Estimation of Viterbi path in Bayesian hidden Markov models

Jüri Lember[1]*, Dario Gasbarra[2], Alexey Koloydenko[3] and Kristi Kuljus[1]

February 7, 2018

[1]*University of Tartu, Estonia;* [2]*University of Helsinki, Finland;*
[3]*Royal Holloway, University of London, UK*

Abstract

The article studies different methods for estimating the Viterbi path in the Bayesian framework. The Viterbi path is an estimate of the underlying state path in hidden Markov models (HMMs), which has a maximum posterior probability (MAP). For an HMM with given parameters, the Viterbi path can be easily found with the Viterbi algorithm. In the Bayesian framework the Viterbi algorithm is not applicable and several iterative methods can be used instead. We introduce a new EM-type algorithm for finding the MAP path and compare it with various other methods for finding the MAP path, including the variational Bayes approach and MCMC methods. Examples with simulated data are used to compare the performance of the methods. The main focus is on non-stochastic iterative methods and our results show that the best of those methods work as well or better than the best MCMC methods. Our results demonstrate that when the primary goal is segmentation, then it is more reasonable to perform segmentation directly by considering the transition and emission parameters as nuisance parameters.

Keywords: HMM, Bayes inference, MAP path, Viterbi algorithm, EM, variational Bayes, simulated annealing

1 Introduction and preliminaries

Hidden Markov models (HMMs) are widely used in several application areas including speech recognition, computational linguistics, computational molecular biology and many more. Recently there has been a continuing interest to apply HMMs in the Bayesian framework, where model parameters are not directly estimated. On the one hand, the Bayesian approach allows researchers to incorporate their prior believes and information in the modeling process, and makes inference more robust with respect to different sets of parameters. On the other hand, the Bayesian framework might heavily complicate the analysis, since a mixture of HMMs is not typically an HMM anymore. Therefore, the algorithms and methods valid for a single HMM might not be applicable in the Bayesian setup. For example, when several HMMs are mixed, the optimality principle is not present anymore and dynamic programming algorithms like the Viterbi algorithm as well as forward-backward algorithms do not work. Therefore, finding the Viterbi path in the Bayesian framework is a difficult task where no simple solution exists.

*Corresponding author, e-mail: jyri.lember@ut.ee

Most of the literature on HMMs in the Bayesian framework (see, e.g. [5, 18, 22, 30, 27, 23, 8]) deals with MCMC methods [30, 5, 18, 27, 3]. When the goal is segmentation, often various methods based on Gibbs sampling, for example simulated annealing, are used. It should be noted that simulated annealing works well only if the applied cooling schedule is correct and the number of sweeps big enough. In this article, we are mainly interested in studying different non-stochastic methods for finding the Viterbi path in the Bayesian framework, because in comparison to MCMC methods non-stochastic methods are computationally less demanding. We introduce a new EM-type segmentation method which we call *segmentation EM*, where the hidden path is considered as a parameter of interest. The segmentation EM method is our main study objective, since it optimizes the studied criterion. The performance of segmentation EM is compared with various other segmentation methods (including the variational Bayes approach and parameter EM estimation procedures) in numerical data examples. According to our numerical examples, the segmentation EM method and a closely related method which we call *segmentation MM*, perform as well as MCMC methods. Moreover, our empirical studies demonstrate that direct Bayesian segmentation approach outperforms the commonly used parameter-first approach, where segmentation is performed after the parameters have been estimated.

The Viterbi path estimation in the Bayesian framework has been studied and applied in speech tagging problems [12, 14, 10, 11]. In these papers, several methods for calculating the Viterbi path have been studied, for example simulated annealing, variational Bayes and also the parameters-first approach. The results of the studies are somewhat contradictory (see, e.g. [11]), showing that further research in this area is indeed needed. An overview of Bayesian HMMs in speech tagging can be found in [7].

The article is organized as follows. In the rest of this section, we introduce the problem of estimating the Viterbi path in the frequentist and Bayesian framework. At the end of the section, the main objectives of the article will be summarized. In Section 2, a brief overview of the methods and algorithms considered in the article is given. In Section 3, the results of numerical examples are presented. First, in Subsection 3.2 we consider the case where the emission parameters are known and transition parameters are unknown having Dirichlet prior distributions. In Subsection 3.3, we consider the case where also emission parameters are unknown. In the example, the emission densities are normal with conjugate priors.

1.1 Segmentation with hidden Markov models

Hidden Markov model. Consider a homogeneous Markov chain $Y^n := Y_1, \ldots, Y_n$ with states $S = \{1, \ldots, K\}$. Let $X^n := X_1, \ldots, X_n$ be random variables taking values on $\mathcal{X}$ such that: 1) given Y^n, the random variables $\{X_t\}$ are conditionally independent; 2) the distribution of X_t depends on Y^n only through Y_t. Since only X^n is observed, the pair (Y^n, X^n) is referred to as a *hidden Markov model*. Because the underlying Markov chain is homogeneous, the model is fully specified by the transition matrix $\mathbb{P} = (p_{lj})$, initial probabilities p_{0k} and *emission distributions* $P(X_t \in \cdot | Y_t = k)$, $k = 1, \ldots, K$. Thus, there are two types of parameters in our model: transition parameters Θ_{tr} where $\theta_{tr} \in \Theta_{tr}$ specifies the transition matrix and the initial probability vector, and emission parameters Θ_{em}. Often initial distribution is fixed or a function of the transition matrix. In this case, any θ_{tr} can be identified with a transition matrix. The whole parameter space is given by $\Theta := \Theta_{em} \times \Theta_{tr}$. Without loss of generality we assume that all emission distributions have *emission densities* f_k with respect to some common reference measure on $\mathcal{X}$. Typically, all emission densities are assumed to belong to the same parametric family $\mathcal{F} = \{f(\cdot|\theta) : \theta \in \Theta_{em}\}$. Thus, for any state $k \in S$, there is a $\theta^k_{em} \in \Theta_{em}$ such that $f_k(\cdot) = f_k(\cdot|\theta^k_{em})$. The joint likelihood of (y^n, x^n) is denoted by $p(y^n, x^n)$. Similarly, $p(x^n)$ and $p(y^n)$ denote the marginal likelihoods of $p(y^n, x^n)$, and $p(x^n|y^n)$ and $p(y^n|x^n)$ stand for the conditional likelihoods. We assume that the length of the observation sequence is fixed and leave it from the notation. Thus, we denote by $x \in \mathcal{X}^n$

a vector x^n of observations and by $y, s \in S^n$ state sequences. Also, X and Y stand for X^n and Y^n, respectively, $p(y, x)$ is used instead of $p(y^n, x^n)$ and so on. To indicate a single entry of a vector x or s, we use x_t or s_t, $t = 1, \ldots, n$. For any $t = 1, \ldots, n$, p_t is used for marginal probability, for example

$$p_t(y_t|x) = \sum_{s:s_t=y_t} p(s|x) = P(Y_t = y_t|X = x).$$

Viterbi path. Suppose $\theta \in \Theta$ is given. The *segmentation problem* consists of estimating the unobserved realization of the underlying Markov chain Y given observations X. Formally, we are looking for a mapping $g : \mathcal{X}^n \to S^n$ called a *classifier*, that maps every sequence of observations into a state sequence. The best classifier g is often defined via a *loss function* $L : S^n \times S^n \to [0, \infty]$, where $L(y, s)$ measures the loss when the actual state sequence is y and the estimated sequence is s. For any state sequence $s \in S^n$, the expected loss is called the conditional risk of s for given θ,

$$R(s|x, \theta) := E[L(Y, s)|X = x, \theta] = \sum_{y \in S^n} L(y, s)p(y|x, \theta).$$

In the present paper, we consider the common loss function L_∞ defined as

$$L_\infty(y, s) := \begin{cases} 1, & \text{if } y \neq s, \\ 0, & \text{if } y = s. \end{cases}$$

Minimizing the risk for given parameters θ corresponds in this case to maximizing the conditional probability $p(s|x, \theta)$. For an overview of risk-based segmentation with HMMs, see [17, 21, 34]. The solution of $\max_{y \in S^n} p(y|x, \theta)$ can be found by a dynamic programming algorithm called the *Viterbi algorithm*. Hence, any path $v = \arg\max_{y \in S^n} p(y|x, \theta)$ is called the *Viterbi path*. Observe that the Viterbi path is not optimal when it comes to minimizing the expected number of misclassification errors. Yet, it is the most popular and most studied hidden path estimate in practice (see e.g. [26, 18, 4, 6]).

