On the Accuracy of the MAP Inference in HMMs

Article *in* Methodology and Computing in Applied Probability · March 2015

DOI: 10.1007/s11009-015-9443-x

2 authors:

Kristi Kuljus
University of Tartu
22 PUBLICATIONS **60** CITATIONS
SEE PROFILE

Jüri Lember
University of Tartu
61 PUBLICATIONS **452** CITATIONS
SEE PROFILE

Methodol Comput Appl Probab
DOI 10.1007/s11009-015-9443-x

On the Accuracy of the MAP Inference in HMMs

Kristi Kuljus · Jüri Lember

Received: 25 November 2013 / Revised: 14 February 2015 / Accepted: 18 February 2015

Abstract In a hidden Markov model, the underlying Markov chain is usually unobserved. Often, the state path with maximum posterior probability (Viterbi path) is used as its estimate. Although having the biggest posterior probability, the Viterbi path can behave very atypically by passing states of low marginal posterior probability. To avoid such situations, the Viterbi path can be modified to bypass such states. In this article, an iterative procedure for improving the Viterbi path in such a way is proposed and studied. The iterative approach is compared with a simple *batch approach* where a number of states with low probability are all replaced at the same time. It can be seen that the iterative way of adjusting the Viterbi state path is more efficient and it has several advantages over the batch approach. The same iterative algorithm for improving the Viterbi path can be used when it is possible to reveal some hidden states and estimating the unobserved state sequence can be considered as an *active learning* task. The batch approach as well as the iterative approach are based on *classification probabilities* of the Viterbi path. Classification probabilities play an important role in determining a suitable value for the threshold parameter used in both algorithms. Therefore, properties of classification probabilities under different conditions on the model parameters are studied.

Keywords Hidden Markov model · Viterbi state path · Segmentation · Active learning · Classification probability

Mathematics Subject Classification (2010) 60J10 · 60J22 · 62M05

K. Kuljus (✉)
Department of Mathematics and Mathematical Statistics, Umeå University, 901 87 Umeå, Sweden
e-mail: kristi.kuljus@math.umu.se

J. Lember
Institute of Mathematical Statistics, University of Tartu, Liivi 2-513, 50409, Tartu, Estonia
e-mail: jyril@ut.ee

1 Introduction and Preliminaries

1.1 Notation

Let $Y = Y_1, Y_2, \ldots$ be a time-homogeneous Markov chain with states $S = \{1, \ldots, K\}$ and irreducible transition matrix $\mathbb{P} = \left(p_{ij}\right)$. Let $X = X_1, X_2, \ldots$ be a process such that: 1) given $\{Y_t\}$ the random variables $\{X_t\}$ are conditionally independent; 2) the distribution of X_j depends on $\{Y_t\}$ only through Y_j. The process X is sometimes called a *hidden Markov process* (HMP) and the pair (Y, X) is referred to as a *hidden Markov model* (HMM). The name is motivated by the assumption that the process Y, which is sometimes called the *regime*, is non-observable. The distributions $P_s := \mathbf{P}(X_1 \in \cdot | Y_1 = s)$ are called *emission distributions*. We shall assume that the emission distributions are defined on a measurable space $(\mathcal{X}, \mathcal{B})$, where $\mathcal{X}$ is usually $\mathbb{R}^d$ and $\mathcal{B}$ is the Borel σ-algebra. Without loss of generality we shall assume that the measures P_s have densities f_s with respect to some reference measure μ. Our notation differs from the one used in the HMM literature, where usually X stands for the regime and Y for the observations. Since our study is mainly motivated by statistical learning, we would like to be consistent with the notation used there and keep X for observations and Y for latent variables. Given a set $\mathcal{A}$ and integers m and n, $m < n$, we shall denote any $(n - m + 1)$-dimensional vector with all the components in $\mathcal{A}$ by $a_m^n := (a_m, \ldots, a_n)$. When $m = 1$, it will be often dropped from the notation and we write $a^n \in \mathcal{A}^n$.

HMMs are widely used in various fields of applications, including speech recognition (Rabiner 1989; Jelinek 1997), bioinformatics (Koski 2001; Durbin et al. 1998), language processing (Och and Ney 2000), image analysis (Li et al. 2000) and many others. For a general overview of HMMs, we refer to Cappé et al. (2005) and Ephraim and Merhav (2002).

1.2 Segmentation and Standard Paths

The *segmentation* problem consists of estimating the unobserved realization of the first n elements of the underlying Markov chain $Y^n = (Y_1, \ldots, Y_n)$, given the first n observations $x^n = (x_1, \ldots, x_n)$ from a hidden Markov process $X^n = (X_1, \ldots, X_n)$. In communications literature segmentation is also known as *decoding* (Viterbi 1967; Bahl et al. 1974) or *state sequence detection* (Hayes et al. 1982). Segmentation is often the primary interest of the HMM-based inference, but it can also be an intermediate step of a larger problem such as estimation of the model parameters (Rabiner 1989; Lember and Koloydenko 2008). Formally, the aim of the segmentation is to look for a mapping $g : \mathcal{X}^n \to S^n$ called a *classifier* or *decoder*, that maps every sequence of observations x^n into a state sequence $g(x^n) = (g_1(x^n), \ldots, g_n(x^n))$, which is typically referred to as a *path*, sometimes also as an *alignment* (Koski 2001; Udupa and Maji 2005). Since it is generally impossible to find the underlying realization of Y^n exactly, the obtained path $g(x^n)$ has to be the best estimate, in a sense. To measure the goodness of the obtained path (or equivalently of the corresponding classifier), it is natural to introduce a task-dependent risk function $R(s^n|x^n)$ that gives a measure of goodness of a path s^n given the data x^n. For a given risk function, the best classifier g is then the one that minimizes the risk: $g(x^n) = \arg\min_{s^n} R(s^n|x^n)$. The path(s) minimizing a risk $R(\cdot|x^n)$ will be referred to as *optimal* path(s) for the particular risk. Such a general risk-based segmentation theory has been introduced by Lember and Koloydenko (2014) and Lember et al. (2011), and independently by Yau and Holmes (2013). The most popular classifier in practice is the so-called *Viterbi classifier* or *maximum a posteriori (MAP) classifier* v that maximizes the posterior probability, i.e.

$$v(x^n) := \arg\max_{s^n} \mathbf{P}(Y^n = s^n | X^n = x^n).$$

The name is inherited from the dynamic programming algorithm (Viterbi algorithm) used for finding it. Obviously, the Viterbi path is not necessarily unique. Despite its popularity, the Viterbi classifier has some major disadvantages. In particular, the Viterbi path does not minimize the expected number of classification errors. The best path in this sense and therefore also often used in practice is the so-called *pointwise maximum a posteriori (PMAP)* path defined as follows:

$$g_t(x^n) := \arg\max_{s \in S} \mathbf{P}(Y_t = s | X^n = x^n), \quad t = 1, \ldots, n.$$

The PMAP path is also known as *marginal posterior mode* (Winkler 2003), *maximum posterior marginals* (Rue 1995) or *posterior decoding* (Brejová et al. 2007) estimate. Because the value of $g_t(x^n)$ does not depend on $g_{t'}$ for any other $t' \neq t$, the PMAP path can be obtained pointwise. Thus, unlike the Viterbi classifier, the PMAP classifier is purely local. The lack of global structure is the biggest disadvantage of the PMAP classifier, since in the presence of zeros in the transition matrix, the path can have zero posterior probability because of the forbidden transitions. As the examples in Lember and Koloydenko (2014) show, PMAP paths with zero posterior probability can indeed happen in practice. Thus, although being the best in the sense of expected number of misclassifications, the PMAP path can be the worst in terms of posterior probability. This problem has already been mentioned in the celebrated tutorial of Rabiner (1989) and is probably one of the main reasons why the Viterbi classifier has become so popular.

Both the Viterbi classifier and the PMAP classifier are commonly used and can be considered the standard classifiers in HMM segmentation (see Lember and Koloydenko (2014) and the references therein). Both the classifiers can be computed with complexity $\mathcal{O}(K^2 n)$: the Viterbi path can be found with the Viterbi algorithm and the *smoothing probabilities* $\mathbf{P}(Y_t = s | X^n = x^n)$ (and hence also the PMAP path) can be calculated with the well-known forward-backward recursions. As mentioned above, both of them are, in a sense, extreme, and they can be very different. The difference of these standard paths becomes more evident, when the number of states K increases. When binary chains ($K = 2$) indeed can leave too little room for the standard paths to differ, then models with bigger K can have very differently behaving standard paths, especially if the transition matrix has many zeros. For such models one would like to have a path that has a reasonably big posterior probability (at least non-zero) and at the same time a rather small number of expected classification errors. To our best knowledge, the first attempt in this direction was made in Brushe et al. (1998), where a hybrid algorithm bridging the Viterbi and PMAP classifiers was proposed. As pointed out in Section 5 of Lember and Koloydenko (2014), the proposed algorithm has many drawbacks and computational issues and is probably not useful in practice. In Lember and Koloydenko (2014) and Lember et al. (2011), the goal of having a path with high posterior probability and small number of errors was achieved by defining new risk functions (measures of goodness), so that the corresponding optimal classifiers would in some sense be between the two standard classifiers and have the properties of both. An example of such a risk function is given below:

$$R(s^n | x^n) = -\frac{C}{n} \ln \mathbf{P}(Y^n = s^n | X^n = x^n) - \frac{1}{n} \sum_{t=1}^{n} \ln \mathbf{P}(Y_t = s_t | X^n = x^n),$$

where $C \geq 0$ is a trade-off parameter. Clearly, if C is big enough, then the optimal path for this risk is the Viterbi path and when $C = 0$, the optimal path is the PMAP path. A

similar approach for designing new classifiers was used in Yau and Holmes (2013). For a more detailed discussion about the difference of the standard paths and for an overview of the different approaches, we refer to Subsection 1.2.1 in Lember and Koloydenko (2014).

In this paper, to obtain a path with high posterior probability and small number of errors, we proceed differently. We take the Viterbi path and try to modify it so that the expected number of classification errors will decrease, but the posterior probability of the modified path will still remain considerably high. This approach is motivated by the study of classification probabilities introduced in the next subsection.

1.3 Overview of the Main Results

1.3.1 Bounds for Classification Probabilities

Given a classifier $g = (g_1, \dots, g_n)$, the main object of interest in this paper is the probability that for a given time point $t = 1, \dots, n$, the path guesses the true state Y_t correctly:

$$\mathbf{P}(Y_t = g_t(x^n)|X^n = x^n). \tag{1.1}$$

Let us call these probabilities *classification probabilities*. Obviously, this probability tends to decrease when the number of hidden states K increases, and for any t the classification probability is biggest when g is the PMAP classifier. For the PMAP classifier (and for any HMM) the following lower bound trivially holds:

$$\mathbf{P}(Y_t = g_t(x^n)|X^n = x^n) \geq \frac{1}{K}, \quad t = 1, \dots, n.$$

Thus, for a two-state HMM one can be sure that given the observations x^n and a time point t, the PMAP classifier guesses the hidden state Y_t correctly with posterior probability $1/2$ at least, even if the overall probability of observing the PMAP state sequence $g(x^n)$ is very small. Given x^n, the normalized expected number of correctly classified states is just the mean of the classification probabilities:

$$A_n(g) := \frac{1}{n} E\left[\sum_{t=1}^{n} I_{\{Y_t = g_t(x^n)\}}|X^n = x^n\right] = \frac{1}{n}\sum_{t=1}^{n} \mathbf{P}(Y_t = g_t(x^n)|X^n = x^n). \tag{1.2}$$

In this paper, $A_n(g)$ is referred to as the *accuracy* of the path $g(x^n)$. The PMAP classifier is the most accurate classifier and the trivial lower bound above gives that its accuracy is at least $1/K$. What about the Viterbi classifier? Can classification probability (1.1) for the Viterbi classifier be arbitrarily low or does there exist a data-independent lower bound just like for the PMAP classifier? Since all together there are at most K^n different state paths, it follows that the Viterbi path must have the posterior probability at least K^{-n}. Since for any t, the classification probability is the sum of the posterior probabilities over all the paths passing v_t at t, we obtain the following trivial lower bound:

$$\mathbf{P}(Y_t = v_t(x^n)|X^n = x^n) \geq \mathbf{P}(Y^n = v(x^n)|X^n = x^n) \geq K^{-n}. \tag{1.3}$$

This bound depends on n and is typically not so useful. Does there exist a positive lower bound not depending on n? These questions are addressed in Section 2. It turns out that the answer depends on the model. We start with an observation that when the transition matrix has only positive entries, then a data-independent lower bound (that depends on the transition matrix) exists (Proposition 2.1). Thereafter we present a counterexample showing that with zeros in the transition matrix this is not necessarily the case, and for such models classification probability (1.1) can be arbitrarily small (Section 2.2.1). This counterexample is

alarming, since it shows that although having the biggest posterior probability, the Viterbi path can (and when n is big enough, then eventually will) sometimes behave highly atypically by passing at certain time t a state that is least expected. Hence, for these models there does not exist a constant data-independent lower bound. However, as follows from Kuljus and Lember (2012), under some mild conditions there still exists a data-dependent lower bound (Lemma 2.1). From this lemma it follows that for a stationary HMM, the tail of the random variable

$$-\ln \mathbf{P}(Y_t = v_t(X^n)|X^n)$$

has an exponential decay independent of t and n. Thus, there exist positive constants r and d so that for any n, any t such that $1 \le t \le n$, and any $u > 0$,

$$\mathbf{P}\left(-\ln \mathbf{P}(Y_t = v_t(X^n)|X^n) > u\right) \le r \exp[-du]$$

(Corollary 2.3). Hence, the classification probability viewed as a random variable can be arbitrarily small, but such events occur with a certain probability only. As shown in Kuljus and Lember (2012), such a lower bound is useful when proving asymptotic results for segmentation.

