = 1 \mid x^{t-1}) = 1$ and $P_2(X_t = 0 \mid x^{t-1}) = 1$, respectively. Then any tracking HMM $\mathbb{S}$ with corresponding Bernoulli HMM $\mathbb{B}$ has $\mathbb{S}(x^T) = \mathbb{B}(y^{T-1})$, where $y_t = 1 - \mathbf{1}_{\{x_t\}}(x_{t+1})$ identifies whether the t-th and $(t+1)$-th outcomes are the same or not. Hence the regret is maximised for x^T such that the corresponding y^{T-1} maximises $\mathbb{B}_a(y^{T-1}) / \mathbb{B}_b(y^{T-1})$.

Section 3 introduces two new Bernoulli HMMs. We already mentioned the first one, $\mathbb{B}_{\text{Bayes}}$, in the example above. In Section 3.2 we provide a uniform bound on its regret compared to any Bernoulli distribution. Then in Section 3.3 we define $\mathbb{B}_{\text{ro}}$, which does not require T to be known in advance, and extend the results from Section 3.2 to bound the regret of $\mathbb{B}_{\text{ro}}$. The `Refine-Online` algorithm is defined using this second Bernoulli HMM. Combining these results with Lemma 1 and the observation that $\mathbb{B}^\alpha_{\text{fixed}}(y^T) = P_\alpha(y^T)$ for all y^T, we directly obtain the main results of this paper:

Theorem 1 (Learning the Switching Rate)**.** *Let $\mathbb{B}_{\text{Bayes}}$ be as in Definition 2 below. Then for any $\alpha \in [0,1]$ and any data x^T such that $T > 1$, the regret of $\mathbb{S}_{\text{Bayes}}$ compared to $\mathbb{F}_\alpha$ is bounded by*

$$\ln \frac{\mathbb{F}_\alpha(x^T)}{\mathbb{S}_{\text{Bayes}}(x^T)} \leq \tfrac{1}{2} \ln(T-1) + 2.8,$$

and the regret of $\mathbb{S}_{\text{ro}}$ is bounded by

$$\ln \frac{\mathbb{F}_\alpha(x^T)}{\mathbb{S}_{\text{ro}}(x^T)} \leq \log 3 \left(\tfrac{1}{2} \ln(T-1) + \ln \ln(T) \right) + 23.1.$$

(For $T = 1$, $\mathbb{S}_{\text{Bayes}}(x) = \mathbb{S}_{\text{ro}}(x) = \mathbb{F}_\alpha(x)$ for any x.)

While this theorem yields a bound for $\mathbb{S}_{\text{Bayes}}$ comparable to that given in [Monteleoni and Jaakkola, 2003], the analysis is different: in the end it is based on Lemma 1, which can only be usefully applied when good *uniform* bounds on the prior probability of the expert sequence, as established in Section 3.2, are available. In contrast, the analysis in [Monteleoni and Jaakkola, 2003] only requires a good bound on the Kullback-Leibler divergence $D(\hat{\alpha} \| \ddot{\alpha})$ between the *optimal* switching rate $\hat{\alpha}$ and the best discretised parameter $\ddot{\alpha} \in \mathcal{A}_T$. In other words, the only region where the discretisation precision actually matters is close to $\hat{\alpha}$. But their analysis does not readily generalise to other Bernoulli HMMs such as $\mathbb{B}_{\text{ro}}$.

3 Discretisation of Bernoulli Sources

In this section we define two Bernoulli HMMs, $\mathbb{B}_{\text{Bayes}}$ and $\mathbb{B}_{\text{ro}}$, and derive bounds on their worst-case regret. The first is based on a fixed discretisation of the set of Bernoulli distributions, where the optimal number of discretisation levels depends on the total number of outcomes T, which therefore has to be known. The resulting tracking HMM, $\mathbb{S}_{\text{Bayes}}$, is similar to `Learn-`$\alpha$, but with the added advantage that the exact number and locations of the discretisation points are explicitly speficied. Moreover, we obtain an explicit constant.

The analysis of $\mathbb{B}_{\text{Bayes}}$ is also an essential stepping stone to the specification of the second Bernoulli HMM $\mathbb{B}_{\text{ro}}$, whose discretisation of the set of Bernoulli distributions is not fixed; instead the discretisation is *refined* every time the number of outcomes gets large enough that it pays to do so.