1.2 Bayesian approach

The Viterbi algorithm is applicable when the transition matrix as well as emission parameters are known. When this is not the case, the standard way is to estimate first the parameters and then perform segmentation. This approach – parameters first, then segmentation – is also applicable in the Bayesian framework, where the parameters of an HMM are considered random. Indeed, one can use a Bayesian point estimate and then perform segmentation. However, if the primary goal is segmentation rather than parameter estimation, one can consider the true underlying path as the actual parameter of interest and the emission and transition parameters as nuisance parameters, and perform segmentation directly. Such approach ensures a high probability of the obtained path over a range of possible parameters and is therefore more robust.

Bayesian framework. Let π be a prior density in Θ with respect to a reference measure $d\theta$. For any $\theta = (\theta_{tr}, \theta_{em})$ and for any pair (x, y),

$$p(x, y|\theta) = p(x|y, \theta_{em})p(y|\theta_{tr}), \quad p(y, x) = \int p(y, x|\theta)\pi(\theta)d\theta. \tag{1.1}$$

It is important to note that although for any parameter set θ the measure $p(y, x|\theta)$ is a distribution of an HMM, then the measure $p(y, x)$ obtained after mixing is a distribution of a process that in general is not

an HMM. This circumstance complicates the whole analysis. The joint likelihood of an underlying path y, observations x and parameter θ is now $p(y, x, \theta) = p(y, x|\theta)\pi(\theta)$, and

$$p(y|x) = \int p(y, \theta|x)d\theta = \int p(y|x, \theta)p(\theta|x)d\theta. \tag{1.2}$$

As previously, the *Viterbi path* v is defined as any state sequence $y \in S^n$ that maximizes the probability $p(y|x)$ over all state sequences. In general, for a given loss function L, segmentation in the Bayesian framework means minimizing the risk $R(s|x) = \sum_y L(y, s)p(y|x)$, where due to (1.2),

$$R(s|x) = \int \sum_y L(y, s)p(y|x, \theta)p(\theta|x)d\theta = \int R(s|x, \theta)p(\theta|x)d\theta.$$

Therefore, our objective function $R(s|x)$ is the expected conditional risk, where the expectation is taken over the posterior $p(\theta|x)$. The Viterbi path minimizes $R(s|x)$ for the loss function L_∞.

Prior distributions. In this article, we assume that the number of states K as well as initial probabilities p_{0k} are known. The prior π is assumed to be such that emission and transition parameters are independent: $\pi(\theta) = \pi_{em}(\theta_{em})\pi_{tr}(\theta_{tr})$, where π_{em} and π_{tr} are marginals. Then

$$p(y, x, \theta) = p(y|\theta_{tr})\pi_{tr}(\theta_{tr})p(x|y, \theta_{em})\pi_{em}(\theta_{em}), \tag{1.3}$$

with

$$p(y|\theta_{tr}) = p_{0y_1} \prod_{lj} \big(p_{lj}(\theta_{tr})\big)^{n_{lj}(y)}, \quad p(x|y, \theta_{em}) = \prod_{k=1}^{K} \prod_{t:y_t=k} f_k(x_t|\theta^k_{em}),$$

where $n_{lj}(y)$ denotes the number of transitions from state l to state j in the state sequence y. In particular, (1.3) ensures that for given y and x, $p(y)$ depends on transition priors and $p(x|y)$ depends on emission priors only. The independence also implies that the posterior of the transition parameters depends only on y: $p(\theta_{tr}|x, y) = p(\theta_{tr}|y)$, and that θ_{em} and θ_{tr} are independent under posterior measure:

$$p(\theta|x, y) = p(\theta_{tr}|y)p(\theta_{em}|y, x). \tag{1.4}$$

We consider the case where emission parameters are componentwise independent, that is $\pi_{em}(\theta_{em}) = \pi^1_{em}(\theta^1_{em}) \cdots \pi^K_{em}(\theta^K_{em})$ for $\theta_{em} = (\theta^1_{em}, \dots, \theta^K_{em})$, which implies the independence under posterior:

$$p(\theta_{em}|x, y) = \prod_{k=1}^{K} p(\theta^k_{em}|x, y). \tag{1.5}$$

Typically transition parameters are the transition probabilities, that is $p_{lj}(\theta_{tr}) = p_{lj}$. The standard approach in this case is to model all the rows of a transition matrix $\mathbb{P} = (p_{lj})$ independently with the l-th row having a Dirichlet prior $\mathrm{Dir}(\alpha_{l1}, \dots, \alpha_{lK})$, see e.g. [18, 5, 22, 8, 13, 12]. Thus,

$$\pi_{tr}\big(\mathbb{P}\big) = \pi_{tr}(p_{11}, \dots, p_{1K})\pi_{tr}(p_{21}, \dots, p_{2K}) \cdots \pi_{tr}(p_{K1}, \dots, p_{KK}) \propto \prod_{lj} p_{lj}^{\alpha_{lj}-1},$$

provided $(p_{l1}, \dots, p_{lK}) \in \mathbb{S}_K$, where $\mathbb{S}_K$ is a unit simplex. Since the rows are independent under the prior, they are also independent under the posterior, so that for a given path y, the l-th row has a Dirichlet distribution:

$$p\big((p_{l1}, \dots, p_{lK})|y\big) \sim \mathrm{D}ir(\alpha_{l1} + n_{l1}(y), \dots, \alpha_{lK} + n_{lK}(y)).$$

Let $n_l(y) = \sum_j n_{lj}(y)$ and $\alpha_l = \sum_j \alpha_{lj}$. Under Dirichlet prior, the marginal probability of any path y can be calculated as (see e.g. (19) in [8])

$$p(y) = \int p(y|\theta_{tr})\pi_{tr}(\theta_{tr})d\theta_{tr} = p_{0,y_1} \prod_l \frac{\Gamma(\alpha_l)}{\Gamma(\alpha_l + n_l(y))} \prod_j \frac{\Gamma(\alpha_{lj} + n_{lj}(y))}{\Gamma(\alpha_{lj})}. \tag{1.6}$$

Uniform Dirichlet priors. Let us briefly discuss the case when $\alpha_{lj} = 1$ for every l and j. Then the rows of the transition matrix are uniformly distributed and therefore, the priors with $\alpha_{lj} = 1$ are considered to be non-informative corresponding to not assuming anything of the transition matrix. In other words, all transition matrices are equiprobable and the expected values of all entries in the transition matrix are $\frac{1}{K}$. This might suggest that the same holds in the sequence space and no particular path structure (like sequences with long blocks or rapid changes) is preferred. But this is not the case – with uniform Dirichlet priors the state sequences are far from being equiprobable and the ones having long blocks are preferred. The following proposition proves that sequences with maximum prior weight are the constant ones.