1.3.2 Modified Viterbi Path: Motivation

As explained above, the classification probabilities of the Viterbi path might be rather small. A small classification probability at t means that in most cases the Viterbi path guesses the hidden state Y_t incorrectly. Hence, to control the accuracy, a natural idea seems to be to modify the Viterbi path by forcing it to bypass such states. This type of the Viterbi algorithm is known as a *constrained Viterbi algorithm* (see, e.g. Cao and Chen 2003). More precisely, one can proceed as follows. Given a threshold parameter $\delta > 0$, find all time points t such that $\mathbf{P}(Y_t = v_t(x^n)|X^n = x^n) \le \delta$. Let that set be $T(\delta, x^n)$. Then, for every $t \in T$, find the PMAP state at t, i.e. find $g_t(x^n) = \arg\max_{s\in S} \mathbf{P}(Y_t = s|X^n = x^n)$. After that determine the *constrained Viterbi path*

$$u(x^n) := \arg \max_{s^n \in S^n : s_t = g_t,\, t\in T} \mathbf{P}(Y^n = s^n|X^n = x^n).$$

Note that the path u coincides with the PMAP path at every $t \in T$, but outside of T, the path u might still differ from the Viterbi path v. In what follows, the described method will be referred to as the *batch approach*. There are two problems connected with the batch approach:

1) Various simulation studies (e.g. Lember and Koloydenko 2014 and Yau and Holmes 2013) have shown that typically a low classification probability entails that the Viterbi path has to be isolated for quite a long time. That is, if the classification probability is low at some time point t, then it is low also in the neighbourhood of t. It is hard to find a general justification to this observation and for some models it might not be true. However, the simulations have shown that for two-state models like the one in the example after Corollary 2.1, the Viterbi path is typically more "inert" and has on average less transitions than the true path and the PMAP path. In particular, when the PMAP path stays in a state for a considerably short time, producing an "island", then typically the Viterbi path will not recognize that island. Hence, the classification probability is low at every time point in that island (see, e.g. Figure 1 in Lember and Koloydenko 2014), and in those cases many consecutive time points should be replaced by the PMAP states. This in turn can involve impossible transitions, so that

the obtained path $u(x^n)$ can have zero posterior probability. We shall see in Section 3 that this can indeed happen.

2) Since the path u equals the PMAP path at every $t \in T$, the classification probability of u at $t \in T$ is biggest possible and hence (given the threshold δ is not too big, that is, $\delta < 1/K$) for every $t \in T$, $\mathbf{P}(Y_t = u_t | X^n = x^n) > \delta$. However, since the path u can differ from the Viterbi path v also outside of T, the probability $\mathbf{P}(Y_t = u_t | X^n = x^n)$ might drop below δ somewhere else.

As a remedy against both the aforementioned disadvantages, in Section 3 we propose a more elaborate iterative modification of the Viterbi path. To understand the idea of the *iterative approach* better, imagine that at some few time points it is possible to figure out the true underlying states of hidden Y. This can be a realistic situation in practice, but since often figuring out true states costs a lot, it can only be done at some few well-chosen time points. One practical example of where revealing the true states is possible is detection of copy number variations (CNVs) in DNA sequences. CNV refers to an alteration (duplication or deletion) of a segment of DNA sequence. One approach for finding CNVs is to incorporate the information into HMM framework (Colella et al. 2007; Wang et al. 2007). Hence, in terms of machine learning, in this paper we consider a special case of *semi-supervised* learning, sometimes also called *active learning* (Sznitman and Jedynak 2010), where the test data can be revealed during the learning process.

In what follows, revealing a hidden state shall be called *probing the true state*. Since we cannot probe the true state often, it is meaningful to do it at some time point t only if the classification probability of the Viterbi classifier at time t is very low, that is, the Viterbi path at time t is likely to be incorrect. Again, one could use the batch approach: figure out the set of time points with lowest misclassification probabilities and reveal them all together, and then find the constrained Viterbi path. Since the revealed states correspond to the true underlying path, all the transitions in the constrained Viterbi path are possible, and therefore it definitely has a positive posterior probability. Thus, problem 1) mentioned above does not arise. However, it turns out that state probing is more efficient when it is done iteratively, as described below.

Start with finding the time point t_1 with the lowest classification probability and probe the true state at t_1. Since the value y_{t_1} of Y_{t_1} is now known, we take this information into consideration. Thus, in addition to finding the constrained Viterbi path, say $v^{(1)}(x^n)$, it is meaningful to recalculate all the smoothing probabilities under the additional condition $Y_{t_1} = y_{t_1}$. Hence, we find the conditional classification probabilities

$$\mathbf{P}(Y_t = v_t^{(1)}(x^n) | X^n = x^n, Y_{t_1} = y_{t_1}), \quad t = 1, \ldots, n.$$

Next, find the time point t_2 with the smallest conditional classification probability, probe the true state at t_2 and determine the constrained Viterbi path, i.e. the maximum posterior probability path that passes y_{t_1} at t_1 and y_{t_2} at t_2. Then calculate again the conditional classification probabilities by conditioning on $Y_{t_1} = y_{t_1}$ and $Y_{t_2} = y_{t_2}$. Thereafter, find t_3 with the lowest conditional classification probability and so on. In Section 3 we present simulations that demonstrate that the iterative approach is more efficient than the batch approach because the same effect, that is a certain decrease in classification errors, can be obtained with considerably fewer state probings. When probing true states is not possible, then instead of the true state we consider the PMAP state as the one being the most likely true state. Thus, in this case the iterative algorithm uses PMAP replacements. Again, the simulation examples in Section 3 demonstrate the advantage of the iterative algorithm over the

batch approach also in the case of PMAP replacements. In machine learning it is known that active learning outperforms its batch counterparts (see, e.g. Sznitman and Jedynak 2010), and the current paper confirms the same in HMM segmentation. The explicit description of the iterative algorithm is given in Section 3.

1.3.3 Unsuccessful State Probing

It turns out that the question of choosing the right points for state probing is more important as it might seem at first sight. Indeed, if we reveal the true state at time point t and see that the Viterbi path guesses Y_t correctly, i.e. $v_t = Y_t$, then the constrained Viterbi path coincides with the original one and hence, nothing changes. On the other hand, if $v_t \neq Y_t$, then the constrained Viterbi path differs from the original one and typically more than just at t. Is the average number of correctly classified states now bigger? Clearly, state probing induces one correctly estimated state, because Y_t is correct. However, in Section 4 we present a counterexample illustrating that it is possible that the constrained Viterbi path behaves so badly in the neighbourhood of t, that the accuracy defined in Eq. 1.2 drops significantly. In other words, despite the fact that the constrained Viterbi path guesses one more state correctly, the average number of correctly classified states for the constrained path is worse than for the unconstrained Viterbi. Therefore, in this example state probing either does not change anything or makes the path even worse, so that the overall effect of revealing the true state is negative! Moreover, we show that the example can be constructed so that the expected number of classification errors introduced by probing the true state can be arbitrarily large. In this example, the badly chosen t has high classification probability. Thus, probing the true state at such t does not make much sense and neither the batch nor the iterative approach would pick t as a possible state probing time. However, it is intriguing to know whether it would be possible to have such counterexamples also with lower classification probabilities. More generally, would it be possible to find out (based on the data x^n and the model) whether the effect of probing the true state at t is non-negative? And are there any models (two-state HMMs or HMMs with positive transitions, perhaps), where state probing is guaranteed to have a non-negative effect only? These questions are the subject of the future research.

The rest of the paper is organized as follows. In Section 2, the classification probabilities and their lower bounds are studied. Section 3 is devoted to the iterative algorithm and to simulations illustrating its behavior. Section 4 presents the counterexample showing that revealing true states can increase the expected number of classification errors.

2 Lower Bounds on Classification Probabilities

In this section, we study classification probabilities (1.1) for the Viterbi path. Classification probabilities play an important role in determining a suitable value for the threshold parameter used in both the batch and the iterative algorithm. Recall that the accuracy of a path is just the mean of the corresponding classification probabilities. At first we note that when all the transition probabilities are positive, then there exists a data-independent lower bound on the classification probabilities of the Viterbi path, hence there exists also a lower bound on the accuracy. Then we present a counterexample showing that in the presence of forbidden transitions this is not the case and the classification probability of the Viterbi path at some point t can be arbitrarily low. Finally, we prove that under an additional condition, low classification probabilities occur with a small probability only.

Throughout the paper we shall use the following notation. For any sequence of observations x^n and for any state sequence y^n, $p(x^n)$ stands for the likelihood and $p(x^n, y^n)$ for the joint likelihood. For any $s \in S$ and $k = 1, \ldots, n$, define the α-variables

$$\alpha(x^k, s) := \sum_{y^k : y_k = s} p(x^k, y^k), \quad \alpha(s, x_k^n) := \sum_{y_k^n : y_k = s} p(x_k^n, y_k^n).$$

Thus

$$p(x^n) = \sum_s \alpha(x^n, s).$$

Let for any $s \in S$ and $t = 1, \ldots, n$,

$$\gamma_t(s) := \mathbf{P}(Y_t = s | X^n = x^n) p(x^n).$$

When the emission distributions are discrete, then

$$\begin{aligned} \alpha(x^k, s) &= \mathbf{P}(X^k = x^k, Y_k = s), \quad \alpha(s, x_k^n) = \mathbf{P}(X_k^n = x_k^n, Y_k = s), \\ p(x^n) &= \mathbf{P}(X^n = x^n), \quad \gamma_t(s) = \mathbf{P}(X^n = x^n, Y_t = s). \end{aligned}$$

2.1 Positive Transitions

We need some additional notation. Recall that $\mathbb{P} = (p_{ij})$ denotes the transition matrix of Y. Let

$$\sigma_1 =: \min_s \frac{\min_{s'} p_{ss'}}{\max_{s'} p_{ss'}}, \quad \sigma_2 =: \min_s \frac{\min_{s'} p_{s's}}{\max_{s'} p_{s's}}. \tag{2.1}$$

Clearly, $\sigma_1 > 0$ if and only if all the transitions are positive and the same holds for σ_2. The following proposition is a special case of Proposition 4.1 in Kuljus and Lember (2012). The proof is given in the Appendix.

Proposition 2.1 *Assume that all the transition probabilities are positive. Let π be an arbitrary initial distribution with K_1 non-zero entries. Then the following bounds hold:*

$$\begin{aligned} \mathbf{P}(Y_t = v_t(x^n) | X^n = x^n) &\geq \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 \sigma_2^2 + (K-1)}, \quad t = 2, \ldots, n-1, \\ \mathbf{P}(Y_1 = v_1(x^n) | X^n = x^n) &\geq \frac{\sigma_1^2}{\sigma_1^2 + (K_1 - 1)}, \\ \mathbf{P}(Y_n = v_n(x^n) | X^n = x^n) &\geq \frac{\sigma_2^2}{\sigma_2^2 + (K-1)}. \end{aligned}$$

Note that for $t = 2, \ldots, n$, the lower bounds on the classification probabilities do not depend on the initial distribution. Hence, the bounds hold also for stationary distribution when it exists. For a stationary chain, the Viterbi path as well as the smoothing probabilities do not depend on whether the forward or backward chain is considered. Hence, for the time-reversed chain, the bounds should remain the same, provided that σ_1 and σ_2 correspond to the time-reversed chain. Let π be now the stationary distribution and let $q_{ss'}$ denote the transition probabilities for the reversed chain, then:

$$q_{ss'} = \frac{p_{s's} \pi_{s'}}{\pi_s}.$$

Let σ_1' and σ_2' be the minimum values as in Eq. 2.1 corresponding to the reversed chain. If the underlying Markov chain is reversible, then $q_{ss'} = p_{ss'}$ and $\sigma_i' = \sigma_i$, $i = 1, 2$. When π

is uniform, then $q_{ss'} = p_{s's}$ ($\mathbb{P}$ is a doubly stochastic matrix), hence $\sigma_1' = \sigma_2$ and $\sigma_2' = \sigma_1$. In both cases $\sigma_1'\sigma_2' = \sigma_1\sigma_2$ and the lower bounds for $t = 2, \ldots, n-1$ remain unchanged. But in general, $\sigma_1'\sigma_2' \neq \sigma_1\sigma_2$, thus the following corollary is meaningful.