Preliminaries As before, let P_α denote the Bernoulli distribution with $P_\alpha(Y = 1) = \alpha$. For any binary sequence y^T, the maximum likelihood parameter is $\hat{\alpha}(y^T) = T^{-1} \sum_{t=1}^T y_t$. When the data sequence is clear from context, we usually abbreviate $\hat{\alpha} \equiv \hat{\alpha}(y^T)$. The maximum likelihood is a sufficient statistic: for any α and T, the probability $P_\alpha(y^T)$ is completely determined by $\hat{\alpha}$. We therefore define $P_\alpha(\hat{\alpha}) := \alpha^{\hat{\alpha}}(1-\alpha)^{1-\hat{\alpha}}$, allowing any $\hat{\alpha} \in [0,1]$, not just rational values. Note that $T \ln P_\alpha(\hat{\alpha}) = \ln P_\alpha(y^T)$.

3.1 Discretisation

The analysis below is based on a different parametrisation of the Bernoulli distributions. For $\alpha \in [0,1]$ and $\phi \in [0,\pi/2]$, let $\phi(\alpha) = \arcsin\sqrt{\alpha}$ and $\alpha(\phi) = \sin^2\phi$. It is convenient to think of ϕ-parameters as points in the first quadrant of the unit circle. The parametrisation has many elegant properties; for example the Fisher information is constant. Similar *arcsine transformations* are well-known in the statistical literature [Anscombe, 1948, Freeman and Tukey, 1950]. In the following we will use $P_\alpha(\hat{\alpha})$ and $P_\phi(\hat{\phi})$ interchangeably, where the intended parametrisation should be clear from the parameter name and the context.

We now describe an explicit discretisation scheme for the ϕ-parameter of Bernoulli distributions that is especially easy to refine incrementally in online settings.

Definition 1 (k-Discretisation)**.** For $k \in \{1,2,\ldots\}$ define the *k-discretisation* as the set $\mathcal{D}_k := \{\delta_k, 2\delta_k, 3\delta_k, \ldots, (2^k-1)\delta_k\} \cup \{\frac{1}{2}\delta_k, \pi/2 - \frac{1}{2}\delta_k\}$ of 2^k+1 discretisation points, where $\delta_k = \pi 2^{-k-1}$.

This is a uniform discretisation made slightly denser at the boundaries. The $(k+1)$-discretisation adds a new point midway between any two points in the k-discretisation, except at the boundaries, which require special care. Thus $\mathcal{D}_k \subset \mathcal{D}_{k+1}$, which will turn out to facilitate incremental refinement in the online setting.

Given k-discretisation $\mathcal{D}_k$, any point $\psi \in [0,\pi/2]$ has a set $N_k(\psi)$ of *neighbours* in $\mathcal{D}_k$, which is defined as

$$N_k(\psi) = \begin{cases} \{\phi_1\} & \text{if } \psi > \pi/2 - \delta_k/2, \\ \{\phi_2\} & \text{if } \psi < \delta_k/2, \\ \{\phi_1, \phi_2\} & \text{otherwise,} \end{cases}$$

where $\phi_1 = \max\{\phi \in \mathcal{D}_k \mid \phi \le \psi\}$ and $\phi_2 = \min\{\phi \in \mathcal{D}_k \mid \phi \ge \psi\}$. (Note that $\phi_1 = \phi_2$ if $\psi \in \mathcal{D}_k$.)

3.2 The Offline Bernoulli HMM $\mathbb{B}_{\text{Bayes}}$

In the example above, we defined the offline Bernoulli HMM $\mathbb{B}_{\text{Bayes}}$ using an unspecified set $\mathcal{A}_T$ of discretisation points. We now complete the definition.

Definition 2. $\mathbb{B}_{\text{Bayes}}$ is the Bernoulli HMM as introduced in (2), defined with respect to $\mathcal{A}_T = \{\alpha(\phi) \mid \phi \in \mathcal{D}_{k(T)}\}$, where $k(T) = \left\lceil \frac{1}{2}\log(T\pi^2(2-\sqrt{2}))\right\rceil$.

As the number of transitions per time step equals $|\mathcal{A}_T|$ for this Bernoulli HMM, the forward algorithm for $\mathbb{B}_{\text{Bayes}}$ runs in $O(T\sqrt{T})$ time.

We proceed to analyse the regret of $\mathbb{B}_{\text{Bayes}}$ in the worst case over all possible binary sequences $y^T \in \{0,1\}^T$. The following lemma is at the basis for all of the following results. Its proof, and the proofs of the other results in this section, are deferred to the appendix.