Proposition 1.1. *Let $\alpha_{lj} = 1$ for every l, j. Then*

$$\arg\max_y p(y) = \{(i, \dots, i), \quad i = \arg\max_i p_{0i}\}.$$

Proof. When $\alpha_{lj} = 1$, then $p(y)$ is according to (1.6) for any sequence y given by

$$p(y) = p_{0y_1}\Gamma(K) \prod_l \frac{\prod_j \Gamma(1 + n_{lj}(y))}{\Gamma(K + n_l(y))} = p_{0y_1}\Gamma(K) \prod_l \frac{\prod_j n_{lj}(y)!}{(n_l(y) + K - 1)!}.$$

For the proof it suffices to show that any constant sequence maximizes the product term in the expression above. Fix y and denote $n_{lj} := n_{lj}(y)$. Since for every $l = 1, \dots, K$, $\sum_j n_{lj} = n_l$ and $\sum_l n_l = n - 1$, the following inequality holds for any integer $a > 0$: $\prod_{l=1}^K (n_l + a) \geq a^{K-1}(n - 1 + a)$, where the equality holds only if $n_l = n - 1$ for some $l = 1, \dots, K$. Therefore,

$$\prod_{l=1}^K \frac{\prod_{j=1}^K n_{lj}!}{(n_l + K - 1)!} \leq \prod_{l=1}^K \frac{n_l!}{(n_l + K - 1)!} \leq \Big(\frac{1}{2 \cdot 3 \cdots (K-1))}\Big)^{(K-1)} \Big(n(n+1)\cdots(n+K-2)\Big)^{-1}.$$

The inequality is strict if $\exists j, k$ such that $n_j > 0$ and $n_k > 0$. Thus, the upper bound is reached only if $n_l = n - 1$ for some $l = 1, \dots, K$, that is we have a constant state sequence. ∎

To summarize: assuming nothing about the transition matrix is not equivalent to not assuming anything about the state sequences. On the contrary, equiprobable paths correspond to the fixed transition matrix with all entries equal to $\frac{1}{K}$, which is a very specific and strong assumption about the transition matrix.

The role of concentration parameter. As is common in the case of Dirichlet priors (see e.g. [13]), we will use factorization $\alpha_{lj} = Mq_{lj}$, where $Q = \big(q_{lj}\big)$ is a transition matrix and $M > 0$ the so-called *concentration parameter*. Thus, Q postulates our belief about the general form of the transition matrix and M shows how strongly we believe in it. The bigger M, the smaller is the variance of p_{lj}. When $q_{lj} = 1/K$ for every l and j, then the concentration parameter M can be considered as a regularization parameter in the optimization problem

$$\max_y \Big(\ln p(x|y) + \ln p_M(y) \Big), \tag{1.7}$$

where the subscript M denotes the dependence on M. Increasing M corresponds to reducing the role of $\ln p(y)$, thus the limit case $M \to \infty$ corresponds to $\ln p(y) = const$ (all paths are equiprobable). Therefore, when $M \to \infty$, (1.7) reduces to $\max_y p(x|y)$. The case with $M = K$ corresponds to the case of uniform Dirichlet priors with $\alpha_{lj} = 1$, and in this case the role of $\ln p(y)$ in (1.7) is to make the output sequences more constant.
Thus, when $q_{lj} = 1/K$, then decreasing M means changing the sequence prior $p_M(y)$ so that the sequences with big blocks have more weight. Interestingly, the same holds for other Q-matrices as well. In this article, we also consider Q-matrices, where the entries on the main diagonal have larger values than the off-diagonal elements. With such Q, for every M the sequence prior $p_M(y)$ puts more weight on the sequences with big blocks and the most probable sequences are constant ones. However, this behavior is more pronounced when M is small. Indeed, if $M \to \infty$, then for every y, $p_M(y) \to p_{0y_1} \prod_{l,j} q_{lj}^{n_{lj}(y)} := p_\infty(y)$. It is easy to see that for constant sequences the convergence is monotone. For example, if $y = 1, \dots, 1$, then as $M \to \infty$, it holds $p_M(y) \searrow p_{01} q_{11}^{n-1} = p_\infty(y)$. Thus, since the entries on the main diagonal have larger values than the off-diagonal elements, the limit measure p_∞ puts more weight on sequences with big blocks. But due to the monotone convergence, we see that for smaller M, the measure $p_M(y)$ concentrates on such sequences even more.

1.3 Objectives of the article

The main goals of the article are the following:

- To introduce segmentation EM method and study its performance in comparison to other Bayesian segmentation methods. Segmentation EM is a standard EM method where the path y is considered as the parameter of interest and θ as a nuisance parameter. Although the idea is universal, it might not be always applicable. We show that in Bayesian HMM-setting considered in this paper, the segmentation EM is applicable. Unlike the other non-stochastic methods considered in the article, it maximizes $p(y|x)$, and is therefore theoretically justified.

- To show that the best non-stochastic iterative methods perform in Bayesian segmentation at least as well as MCMC methods like simulated annealing, being at the same time computationally faster and less demanding.

- To show how largely the objective function $p(y|x)$ as well as the optimal path actually depend on the choice of hyperparameters.

2 Bayesian segmentation methods

2.1 Segmentation EM

Since our goal is to find a state sequence that maximizes $p(y|x)$, the main parameter of interest is the hidden path rather than the model parameters θ. Therefore, it is natural to change their roles in the EM procedure in order to maximize $p(y|x)$. We start with an initial sequence $y^{(0)}$ and update then the state sequences according to the following rule:

$$y^{(i+1)} = \arg\max_y \int \ln p(y, \theta|x) p(\theta|y^{(i)}, x) d\theta = \arg\max_y \int \ln p(y, x|\theta) p(\theta|y^{(i)}, x) d\theta. \tag{2.1}$$

Every iteration step increases the probability $p(y|x)$ and the algorithm stops when there are no more changes in the estimated state sequence. We call this procedure *segmentation EM*, the output is denoted by $\hat{v}_{\mathrm{sEM}}$.

Lemma 2.1. *Every iteration step in the segmentation EM procedure increases the posterior probability:* $p(y^{(i+1)}|x) \geq p(y^{(i)}|x)$. *Furthermore, the objective function in (2.1) can be maximized with the Viterbi algorithm considering the matrix* (u_{lj}) *and the functions* h_k *as the transition and emission parameters, where*

$$u_{lj} := \exp\Big[\int \ln p_{lj}(\theta_{tr}) p(\theta_{tr}|y^{(i)}) d\theta_{tr}\Big], \quad h_k(x_t) := \exp\Big[\int \ln f_k(x_t|\theta^k_{em}) p(\theta^k_{em}|y^{(i)}, x) d\theta^k_{em}\Big]. \tag{2.2}$$

Proof. It is well known that the standard EM algorithm increases the likelihood at every iteration step (see, e.g. [24, 5]). Change the roles of θ and y to obtain $p(y^{(i+1)}|x) \geq p(y^{(i)}|x)$. To see that $y^{(i)}$ can be found with the Viterbi algorithm, note that by (1.1), (1.4) and (1.5) we have

$$\begin{aligned}
&\int \ln p(y, x|\theta) p(\theta|y^{(i)}, x) d\theta = \int \ln p(y|\theta_{tr}) p(\theta_{tr}|y^{(i)}) d\theta_{tr} + \int \ln p(x|y, \theta_{em}) p(\theta_{em}|y^{(i)}, x) d\theta_{em} \\
&= \ln p_{0y_1} + \sum_{l,j \in S} n_{lj}(y) \int \ln p_{lj}(\theta_{tr}) p(\theta_{tr}|y^{(i)}) d\theta_{tr} + \sum_{k=1}^{K} \sum_{t:y_t=k} \int \ln f_k(x_t|\theta^k_{em}) p(\theta^k_{em}|y^{(i)}, x) d\theta^k_{em} \\
&= \ln p_{0y_1} + \sum_{l,j \in S} n_{lj}(y) \ln u_{lj} + \sum_{k=1}^{K} \sum_{t:y_t=k} \ln h_k(x_t).
\end{aligned}$$

Thus, the objective function is in the form of $\ln p(y, x)$ of an HMM with 'transition matrix' (u_{lj}) and 'emission densities' h_k. By Jensen's inequality we know that the rows of (u_{lj}) do not sum up to one, thus (u_{lj}) is not a transition matrix. Similarly, the functions h_k do not integrate to one, thus the functions h_k are not probability densities. However, the Viterbi algorithm can still be applied to find the path with maximum probability. To see that, note first that the functions h_k enter into the Viterbi algorithm only via values $h_k(x_t)$, so it really does not matter whether they integrate to one or not. Similarly, the optimality principle – if a maximum probability path passes state k at time m, the first m elements of that path must form a maximum likelihood path amongst those paths that end in state k at time m – does not depend on whether the probabilities sum up to one or not. If the optimality principle holds, then the Viterbi algorithm as a dynamic programming algorithm finds the maximum probability path. □