Corollary 2.1 *Assume that all the transition probabilities are positive. Then, if the initial distribution is stationary, the following bounds hold:*

$$\mathbf{P}(Y_t = v_t(x^n)|X^n = x^n) \geq \frac{(\sigma_1\sigma_2 \vee \sigma_1'\sigma_2')^2}{(\sigma_1\sigma_2 \vee \sigma_1'\sigma_2')^2 + (K-1)}, \quad t = 2, \ldots, n-1,$$

$$\mathbf{P}(Y_1 = v_1(x^n)|X^n = x^n) \geq \frac{(\sigma_1 \vee \sigma_2')^2}{(\sigma_1 \vee \sigma_2')^2 + (K-1)},$$

$$\mathbf{P}(Y_n = v_n(x^n)|X^n = x^n) \geq \frac{(\sigma_2 \vee \sigma_1')^2}{(\sigma_2 \vee \sigma_1')^2 + (K-1)}.$$

Proof The proof follows from the fact that when $a > b > 0$, then

$$\frac{a}{a + (K-1)} > \frac{b}{b + (K-1)}.$$

□

Example An important two-state HMM is the model with transition matrix

$$\mathbb{P} = \begin{pmatrix} 1-\epsilon_1 & \epsilon_1 \\ \epsilon_2 & 1-\epsilon_2 \end{pmatrix},$$

where $0 < \epsilon_1, \epsilon_2 \leq 0.5$. Without loss of generality, let $\epsilon_1 \leq \epsilon_2$. Then $\sigma_1 = \frac{\epsilon_1}{1-\epsilon_1}$ and $\sigma_2 = \frac{\epsilon_1}{1-\epsilon_2}$. The transition matrix of the reversed chain remains the same, hence $\sigma_i' = \sigma_i$, $i = 1, 2$. Thus, the obtained bounds are

$$\mathbf{P}(Y_1 = v_1(x^n)|X^n = x^n) \geq \frac{\epsilon_1^2}{\epsilon_1^2 + (1-\epsilon_1)^2},$$

$$\mathbf{P}(Y_n = v_n(x^n)|X^n = x^n) \geq \frac{\epsilon_1^2}{\epsilon_1^2 + (1-\epsilon_2)^2},$$

$$\mathbf{P}(Y_t = v_t(x^n)|X^n = x^n) \geq \frac{\epsilon_1^4}{\epsilon_1^4 + (1-\epsilon_1)^2(1-\epsilon_2)^2}, \quad t = 2, \ldots, n-1.$$

When $\epsilon_1 = \epsilon_2 = 0.5$, then the underlying Markov chain consists of iid Bernoulli random variables with parameter 0.5. In this case the Viterbi and the PMAP path are the same. Given that the ties are broken in favor of 1, $v_t(x^n) = 1$ if and only if $f_1(x_t) \geq f_2(x_t)$. All the bounds above equal $\frac{1}{2}$, which is clearly a tight bound. Without loss of generality, let $v_t(x^n) = 1$. The classification probability in this trivial case can be calculated as

$$\mathbf{P}(Y_t = v_t(x^n)|X^n = x^n) = \mathbf{P}(Y_t = 1|X_t = x_t) = \frac{f_1(x_t)}{f_1(x_t) + f_2(x_t)} \geq \frac{1}{2}.$$

2.2 General Case

The proof of Proposition 2.1 holds only in the case of transition matrices with non-zero entries. The following counterexample shows that if the transition matrix contains zeros,

then a data-independent lower bound on the classification probabilities of the Viterbi path does not exist.

2.2.1 Counterexample

Consider a 4-state model with the transition matrix and initial distribution given by

$$\mathbb{P} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}, \quad \pi = (1/4, 1/4, 1/4, 1/4)'.$$

Suppose the emission distributions are all discrete, hence μ is the counting measure and $f_i(x), i = 1, \ldots, 4$, are all probabilities. Suppose there exist atoms x and y so that emission probabilities satisfy the following conditions for some positive A and B:

1) $f_2(x) = 0$, $f_1(x) = f_3(x) = f_4(x) = A$,
2) $f_1(y) = f_3(y) = f_4(y) = 0$, $f_2(y) = B$.

Let $\epsilon > 0$ be arbitrary. We shall show that for n big enough, there exists a sequence of observations x^n with $p(x^n) > 0$ such that for some time point t,

$$\mathbf{P}(Y_t = v_t(x^n) | X^n = x^n) < \epsilon. \tag{2.2}$$

Let $m \in \mathbb{N}$ be so big that

$$\frac{1}{1 + \left(\frac{4}{3}\right)^m} < \epsilon.$$

Consider a sequence of observations $x_1, \ldots, x_n, n > m$, such that $x_1 = x_2 = \ldots = x_m = x$ and $x_{m+1} = y$. By assumptions, the probability of having such observations is strictly positive. Let the rest of the observations, that is $x_{m+2}, \ldots, x_n$, be arbitrary with the only requirement that the probability of emitting x^n is positive, i.e. $p(x^n) > 0$. Note that since all the paths with positive posterior probability, including the Viterbi path, pass state 2 at time $m + 1$, then for any $s \in \{1, 2, 3, 4\}$,

$$\mathbf{P}(Y_t = s | X^n = x^n) = \mathbf{P}(Y_t = s | X^{m+1} = x^{m+1}), \quad t = 1, \ldots, m + 1.$$

Observe also that any path passing state 2 before the time point $m+1$ will have zero posterior probability. Hence, the only path passing state 1 at any $t \leq m$ is the path that is constantly in state 1 up to time m. Therefore, for any $t = 1, \ldots, m$,

$$\mathbf{P}(Y_t = 1, X^{m+1} = x^{m+1}) = \left(\frac{1}{4}\right) A^m \left(\frac{1}{2}\right)^m B = A^m \left(\frac{1}{2}\right)^{m+2} B.$$

Note also that there is no path with transition from state 3 or 4 into state 1 that would have positive posterior probability. Hence, for $s = 3, 4$ and for any $t \leq m$,

$$\alpha(x^t, s) = 2^{t-1} \left(\frac{1}{4}\right) A^t \left(\frac{1}{3}\right)^{t-1},$$

implying that

$$\mathbf{P}(Y_m = s, X^{m+1} = x^{m+1}) = 2^{m-1} \left(\frac{1}{4}\right) A^m \left(\frac{1}{3}\right)^m B.$$

It follows that

$$\mathbf{P}(Y_m = 1 | X^n = x^n) = \frac{A^m \left(\frac{1}{2}\right)^{m+2}}{A^m \left(\frac{1}{2}\right)^{m+2} + 2^m \left(\frac{1}{4}\right) A^m \left(\frac{1}{3}\right)^m} = \frac{1}{1 + \left(\frac{4}{3}\right)^m} < \epsilon.$$

Thus, if $v_m(x^n) = 1$, then Eq. 2.2 holds. Let us show that up to the time point m, the Viterbi path is given by $v_1 = v_2 = \cdots = v_m = 1$. Since the Viterbi path passes state 2 at $m + 1$, by optimality principle the observations $x_{m+2}, \ldots, x_n$ do not affect the path up to $m + 1$. Therefore, it is sufficient to consider the joint likelihood up to $m + 1$. For $u_1 = \cdots = u_m = 1, u_{m+1} = 2$,

$$p(x^{m+1}, u^{m+1}) = \left(\frac{1}{4}\right)\left(\frac{1}{2}\right)^m A^m B.$$

All other paths with positive posterior probability can up to time m pass states 3 and 4 only. For any such path s^{m+1},

$$p(x^{m+1}, s^{m+1}) = \left(\frac{1}{4}\right)\left(\frac{1}{3}\right)^m A^m B, \quad s_t \in \{3, 4\}, \quad t = 1, \ldots, m, \quad s_{m+1} = 2.$$

Since $\frac{1}{2} > \frac{1}{3}$, we have $p(x^{m+1}, u^{m+1}) > p(x^{m+1}, s^{m+1})$, and therefore $v(x^{m+1}) = u^{m+1}$. Thus, Eq. 2.2 holds for $t = m$.

2.2.2 *Data-Dependent Lower Bound*

Cluster Assumption We shall relax the assumption of positive transitions by the following much weaker assumption. Let G_j denote the support of the emission distribution P_j. We call a non-empty subset $C \subset S$ *a cluster* if the following conditions are satisfied:

$$\min_{j \in C} P_j(\cap_{s \in C} G_s) > 0 \quad \text{and} \quad \max_{j \notin C} P_j(\cap_{s \in C} G_s) = 0.$$

Here the maximum over an empty set equals zero. Hence, a cluster is a maximal subset of states such that $G_C := \cap_{s \in C} G_s$, the intersection of the supports of the corresponding emission distributions, is "detectable". Every state belongs to at least one cluster. Distinct clusters need not be disjoint and a cluster can consist of a single state. In this latter case such a state is not hidden, since it is exposed by any observation it emits. If $K = 2$, then S is the only cluster possible, because otherwise the underlying Markov chain would cease to be hidden. The existence of C implies the existence of a set $\mathcal{X}_o \subset \cap_{s \in C} G_s$ and $\epsilon > 0, M < \infty$, such that $\mu(\mathcal{X}_o) > 0$ and $\forall x \in \mathcal{X}_o$ the following statements hold: (i) $\epsilon < \min_{s \in C} f_s(x)$; (ii) $\max_{s \in C} f_s(x) < M$; (iii) $\max_{s \notin C} f_s(x) = 0$. For proof, see Lember and Koloydenko (2010).

A1 (cluster-assumption): There exists a cluster $C \subset S$ such that the sub-stochastic matrix $R = (p_{ij})_{i,j \in C}$ is primitive, i.e. there is a positive integer r such that the r-th power of R is strictly positive (all its elements are strictly positive).

The cluster assumption **A1** is often met in practice. It is clearly satisfied if all the elements of $\mathbb{P}$ are positive. Since any irreducible aperiodic matrix is primitive, assumption **A1** is also satisfied if the densities f_s satisfy the following condition: for every $x \in \mathcal{X}$, $\min_{s \in S} f_s(x) > 0$, i.e. for all $s \in S$, $G_s = \mathcal{X}$. Thus, **A1** is more general than the *strong mixing condition* (Assumption 4.3.21 in Cappé et al. 2005) and also weaker than Assumption 4.3.29 in Cappé et al. (2005). Note that **A1** implies the aperiodicity of Y, but not vice versa.

Example Let us reconsider the counterexample from Section 2.2.1. The example is very easy to modify so that **A1** holds. It suffices to have one atom, say z, so that $f_j(z) > 0$ for every $j = 1, 2, 3, 4$. Then $z \in \cap_{j \in S} G_j$ so that the cluster consists of all states, i.e. $C = \{1, 2, 3, 4\}$. Note that $\mathbb{P}^2$ is primitive, so that $r = 2$. The set $\mathcal{X}_o$ can be taken as $\{z\}$.

Let x^n be fixed and $\mathcal{X}_o$ and r be as in **A1**. Define for any $t \in \{1, \ldots, n\}$,

$$w_t(x^n) := \min\{t + r < w \leq n : x_{w-r}^w \in \mathcal{X}_o^{r+1}\} \wedge n,$$
$$u_t(x^n) := \max\{1 \leq u < t - r : x_u^{u+r} \in \mathcal{X}_o^{r+1}\} \vee 1,$$

where minimum over the empty set is set to ∞ and maximum over the empty set is set to $-\infty$. Thus, w_t is the first time after t when a word from $\mathcal{X}_o^{r+1}$ is fully observed, and $w_t = n$ if there is no such word up to time n. Similarly, u_t is the last time before t when a word from $\mathcal{X}_o^{r+1}$ is fully observed, and $u_t = 1$ if there is no such word up to time t. The following lemma follows from Proposition 4.1 and Corollary 4.1 in Kuljus and Lember (2012).

Lemma 2.1 *There exist positive constants c and A such that for every $t = 1, \ldots, n$,*

$$\boldsymbol{P}(Y_t = v_t(x^n) | X^n = x^n) \geq c \exp[-A(w_t - u_t)]. \tag{2.3}$$

The bound in Eq. 2.3 depends on x^n, because w_t and u_t depend on x^n. If there is no word from $\mathcal{X}_o^{r+1}$ in the observation sequence x^n, then $w_t = n$ and $u_t = 1$, so that the bound is $c \exp[-A(n-1)]$. Hence, Eq. 2.3 clearly improves the trivial bound given in Eq. 1.3.

Stochastic Bounds that are Independent of n Letting now the data X^n be random, we get that W_t and U_t are random stopping times, and the bound in Eq. 2.3 can be written as

$$\mathbf{P}(Y_t = v_t(X^n) | X^n) \geq c \exp[-A(W_t - U_t)]. \tag{2.4}$$

Let us study the distribution of the random variables $W_t - U_t$. Obviously, $W_t - U_t \leq n - 1$, and the distribution of $W_t - U_t$ depends on both t and n. We would, however, like to have an upper bound on $W_t - U_t$ that is independent of n and if possible, also independent of t. Consider the observation process $X_1, X_2, \ldots$, and let

$$W_t^* := \min\{w > t + r : X_{w-r}^w \in \mathcal{X}_o^{r+1}\}.$$

Thus, $W_t = W_t^* \wedge n$, so that $W_t \leq W_t^*$. The random variable W_t^* is independent of n and as the following proposition shows, $W_t^* - t$ has an exponential tail that can be chosen independently of t.

Proposition 2.2 *Assume* **A1**. *There exist constants $a > 0$ and $b > 0$ such that for any initial distribution π and for any t,*

$$\mathbf{P}(W_t^* - t > k) \leq a \exp[-bk] \quad \textit{for all} \quad k = 0, 1, 2, \ldots.$$

The proof is given in the Appendix. Because of the proposition, the following corollary holds.

Corollary 2.2 *Assume* **A1.** *Then for any initial distribution, the following lower bound holds:*

$$\mathbf{P}(Y_t = v_t(X^n) | X^n) \geq c \exp[-At] Z_t.$$

Here Z_t is a $\sigma(X_1, X_2, \ldots)$-measurable random variable such that $-\ln Z_t$ has an exponential tail independent of t, that is for some positive constants r and d and for every $u > 0$, $\mathbf{P}(-\ln Z_t > u) \leq r \exp[-du]$.