Lemma 3 (Generalised Divergence Bound)**.** *Suppose ϕ_1, ϕ_2 and ϕ_3 all lie in $[0,\pi/4]$ and $\phi_2 > 0$. Then*

$$\begin{aligned} \ln\frac{P_{\phi_1}(\phi_3)}{P_{\phi_2}(\phi_3)} &= D(\phi_3\|\phi_2) - D(\phi_3\|\phi_1) \\ &\le \begin{cases} 4(\phi_2-\phi_1)(\phi_2-\phi_3) & \text{if } \phi_3 \le \phi_2, \\ 4(\phi_2-\phi_1)(\phi_2-\phi_3)\dfrac{\phi_3}{\phi_2} & \text{otherwise.} \end{cases} \end{aligned} \tag{3}$$

Note that by symmetry in $\pi/4$ the lemma can also be applied to $\phi_i' = \pi/2 - \phi_i$ for $i = 1,2,3$. Although it provides a bound on the Kullback-Leibler divergence, which is an expected quantity, we use it to prove results on individual sequence regret. In particular, Lemma 3 will typically be applied with ϕ_3 set to the maximum likelihood $\hat{\phi}$ for some binary sequence. As a notational reminder, ϕ_3 will be called $\hat{\phi}$ in the remainder.

The following consequence of Lemma 3 is an important intermediate result. It expresses that the regret of using the best discretisation point rather than the maximum likelihood is $O(\delta_k^2)$, which means that $O(\sqrt{T})$ uniformly spaced discretisation points suffice to achieve an $O(1)$ overall worst-case regret. Using the ϕ-parametrisation is crucial; in the α-parametrisation the discretisation points must be packed extra densely near the boundaries of the parameter space.

Lemma 4 (Discretisation Lemma)**.** *For any $\hat{\phi} \in [0,\pi/2]$ and $\phi \in (0,\pi/2)$ it holds that*

$$\min_{\phi \in N_k(\hat{\phi})} \ln\frac{P_{\hat{\phi}}(\hat{\phi})}{P_\phi(\hat{\phi})} \le (8-4\sqrt{2})\delta_k^2 \le 2.4\ \delta_k^2.$$

Specifically, for $\mathbb{B}_{\text{Bayes}}$ we obtain the following worst-case regret bound.

Theorem 2 (Offline Discretisation)**.** *For any binary sequence $y^T \in \{0,1\}^T$ and any $\alpha \in [0,1]$*

$$\ln\frac{P_\alpha(y^T)}{\mathbb{B}_{\text{Bayes}}(y^T)} \le \tfrac{1}{2}\ln T + 2.8.$$

3.3 The Online Bernoulli HMM $\mathbb{B}_{\text{ro}}$

We shall now define the remaining properties of the Refine-Online Bernoulli HMM, $\mathbb{B}_{\text{ro}}$, using $\mathcal{D}_k$ as before. But since we do not know T, rather than choosing a fixed k as a function of T, we let k increase by one every time the precision threatens to become insufficient, roughly doubling the number of discretisation points. The critical step in the definition of $\mathbb{B}_{\text{ro}}$ will describe how to patch things up whenever k increases.

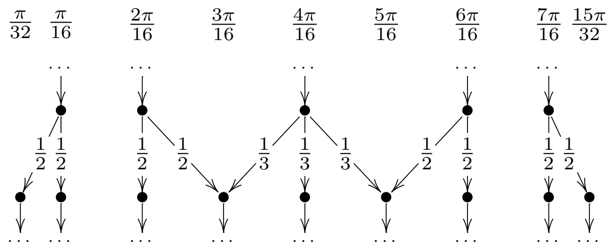

Figure 2: Refinement from $\mathcal{D}_2$ to $\mathcal{D}_3$.

Our approach is more subtle than the *doubling trick*, which is often used to deal with unknown T [Cesa-Bianchi and Lugosi, 2006]. Naive doubling can be done in two ways. The simplest is to restart the algorithm completely each time the precision needs to be increased. But then the Bernoulli parameter has to be relearned in each segment, which results in a significantly worse loss bound of order $O(\ln^2 T)$. Alternatively, one might revisit previous data and continue by setting the algorithm's weights as if the increased precision had been used from the start. But this requires the algorithm to store all data indefinitely; moreover, we have not been able to improve our loss bound using this approach. In the following we therefore suggest a more advanced way of doubling, which redistributes the weights of the algorithm without looking at old data whenever the precision is increased.