The fact that the Viterbi algorithm can be applied for maximizing $\ln p(y, x)$ makes the segmentation EM possible as soon as (u_{lj}) and $h_k(x_t)$ can be calculated. In our case with Dirichlet priors for the transition parameters the posterior measure $p(\theta_{tr}|y)$ is the product of the row posteriors, and the posterior of the l-th row is $\mathrm{D}ir(\alpha_{l1} + n_{l1}(y), \ldots, \alpha_{lK} + n_{lK}(y))$. Then

$$p_{lj} \sim \mathrm{Be}\big(\alpha_{lj} + n_{lj}(y), \alpha_l + n_l(y) - \alpha_{lj} - n_{lj}(y)\big).$$

It is known that when $X \sim \mathrm{Be}(\alpha, \beta)$, then $E(\ln X) = \psi(\alpha) - \psi(\alpha + \beta)$, where ψ is the digamma function. Thus, for any sequence y, the quantities u_{lj} can be calculated with the following formula:

$$\ln u_{lj}(y) = \int \ln p_{lj}(\theta_{tr}) p(\theta_{tr}|y) d\theta_{tr} = \psi(\alpha_{lj} + n_{lj}(y)) - \psi(\alpha_l + n_l(y)). \tag{2.3}$$

Computing h_k depends on the family of emission densities. If emission distributions belong to an exponential family, that is

$$f(x|\theta_{em}) = \exp[\theta_{em}^t T(x) + A(\theta_{em}) + B(x)],$$

then calculation of $h_k(x)$ reduces to evaluating the moments

$$\int \theta_{em} p(\theta_{em}|y,x)d\theta_{em}, \quad \int A(\theta_{em})p(\theta_{em}|y,x)d\theta_{em}.$$

For conjugate priors, this kind of integration is often feasible.

2.2 Other segmentation methods

Segmentation MM. The segmentation MM algorithm is just like the segmentation EM algorithm, except that the expectation step is replaced by the maximization step. We start with an initial path $y^{(0)}$. Then, given $y^{(i)}$, find

$$\theta^{(i+1)} = \arg\max_{\theta} p(\theta|y^{(i)},x), \quad y^{(i+1)} = \arg\max_{y} p(y|\theta^{(i+1)},x).$$

The algorithm converges when there are no changes in the two consecutive path estimates. Every iteration step increases the joint likelihood, that is

$$p(y^{(i+1)},\theta^{(i+1)}|x) \geq p(y^{(i)},\theta^{(i+1)}|x) \geq p(y^{(i)},\theta^{(i)}|x),$$

but the objective function $p(y|x)$ is not guaranteed to increase. In the context of parameter estimation in the non-Bayesian setting (that is when the prior is non-informative) this algorithm is sometimes called the *Viterbi training* [21, 19, 16, 20] or *classification EM* [29, 28]. It should move on faster than segmentation EM. The advantage of the segmentation MM procedure over the segmentation EM procedure is that it does not require calculation of u_{lj} and h_k. For given $\theta^{(i)}$ the path $y^{(i)}$ can be found by the standard Viterbi algorithm, and $\theta^{(i+1)}$ is just the posterior mode that can in our case due to the independence be calculated separately for the emission and transition parameters: $\theta^{(i+1)} = (\theta_{tr}^{(i+1)}, \theta_{em}^{(i+1)})$, where

$$\theta_{tr}^{(i+1)} = \arg\max_{\theta_{tr}} p(\theta_{tr}|y^{(i)}), \quad \theta_{em}^{(i+1)} = \arg\max_{\theta_{em}} p(\theta_{em}|y^{(i)},x).$$

Bayesian EM. The *parameters-first* approach in segmentation consists of estimating the unknown parameters first and then performing segmentation. The most common parameter estimate in the Bayesian setup is the *MAP estimate* defined as

$$\hat{\theta} = \arg\max_{\theta} p(\theta|x) = \arg\max_{\theta} p(x|\theta)\pi(\theta).$$

The standard method for finding $\hat{\theta}$ is the EM algorithm [24, 18]. The EM procedure in the Bayesian setup starts with an initial parameter $\theta^{(0)}$ and updates the parameters iteratively as follows:

$$\theta^{(i+1)} = \arg\max_{\theta} \sum_{y} \ln p(y,\theta|x)p(y|\theta^{(i)},x) = \arg\max_{\theta} \Big[\sum_{y} \ln p(y,x|\theta)p(y|\theta^{(i)},x) + \ln \pi(\theta)\Big]. \tag{2.4}$$

Every iteration increases the posterior probability, that is $p(\theta^{(i+1)}|x) \geq p(\theta^{(i)}|x)$. We call this estimation procedure *Bayesian EM* and denote the resulting parameter estimate by $\hat{\theta}_{\mathrm{B(EM)}}$. The EM procedure

in the non-Bayesian setup is the same, except that $\ln \pi(\theta)$ is missing on the right hand side of (2.4). This procedure will be called *standard EM* and the output of the procedure will be denoted by $\hat{\theta}_{\mathrm{EM}}$. Thus, the standard EM algorithm can be considered as a special case of the Bayesian EM algorithm with a non-informative prior ($\ln \pi(\theta) = const$). In the case of Dirichlet transition priors, noninformative priors correspond to the case $\alpha_{lj} = 1$. The Viterbi path estimates $\hat{v}_{\mathrm{B(EM)}}$ and $\hat{v}_{\mathrm{EM}}$ are obtained by applying the Viterbi algorithm with the respective parameter estimate: $\hat{v}_{\mathrm{B(EM)}} := \arg\max_y p(y|x, \hat{\theta}_{\mathrm{B(EM)}})$, $\hat{v}_{\mathrm{EM}} := \arg\max_y p(y|x, \hat{\theta}_{\mathrm{EM}})$.

Variational Bayes approach. The idea behind the variational Bayes (VB) approach (see, e.g. [32, 31, 9, 15, 1, 2, 4]) is to approximate the posterior $p(\theta, y|x)$ with a product $\hat{q}_\theta(\theta)\hat{q}_Y(y)$, where $\hat{q}_\theta$ and $\hat{q}_Y$ are probability measures on the parameter space and S^n, that minimize the Kullback-Leibler divergence $D\big(q_\theta \times q_Y || p(\theta, y|x)\big)$ over all product measures $q_\theta \times q_Y$, that is

$$\hat{q}_\theta \times \hat{q}_Y = \arg \inf_{q_\theta \times q_Y} D\big(q_\theta \times q_Y || p(\theta, y|x)\big).$$

It is known that the measures $\hat{q}_\theta$ and $\hat{q}_Y$ satisfy the following equations:

$$\ln \hat{q}_\theta(\theta) = c_1 + \int \ln p(\theta, y|x)\, \hat{q}_Y(dy) = c_1 + \sum_y \ln p(\theta, y|x) \hat{q}_Y(y),$$
$$\ln \hat{q}_Y(y) = c_2 + \int \ln p(\theta, y|x)\, \hat{q}_\theta(d\theta),$$

where c_1 and c_2 are constants. This suggests the following iterative algorithm for calculating $\hat{q}_\theta(\theta)$ and $\hat{q}_Y(y)$. Start with an initial sequence $y^{(0)}$ and take $q_Y^{(0)} = \delta_{y^{(0)}}$. Given $q_Y^{(i)}$, update the measures as

$$\ln q_\theta^{(i+1)}(\theta) = c_1^{(i+1)} + \sum_y \ln p(\theta, y|x) q_Y^{(i)}(y),$$
$$\ln q_Y^{(i+1)}(y) = c_2^{(i+1)} + \int \ln p(\theta, y|x) q_\theta^{(i+1)}(d\theta).$$

In [1], the algorithm is called the *variational Bayes EM* and it is argued (Theorem 2.1) that it decreases the Kullback-Leibler divergence in the following sense:

$$D\big(q_\theta^{(i)} \times q_Y^{(i)} || p(\theta, y|x)\big) \geq D\big(q_\theta^{(i+1)} \times q_Y^{(i)} || p(\theta, y|x)\big) \geq D\big(q_\theta^{(i+1)} \times q_Y^{(i+1)} || p(\theta, y|x)\big).$$

Suppose the VB algorithm described above has converged and its final output is $\hat{q}_\theta \times \hat{q}_Y$. Then $\hat{q}_Y$ is the approximation of $p(y|x)$ and the Viterbi path estimate $\hat{v}_{\mathrm{VB}}$ is obtained as $\hat{v}_{\mathrm{VB}} := \arg\max_y \hat{q}_Y(y)$.