Proof Let $\lfloor\cdot\rfloor$ denote the floor function. From Eq. 2.4 it follows that

$$\mathbf{P}(Y_t = v_t(X^n)|X^n) \geq c\exp[-AW_t^*] = c\exp[-At]\exp[-A(W_t^* - t)] = c\exp[-At]Z_t,$$

where $Z_t = \exp[-A(W_t^* - t)]$. Thus, $-\ln Z_t = A(W_t^* - t)$, and for any $u > 0$,

$$\begin{aligned}\mathbf{P}(-\ln Z_t > u) &= \mathbf{P}(W_t^* - t > A^{-1}u) = \mathbf{P}(W_t^* - t > \lfloor A^{-1}u\rfloor) \leq a\exp[-b\lfloor A^{-1}u\rfloor] \\ &\leq a\exp[-b(A^{-1}u - 1)] = r\exp[-du],\end{aligned}$$

where $r := ae^b$ and $d := bA^{-1}$. □

Stationary Case Let now the initial distribution be stationary. Then it is convenient to embed X into a two-way infinite stationary hidden Markov process $\{X_t\}_{t=-\infty}^{\infty}$. Now, besides the stopping time W_t^* we can also define the time U_t^* as follows:

$$U_t^* := \max\{u < t - r : X_u^{u+r} \in \mathcal{X}_o^{r+1}\}.$$

Thus, $U_t = U_t^* \vee 1$, so that $U_t \geq U_t^*$. Proposition 2.2, possibly with some other constants, holds also for $t - U_t^*$. Therefore, for any t, the random variable $W_t^* - U_t^*$ has an exponentially decreasing tail:

$$\mathbf{P}(W_t^* - U_t^* > k) = \mathbf{P}\left((W_t^* - t) + (t - U_t^*) > k\right)$$

$$\leq \mathbf{P}(W_t^* - t > k/2) + \mathbf{P}(t - U_t^* > k/2) \leq a_o e^{-b_o k},$$

where a_o and b_o are some positive constants. Thus, we have the following lower bound.

Corollary 2.3 *Assume* **A1** *and let the initial distribution π be stationary. Let $\{X_t\}_{t=-\infty}^{\infty}$ be the bi-infinite embedding of X^n. Then*

$$\mathbf{P}(Y_t = v_t(X^n)|X^n) \geq Z_t, \tag{2.5}$$

where Z_t, $t = 1, \ldots, n$, are $\sigma(\{X_t\}_{t=-\infty}^{\infty})$-measurable identically distributed random variables such that $-\ln Z_t$ has an exponential tail, that is for some positive constants r and d and for every $u > 0$, $\mathbf{P}(-\ln Z_t > u) \leq r\exp[-du]$. Hence, $E[-\ln Z_t] < \infty$.

Proof From Eq. 2.4 it follows that

$$\mathbf{P}(Y_t = v_t(X^n)|X^n) \geq c\exp[-A(W_t^* - U_t^*)] =: Z_t.$$

By stationarity, the random variables Z_t are identically distributed. The rest of the proof is the same as the one of Corollary 2.2. □

Recall the definiton of the accuracy A_n given in Eq. 1.2. For a stationary chain, thus, inequality (2.5) gives an upper bound on the probability that the accuracy of the Viterbi path is less than $\alpha \in (0, 1)$. Indeed, with $M := E[-\ln Z_t] < \infty$, we obtain

$$\mathbf{P}\left(A_n(v) \leq \alpha\right) \leq \mathbf{P}\left(\frac{1}{n}\sum_{t=1}^{n} Z_t \leq \alpha\right) = \mathbf{P}\left(-\ln\left(\frac{1}{n}\sum_{t=1}^{n} Z_t\right) \geq -\ln\alpha\right)$$

$$\overset{(Jensen)}{\leq} \mathbf{P}\left(\frac{1}{n}\sum_{t=1}^{n}(-\ln Z_t) \geq \ln\frac{1}{\alpha}\right) \overset{(Markov)}{\leq} \frac{E\left[-\ln Z_t\right]}{\ln\left(\frac{1}{\alpha}\right)} = \frac{M}{\ln\left(\frac{1}{\alpha}\right)}.$$

3 Iterative Algorithm

Recall that we aim to improve the accuracy of the Viterbi path. Since the accuracy is just the mean of classification probabilities, a straightforward idea for doing this is to find the time points with lowest classification probabilities, replace them by the PMAP states (or by the true states when probing the true states is possible) and replace the original Viterbi path by the constrained Viterbi path. As explained in the introduction, such a batch approach has a big drawback, since typically the time points with low classification probabilities are situated next to each other and substituting a number of consecutive states with the corresponding PMAP states can make the adjusted path inadmissible. The following iterative algorithm ensures that the adjusted path remains admissible.

3.1 Description of the Iterative Algorithm

Input: observations x^n, a threshold parameter δ, $0 < \delta < \frac{1}{K}$, and the maximum number of iterations M.

Initialization: find the Viterbi path $v(x^n)$ and calculate the classification probabilities

$$\rho_t^{(0)} := \mathbf{P}(Y_t = v_t | X^n = x^n), \quad t = 1, \dots, n.$$

Define $v^* := v$.

For $m = 1, \dots, M$ **do**: if $\min_t \rho_t^{(m-1)} \geq \delta$, then quit, else

1) find the time point t_m with lowest conditional classification probability and the state w_m that maximizes the corresponding conditional classification probability:

$$t_m := \arg\min\{\rho_t^{(m-1)} : t = 1, \dots, n\},$$

$$w_m := \arg\max_{s \in S} \mathbf{P}(Y_{t_m} = s | X^n = x^n; Y_{t_i} = w_i, i = 1, \dots, m-1);$$

2) let $S^n(m) := \{s^n \in S^n : s_{t_1} = w_1, \dots, s_{t_m} = w_m\}$, find the new constrained Viterbi path $v^{(m)}$,

$$v^{(m)} := \arg\max_{s^n \in S^n(m)} \mathbf{P}(Y^n = s^n | X^n = x^n)$$

$$= \arg\max_{s^n} \mathbf{P}(Y^n = s^n | X^n = x^n; Y_{t_i} = w_i, i = 1, \dots, m),$$

define $v^* := v^{(m)}$;

3) calculate the new conditional classification probabilities $\rho_t^{(m)}$,

$$\rho_t^{(m)} := \mathbf{P}(Y_t = v_t^{(m)} | X^n = x^n; Y_{t_i} = w_i, i = 1, \dots, m), \quad t = 1, \dots, n. \quad (3.1)$$

Output: the path $v^*(x^n)$.

In the algorithm described above, at first the time point t_1 with the lowest classification probability is found. Then, at this point, the state w_1 with maximum posterior probability—the PMAP state—is found. The state w_1 at time point t_1 is taken as if it were the true state, and in all what follows, only the paths passing w_1 at t_1 are considered. The conditional classification probabilities in the next step are computed given the event $\{Y_{t_1} = w_1\}$. The time t_2 has the smallest conditional classification probability and the state w_2 is the state that at t_2 has the maximum posterior probability given $\{Y_{t_1} = w_1\}$. This means that the probability $\mathbf{P}(Y_{t_1} = w_1, Y_{t_2} = w_2 | X^n = x^n)$ is strictly positive, thus the algorithm guarantees that the path remains admissible, i.e. it has positive posterior probability. In what follows, the states w_1 and w_2 at time points t_1 and t_2 are taken as if they were the true states, and all probabilities are calculated conditional on $\{Y_{t_1} = w_1, Y_{t_2} = w_2\}$. Therefore, the output v^* is always

of positive probability. Moreover, this probability is decreasing (non-increasing) with m due to the increasing constraints.

As explained in the introduction, the other problem with the batch approach is that replacing the states with low classification probability by the PMAP states can change the path, so that the classification probabilities of the constrained Viterbi path can drop below the threshold somewhere else. As the example in the next subsection shows, this can indeed be the case. The iterative algorithm does not necessarily exclude such a possibility, but we have a reason to believe that such a phenomenon is less likely to happen with the iterative approach. The reasoning is as follows. As is shown in Lember and Koloydenko (2008, 2010) and Koloydenko and Lember (2008), (under some conditions) the influence of changing the Viterbi path is local. This means that (with high probability) there exist time points $1 = u_0 < u_1 < u_2 < \cdots < u_k = n$, such that if $t \in (u_{j-1}, u_j)$, then forcing the path to pass a prescribed state at time t changes the Viterbi path in the range (u_{j-1}, u_j) only (see also Lember et al. 2011). Hence all classification probabilities outside the piece remain unchanged.

The piecewise structure of the Viterbi path also (at least partially) explains why the iterative algorithm achieves the same performance as the batch algorithm with a considerably smaller number of replacements. Suppose the classification probability $\mathbf{P}(Y_t = v_t | X^n = x^n)$ is very low. Then as explained before, due to the "inertness" of the Viterbi path, it is often low also for the neighbors, meaning that the segment (u_{j-1}, u_j) containing t is somehow abnormal. However, due to the same inertness, changing the path at t changes it also in the neighborhood, so that the classification probabilities of $v^{(1)}$ are now bigger not only at t but also in the neighborhood. In particular, it might happen that the whole abnormal segment will be adjusted with only one replacement. If there is now another abnormal segment (u_{l-1}, u_l) $(l \neq j)$, then the previous changes do not influence the Viterbi path in that segment, so that at some $t_2 \in (u_{l-1}, u_l)$, the (unconditional) classification probability of $v^{(1)}$ is still below threshold. The question is whether the algorithm still finds t_2, since it uses the conditional (given $\{Y_{t_1} = w_1\}$) smoothing probabilities. However, for many models the smoothing probabilities $\mathbf{P}(Y_t = s | X^n)$ have the so-called exponential forgetting properties (Lember et al. 2011a, b; Gerencsér and Molnár-Sáska 2002; Le Gland and Mevel 2000), implying that for some constant $0 < \rho_o < 1$, for a non-negative finite constant C (depending on $X^n = x^n$) and for any state s,

$$\left|\mathbf{P}(Y_{t_2} = s | X^n = x^n) - \mathbf{P}(Y_{t_2} = s | X^n = x^n, Y_{t_1} = w_1)\right| \leq C\rho_o^{|t_1 - t_2|}.$$

This inequality implies that when t_1 and t_2 are sufficiently far from each other, then conditioning on $\{Y_{t_1} = w_1\}$ does not influence much the classification probability at t_2, and the algorithm finds the next abnormal segment. For a similar result, see Corollary 2.1 in Lember (2011b).

If state probing is possible, then instead of revealing a batch of true states at once, one can also perform state probing iteratively. Although (computationally) more costly, the iterative way of adjusting the Viterbi path has several advantages over the batch approach. The iterative algorithm tends to adjust the Viterbi path piecewise. Since the number of abnormal segments is usually smaller than the number of time points with low classification probability, the number of replacements (iterations) needed to reach a certain improvement is considerably smaller for the iterative approach compared to the batch approach.

Choosing the Threshold Parameter The performance of the iterative as well as of the batch approach depends on the chosen threshold parameter δ. This parameter can be regarded as a regularization parameter that controls the trade-off between the Viterbi and PMAP path.

The bigger δ is, the closer the adjusted path is to the PMAP path. The choice of δ depends on the concrete segmentation problem and the underlying model. Selection of the threshold parameter can for example be related to the size of improvement we aim to achieve in the accuracy of the adjusted path. With the iterative algorithm, one can fix the number of iterations instead of δ. This is especially good in the case of revealing true states that usually costs a lot. To make a suitable choice for δ, the information about the underlying model should be combined with given data. For the given data, the threshold parameter can for example be chosen based on percentiles of the empirical distribution of the calculated classification probabilities.

3.2 Comparison of the Batch and Iterative Approaches

3.2.1 A Case Study

In this example, we consider a model that is used in Lember and Koloydenko (2014) for illustrating the task of predicting protein secondary structure in single amino-acid sequences. The underlying Markov chain has six possible states. The transition matrix and initial distribution are as follows:

$$\mathbb{P} = \begin{pmatrix} 0.8360 & 0.0034 & 0.1606 & 0 & 0 & 0 \\ 0.0022 & 0.8282 & 0.1668 & 0.0028 & 0 & 0 \\ 0.0175 & 0.0763 & 0.8607 & 0.0455 & 0 & 0 \\ 0 & 0 & 0 & 0.7500 & 0.2271 & 0.0229 \\ 0 & 0 & 0 & 0 & 0.8450 & 0.1550 \\ 0 & 0.0018 & 0.2481 & 0 & 0 & 0.7501 \end{pmatrix},$$

$$\pi = (0.0016, 0.0041, 0.9929, 0.0014, 0, 0)'.$$

Many transitions are impossible and this can make a PMAP sequence inadmissible. The observations come from a 20-symbol emission alphabet of amino-acids, the emission matrix is given in the Appendix. In order to compare the batch approach and the iterative approach, we have generated an observation sequence (together with the underlying Markov chain) of length $n = 1000$ from this model. Furthermore, to understand better the behaviour of the studied algorithms for a given observation sequence, we have re-sampled from the posterior state sequence distribution. Thus, the comparisons of the two approaches for both PMAP replacements and when revealing true states is possible are based on one hundred underlying state sequences.