We first define $\mathbb{B}_{\text{ro}}^k$ with respect to a function $k\colon \mathbb{Z}^+ \to \mathbb{Z}^+$, called the *discretisation function*. It identifies the discretisation set $\mathcal{D}_{k(t)}$ to be used at time t, and should have the property that $k(t+1) = k(t)$ or $k(t+1) = k(t)+1$ for all t. Thus $\mathcal{A}_t = \{\alpha(\phi) \mid \phi \in \mathcal{D}_{k(t)}\}$. The discretisation function for `Refine-Online` is

$$\kappa(t) = \left\lfloor \tfrac{1}{2}\log t + \log\log(t+1)\right\rfloor + 1,$$

and we simply write $\mathbb{B}_{\text{ro}}$ for $\mathbb{B}_{\text{ro}}^\kappa$.

The initial weights of the states are $\mathbb{B}_{\text{ro}}^k(\langle\alpha_1, 1\rangle) = 1/|\mathcal{D}_{k(1)}|$. It remains to define the transition probabilities between states. For consecutive times t and $t+1$ when the discretisation does not change, i.e. $k(t) = k(t+1)$, these transitions are similar to those for the Bayesian Bernoulli HMM in (2); for times when the discretisation does change, the probabililities are given by a *refinement function* $d_k\colon \mathcal{D}_k \times \mathcal{D}_{k+1} \to [0,1]$. Thus, $\mathbb{B}_{\text{ro}}^k(\langle\alpha_{t+1}, t+1\rangle \mid \langle\alpha_t, t\rangle)$

$$= \begin{cases} \mathbf{1}_{\{\alpha_t\}}(\alpha_{t+1}) & \text{if } k(t) = k(t+1),\\ d_{k(t)}(\phi(\alpha_t), \phi(\alpha_{t+1})) & \text{otherwise.}\end{cases}$$

The refinement function d_k, which determines our patch-up strategy, is chosen such that ϕ_{t+1} gets some mass from each of its neighbours in $N_{k(t)}(\phi_{t+1})$:

$$d_k(\phi_t, \phi_{t+1}) = \mathbf{1}_{N_k(\phi_{t+1})}(\phi_t) \cdot \begin{cases} \frac{1}{2} & \text{if } \phi_t \le \delta_k \text{ or } \phi_t \ge \frac{1}{2}\pi - \delta_k,\\ \frac{1}{3} & \text{otherwise.}\end{cases}$$

The refinement function is illustrated by Figure 2 for $k = 2$, but note that as $k(t)$ gets larger, the case that $d_k(\phi_t, \phi_{t+1}) = 1/3$ becomes most important. Also note there are at most three transitions for each discretisation point per time step. The forward algorithm therefore runs in time proportional to $\sum_{t=1}^T |\mathcal{D}_{k(t)}| \le T\,|\mathcal{D}_{k(T)}|$. In particular for $\mathbb{B}_{\text{ro}}$ ($k = \kappa$) its running time is $O(T\sqrt{T}\log T)$.

While it may seem redundant to allow for converging paths in the HMM, we do need such a structure for the proof of the lemma below, which bounds the weights of the newly introduced discretisation points. The idea is to compare the weight that is accumulated in any state $\langle\alpha_t, t\rangle$ after observing y^t, to $P_{\alpha_t}(y^t)$. Let $t(k) = \min\{t \in \mathbb{Z}^+ \mid k(t) = k\}$ be the first time at which the k-discretisation is used. If the discretisation function k were strictly increasing, this would be its inverse.

Lemma 5 (Refinement Lemma)**.** *For any $y^t \in \{0,1\}^t$, any $\phi \in \mathcal{D}_{k(t)}$ it holds that*

$$\ln\frac{P_\phi(y^t)}{\mathbb{B}_{\text{ro}}^k(y^t, \langle\alpha(\phi), t\rangle)} \le \ln|\mathcal{D}_{k(1)}| + \sum_{k=k(1)+1}^{k(t)} \ln 3 + (4 - 2\sqrt{2})\pi^2\frac{t(k)-1}{4^k}. \quad (4)$$

In particular for the discretisation function κ we get

$$\ln\frac{P_\phi(y^t)}{\mathbb{B}_{\text{ro}}(y^t, \langle\alpha(\phi), t\rangle)} \le \log 3\Big(\tfrac{1}{2}\ln t + \ln\ln(t+1)\Big) + 20.7.$$

Using this lemma it is not hard to provide a worst-case regret bound for $\mathbb{B}_{\text{ro}}$.