Applying the variational Bayes method for estimating the Viterbi path is certainly not a trivial task. All the formulas needed for updating $q_Y^{(i+1)}$ and $q_\theta^{(i+1)}$ with explanations about technical details are presented in the Appendix.

Simulated annealing. Let $1 \leq \beta_1 < \beta_2 \ldots < \beta_r$ be a cooling schedule. Since direct sampling from distribution $p_\beta(y|x) \propto p^\beta(y|x)$ is not possible, for every β we sample $y_\beta^{(1)}, \theta_\beta^{(1)}, y_\beta^{(2)}, \theta_\beta^{(2)}, \ldots, y_\beta^{(n_\beta)}$ alternately from a probability measure $p_\beta(\theta, y|x) \propto p(\theta, y|x)^\beta$ in the Metropolis-Hastings sense as follows. For given β and path $y^{(i)}$, generate the parameter $\theta^{(i)}$ from the distribution $p_\beta(\theta|y^{(i)}, x) \propto p(\theta|y^{(i)}, x)^\beta$. Then, given

$\theta^{(i)}$, generate a path y from $p_\beta(y|\theta^{(i)},x) \propto p(y|\theta^{(i)},x)^\beta$. The generated path y will be accepted as $y^{(i+1)}$ with the probability

$$\frac{p(y|x)^\beta/[p_\beta(y|\theta^{(i)},x)p_\beta(\theta^{(i)}|y^{(i)},x)]}{p(y^{(i)}|x)^\beta/[p_\beta(y^{(i)}|\theta^{(i)},x)p_\beta(\theta^{(i)}|y,x)]} \wedge 1 = \frac{p(y|x)^\beta/p_\beta(y,x)}{p(y^{(i)}|x)^\beta/p_\beta(y^{(i)},x)} \wedge 1,$$

where $J_\beta(y|y^{(i)}) = p_\beta(y|\theta^{(i)},x)p_\beta(\theta^{(i)}|y^{(i)},x)$ is the proposal distribution and $p_\beta(y,x) \propto \int p(y,x|\theta)^\beta \pi(\theta)^\beta d\theta$. Note that the ratio actually does not depend on $\theta^{(i)}$. If the candidate path y is not accepted, then a new parameter $\theta^{(i)}$ from $p_\beta(\theta|y^{(i)},x)$ and a new path y from the distribution $p_\beta(y|\theta^{(i)},x)$ will be generated. At the end of the sampling, the path with highest probability is found:

$$\hat{v}_{\mathrm{SA}} := \arg \max_{k=1,\ldots,r;\, i=1,\ldots,n_{\beta_k}} p(y^{(i)}_{\beta_k}|x).$$

Iterative conditional mode algorithm. As already mentioned, sampling from $p(y|x)$ is in general not possible even if the model is simple. Since for any path y the probability $p(y|x)$ can be found, also for any site t the probability $p_t(y_t|y_{-t},x)$ can be calculated, where $p_t(y_t|y_{-t},x)$ stands for the probability of observing y_t at site t given the rest of the sequence and x. Note that because $p(y|x)$ is not a Markov measure, $p_t(y_t|y_{-t},x)$ is not necessarily the same as $p_t(y_t|y_{t-1},y_{t+1},x)$. The *iterative conditional mode (ICM)* updates paths iteratively as follows. It starts from a sequence $y^{(0)}$. To obtain $y^{(i+1)}$, the sequence $y^{(i)}$ is updated site-by-site by the following rule:

$$y_t^{(i+1)} = \arg\max_{k\in S} p_t(k|y_1^{(i+1)},\ldots,y_{t-1}^{(i+1)},y_{t+1}^{(i)},\ldots,y_n^{(i)},x).$$

Thus, the ICM algorithm acts similarly to single site sampling ([5], [11]), but instead of generating a random state, it picks at every step a state with maximum probability. In [8], the ICM algorithm is used under the name 'greedy algorithm'. It is indeed greedy in the sense that the update of every site increases the probability $p(y|x)$. The ICM algorithm converges when no further changes occur in the estimated sequence, the output will be denoted by $\hat{v}_{\mathrm{ICM}}$.

It is well known from the theory of simulated annealing that such a greedy update can cause the output being trapped into a local maximum (see, e.g. [33]) and also our numerical examples confirm that. However, since [8] is one of the few papers that considers segmentation in the Bayesian framework by non-stochastic methods, we include this method in our study.

3 Numerical examples

To illustrate the behaviour of the segmentation methods described in Section 2, we will present the results of two examples. In the first example we study the case with known emission distributions and transition probabilities following Dirichlet priors. Thus, $\theta = \theta_{tr}$, $\pi = \pi_{tr}$ and under π, the rows of the transition matrix are independent with the l-th row having a Dirichlet distribution $\mathrm{Dir}(\alpha_{l1},\cdots,\alpha_{l4})$. In the second example also emission parameters are assumed to be unknown and normal emissions with conjugate priors are studied. Since the estimation criterion is $\arg\max_y p(y|x)$, the main measure of goodness is $p(y|x)$ or equivalently, $\ln p(y,x)$. We will also study how the methods perform in regard to initial state sequences and how the estimated state paths differ for different sets of prior parameters.

3.1 General framework

The data is generated from an HMM with four underlying states, thus $S = \{1, 2, 3, 4\}$. The emission distributions are normal with common variance $\sigma^2 = 0.25$, the emission distribution corresponding to state k is $\mathcal{N}(\mu_k, \sigma^2)$ with $\mu_1 = -0.7$, $\mu_2 = 0$, $\mu_3 = 0.7$ and $\mu_4 = 1.4$, respectively. The transition matrix is given by $\mathbb{P} = (p_{lj})$ with $p_{ll} = 0.6$, $l = 1, \ldots, 4$, and $p_{lj} = 0.4/3$, otherwise. The initial distribution (p_{0k}) is uniform. The length of the generated data sequence x is $n = 600$.

Hyperparameters. Recall that we use the parametrization $\alpha_{lj} = Mq_{lj}$, where $Q = \big(q_{lj}\big)$ is a transition matrix and $M > 0$ concentration parameter. We will consider three Q-matrices:

$$Q_1 = (q_{lj}) \text{ with } q_{lj} = 0.25 \ \forall l, j; \quad Q_2 = \mathbb{P}; \quad Q_3 = (q_{lj}) \text{ with } q_{ll} = 0.4, \ q_{lj} = 0.2 \text{ for } l \neq j.$$