To compare the behaviour of the batch and the iterative algorithm, we provide for both algorithms tables with some summary characteristics that have been calculated for different numbers of replacements or iterations m, respectively. In Tables 1, 2, 3 and 4, the simulation results for the generated observation sequence are presented for two different state sequences: y_{typ} and y_{atyp}. The sequence y_{atyp} represents atypical state sequences, where the number of classification errors rises due to PMAP replacements to a higher level compared to the unconstrained Viterbi path. In the tables, *Errors* denotes the real number of classification errors and *E(Errors)* the expected number of classification errors, *mean(Errors)* is the average number of classification errors over the hundred state sequences, $\rho_{min}^{uncond} := \min_t \mathbf{P}(Y_t = v_t^{(m)} | X^n = x^n)$ and $\rho_{min}^{cond} := \min_t \rho_t^{(m)}$ (see Eq. 3.1) give respectively the

Table 1 PMAP replacements with the batch algorithm

m	Errors		mean(Errors)	E(Errors)	ρ_{min}^{uncond}	Log-likelihood
	y_{typ}	y_{atyp}				
0	583	481	544.22	544.02	0.0113	−168.18
1	572	452	528.59	528.08	0.0279	−168.58
2	572	452	528.59	528.08	0.0279	−168.58
3	572	452	528.59	528.08	0.0279	−168.58
4	572	452	528.59	528.08	0.0279	−168.58
5	572	452	528.59	528.08	0.0279	−168.58
10	561	449	523.07	*522.16*	0.0437	−169.44
15	555	445	522.90	521.83	0.0437	−172.13
20	555	445	522.90	521.83	0.0437	−172.13
25	556	445	522.86	521.81	0.0437	−172.18
30	556	445	522.86	521.81	0.0437	−172.18
35	558	433	518.88	519.42	0.0448	−172.50
40	554	429	514.92	516.38	0.0448	*−172.80*
50	*550*	455	506.81	508.27	0.0448	−175.39
60	*550*	461	503.90	504.72	0.0448	−177.65
70	529	487	494.59	*496.20*	0.0448	−177.89
77	525	483	492.74	494.09	0.0448	−178.90
78	525	483	492.58	493.91	0.0448	$-\infty$
140	517	486	487.20	488.49	0.0448	$-\infty$

Table 2 PMAP replacements with the iterative algorithm

m	Errors		mean(Errors)	E(Errors)	ρ_{min}^{cond}	ρ_{min}^{uncond}	Log-likelihood
	y_{typ}	y_{atyp}					
0	583	481	544.22	544.02	0.0113	0.0113	−168.18
1	572	452	528.59	528.08	0.0279	0.0279	−168.58
2	559	451	523.15	522.52	0.0437	0.0437	−169.37
3	561	439	519.17	520.13	0.0439	0.0448	−169.69
4	*550*	428	513.91	514.73	0.0103	0.0458	*−171.28*
5	546	433	511.46	512.24	0.0453	0.0458	−172.64
10	542	452	499.51	*500.55*	0.0451	0.0576	−176.19
15	*532*	*458*	496.96	*497.38*	0.0459	0.0608	−179.16
18	510	485	486.15	486.65	*0.1094*	*0.1094*	−181.85
77	472	498	478.70	480.73	0.2779	0.0947	−215.19
78	475	502	479.43	481.32	0.3105	0.0947	−215.38

Table 3 State probing with the batch algorithm

m	Errors		E(Errors)		ρ_{min}^{cond}		Log-likelihood	
	y_{typ}	y_{atyp}	y_{typ}	y_{atyp}	y_{typ}	y_{atyp}	y_{typ}	y_{atyp}
0	583	481	544	544	0.0113	0.0113	−168.18	−168.18
1	572	452	527	527	0.0279	0.0279	−168.58	−168.58
2	572	485	527	516	0.0279	0.0319	−168.58	−170.52
3	572	485	527	515	0.0279	0.0319	−168.58	−170.52
4	559	451	516	527	0.0437	0.0238	−169.37	−175.24
5	559	450	516	524	0.0437	0.0238	−169.37	−175.29
10	543	442	510	512	0.0437	0.0437	−174.23	−181.32
15	536	436	508	506	0.0437	0.0437	−177.26	−187.53
20	536	435	508	505	0.0437	0.0437	−177.26	−187.92
25	536	429	505	503	0.0437	0.0437	−177.26	−189.43
30	536	429	503	501	0.0437	0.0437	−177.26	−189.43
35	526	416	493	491	0.0458	0.0439	−178.57	−190.28
40	517	423	486	483	0.0394	0.0394	−179.92	−191.26
50	504	415	463	457	0.0555	0.0447	−183.69	−192.12
60	486	408	453	450	0.0580	0.0452	−193.48	−193.28
70	465	406	437	435	0.0580	0.0452	−194.28	−193.50
77	463	404	433	429	0.0580	0.0452	−195.13	−194.61
78	463	404	432	429	*0.0580*	*0.0452*	−195.13	−194.61
140	423	369	396	383	0.0571	0.1094	−213.11	−215.54

minimum unconditional and conditional classification probability for the constrained path after m replacements/iterations, and *Log-likelihood* gives the logarithm of the posterior probability of the constrained path. Observe that $m = 0$ gives the characteristics for the unconstrained paths.

PMAP Replacements Suppose that the threshold parameter δ is set to 0.1. There are 140 classification probabilities smaller than 0.1 for the Viterbi path of this sequence. Using the batch algorithm would mean that we substitute the states corresponding to these 140 low probabilities with the respective PMAP states and find then the constrained Viterbi path. From Table 1 we can see that the posterior probability of the constrained path is zero. The posterior probability of the constrained Viterbi will be zero after 78 replacements. This depends on replacement of many consecutive states. Indeed, all the states from time point 712 to 754, except at 728, are substituted, whereas from 753 to 754 we obtain an inadmissible transition $3 \rightarrow 5$. If we would use the iterative algorithm with the same threshold instead, we would stop after 18 iterations because $\min_t \rho_t^{(18)} = 0.1094$. The 11 lowest unconditional classification probabilities for the constrained paths are:

1) Batch 0.0448, 0.0449, 0.0474, 0.0506, 0.0558, 0.0655, 0.0671, 0.0771, 0.0880, 0.0944, 0.1018;
2) Iterative 0.1094, 0.1149, 0.1184, 0.1227, 0.1247, 0.1276, 0.1305, 0.1383, 0.1426, 0.1428, 0.1460.

Table 4 State probing with the iterative algorithm

m	Errors		E(Errors)		ρ_{min}^{cond}		Log-likelihood	
	y_{typ}	y_{atyp}	y_{typ}	y_{atyp}	y_{typ}	y_{atyp}	y_{typ}	y_{atyp}
0	583	481	544	544	0.0113	0.0113	−168.18	−168.18
1	572	452	527	527	0.0279	0.0279	−168.58	−168.58
2	556	448	514	514	0.0437	0.0437	−170.36	−170.36
3	547	436	507	506	0.0458	0.0439	−171.18	−170.69
4	540	*423*	501	495	0.0484	0.0458	−172.28	−173.18
5	526	*430*	492	488	0.0506	0.0484	−173.88	−174.16
10	484	422	463	445	0.0846	0.1256	−180.70	−179.06
15	*453*	*395*	441	429	*0.1428*	*0.1152*	−189.00	−183.16
18	446	393	432	414	0.1493	0.1435	−192.57	−183.33
77	310	299	314	299	0.2879	0.3146	−245.89	−228.85
78	307	298	313	298	0.3084	0.2970	−246.53	−228.97

We can see that in the case of the batch algorithm, after fixing the preliminary set of 140 time points, the classification probability has dropped below δ for ten time points. For the iterative algorithm, all the probabilities are above the threshold.

In the case of PMAP replacements, only Errors depends on the realization of the underlying hidden Markov chain. For both the batch and the iterative algorithm, the average number of classification errors over 100 state sequences is close to the expected number of errors (see mean(Errors) and E(Errors) in Tables 1 and 2). For PMAP replacements, there are 16 and 19 state sequences out of 100, where the number of classification errors for the batch and the iterative algorithm is higher for some m considered in the tables than for the unconstrained Viterbi. We call these paths "atypical" as opposed to the remaining set of "typical" paths. In the case of the batch algorithm, for most of atypical cases, the number of errors decreases at first, then it increases and then starts to decrease again as the number of replacements/iterations continues to grow. In the case of the iterative algorithm, for some of these 19 atypical sequences the number of errors fluctuates up and down, whereas for other sequences it increases at first and then starts to decrease. In Tables 1 and 2 we can see how for a fixed atypical path y_{atyp} the number of classification errors decreases at first, but then it starts to increase again, whereas for a fixed typical path y_{typ} it has a decreasing trend. Consider the number of errors 550 for y_{typ} in Tables 1 and 2. The iterative algorithm reaches this number after four iterations. To obtain the same error rate with the batch algorithm, we need to make between 40 and 50 replacements. The posterior probability of the constrained Viterbi path after four iterations is higher compared to the posterior probability of the constrained sequence obtained after 40 substitutions with the batch algorithm (log-posterior probabilities are −171.28 and −172.80, respectively). This shows that the iterative algorithm is more effective since it works piecewise. If we would use the batch algorithm with four replacements, the replacements would occur at time points 723, 724, 725 and 733, which give the four lowest classification probabilities. This means that we would make adjustments at three consecutive time points. With the iterative algorithm, the substitutions would be made at 723, 752, 582 and 557. Thus, with the iterative algorithm the available information for making adjustments is used more efficiently by fixing the problematic segments in turn.

Observe that E(Errors) is just $n(1-A_n)$. For the Viterbi path this number is 544. The best possible expected number of errors, which corresponds to the PMAP path, is 459. Again, to reach a certain decrease in the expected number of errors, the iterative algorithm needs a smaller number of replacements than the batch algorithm. After ten replacements/iterations, for example, E(Errors) is 522 (batch) and 501 (iterative). To achieve E(Errors) $=$ 497, 15 iterations are needed, whereas the batch algorithm requires about 70 replacements. The decrease from 544 to 497 might not seem that big, but one should take into consideration that the maximum possible improvement is $544 - 459 = 85$. Hence, the improvement $544 - 497 = 47$ that the iterative algorithm achieves with 15 replacements is more than half of the possible improvement.

State Probing Tables 3 and 4 compare the batch and the iterative approach in the case of state probing. In this case, we take into account the additional information obtained when revealing states. Thus, E(Errors) is calculated with the help of conditional classification probabilities:

$$\mathrm{E(Errors)} = n - \sum_{t=1}^{n} \mathbf{P}(Y_t = v_t^{(m)} | X^n = x^n, Y_{t_1} = y_{t_1}, \ldots, Y_{t_m} = y_{t_m}). \tag{3.2}$$

Observe that when revealing true states, all the characteristics in Tables 3 and 4 depend on the underlying state sequence. Again, the iterative algorithm is more efficient than the batch algorithm. After 78 replacements with the batch approach, the minimum (conditional) classification probability for the constrained sequences is still 0.0580 and 0.0452 for the state sequences y_{typ} and y_{atyp}, respectively. For iterative state probing, these probabilities are 0.1428 and 0.1152 after 15 iterations. With the batch algorithm, the minimum conditional classification probability ρ_{min}^{cond} has reached the threshold level $\delta = 0.1$ after 140 substitutions for 45 sequences. With the iterative algorithm, this threshold level has been mainly reached after 15 iterations for these 45 sequences (10 iterations [13 sequences]; 15 [29]; 18 [1]; 20 [1]; 25 [1]). For both y_{typ} and y_{atyp}, the first replacement has a positive effect: the number of errors decreases from 583 and 481 to 572 and 452, respectively (apparently a whole segment is corrected). But the subsequent replacements with the batch method give an additional decrease in the number of errors that is generally smaller than the increase in the number of replacements m, or the subsequent replacements have no effect or a negative effect (that is, causing additional errors). As Table 4 shows, adjusting the path iteratively is much more efficient in this sense (especially for y_{typ}), since additional replacements after the first one decrease the number of errors by more than the number of additional replacements itself. The number of errors for y_{atyp} in the case $m = 4$ and $m = 5$ in Table 4 shows that iterative state probing can also have a negative effect. There are seven and six state sequences out of one hundred for the batch and the iterative algorithm respectively, where the number of classification errors increases for some m to the same level as for the unconstrained Viterbi. We can also study the effect of the iterative approach when states are substituted with the corresponding PMAP states or true states. Tables 2 and 4 show that after 15 iterations for example, the constrained sequences have 532 and 458 errors when replacements are done with the PMAP states, and 453 and 395 errors when replacements are done with the true states, respectively. Note that E(Errors) might increase with m (see Table 3). For the batch algorithm, there are 46 such sequences, whereas for the iterative algorithm only 6. Thus, that E(Errors) increases with m is not a typical behaviour of the iterative state probing. We shall address this issue more closely in Section 4.

Table 5 PMAP replacements: mean behaviour of the constrained paths for the batch algorithm

δ	Replacements	Errors	E(Errors)	ρ_{min}^{uncond}	Log-likelihood
0.20	7.5 (5.0)	341	344	0.1911	−107.38
0.25	19.7 (9.4)	338	340	0.1823	−109.69
0.30	39.1 (14.6)	340	340	0.1616	−112.65

3.2.2 Behaviour of the Batch and the Iterative Approach Depending on the Threshold

In this example, we consider the following two-state hidden Markov model. The transition matrix and initial probabilities are given by

$$\mathbb{P} = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}, \qquad \pi' = (0.5, 0.5),$$

and the emission distributions are given by $\mathcal{N}(0, 1)$ and $\mathcal{N}(0.5, 1)$. Thus, the underlying Markov chain is stationary. We have generated 100 observation sequences (together with the underlying state paths) of length $n = 1000$ from this HMM and studied the mean behaviour of the constrained Viterbi sequences for different threshold parameters δ. We study threshold-based adjustments. For the batch approach this means that for all the time points with classification probability lower than δ, the Viterbi state is substituted with the corresponding PMAP state (or if probing the true state is possible, with the true state), and thereafter constrained segmentation is performed. In the case of the iterative algorithm, replacements are based on conditional classification probabilities and performed iteratively. For every constrained path, we have calculated the real number of classification errors, the expected number of classification errors, the minimum conditional and unconditional classification probability, and the log-likelihood of the constrained Viterbi path. The mean values of these characteristics over the hundred replicates for the unconstrained Viterbi are as follows: 350, 354, 0.15 and −105.8. The average values of the characteristics (over 100 replicates) for the constrained sequences are given in Tables 5, 6, 7 and 8. The average number of substitutions made and the standard deviation of substitutions can be seen in columns *Replacements* and *Iterations* for the batch and the iterative algorithm, respectively. The average number of PMAP errors for the studied sequences is 306.