Theorem 3 (Online Discretisation)**.** *For any binary sequence $y^t \in \{0,1\}^t$ and any $\alpha \in [0,1]$*

$$\ln\frac{P_\alpha(y^t)}{\mathbb{B}_{\text{ro}}(y^t)} \le \log 3\left(\tfrac{1}{2}\ln t + \ln\ln(t+1)\right) + 23.1.$$

Here the constant is the sum of the constants appearing in Lemmas 4 and 5. The proof of this theorem is based on the regret of the discretisation point $\ddot{\phi}(y^t) \in \mathcal{D}_{k(t)}$ that is closest to the unconstrained maximum likelihood $\hat{\phi}(y^t)$. There are $O(\log t)$ discretisation points sufficiently close to $\hat{\phi}(y^t)$. Taking this into account would result in an improved constant in front of the $\ln\ln(t+1)$ term, but the term would not vanish and the proof would become more complex.

4 Conclusion

We have presented a new discretisation scheme for Bernoulli sources that achieves a regret bound of

$\frac{1}{2}\ln T + 2.8$ nats if the final number of outcomes, T, is known in advance, but unlike the approach in [Monteleoni and Jaakkola, 2003] specifies the exact number and positions of the discretisation points explicitly. This scheme is most useful, however, when T is not known in advance: in Section 3.3 the HMM $\mathbb{B}_{\text{ro}}$ was presented that achieves a regret of $\frac{1}{2}\log 3 \ln T + \log 3 \ln\ln(T+1) + 23.1$ nats without knowing T in advance. The predictions of $\mathbb{B}_{\text{ro}}$ can be computed in $O(T\sqrt{T}\log T)$ time using the standard forward algorithm for HMMs.

Our interest in Bernoulli sources stems from Lemma 1, which shows that these bounds directly translate into regret bounds for learning the switching rate for the `Fixed-Share` algorithm. As discussed in Section 2.1, running times also carry over. We call the new algorithm for the case where T is not known `Refine-Online`.

Analogues to Lemma 1 may easily be proved for any expert algorithm that involves a repeated binary choice with fixed probability, like *elementwise mixtures* [Koolen and de Rooij, 2008].

Future Research The worst-case regret for Bernoulli sources is $\frac{1}{2}\log T + O(1)$ [Cesa-Bianchi and Lugosi, 2006, Thm 9.2]. This provides a lower bound on the worst-case regret for tracking HMMs, because Lemma 1 is tight. The lower bound is achieved by $\mathbb{S}_{\text{Bayes}}$, but for $\mathbb{S}_{\text{ro}}$ a $\log 3$ factor appears. This factor can be explained as follows. When the discretisation is refined, each new point gets mass from two neighbours, but our analysis in Lemma 4 only takes the best neighbour into account. It is an interesting open question whether the optimal bound could be achieved, at least up to $O(\log\log T)$, by improving either the refinement function or the analysis.

Acknowledgements

We thank Wouter Koolen, who helped improve the crucial Lemma 3, and Peter Harremoës and the anonymous referees for useful suggestions. This research was carried out while the first author was at Eindhoven University of Technology (TU/e) and the University of Cambridge, and the second author was at the Centrum Wiskunde & Informatica (CWI) in Amsterdam. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

References

F. J. Anscombe. The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35:246–254, 1948.

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games.* Cambridge University Press, 2006.

M. Freeman and J. Tukey. Transformations related to the angular and the square root. *Annals of Mathematical Statistics*, 21:607–611, 1950.

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

W. Koolen and S. de Rooij. Expert automata for efficient tracking. In *Proc. of the 21st Annual Conf. on Computational Learning Theory*, pages 275–286, 2008.

C. Monteleoni and T. Jaakkola. Online learning of nonstationary sequences. In *Advances in NIPS*, volume 16, Cambridge, MA, 2003. MIT Press.

T. van Erven, P. Grünwald, and S. de Rooij. Catching up faster in Bayesian model selection and model averaging. In *Advances in NIPS*, volume 20, 2008.

P. Volf and F. Willems. Switching between two universal source coding algorithms. In *Proc. of the Data Compression Conf., Snowbird, Utah*, pages 491–500, 1998.