Thus, the combination Q_1 and $M = 4$ corresponds to uniform priors on transition parameters and Q_1 together with very large M puts uniform prior $p(y)$ on sequences. The matrices Q_2 and Q_3 favour sequences with long blocks, the smaller M, the more such behavior is pronounced. To explain our choices of M in simulations, let us give some intuition about the role of M in some procedures. First, the Bayesian EM updates (4.4) for this parametrization are given by

$$p_{lj}^{(i+1)} = \frac{\xi^{(i)}(l, j) + (Mq_{lj} - 1)}{\sum_j \xi^{(i)}(l, j) + (M - K)}, \tag{3.1}$$

where $\xi^{(i)}(l, j)$ is the expected number of transitions from state l to j at iteration i, which varies between 0 and $n - 1$. If all transitions are equally likely, with our $n = 600$ it is approximatively of order 37. If M is a way bigger than n, then the influence of data in (3.1) is negligible and the output of the procedure is very close to Q. On the other hand, a necessary condition in (3.1) is that $Mq_{lj} > 1$, which gives a lower bound to M. Similar argumentation holds for segmentation EM. Since for any integer n large enough (see e.g. [14]), $\psi(n) \approx \ln(n - 0.5)$, where $\gamma \approx 0.577$, we can for large $n \ll m$ use the approximation $\psi(m) - \psi(n) \approx \ln(m - 0.5) - \ln(n - 0.5)$. Disregarding the fact that Mq_{lj} might not be an integer, (2.3) gives that for a given state sequence y,

$$u_{lj}(y) \approx \frac{Mq_{lj} + n_{lj}(y) - 0.5}{M + n_l(y) - 0.5}.$$

If M is very small in comparison to n_{lj}, then $u_{lj} \approx \frac{n_{lj}(y)}{n_l(y)}$ and the segmentation EM algorithm is practically the same as the segmentation MM algorithm. If on the other hand M is too big, then the data are negligible and the output is close to the Viterbi path with Q. Based on these arguments, we consider the following constants M: 600, 150, 50, 10, 5. Observe that segmentation MM and Bayesian EM are applicable when $Mq_{lj} > 1$, which is restrictive when hyperparameters $\alpha_{lj} \leq 1$ are of interest.

Initial sequences. Since the studied non-stochastic methods depend on initial path values, the choice of initial paths has an important role in our numerical examples. All our procedures are designed to start with initial sequence, but a closer inspection of formulas (2.3), (4.1) and (4.5) reveals that when emission parameters are known and transition probabilities have Dirichlet priors, then the segmentation EM, segmentation MM, Bayesian EM and variational Bayes algorithms actually depend on $y^{(0)}$ only through the frequency matrix or empirical transition matrix $(n_{lj}(y^{(0)}))$. The only deterministic algorithm that really depends on initial sequence is ICM. Therefore, it is expected that ICM is more sensitive with

respect to initial sequences, because there are many more actual sequences than frequency matrices. For MCMC methods as simulated annealing the initial value does not matter, because the number of sweeps is typically large.

Since our goal is to find the global maximum of $p(y|x)$ and the output of a method depends typically on the frequency matrix of the initial sequence, we try to choose initial sequences so that the corresponding frequency matrices will be different. Theoretically we would somehow like to cover the whole space of transition matrices. In the simplest case, that is for a two-state model, we could for example choose transition matrices as follows:

$$\begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix}, \quad \text{where} \quad p, q \in \{0.25, 0.5, 0.75\}.$$

This would provide us with nine different transition matrices which could then be used to generate random sequences as realizations of a Markov chain with initial distribution being the stationary one. In the case of four states applying the described approach becomes more complicated. Therefore, in our examples we have considered 15 transition matrices $B_1, \ldots, B_{15}$ for generating initial sequences, which are obtained as follows. The first three matrices are just our Q_1, Q_2 and Q_3. The rest, $B_4, \ldots, B_{15}$, have been randomly generated as follows: each row of B_l has been independently generated from $\mathrm{Dir}(\alpha, \alpha, \alpha, \alpha)$, where the following 12 constants $\alpha = (0.3, 0.5, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.5, 1.7, 1.9)$ have been used. From each matrix we have generated three random sequences as realizations of a Markov chain with initial distribution being the stationary one. We study also the initial sequence $y^{(0)}$ that corresponds to maximizing emissions pointwise, that is

$$y_t^{(0)} = \arg \max_{k=1,2,3,4} f_k(x_t), \quad t = 1, \ldots, n,$$

which was suggested in [8]. For given Q, a good candidate for initial path is always the Viterbi path obtained using Q, therefore the last initial sequence considered is the Viterbi path obtained with the transition matrix Q. Thus, alltogether we have studied 47 initial sequences. Given a set of hyperparameters and a non-stochastic iterative method, every initial sequence produces an output sequence. The maximum number of different output sequences is 47. The smaller that number, the more *robust* or *less sensitive* with respect to initial sequences the method is. The end result of the method is given by the best output sequence, i.e. the one that has the biggest likelihood $\ln p(y, x)$.

3.2 Example 1: fixed emission distributions

The main purpose of the first example is to compare the general performance of the presented algorithms. All non-stochastic methods (segmentation EM, segmentation MM, VB, ICM, Bayesian EM, standard EM) were run with all 47 initial sequences, whereas for simulated annealing one initial sequence was used. In the case of segmentation EM, segmentation MM and ICM the algorithm stopped when there were no further changes in the estimated state sequence. In simulated annealing a cooling schedule with inverse temperatures equally spaced in the range [1,10.2] was used, where for every inverse temperature 15 paths were generated.

In Table 1, for every method the best log-likelihood value $\ln p(\hat{v}, x)$ over the outcomes corresponding to 47 initial sequences is presented, where $\hat{v}$ denotes the best output sequence for the corresponding method. The number in the brackets gives the number of different outputs out of 47 possible. The best log-likelihood value over all the methods for each set of hyperparameters is given in bold. As the table shows, the best results are generally obtained by the segmentation EM and segmentation MM methods.

The results for those methods differ for five sets of hyperparameters, and then sometimes the segmentation EM performs slightly better and sometimes the other way around. The similarity of the segmentation EM and MM methods is explained in Subsection 3.5. It is a bit surprising that the segmentation EM method that is designed to maximize $p(y|x)$ gives sometimes a slightly worse result than the segmentation MM method that maximizes something else than $p(y|x)$. In Table 1, we can also see that VB and Bayesian EM behave quite similarly, this will also be clarified in Subsection 3.5. Observe that Bayesian EM is independent of initial sequence, while VB can result in different path outcomes. We can see that for EM-type methods the number of different outputs (sensitivity) increases when M decreases and this makes sense, because smaller M means that data has more influence. Notice that ICM is most sensitive among the studied methods, resulting in a different outcome for basically every initial sequence. It can also be remarked that the number of different outputs in the table for segmentation EM and segmentation MM shows that the initial state sequences generated from the same transition matrix result often in different output sequences.

Q	M	sEM	sMM	ICM	VB	B(EM)	EM	SA
Q_1	600	**-1071.76** (6)	**-1071.76** (6)	**-1071.76** (11)	-1072.93 (2)	-1072.98	-1127.27 (4)	-1072.02
	150	**-1017.57** (30)	-1017.59 (27)	-1030.48 (46)	-1051.82 (4)	-1051.92	-1038.00 (4)	-1031.63
	50	-940.46 (31)	**-940.42** (31)	-971.08 (46)	-1019.84 (8)	-1019.07	-955.78 (4)	-945.68
	10	**-861.81** (38)	-861.84 (37)	-909.67 (46)	-932.17 (4)	-924.27	-878.13 (4)	-865.10
	5	**-842.27** (33)	-842.30 (35)	-901.81 (46)	-909.12 (1)	-899.53	-863.33 (4)	-860.88
Q_2	600	**-898.31** (1)	**-898.31** (1)	-898.78 (47)	-899.28 (1)	-899.28	-927.24 (4)	**-898.31**
	150	**-882.51** (8)	**-882.51** (7)	-887.10 (47)	-888.32 (2)	-887.35	-901.07 (4)	-882.68
	50	**-862.31** (18)	**-862.31** (19)	-877.19 (47)	-878.32 (4)	-877.91	-876.46 (4)	-865.63
	10	**-831.71** (32)	**-831.71** (36)	-873.11 (47)	-871.52 (3)	-869.18	-853.36 (4)	-851.10
	5	**-825.07** (36)	na	-875.43 (47)	-870.10 (2)	na	-849.63 (4)	-834.38
Q_3	600	**-985.91** (6)	**-985.91** (6)	-988.46 (47)	-989.51 (1)	-989.22	-1010.59 (4)	-985.91
	150	**-945.93** (13)	**-945.93** (14)	-961.60 (47)	-966.80 (4)	-966.80	-964.31 (4)	-946.58
	50	-901.42 (24)	**-901.38** (27)	-936.78 (47)	-940.26 (2)	-938.31	-913.28 (4)	-905.46
	10	**-846.64** (34)	**-846.64** (32)	-901.30 (47)	-895.44 (1)	-892.72	-865.09 (4)	-865.05
	5	**-833.49** (34)	na	-895.74 (47)	-888.02 (4)	na	-855.95 (4)	-843.05

Table 1: The best log-likelihood value obtained for every method in Example 1. The best result(s) for every set of hyperparameters is presented in bold. In the brackets, the number of different output sequences out of 47 possible is given.