Compare the batch and the iterative algorithm for $\delta = 0.25$, for example. On average, there are 20 classification probabilities lower than 0.25. After substituting the states with low classification probability according to the batch algorithm, the average minimum classification probability for the constrained Viterbi paths is 0.18 and the average number of classification errors is 338. For the iterative algorithm with the same threshold, we need 7 iterations on average. The average minimum classification probability for the constrained paths is 0.26, which is above the threshold, and the average number of classification errors is 327. This demonstrates that the iterative algorithm is more efficient.

Table 6 PMAP replacements: mean behaviour of the constrained paths for the iterative algorithm

δ	Iterations	Errors	E(Errors)	ρ_{min}^{uncond}	ρ_{min}^{cond}	Log-likelihood
0.20	3.3 (2.0)	336	341	0.2181	0.2182	−107.67
0.25	7.4 (3.4)	327	330	0.2626	0.2633	−110.83
0.30	13.9 (4.9)	321	321	0.3066	0.3093	−115.62

Table 7 State probing: mean behaviour of the constrained paths for the batch algorithm

δ	Replacements	Errors	E(Errors)	ρ_{min}^{cond}	Log-likelihood
0.20	7.5 (5.0)	335	339	0.1949	−107.84
0.25	19.7 (9.4)	324	325	0.1932	−111.51
0.30	39.1 (14.6)	307	310	0.1782	−117.14

In the same way, we can compare the threshold-based adjustment procedure for the batch and the iterative algorithm in the case of probing the true states. To take into account the information obtained through revealing states, we consider conditional probabilities when calculating the classification probabilities and the expected number of classification errors for the constrained Viterbi paths.

Consider again $\delta = 0.25$. When using the batch algorithm, we would need to probe the true state at 20 time points on average, whereas with the iterative algorithm the average number of state probings would be 7. For the batch algorithm, the mean minimum classification probability for the constrained sequences is 0.19, which is below the threshold, and the average number of errors is 324. The same characteristics in the case of iterative state probing are 0.26 and 319, respectively.

To summarize, both our simulation studies demonstrate that the iterative algorithm is more efficient for adjusting Viterbi paths. We need fewer replacements to achieve a certain decrease in the number of errors and the constrained sequences behave better when obtained iteratively.

4 Unsuccessful State Probing

Recall Table 3 and E(Errors) for y_{atyp}. The number of expected errors for $m = 4$ is bigger than for $m = 3$ (527 and 515, respectively). This means that probing the true state at four points is worse than doing it at three points—an additional state probing at t_4 has a negative effect. According to Eq. 3.2, E(Errors) when m hidden states are revealed is conditional on x^n as well as on $y_{t_1}, \ldots y_{t_m}$, implying that the negative effect we see in this particular example might be due to "bad" value of Y_{t_4} that in our simulations happens to be very atypical. When we take the expectation over Y_{t_4}, the average effect can still be positive, because the atypical value has very small probability and for the rest of the values everything is normal. This speculation raises the following question: is it possible to probe the true state at some fixed time point t_1 so that E(Errors) increases also when averaging over Y_{t_1}? Formally, the question is the following: do there exist an HMM, a sequence of observations x^n having a positive likelihood and a fixed time point t_1 such that the following inequality holds:

Table 8 State probing: mean behaviour of the constrained paths for the iterative algorithm

δ	Iterations	Errors	E(Errors)	ρ_{min}^{cond}	Log-likelihood
0.20	3.2 (1.9)	333	337	0.2192	−107.34
0.25	6.9 (3.2)	319	322	0.2640	−109.55
0.30	12.2 (4.1)	304	306	0.3102	−112.79

$$\sum_{t=1}^{n} \mathbf{P}(Y_t = v_t(x^n)|X^n = x^n) > \sum_{t=1}^{n} \mathbf{P}(Y_t = v_t^{(1)}(x^n, Y_{t_1})|X^n = x^n)? \tag{4.1}$$

Here $v^{(1)}$, as previously, stands for the constrained Viterbi path given the value of Y_{t_1}. Inequality (4.1) states that the accuracy of the unconstrained Viterbi path is strictly bigger than that of the constrained Viterbi path after probing Y_{t_1}, i.e. $A_n(v) > A_n(v^{(1)})$. In what follows, we present an example showing that such an *unsuccessful state probing* is possible and Eq. 4.1 can happen.

The Model and Observations Consider a 3-state HMM with the transition matrix

$$\mathbb{P} = \begin{pmatrix} \frac{2}{3}(1-\epsilon) & \frac{2}{3}\epsilon & \frac{1}{3} \\ \frac{2}{3}\epsilon & \frac{2}{3}(1-\epsilon) & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix},$$

where $0 < \epsilon < \frac{1}{4}$, implying that $\frac{2}{3}(1-\epsilon) > \frac{1}{2}$. Let the initial distribution be stationary, i.e.

$$\pi_1 = \frac{3}{5}\left(\frac{1+2\epsilon}{1+4\epsilon}\right), \quad \pi_2 = \frac{6}{5}\left(\frac{\epsilon}{1+4\epsilon}\right), \quad \pi_3 = \frac{2}{5}.$$

Let $\delta > 0$ be so small that

$$(1+\delta)\epsilon < (1-\epsilon) \tag{4.2}$$

and let $m \in \mathbb{N}$ be big enough (to be specified later). Suppose $x, y, z, a \in \mathcal{X}$ are such that

1) $f_1(x) = 1$ and $f_2(x) = f_3(x) = 0$;
2) $f_2(y) = 1 + \delta$ and $f_1(y) = f_3(y) = 1$;
3) $f_3(a) = 0$, $f_1(a) = f_2(a) = 1$;
4) $f_1(z) = f_2(z) = f_3(z) = 1$.

Let the observations $x_1, \ldots, x_n$ be as follows: $n = m + 2$ and

$$x_1 = x, \quad x_2 = y, \quad x_3 = x_4 = \cdots = x_m = z, \quad x_{m+1} = a, \quad x_{m+2} = x.$$

Viterbi Path By condition 1), all the state paths with positive posterior probability begin and end in state 1. From Eq. 4.2 it follows that

$$\left(\frac{2}{3}(1-\epsilon)\right)^{m+1} > \left(\frac{2}{3}(1-\epsilon)\right)^{m-1} \left(\frac{2}{3}\epsilon\right)^2 (1+\delta),$$

implying that

$$\mathbf{P}(Y_1 = \cdots = Y_n = 1|X^n = x^n) > \mathbf{P}(Y_1 = 1, Y_2 = \cdots = Y_{n-1} = 2, Y_n = 1|X^n = x^n).$$

From $\frac{2}{3}(1-\epsilon) > \frac{1}{2}$ it follows that the posterior probability to remain in state 1 is bigger than jumping from state 1 to state 3, remaining then there and jumping thereafter back to state 1. Formally, for any $1 \le k < l < m+1$,

$$\begin{aligned} &\mathbf{P}(Y_1 = \cdots = Y_n = 1 \big| X^n = x^n) \\ &> \mathbf{P}\left(Y_1 = \cdots = Y_k = 1, Y_{k+1} = \cdots = Y_l = 3, Y_{l+1} = \cdots = Y_n = 1 \big| X^n = x^n\right). \end{aligned}$$

This means that the Viterbi path remains in state 1 all the time.

Constrained Viterbi Path We now take $t_1 := m + 1 = n - 1$. Thus, we will look at the value of Y_{n-1}. Since by 3), $\mathbf{P}(Y_{n-1} = 3|X^n = x^n) = 0$, the constrained Viterbi path will

differ from the original one only if $Y_{n-1} = 2$. Let us find the constrained Viterbi path given it passes state 2 at time $n-1$, i.e., let us find

$$\begin{aligned} v^{(1)}(x^n, 2) &= \arg\max_{s^n} \mathbf{P}(Y^n = s^n | X^n = x^n, Y_{n-1} = 2) \\ &= \arg\max_{s^n : s_{n-1}=2} \mathbf{P}(Y^n = s^n | X^n = x^n). \end{aligned}$$

Because of condition 2) it follows that for any $k > 2$,

$$\begin{aligned} &\mathbf{P}(Y_1 = 1, Y_2 = \cdots = Y_{n-1} = 2, Y_n = 1 | X^n = x^n) \\ &> \mathbf{P}(Y_1 = \cdots = Y_{k-1} = 1, Y_k = \cdots = Y_{n-1} = 2, Y_n = 1 | X^n = x^n). \end{aligned}$$

Secondly, note that leaving state 2 for either state 1 or 3 and returning to state 2 afterwards (since the constrained path has to do so) is less probable than remaining in state 2. Hence, when the constrained path is in state 2 at some time $t < m+1$, then it remains in state 2 until time $m+1$. Also, jumping from state 1 to state 3 at time 2, remaining there for a while and jumping to state 1 (since the transition from 3 to 2 is forbidden), say at time k, is less likely than remaining in state 1 from time 1 to k. Therefore, $v^{(1)}(x^n, 2)$ is constantly in state 2 except for the times 1 and n, when it is in state 1. Thus, if $Y_{n-1} = 2$, then the Viterbi and constrained Viterbi path differ at every time from 2 to $n-1$: the Viterbi stays in 1 and the constrained Viterbi stays in 2.

Checking Eq. 4.1 Since given our data, Y_{t_1} can take on two values only, we have for every $t = 1, \ldots, n$,

$$\begin{aligned} \mathbf{P}(Y_t = v_t^{(1)}(x^n, Y_{t_1}) | X^n = x^n) = \sum_{s=1}^{2} &\mathbf{P}(Y_t = v_t^{(1)}(x^n, s) | X^n = x^n, Y_{t_1} = s) \\ &\times \mathbf{P}(Y_{t_1} = s | X^n = x^n). \end{aligned}$$

On the other hand, obviously

$$\mathbf{P}(Y_t = v_t(x^n) | X^n = x^n) = \sum_{s=1}^{2} \mathbf{P}(Y_t = v_t(x^n) | X^n = x^n, Y_{t_1} = s) \mathbf{P}(Y_{t_1} = s | X^n = x^n).$$

Because $v^{(1)}(x^n, 1) = v(x^n)$ and $\mathbf{P}(Y_{t_1} = 2 | X^n = x^n) > 0$, it immediately follows that inequality (4.1) holds if and only if

$$\sum_{t=1}^{n} \mathbf{P}(Y_t = v_t(x^n) | X^n = x^n, Y_{t_1} = 2) > \sum_{t=1}^{n} \mathbf{P}(Y_t = v_t^{(1)}(x^n, 2) | X^n = x^n, Y_{t_1} = 2). \tag{4.3}$$

Recall that $t_1 = n - 1 = m + 1$. Let for every $i = 1, 2, 3$,

$$Q_t(i) := \mathbf{P}(Y_t = i | X^n = x^n, Y_{n-1} = 2), \quad t = 1, \ldots, n.$$

With this notation, Eq. 4.3 holds if and only if

$$\sum_{t=2}^{m} Q_t(1) > 1 + \sum_{t=2}^{m} Q_t(2). \tag{4.4}$$

This is indeed so in our example. Let $\delta = 1$, consider $\epsilon = 0.2$ and $\epsilon = 0.01$. In Table 9, the values of the right-hand side and left-hand side of inequality (4.4) have been calculated for some values of m. Observe that already for $m = 7$, inequality (4.4) holds. The difference

Table 9 Comparison of accuracy before and after state probing

$\epsilon = 0.2$			$\epsilon = 0.01$		
m	$\sum_{t=2}^{m} Q_t(1)$	$\sum_{t=2}^{m} Q_t(2)+1$	m	$\sum_{t=2}^{m} Q_t(1)$	$\sum_{t=2}^{m} Q_t(2)+1$
3	0.79	2.09	3	0.77	2.14
5	1.80	2.46	5	1.82	2.66
6	2.29	2.58	6	2.38	2.79
7	2.78	2.70	7	2.96	2.87
98	45.26	14.82	98	56.52	4.00
998	465.26	134.82	998	586.13	14.38

$\sum_{t=2}^{m} Q_t(1) - \sum_{t=2}^{m} Q_t(2)$ grows with m and we will show that it can be made arbitrarily large.