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

A Proofs

Generalised Divergence Bound (Lemma 3) The equality follows by rewriting definitions. The inequality is proved as follows. For any concave function f with derivative f', and any x and y, it holds that

$$(x-y)f'(x) \le f(x) - f(y) \le (x-y)f'(y). \tag{5}$$

In particular for $\ln P_\phi(\phi_3)$ as a function of ϕ:

$$\begin{aligned}\ln\frac{P_{\phi_1}(\phi_3)}{P_{\phi_2}(\phi_3)} &\le (\phi_1-\phi_2)\left(2\alpha_3\frac{\cos\phi_2}{\sin\phi_2} - 2(1-\alpha_3)\frac{\sin\phi_2}{\cos\phi_2}\right)\\ &= 2(\phi_1-\phi_2)\frac{\cos^2\phi_2 - \cos^2\phi_3}{\sin\phi_2\cos\phi_2}.\end{aligned} \tag{6}$$

Since $\cos^2\phi$ is a concave function of ϕ as well, we can use (5) once more to find

$$\begin{aligned}&-2(\phi_2-\phi_3)\sin\phi_2\cos\phi_2 \le \cos^2\phi_2 - \cos^2\phi_3\\ &\le -2(\phi_2-\phi_3)\sin\phi_3\cos\phi_3.\end{aligned} \tag{7}$$

If $\phi_2 - \phi_3 \ge 0$, then plugging the left-hand side into (6) gives the first case of (3). For $\phi_2 - \phi_3 < 0$ we first combine the inequality on the right hand side of (7) with (6) to find

$$\ln\frac{P_{\phi_1}(\phi_3)}{P_{\phi_2}(\phi_3)} \le 4(\phi_1-\phi_2)(\phi_3-\phi_2)\frac{\sin\phi_3\cos\phi_3}{\sin\phi_2\cos\phi_2}. \tag{8}$$

As $\sin x\cos x = \sin 2x$ and $\sin x$ is concave on $[0,\pi/2]$, we also get by (5) that $\frac{\sin\phi_3\cos\phi_3}{\sin\phi_2\cos\phi_2} \le 1 + \frac{2(\phi_3-\phi_2)}{\tan(2\phi_2)} \le \frac{\phi_3}{\phi_2}$, where the second inequality follows by $\tan x \ge x$ for $x \in [0,\pi/2]$. With (8) this completes the proof. □

Discretisation Lemma (Lemma 4) We first show that for any $0 < \phi_1 \le \hat{\phi} \le \phi_2 \le \pi/4$ it holds that

$$\min_{\phi\in\{\phi_1,\phi_2\}} \ln\frac{P_{\hat{\phi}}(\hat{\phi})}{P_\phi(\hat{\phi})} \le 4(\phi_2 - \sqrt{\phi_1\phi_2})^2. \tag{9}$$

This follows by relaxing Lemma 3 to get $\ln P_{\hat\phi}(\hat\phi) - \ln P_{\phi_1}(\hat\phi) \leq 4(\phi_1 - \hat\phi)^2(\hat\phi/\phi_1)^2$ (strictly monotonically increasing in $\hat\phi$) and $\ln P_{\hat\phi}(\hat\phi) - \ln P_{\phi_2}(\hat\phi) \leq 4(\phi_2 - \hat\phi)^2$ (strictly decreasing). At the maximising $\hat\phi = \sqrt{\phi_1\phi_2}$ the bounds are equal. Substitution completes the proof of (9).

To prove Lemma 4, assume w.l.o.g. that $\hat\phi \leq \pi/4$; the other case is symmetric. Then $\phi \leq \pi/4$ for all $\phi \in N_k(\hat\phi)$. If $N_k(\hat\phi) = \{\hat\phi\}$, the lemma is trivially true. If $N_k(\hat\phi) = \{\delta_k/2\}$, then $\hat\phi \leq \delta_k/2$ and from Lemma 3 we get $\ln P_{\hat\phi}(\hat\phi) - \ln P_{\delta_k/2}(\hat\phi) \leq 4(\frac{1}{2}\delta_k - \hat\phi)^2 \leq \delta_k^2$. If $N_k(\hat\phi) = \{\frac{1}{2}\delta_k, \delta_k\}$ we similarly obtain a bound of δ_k^2. Finally suppose that $N_k(\hat\phi) = \{i\delta_k, (i+1)\delta_k\}$ for some integer $i \geq 1$. Then application of (9) yields

$$\min_{\phi \in \{i\delta_k, (i+1)\delta_k\}} \ln \frac{P_{\hat\phi}(\hat\phi)}{P_\phi(\hat\phi)} \leq 4\big((i+1) - \sqrt{i(i+1)}\big)\delta_k^2,$$

which is maximised by $i = 1$. □

Offline Discretisation (Theorem 2) Let $\hat\phi$ denote the maximum likelihood and $\ddot\phi = \operatorname{argmax}_{\phi \in \mathcal{D}_k} P_\phi(y^T)$ denote the maximum likelihood in $\mathcal{D}_k$. The theorem follows by $\ln \mathbb{B}_{\text{Bayes}}(y^T) \geq \ln P_{\ddot\phi}(y^T) + \ln w(\ddot\phi)$ and Lemma 4. □