The log-likelihood values in Table 1 give a general summary measure for comparing the best paths over the studied methods. To understand better how different these best paths really are, we have counted the pointwise differences in comparison to the best path and summarized these in Table 2.

Tables 1 and 2 summarize the results of different methods for fixed observation sequence x. To observe the general behavior of the studied algorithms, we have rerun these simulations for 20 different observation sequences. Different observation sequences show the similar pattern as in Table 1. We now study how segmentation EM and segmentation MM perform in comparison to simulated annealing. Table 3 summarizes the results for our 15 sets of hyperparameters and 20 observation sequences. The counts in the first half of the table (columns SA_{max}, sEM_{max}, sMM_{max}) present for each set of hyperparameters the number of best scores over 20 observation sequences. Each time a considered method is counted as best when it reaches the maximum log-likelihood for a given set of hyperparameters. Thus, for example, if all the three methods resulted in the same state path estimate, every method is counted as the best or

Q	M	sEM	sMM	ICM	VB	B(EM)	EM	SA
Q_1	600	0	0	0	19	20	167	10
	150	0	5	75	178	177	133	51
	50	6	0	119	198	198	108	24
	10	0	1	135	158	150	79	114
	5	0	2	151	141	135	81	87
Q_2	600	0	0	12	19	19	117	0
	150	0	0	60	53	47	117	6
	50	0	0	92	66	67	101	42
	10	0	0	170	113	110	81	86
	5	0	na	170	118	na	81	28
Q_3	600	0	0	38	41	37	90	9
	150	0	0	85	92	92	82	15
	50	7	0	130	125	122	83	163
	10	0	0	160	138	132	80	44
	5	0	na	151	129	na	81	54

Table 2: Comparison of the estimated state sequences with the best Viterbi path estimate for each set of hyperparameters in Example 1. The number of pointwise differences compared to the best path estimate is presented.

'winner'. In our example we can conclude that EM-type methods perform better than simulated annealing, since simulated annealing does not give the maximum log-likelihood value as often as EM-type methods. The second part of the table (columns SA_{min}, sEM_{min}, sMM_{min}) presents for each method and for a given set of hyperparameters the count over 20 observation sequences of when this method was strictly worse than the other two. Here we can see that simulated annealing gives the lowest log-likelihood value most often. Thus, our example demonstrates that for a given cooling schedule and given set of initial state paths, EM-type algorithms perform actually better than simulated annealing.

3.3 Example 2: priors on transition probabilities and emission parameters

In the second example we assume that also the parameters of emission densities are unknown. The transition probabilities are modeled with Dirichlet priors like previously. For emissions we consider normal distributions with conjugate prior distributions. The emission distribution corresponding to state k is $\mathcal{N}(\mu_k, \sigma_k^2)$, where prior distributions for μ_k and σ_k^2 are given by a normal and inverse chi-square distribution, respectively:

$$\pi_{em}(\theta_{\mathrm{em}}) = \prod_{k=1}^{K} \pi_{em}(\theta_{\mathrm{em}}^k), \quad \pi_{em}(\theta_{\mathrm{em}}^k) = \pi(\mu_k, \sigma_k) = \pi(\sigma_k^2)\pi(\mu_k|\sigma_k^2),$$

where

$$\mu_k|\sigma_k^2 \sim \mathcal{N}\big(\xi_k, \frac{\sigma_k^2}{\kappa_0}\big), \quad \sigma_k^2 \sim \mathrm{Inv}{-}\chi^2(\nu_0, \tau_0^2).$$

Here κ_0, ν_0 and τ_0^2 are hyperparameters that might depend on k, but in our example we assume they are equal. The calculations have been performed using the same 20 observation sequences x and the same 47

Q	M	SA_{max}	sEM_{max}	sMM_{max}	SA_{min}	sEM_{min}	sMM_{min}
Q_1	600	0	19	20	20	0	0
	150	0	18	15	20	0	0
	50	0	16	16	20	0	0
	10	1	14	14	18	1	0
	5	5	11	12	14	2	2
Q_2	600	16	20	20	4	0	0
	150	1	18	20	19	0	0
	50	2	16	15	18	1	1
	10	4	11	17	15	2	0
	5	4	16	na	16	4	na
Q_3	600	6	17	18	13	1	0
	150	0	19	20	20	0	0
	50	1	17	16	19	0	0
	10	5	13	11	14	1	4
	5	3	17	na	17	3	na

Table 3: The counts over the 20 sequences when the simulated annealing, segmentation EM and segmentation MM methods reached the maximum and minimum values of log-likelihood in Example 1. The minimum count shows how many times the respective method performs worst of the three methods according to the log-likelihood value.

initial sequences $y^{(0)}$ as in Example 1. We will also refer to this example as the Dirichlet-NIX case. The necessary formulas and computational details about the algorithms needed for the Dirichlet-NIX example can be found in the Appendix.

In the Dirichlet-NIX example, the choice of emission hyperparameters affects segmentation results strongly, see also Subsection 3.4. The hyperparameters we consider are as follows: $\xi = (-0.7, 0, 0.7, 1.4)$, $\tau_0^2 = 0.25$, $\kappa_0 = 10$, $\nu_0 = 50$. In simulated annealing a cooling schedule with inverse temperatures equally spaced in the range $[1, 21]$ was used, for every inverse temperature 15 paths were generated.

Since we are very much interested in how computationally faster non-stochastic methods perform in comparison to MCMC methods, we start with presenting in Table 4 a summary of the behaviour of log-likelihood values over 20 sequences and our 15 sets of transition hyperparameters just as in Table 3. The counts in Table 4 show that simulated annealing gives often the maximum log-likelihood value for $M = 600$, otherwise EM-type algorithms perform generally better.

In Table 5, the log-likelihood values $\ln p(\hat{v}, x)$ of the best output sequence for the studied methods are presented for the same observation sequence as in Table 1. Again, the number of different outcome sequences out of 47 possible can be seen in the brackets. In general Table 5 shows the same pattern as Table 1: the best methods are segmentation MM and segmentation EM and they both outperform VB and Bayesian EM. The log-likelihood values for segmentation MM are slightly better for this observation sequence compared to segmentation EM. But this is not a rule, studying the other 19 observation sequences shows that in general sometimes segmentation EM is better and sometimes the other way around.

The log-likelihood values in Table 5 give again a general summary measure for comparing the best paths over the studied methods. Relatively small differences in log-likelihood values can incorporate large pointwise differences in the respective sequences. For example, the log-likelihood values of the best state sequences for segmentation EM and segmentation MM when $Q = Q_2$ and $M = 50$ are -855.26 and -854.91,

respectively. The pointwise difference between the state paths is 163, the transition frequency matrices are given by

$$\begin{pmatrix} 44 & 1 & 2 & 12 \\ 0 & 333 & 0 & 16 \\ 2 & 0 & 5 & 0 \\ 14 & 15 & 0 & 155 \end{pmatrix}, \qquad \begin{pmatrix} 37 & 3 & 0 & 4 \\ 2 & 485 & 0 & 6 \\ 0 & 0 & 0 & 0 \\ 5 & 6 & 0 & 51 \end{pmatrix}.$$

Q	M	SA_{max}	sEM_{max}	sMM_{max}	SA_{min}	sEM_{min}	sMM_{min}
Q_1	600	18	12	11	1	3	6
	150	12	9	8	7	6	5
	50	2	15	12	18	0	0
	10	5	12	11	15	2	0
	5	5	11	12	15	1	0
Q_2	600	17	4	6	3	7	2
	150	4	10	12	15	3	1
	50	1	10	14	19	0	1
	10	0	9	14	17	1	2
	5	4	16	na	16	4	na
Q_3	600	20	11	13	0	4	1
	150	7	13	10	13	1	2
	50	1	14	13	18	1	1
	10	3	10	14	16	2	1
	5	5	15	na	15	5	na

Table 4: The counts over the 20 sequences when the simulated annealing, segmentation EM and segmentation MM methods reached the maximum and minimum values of log-likelihood in Example 2. The minimum count shows how many times the respective method performs worst of the three methods according to the log-likelihood value.