The difference $\sum_{t=2}^{m} Q_t(1) - \sum_{t=2}^{m} Q_t(2)$ goes to infinity with m At first we will show that the probabilities $Q_t(i)$ can be calculated recursively. Let $\alpha_t(i)$ and $\beta_t(j)$ denote the usual forward and backward probabilities, i.e.

$$\alpha_t(i) = p(x^t, Y_t = i), \quad \beta_t(j) = p(x_{t+1}^n | Y_t = j).$$

Let

$$\gamma_{t_1,t_2}(i,j) := p(x_{t_1+1}^{t_2}, Y_{t_2} = j | Y_{t_1} = i).$$

Then for $t = 2, \ldots, n-2$, $Q_t(i)$ can be expressed as

$$Q_t(i) = \frac{\alpha_t(i)\gamma_{t,n-1}(i,2)\beta_{n-1}(2)}{\sum_i \alpha_t(i)\gamma_{t,n-1}(i,2)\beta_{n-1}(2)} = \frac{\alpha_t(i)\gamma_{t,n-1}(i,2)}{\sum_i \alpha_t(i)\gamma_{t,n-1}(i,2)}.$$

Observe that $Q_{n-1}(1) = 0$ and $Q_{n-1}(2) = Q_1(1) = Q_n(1) = 1$. The quantities $\gamma_{t_1,t_2}(i,j)$ can be seen as constrained backward probabilities. Because $x_3 = x_4 = \ldots = x_m = z$ and $f_1(z) = f_2(z) = f_3(z) = 1$, we can calculate the forward and constrained backward probabilities recursively. Let $u := \frac{2}{3}(1-\epsilon)$ and $v := \frac{2}{3}\epsilon$. Let for any t,

$$\alpha_t := (\alpha_t(1), \alpha_t(2), \alpha_t(3))',$$

and for any $t < n-1$,

$$\gamma_{t,n-1} := (\gamma_{t,n-1}(1,2), \gamma_{t,n-1}(2,2), \gamma_{t,n-1}(3,2))'.$$

Then the α-recursion is given as follows:

$$\alpha_2(1) = \pi_1 u\,, \quad \alpha_2(2) = \pi_1 v(1+\delta)\,, \quad \alpha_2(3) = \frac{\pi_1}{3},$$

and for any $t = 3, \ldots, m$,

$$\alpha_t' = \alpha_{t-1}'\mathbb{P}, \quad \text{thus} \quad \alpha_t' = \alpha_2'\mathbb{P}^{t-2}\,.$$

The recursion for the γ-probabilities is given as follows:

$$\gamma_{n-2,n-1}(1,2) = v, \quad \gamma_{n-2,n-1}(2,2) = u, \quad \gamma_{n-2,n-1}(3,2) = 0,$$

and for any $t = 2, \ldots, n-3$,

$$\gamma_{t,n-1} = \mathbb{P}^{n-2-t}\gamma_{n-2,n-1}.$$

Therefore, for any $t = 2, \ldots, m$,

$$Q_t(i) = \frac{\alpha_2' \mathbb{P}^{t-2} A_i \mathbb{P}^{n-2-t} \gamma_{n-2,n-1}}{\sum_i \alpha_2' \mathbb{P}^{t-2} A_i \mathbb{P}^{n-2-t} \gamma_{n-2,n-1}} = \frac{\alpha_2' \mathbb{P}^{t-2} A_i \mathbb{P}^{m-t} \gamma_{n-2,n-1}}{\alpha_2' \mathbb{P}^{m-2} \gamma_{n-2,n-1}},$$

where A_i is a 3×3 matrix having all entries zero except $a_{ii} = 1$. If $m \to \infty$, then

$$\mathbb{P}^m \to \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 \\ \pi_1 & \pi_2 & \pi_3 \\ \pi_1 & \pi_2 & \pi_3 \end{pmatrix} =: \mathbb{P}^\infty.$$

Hence, if t is large and $m - t$ is large as well, then

$$\alpha_t' \approx \alpha_2' \mathbb{P}^\infty, \quad \gamma_{t,n-1} \approx \mathbb{P}^\infty \gamma_{n-2,n-1},$$

so that

$$\alpha_t(s) \approx \left(\sum_i \alpha_2(i) \right) \pi_s, \quad \gamma_{t,n-1}(s,2) \approx \sum_i \gamma_{n-2,n-1}(i,2) \pi_i.$$

Hence, if t is far from the beginning and from the end, then

$$Q_t(s) \approx \frac{\left(\sum_i \alpha_2(i) \right) \pi_s \left(\sum_i \gamma_{n-2,n-1}(i,2) \pi_i \right)}{\left(\sum_i \alpha_2(i) \right) \left(\sum_i \gamma_{n-2,n-1}(i,2) \pi_i \right)} = \pi_s.$$

The argument above shows that

$$\frac{1}{m} \left(\sum_{t=2}^m Q_t(1) - \sum_{t=2}^m Q_t(2) \right) \to (\pi_1 - \pi_2), \tag{4.5}$$

and since $\pi_1 > \pi_2$, the difference $\sum_{t=2}^m Q_t(1) - \sum_{t=2}^m Q_t(2)$ can be arbitrarily large when choosing m big enough. Hence, given that m is big enough, in this example state probing has definitely a negative effect.

The Difference of the Accuracies In terms of accuracy and Q_t-variables, inequality (4.1) in our example becomes

$$n \left(A_n(v) - A_n(v^{(1)}) \right) = \mathbf{P}(Y_{n-1} = 2 | X^n = x^n) \sum_{t=2}^{n-1} (Q_t(1) - Q_t(2)).$$

Let us now show that there exists $\alpha_o > 0$ such that

$$\mathbf{P}(Y_{n-1} = 2 | X^n = x^n) \to \alpha_o.$$

Since $\alpha_{n-1}' = \alpha_{n-2}' \mathbb{P}$ and $\alpha_{n-2} \to \left(\sum_i \alpha_2(i) \right) \pi$ as $n \to \infty$, we have

$$\alpha_{n-1}(1) \to \left(\sum_i \alpha_2(i) \right) (\pi_1 u + \pi_2 v + \pi_3 \frac{1}{2}), \quad \alpha_{n-1}(2) \to \left(\sum_i \alpha_2(i) \right) (\pi_1 v + \pi_2 u).$$

Because $\alpha_{n-1}(3) = 0$, we therefore obtain

$$\mathbf{P}(Y_{n-1} = 2 | X^n = x^n) = \frac{\alpha_{n-1}(2) v}{\alpha_{n-1}(1) u + \alpha_{n-1}(2) v}$$

$$\to \frac{(\pi_1 v + \pi_2 u) v}{(\pi_1 u + \pi_2 v + \pi_3 \frac{1}{2}) u + (\pi_1 v + \pi_2 u) v} =: \alpha_o > 0.$$

The limit α_o is 0.066667 and 0.000198 for $\epsilon = 0.2$ and $\epsilon = 0.01$, for example. Therefore, from the convergence in Eq. 4.5 it follows that

$$A_n(v) - A_n(v^{(1)}) \to \alpha_o (\pi_1 - \pi_2) > 0.$$

Hence we can conclude that in our example the difference between the left- and right-hand side of Eq. 4.1 goes to infinity as n grows (even with linear rate), implying that the expected number of additional classification errors caused by unsuccessful state probing can be arbitrarily large.

Acknowledgments The authors would like to thank the anonymous referee for many comments and remarks that helped to improve the presentation of the article. The research of K. Kuljus was supported by Swedish University of Agricultural Sciences financing project KON 50210. The research of J. Lember was supported by Estonian Science Foundation grant no 9288 and targeted financing project SF 0180015s12.

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

Proof of Proposition 2.1

Let x^n and $t \in \{2, \ldots, n-1\}$ be fixed. Recall that $S = \{1, \ldots, K\}$. Let us bound $\gamma_t(s)$ for any state $s \in S$ from below and from above. Since

$$\gamma_t(s) = \sum_{s'} \sum_{s''} \alpha(x^{t-1}, s') p_{s's} f_s(x_t) p_{ss''} \alpha(s'', x_{t+1}^n), \tag{5.1}$$

we have

$$\begin{aligned}
\gamma_t(s) &\geq p(x^{t-1})(\min_{s'} p_{s's}) f_s(x_t)(\min_{s'} p_{ss'}) p(x_{t+1}^n), \\
\gamma_t(s) &\leq p(x^{t-1})(\max_{s'} p_{s's}) f_s(x_t)(\max_{s'} p_{ss'}) p(x_{t+1}^n).
\end{aligned}$$

Assume without loss of generality that the Viterbi path passes state 1 at time point t, that is $v_t = 1$. Let $v_{t-1} = a$ and $v_{t+1} = b$. Then for any other state $s \neq 1$ it holds that

$$p_{a1} f_1(x_t) p_{1b} \geq p_{as} f_s(x_t) p_{sb},$$

or equivalently,

$$f_1(x_t) \geq \left(\frac{p_{as}}{p_{a1}}\right) f_s(x_t) \left(\frac{p_{sb}}{p_{1b}}\right). \tag{5.2}$$

Let state $s \neq 1$ but arbitrary otherwise. Using the upper bound for $\gamma_t(s)$ and the lower bound for $\gamma_t(1)$ together with Eq. 5.2, we get

$$\begin{aligned}
\frac{\gamma_t(1)}{\gamma_t(s)} &\geq \frac{(\min_{s'} p_{s'1})}{(\max_{s'} p_{s's})} \frac{p_{as}}{p_{a1}} \frac{p_{sb}}{p_{1b}} \frac{(\min_{s'} p_{1s'})}{(\max_{s'} p_{ss'})} \\
&\geq \frac{(\min_{s'} p_{s'1})}{(\max_{s'} p_{s's})} \frac{(\min_{s'} p_{s's})}{(\max_{s'} p_{s'1})} \frac{(\min_{s'} p_{ss'})}{(\max_{s'} p_{1s'})} \frac{(\min_{s'} p_{1s'})}{(\max_{s'} p_{ss'})} \geq \sigma_1^2 \sigma_2^2.
\end{aligned}$$

Hence, for $t \in \{2, \ldots, n-1\}$, the classification probability has the following lower bound:

$$\mathbf{P}(Y_t = v_t(x^n) | X^n = x^n) = \mathbf{P}(Y_t = 1 | X^n = x^n) = \frac{\gamma_t(1)}{\sum_s \gamma_t(s)} \geq \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 \sigma_2^2 + (K-1)}.$$

Consider now the cases $t = 1$ and $t = n$. For $t = 1$, only the states with positive initial probability are considered. For such a state s, Eq. 5.1 becomes

$$\gamma_1(s) = \sum_{s''} \pi_s f_s(x_1) p_{ss''} \alpha(s'', x_2^n).$$

For $t = n$ and any $s \in S$,

$$\gamma_n(s) = \sum_{s'} \alpha(x^{n-1}, s') p_{s's} f_s(x_n).$$

Similarly, the ratios in Eq. 5.2 become for $t = 1$ and $t = n$, respectively,

$$f_1(x_1) \geq \left(\frac{\pi_s}{\pi_1}\right) f_s(x_1) \left(\frac{p_{sb}}{p_{1b}}\right), \quad f_1(x_n) \geq \left(\frac{p_{as}}{p_{a1}}\right) f_s(x_n).$$

Thus,

$$\frac{\gamma_1(1)}{\gamma_1(s)} \geq \frac{\pi_1}{\pi_s}\frac{\pi_s}{\pi_1}\frac{p_{sb}}{p_{1b}}\frac{(\min_{s'} p_{1s'})}{(\max_{s'} p_{ss'})} \geq \frac{(\min_{s'} p_{ss'})}{(\max_{s'} p_{1s'})}\frac{(\min_{s'} p_{1s'})}{(\max_{s'} p_{ss'})} \geq \sigma_1^2,$$
$$\frac{\gamma_n(1)}{\gamma_n(s)} \geq \frac{(\min_{s'} p_{s'1})}{(\max_{s'} p_{s's})}\frac{p_{as}}{p_{a1}} \geq \frac{(\min_{s'} p_{s'1})}{(\max_{s'} p_{s's})}\frac{(\min_{s'} p_{s's})}{(\max_{s'} p_{s'1})} \geq \sigma_2^2,$$

and the corresponding bounds for the classification probabilites are

$$\mathbf{P}(Y_1 = v_1(x^n)|X^n = x^n) \geq \frac{\sigma_1^2}{\sigma_1^2 + (K_1 - 1)}, \quad \mathbf{P}(Y_n = v_n(x^n)|X^n = x^n) \geq \frac{\sigma_2^2}{\sigma_2^2 + (K - 1)}.$$

Proof of Proposition 2.2

To prove the proposition, we use Lemma 5.1 from Doob (1953). We present the lemma using the same notation as in Doob (1953). The random variables of the Markov chain are denoted by $x_1, x_2, \ldots$, the state space is denoted by X, and $\mathcal{F}_X$ is a Borel field of X sets. For $A \in \mathcal{F}_X$, let $p(\xi, A) = \sum_{\eta \in A} p_{\xi\eta}$, and let $p^{(n)}(\xi, A)$ denote the corresponding n-step probability. The conditional probability (from initial point ξ) that the system will be in a state within A at some time during the first n transitions, is denoted by $\tilde{p}^{(n)}(\xi, A)$, that is

$$\tilde{p}^{(n)}(\xi, A) = \mathbf{P}\{\cup_{j=2}^{n+1}[x_j(\omega) \in A]|x_1(\omega) = \xi\}.$$

Hypothesis (D) in Doob (1953) is the Doeblin condition.

Hypothesis (D) *There is a (finite-valued) measure φ of sets $A \in \mathcal{F}_X$ with $\varphi(X) > 0$, an integer $\nu \geq 1$, and a positive ε, such that*

$$p^{(\nu)}(\xi, A) \leq 1 - \varepsilon \quad \text{if} \quad \varphi(A) \leq \varepsilon.$$

Hypothesis (D) is always satisfied in the case of finite state space, thus it imposes no restriction on finite dimensional stochastic matrices.