Lemma 6. *Suppose that $0 < \phi_1 \leq \phi_2 \leq \pi/4$ and define $\psi = \frac{1}{2}(\phi_1 + \phi_2)$. Then for any $\hat\phi \in [0, \pi/2]$,*

$$\min_{\phi \in \{\phi_1, \phi_2\}} \ln \frac{P_\psi(\hat\phi)}{P_\phi(\hat\phi)} \leq 2(\phi_2 - \phi_1)(\phi_2 - \sqrt{\phi_1\phi_2}). \quad (10)$$

Proof. As $\ln P_\phi(\hat\phi)$ is a concave function of ϕ achieving its maximum at $\phi = \hat\phi$, we have for $\hat\phi < \phi_1$ or $\hat\phi > \phi_2$ that $\min_{\phi \in \{\phi_1, \phi_2\}} \ln P_\psi(\hat\phi)/P_\phi(\hat\phi) \leq 0$, such that (10) is satisfied. Therefore assume w.l.o.g. that $\phi_1 \leq \hat\phi \leq \phi_2$. At the worst-case $\hat\phi$, the bounds from Lemma 3 must be equal; solving yields $\hat\phi = \sqrt{\phi_1\phi_2}$. Substitution in one of the bounds completes the proof. □

Lemma 7. *For all $\psi \in \mathcal{D}_{k+1}$ and any $\hat\phi \in [0, \pi/2]$,*

$$\min_{\phi \in N_k(\psi)} \ln \frac{P_\psi(\hat\phi)}{P_\phi(\hat\phi)} \leq (4 - 2\sqrt{2})\delta_k^2 = \frac{(4 - 2\sqrt{2})\pi^2}{4^{k+1}}. \quad (11)$$

Proof. Assume w.l.o.g. that $\psi < \pi/4$. Then $\phi \leq \pi/4$ for all $\phi \in N_k(\psi)$. If $\psi \in \mathcal{D}_k$, then the lemma is trivially true. If $\psi = \delta_{k+1}/2$, then $N_k(\psi) = \{\delta_k/2\}$, and as $\ln P_\phi(\hat\phi)$ is concave in ϕ and achieves its maximum at $\phi = \hat\phi$, (11) is satisfied if $\hat\phi > \delta_k/2$. If $\hat\phi \leq \delta_k/2$ it follows by Lemma 3 that $\ln P_{\delta_{k+1}/2}(\hat\phi)/P_{\delta_k/2}(\hat\phi) \leq 4(\delta_{k+1}/2)(\delta_k/2 - \hat\phi(y^t)) \leq \frac{1}{2}\delta_k^2$. If neither of these cases apply, we must have $N_k(\psi) = \{\phi_1, \phi_2\}$ with $\phi_1 = i\delta_k$ and $\phi_2 = (i+1)\delta_k$ for some integer $i \geq 1$, and $\psi = (\phi_1 + \phi_2)/2$. In that case we apply Lemma 6 to find

$$\min_{\phi \in N_k(\psi)} \ln \frac{P_\psi(\hat\phi)}{P_\phi(\hat\phi)} \leq 2\delta_k^2(i + 1 - \sqrt{i(i+1)}),$$

which is maximised by $i = 1$. □

Refinement Lemma (Lemma 5) Abbreviate $\langle \alpha(\phi), t\rangle$ to $\langle \phi, t\rangle$ and let $b(t)$ denote the right-hand side of (4). The proof of the first part of the lemma is by induction on t. The case $t = 1$, for which $b(t) = \ln|\mathcal{D}_{k(1)}|$, is verified by noting that $\mathbb{B}_{\text{ro}}(y^1, \langle \phi, 1\rangle) = P_\phi(y^1)/|\mathcal{D}_{k(1)}|$. Suppose the bound is valid for some t. To show that it is also valid for $t+1$, we use that

$$\begin{aligned}
&\ln \frac{P_{\phi_{t+1}}(y^{t+1})}{\mathbb{B}_{\text{ro}}(y^{t+1}, \langle \phi_{t+1}, t+1\rangle)} - \ln \frac{P_{\phi_{t+1}}(y^t)}{\mathbb{B}_{\text{ro}}(y^t, \langle \phi_t, t\rangle)} \\
&\leq \min_{\phi_t \in \mathcal{D}_{k(t)}} \ln \frac{P_{\phi_{t+1}}(y_{t+1})}{\mathbb{B}_{\text{ro}}(y_{t+1}, \langle \phi_{t+1}, t+1\rangle \mid y^t, \langle \phi_t, t\rangle)} \\
&= \min_{\phi_t \in \mathcal{D}_{k(t)}} -\ln \mathbb{B}_{\text{ro}}(\langle \phi_{t+1}, t+1\rangle \mid \langle \phi_t, t\rangle).
\end{aligned}$$

In case $k(t+1) = k(t)$ the bound does not change (i.e. $b(t+1) = b(t)$), because for $\phi_t = \phi_{t+1} \in \mathcal{D}_{k(t)}$ it holds that $\mathbb{B}_{\text{ro}}(\langle \phi_{t+1}, t+1\rangle \mid \langle \phi_t, t\rangle) = 1$, and by induction $\ln P_{\phi_t}(y^t) - \ln \mathbb{B}_{\text{ro}}(y^t, \langle \phi_t, t\rangle) \leq b(t)$. Now suppose that $k(t+1) = k(t) + 1$. Then

$$\begin{aligned}
&\min_{\phi_t \in \mathcal{D}_{k(t)}} -\ln \mathbb{B}_{\text{ro}}(\langle \phi_{t+1}, t+1\rangle \mid \langle \phi_t, t\rangle) + \ln \frac{P_{\phi_{t+1}}(y^t)}{\mathbb{B}_{\text{ro}}(y^t, \langle \phi_t, t\rangle)} \\
&= \min_{\phi_t \in N_{k(t)}(\phi_{t+1})} -\ln d_{k(t)}(\phi_t, \phi_{t+1}) + \ln \frac{P_{\phi_{t+1}}(y^t)}{\mathbb{B}_{\text{ro}}(y^t, \langle \phi_t, t\rangle)} \\
&\leq \ln 3 + \min_{\phi_t \in N_{k(t)}(\phi_{t+1})} \ln \frac{P_{\phi_{t+1}}(y^t)}{P_{\phi_t}(y^t)} + b(t) \\
&\leq \ln 3 + (4 - 2\sqrt{2})\pi^2 \frac{t}{4^{k(t)+1}} + b(t) = b(t+1),
\end{aligned}$$

where the first inequality holds by induction and the last inequality follows from Lemma 7.

For the second part of the lemma we bound $t(k)$ using

$$\sqrt{t(k)} \log(t(k)+1) \leq 2^k \leq 2\sqrt{t(k)} \log(t(k)+1). \quad (12)$$

From the left-hand side of (12) we get $\sqrt{t(k)} \leq 2^k/(\log(t(k)+1)) \leq 2^k/(\frac{1}{2}\log t(k) + \log\log(t(k)+1))$. (We omit the tedious proof of the last inequality.) Together with the right-hand side of (12) it follows that $\sqrt{t(k)} \leq 2^k(k-1)^{-1}$, which implies $t(k) \leq 4^k(k-1)^{-2} \leq (4^k + 1)(k-1)^{-2}$. The result follows by plugging this bound into $b(t)$. □

Online Discretisation (Theorem 3) Fix an arbitrary sequence y^t, and define the global maximum likelihood $\hat\phi = \hat\phi(y^t)$ and the nearest discretisation point $\ddot\phi = \operatorname{argmax}_{\phi \in \mathcal{D}_{k(t)}} P_\phi(y^t)$. Then $\ln \frac{P_{\hat\phi}(y^t)}{\mathbb{B}_{\text{ro}}(y^t)} \leq t \ln \frac{P_{\hat\phi}(\hat\phi)}{P_{\ddot\phi}(\hat\phi)} + \ln \frac{P_{\ddot\phi}(y^t)}{\mathbb{B}_{\text{ro}}(y^t, \langle \alpha(\ddot\phi), t\rangle)}$. The latter two terms can be bounded using Lemmas 4 and 5, respectively. □