Lemma 5.1 (Doob 1953) *Under Hypothesis (D), if a set $A \in \mathcal{F}_X$ has the property that*

$$\lim_{n\to\infty} \tilde{p}^{(n)}(\xi, A) = \sup_n \tilde{p}^{(n)}(\xi, A) > 0, \quad \forall \xi \in X, \tag{5.3}$$

then there is a positive integer μ and a positive $\rho < 1$ for which

$$\tilde{p}^{(n)}(\xi, A) \geq 1 - \rho^{(n/\mu)-1}, \quad \xi \in X.$$

Recall that $W_t^* = \min\{w > t + r : X_{w-r}^w \in \mathcal{X}_o^{r+1}\}$. Consider an arbitrary t. Then $\mathbf{P}(W_t^* - t > k) = 1 - \mathbf{P}(W_t^* \leq t + k)$. Suppose $W_t^* = t + l$ for some $l > r$. Then $X_{t+l-r}^{t+l} \in \mathcal{X}_o^{r+1}$. Since $\forall x \in \mathcal{X}_o$, $\max_{s \notin C} f_s(x) = 0$, we are interested in only those state paths, where $Y_{t+l-r}^{t+l} \in C^{r+1}$. To prove the proposition, we define two new Markov chains

U and Z, and consider an equivalent event to $\{W_t^* \le t+k\}$ for the chain Z. To Z, we can apply Doob's lemma.

We start with defining a new Markov chain $U = \{U_t\}_{t=1}^{\infty} := \{Y_t, I_{X_t}(\mathcal{X}_o)\}_{t=1}^{\infty}$, where

$$I_{X_t}(\mathcal{X}_o) = \begin{cases} 1, & \text{if } X_t \in \mathcal{X}_o; \\ 0, & \text{if } X_t \notin \mathcal{X}_o. \end{cases}$$

Since $\forall x \in \mathcal{X}_o$, $f_s(x) = 0$ when $s \notin C$, the states where $Y_t \notin C$ and $X_t \in \mathcal{X}_o$ are not possible. Thus, the state space of U has $K + |C| = S_U$ possible states. The transition probabilities for U are given by a matrix $\mathcal{P}$ as follows: let $u_t = (i,k)$ and $u_{t+1} = (j,l)$, then

$$\mathcal{P}(u_t, u_{t+1}) = \mathbf{P}(U_{t+1} = u_{t+1} | U_t = u_t) = \mathbf{P}(Y_{t+1} = j, I_{X_{t+1}}(\mathcal{X}_o) = l | Y_t = i, I_{X_t}(\mathcal{X}_o) = k)$$

$$= \mathbf{P}(I_{X_{t+1}}(\mathcal{X}_o) = l | Y_{t+1} = j)\mathbf{P}(Y_{t+1} = j | Y_t = i) = \begin{cases} p_{ij} P_j(\mathcal{X}_o), & \text{if } l = 1; \\ p_{ij} P_j(\mathcal{X}_o^c), & \text{if } l = 0. \end{cases}$$

Observe that if $j \notin C$, then $P_j(\mathcal{X}_o^c) = 1$. Define now the Markov chain $Z = \{Z_t\}_{t=1}^{\infty}$ as

$$Z_t := (U_t, U_{t+1}, \ldots, U_{t+r}).$$

This chain has S_U^{r+1} possible states and the transitions for Z are determined by the transition probabilities for U. A transition from Z_t to Z_{t+1} is possible only if the last r elements of Z_t and the first r elements of Z_{t+1} coincide. The transition probability in this case is given by $\mathcal{P}(u_{t+r}, u_{t+r+1})$.

Let H denote the subset of states of Z, such that for all U_j in Z_t, $j = t, \ldots, t+r$, $Y_j \in C$ and $I_{X_j}(\mathcal{X}_o) = 1$, i.e. $Y_t^{t+r} \in C^{r+1}$ and $X_t^{t+r} \in \mathcal{X}_o^{r+1}$. There are $|C|^{r+1}$ such possible states. Then the event $\left\{\bigcup_{i=t+1}^{t+k-r}(Z_i \in H)\right\}$ is equivalent to the event $\{W_t^* \le t+k\}$.

To apply Doob's lemma, we have to check that property (5.3) holds for Z and our set H. We have for large n:

$$\begin{aligned} \mathbf{P}\left\{\cup_{j=2}^{n+1}(Z_j \in H) | Z_1 = \xi\right\} &\ge \mathbf{P}(Z_n \in H | Z_1 = \xi) \\ &= \mathbf{P}(Y_n^{n+r} \in C^{r+1}, X_n^{n+r} \in \mathcal{X}_o^{r+1} | Z_1 = \xi) \\ &= \mathbf{P}(X_n^{n+r} \in \mathcal{X}_o^{r+1} | Y_n^{n+r} \in C^{r+1}) \mathbf{P}(Y_n^{n+r} \in C^{r+1} | Z_1 = \xi) \\ &=: Prob_1 \cdot Prob_2. \end{aligned}$$

Consider at first $Prob_1$. According to the cluster definition, $\min_{s \in C} f_s(x) > \epsilon$ for some $\epsilon > 0$ for every $x \in \mathcal{X}_o$. Therefore, $\int_{\mathcal{X}_o} f_s(x) d\mu > \epsilon \mu(\mathcal{X}_o) = m$ if $s \in C$. Thus,

$$Prob_1 = \prod_{t=n}^{n+r} \mathbf{P}(X_t \in \mathcal{X}_o | Y_t \in C) > m^{r+1}.$$

Consider now $Prob_2$. Recall that $R = (p_{ij})_{i,j \in C}$ and due to **A1**, R^r is strictly positive. Therefore, $\min_i \sum_j r_{ij}^{(r)} > \delta$ for some $\delta > 0$. Let the state of Y_{r+1} in $Z_1 = \xi$ be s. We obtain:

$$\begin{aligned} Prob_2 &= \mathbf{P}(Y_n \in C, Y_{n+1} \in C, \ldots, Y_{n+r} \in C, Z_1 = \xi) / \mathbf{P}(Z_1 = \xi) \\ &= \sum_{i \in C} \sum_{j \in C} \mathbf{P}(Y_{n+1} \in C, \ldots, Y_{n+r-1} \in C, Y_{n+r} = j | Y_n = i, Z_1 = \xi) \\ &\quad \times \mathbf{P}(Y_n = i | Z_1 = \xi) \end{aligned}$$

$$> \delta \sum_{i \in C} \mathbf{P}(Y_n = i | Y_{r+1} = s) = \delta \sum_{i \in C} p_{si}^{(n-r-1)}.$$

Since Y is irreducible, there exist n_s and $\eta_s > 0$ for every $s \in S$ such that $\sum_{i \in C} p_{si}^{(n_s - r - 1)} = \eta_s > 0$. Take $\eta^* = \min_s \eta_s$ and $n^* = \max_s n_s$. Then since $\mathbf{P}\left\{\cup_{j=2}^{n+1}(Z_j \in H) | Z_1 = \xi\right\}$ is monotone and nondecreasing, we have that for $n > n^*$,

$$\mathbf{P}\left\{\cup_{j=2}^{n+1}(Z_j \in H) | Z_1 = \xi\right\} > m^{r+1}\delta\eta^*.$$

Observe that this holds for every t, i.e. when we condition on Z_t and take the union over $\{t+1, \ldots, t+n\}$. Now we can prove Proposition 2.2.

Proof of Proposition 2.2 We have:

$$\mathbf{P}(W_t^* - t > k) = 1 - \mathbf{P}(W_t^* \leq t + k) = 1 - \mathbf{P}\left\{\cup_{i=t+1}^{t+k-r}(Z_i \in H)\right\}$$

$$= 1 - \sum_{\xi} \mathbf{P}\left\{\cup_{i=t+1}^{t+k-r}(Z_i \in H) | Z_t = \xi\right\} \mathbf{P}(Z_t = \xi) \overset{\text{(Lemma 5.1)}}{\leq} \rho^{(k-r)/\mu - 1} = a \exp[-bk],$$

where $a = \rho^{-r/\mu - 1}$ and $b = -\frac{1}{\mu} \ln \rho$.

Emission matrix for Section 3.2.1

P_1	P_2	P_3	P_4	P_5	P_6
0.1059	0.0636	0.0643	0.1036	0.1230	0.1230
0.0107	0.0171	0.0135	0.0081	0.0111	0.0128
0.0538	0.0319	0.0775	0.0634	0.0415	0.0345
0.0973	0.0477	0.0620	0.1120	0.0852	0.0848
0.0436	0.0576	0.0330	0.0371	0.0386	0.0399
0.0303	0.0484	0.1133	0.0447	0.0321	0.0229
0.0203	0.0227	0.0259	0.0188	0.0197	0.0221
0.0564	0.1010	0.0372	0.0577	0.0694	0.0593
0.0672	0.0443	0.0574	0.0540	0.0671	0.0810
0.1227	0.1068	0.0674	0.0994	0.1279	0.1477
0.0240	0.0219	0.0181	0.0214	0.0293	0.0304
0.0299	0.0252	0.0561	0.0259	0.0338	0.0336
0.0333	0.0208	0.0757	0.0472	0.0067	0.0031
0.0443	0.0270	0.0330	0.0469	0.0497	0.0472
0.0594	0.0464	0.0470	0.0522	0.0677	0.0697
0.0496	0.0496	0.0744	0.0485	0.0422	0.0491
0.0395	0.0641	0.0572	0.0465	0.0412	0.0375
0.0591	0.1386	0.0473	0.0685	0.0677	0.0545
0.0168	0.0170	0.0111	0.0135	0.0130	0.0124
0.0359	0.0483	0.0286	0.0306	0.0331	0.0345

References

Bahl LR, Cocke J, Jelinek F, Raviv J (1974) Optimal decoding of linear codes for minimizing symbol error rate (Corresp.) IEEE Trans Inf Theory 20(2):284–287

Brejová B, Brown DG, Vinař T (2007) The most probable annotation problem in HMMs and its application to bioinformatics. J Comput Syst Sci 73(7):1060–1077
Brushe GD, Mahony RE, Moore JB (1998) A soft output hybrid algorithm for ML/MAP sequence estimation. IEEE Trans Inf Theory 44(7):3129–3134
Cao L, Chen CW (2003) A novel product coding and recurrent alternate decoding scheme for image transmission over noisy channels. IEEE Trans Commun 51(9):1426–1431
Cappé O, Moulines E, Rydén T (2005) Inference in hidden Markov models. Springer, New York
Colella S, Yau C, Taylor JM, Mirza G, Butler H et al (2007) QuantiSNP: an objective Bayes Hidden-Markov Model to detect and accurately map copy number variation using SNP genotyping data. Nucl Acids Res 35(6):2013–2025
Doob JL (1953) Stochastic processes. Wiley, New York
Durbin R, Eddy SR, Krogh A, Mitchison G (1998) Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge
Ephraim Y, Merhav N (2002) Hidden Markov processes. IEEE Trans Inf Theory 48(6):1518–1569
Gerencsér L, Molnár-Sáska G (2002) A new method for the analysis of hidden Markov model estimates. In: Proceedings of the 15th IFAC world congress, Barcelona, Spain
Hayes JF, Cover TM, Riera JB (1982) Optimal sequence detection and optimal symbol-by-symbol detection: similar algorithms. IEEE Trans Commun 30(1):152–157
Jelinek F (1997) Statistical methods for speech recognition. The MIT Press, Cambridge
Koloydenko A, Lember J (2008) Infinite Viterbi alignments in the two state hidden Markov models. Acta Comment Univ Tartu Math 12:109–124
Koski T (2001) Hidden Markov models for bioinformatics, volume 2 of computational biology series. Kluwer Academic Publishers, Dordrecht
Kuljus K, Lember J (2012) Asymptotic risks of Viterbi segmentation. Stoch Process Appl 122(9):3312–3341
Le Gland F, Mevel L (2000) Exponential forgetting and geometric ergodicity in hidden Markov models. Math. Control Signals Systems 13(1):63–93
Lember J (2011a) A correction on approximation of smoothing probabilities for hidden Markov models. Stat Probab Lett 81(9):1463–1464
Lember J (2011b) On approximation of smoothing probabilities for hidden Markov models. Stat Probab Lett 81(2):310–316
Lember J, Koloydenko A (2008) The adjusted Viterbi training for hidden Markov models. Bernoulli 14(1):180–206
Lember J, Koloydenko A (2010) A constructive proof of the existence of Viterbi processes. IEEE Trans Inf Theory 56(4):2017–2033
Lember J, Koloydenko A (2014) Bridging Viterbi and posterior decoding: a generalized risk approach to hidden path inference based on hidden Markov models. J Mach Learn Res 15: 1–58
Lember J, Kuljus K, Koloydenko A (2011) Theory of segmentation. In: Dymarsky P (ed) Hidden Markov models, theory and applications. InTech, pp 51–84
Li J, Gray RM, Olshen RA (2000) Multiresolution image classification by hierarchical modeling with two-dimensional hidden Markov models. IEEE Trans Inform Theory 46(5):1826–1841
Och FJ, Ney H (2000) Improved statistical alignment models. In: Proc 38th ann meet assoc comput linguist, pp 440–447
Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286
Rue H (1995) New loss functions in Bayesian imaging. J Am Stat Assoc 90(431):900–908
Sznitman R, Jedynak B (2010) Active testing for face detection and localization. IEEE Trans Pattern Anal Mach Intell 32(10):1914–1920
Udupa RU, Maji HK (2005) Theory of alignment generators and applications to statistical machine translation. In: Kaelbling LP, Saffiotti A (eds) Proceedings of the 19th international joint conference on artificial intelligence (IJCAI-05), Edinburgh, Scotland, pp 1142–1147
Viterbi AJ (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans Inf Theory 13(2):260–269
Wang K, Li M, Hadley D, Liu R, Glessner J et al (2007) PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. Genome Res 17:1665–1674
Winkler G (2003) Image analysis, random fields and Markov Chain Monte Carlo methods, volume 27 of stochastic modelling and applied probability. Springer, Berlin
Yau C, Holmes CC (2013) A decision-theoretic approach for segmental classification. Ann Appl Stat 7(3):1814–1835