3.4 Clustering under NIX model

To understand the role of emission hyperparameters, it is instructive to consider the optimization problem $\max_y p(x|y)$. In the Bayesian HMM setup this corresponds to the limit case with $Q = Q_1$ and $M \to \infty$, thus $p(y) = const.$ Since $p(y)$ is not involved in segmentation anymore, it is more correct to refer to the problem as *clustering*. We will show that the nature of the clustering problem and its solutions depend heavily on the hyperparameters. It turns out that under NIX priors, the family of possible clustering problems is very big including many familiar k-means related problems. We will briefly discuss some of them. Typically, 'standard' problems are obtained when the hyperparameters approach their extreme values 0 or ∞. The details about the formulas are given in the Appendix.

Q	M	sEM	sMM	ICM	VB	B(EM)	EM	SA
Q_1	600	-984.19 (34)	**-984.19** (33)	-984.19 (33)	-988.38 (8)	-988.33	-1129.93 (35)	-984.19
	150	-964.23 (25)	-964.18 (25)	-964.17 (45)	-975.41 (4)	-974.82	-1031.09 (35)	**-963.80**
	50	-933.55 (22)	**-933.47** (23)	-938.50 (45)	-964.38 (2)	-964.28	-950.33 (35)	-936.15
	10	**-854.69** (20)	**-854.69** (24)	**-854.69** (46)	-915.08 (1)	-910.94	-881.74 (35)	-857.87
	5	**-839.89** (20)	**-839.89** (25)	**-839.89** (46)	-890.10 (9)	-887.64	-869.43 (35)	-860.79
Q_2	600	-891.57 (12)	-891.57 (10)	-895.73 (47)	-900.09 (1)	-898.46	-927.46 (35)	**-891.53**
	150	**-881.36** (15)	**-881.36** (17)	-884.86 (47)	-887.66 (1)	-887.26	-900.19 (35)	-881.37
	50	-855.26 (14)	**-854.91** (16)	-875.71 (47)	-873.85 (1)	-874.64	-877.69 (35)	-866.92
	10	-826.88 (20)	**-822.18** (29)	-857.62 (47)	-864.01 (1)	-858.24	-859.59 (35)	-841.45
	5	**-818.95** (24)	na	-841.14 (47)	-857.46 (1)	na	-857.55 (35)	-841.52
Q_3	600	-938.34 (22)	-938.12 (23)	-938.34 (47)	-954.98 (2)	-950.91	-1014.17 (35)	**-938.08**
	150	**-927.62** (17)	**-927.62** (18)	-935.46 (47)	-936.39 (4)	-936.62	-958.13 (35)	-929.32
	50	-897.59 (20)	**-897.14** (18)	-908.92 (47)	-918.83 (2)	-919.05	-910.96 (35)	-905.39
	10	-840.84 (21)	**-834.19** (24)	-857.11 (47)	-888.23 (4)	-883.58	-869.77 (35)	-857.09
	5	**-826.57** (21)	na	-841.30 (47)	-873.73 (1)	na	-862.78 (35)	-849.91

Table 5: The best log-likelihood value for segmentation EM, segmentation MM, ICM, variational Bayes, Bayesian EM, standard EM and simulated annealing methods in Example 2. The best result(s) for every set of hyperparameters is presented in bold. The number of different output sequences out of 47 possible is given in the brackets.

I. The case $\nu_0 \to \infty$. When $\nu_0 \to \infty$, then the problem $\max_y p(x|y)$ approaches for given $\tau_0^2 > 0$ the following clustering problem: find clusters $S_1, \ldots, S_K$ that minimize

$$\sum_{k=1}^{K} \min_{\mu_k \in \mathcal{X}} \Big[\sum_{t \in S_k} (x_t - \mu_k)^2 + \kappa_0 (\mu_k - \xi_k)^2 + \tau_0^2 \ln(\kappa_0 + m_k) \Big] = \sum_{k=1}^{K} \Big[\sum_{t \in S_k} (x_t - \bar{x}_k)^2 + \frac{\kappa_0 m_k}{\kappa_0 + m_k} (\bar{x}_k - \xi_k)^2 + \tau_0^2 \ln(\kappa_0 + m_k) \Big], \tag{3.2}$$

where $m_k = |S_k|$. Here the first term is the sum of least squares, the second term tries to form clusters around ξ_k and the third term tries to build clusters of unequal length. Thus, if τ_0^2 is very big, then one cluster gets very big and the others are empty or very small. For small τ_0^2, the influence of the third term is small. When $\kappa_0 \to 0$, the second term disappears. This corresponds to the case where the variance of μ_k is infinite (uninformative prior for μ_k). The case with $\kappa_0 \to 0$ and $\tau_0^2 \to 0$ gives the classical k-means optimization problem.

The segmentation MM algorithm acts in the case $\nu_0 \to \infty$ for any τ_0^2 as follows: given clusters $S_1^{(i)}, \ldots, S_K^{(i)}$, find the corresponding cluster centres

$$\mu_k^{(i)} = (m_k^{(i)} \bar{x}_k^{(i)} + \kappa_0 \xi_k)/(\kappa_0 + m_k^{(i)}).$$

Given these centres, find new clusters corresponding to the Voronoi partition:

$$S_k^{(i+1)} = \{x_t : |x_t - \mu_k^{(i)}| = \min_l |x_t - \mu_l^{(i)}|\}.$$

In the case of segmentation EM algorithm the cluster centres $\mu_k^{(i)}$ are calculated in the same way as for segmentation MM, but the clustering rule is different:

$$S_k^{(i+1)} = \Big\{ x_t : \big(x_t - \mu_k^{(i)}\big)^2 + \frac{\tau_0^2}{m_k^{(i)} + \kappa_0} = \min_l \big(x_t - \mu_l^{(i)}\big)^2 + \frac{\tau_0^2}{m_l^{(i)} + \kappa_0} \Big\}. \tag{3.3}$$

The term $\tau_0^2/(m_l^{(i)}+\kappa_0)$ in (3.3) affects cluster size. When τ_0^2 is small, then segmentation EM and MM give the same result, but when τ_0^2 increases, then segmentation EM tends to produce clusters of unequal length, whereas segmentation MM remains unaffected by τ_0^2. When $\kappa_0 \to 0$ and $\tau_0^2 \to 0$, then both algorithms converge to the standard Lloyd algorithm.

When $\kappa_0 \to \infty$, the clustering problem in (3.2) reduces to minimizing $\sum_{k=1}^K \sum_{t\in S_k}(x_t-\xi_k)^2$ with the solution given by $S_k = \{x_t : |x_t-\xi_k| = \min_l |x_t-\xi_l|\}$. The solution matches fully with intuition, because $\nu_0 \to \infty$ and $\kappa_0 \to \infty$ corresponds to the case with fixed normal emissions with means ξ_k and variances τ_0^2, thus clustering is trivial.

II. The case with finite ν_0. For a given ν_0, the general optimization problem we have is the following: find clusters $S_1,\dots,S_k$ minimizing the sum

$$-\sum_k \ln\Gamma(\frac{\nu_0+m_k}{2})+\frac{1}{2}\sum_k \ln(\kappa_0+m_k)+\sum_k \frac{\nu_0+m_k}{2}\min_{\mu_k\in\mathcal{X}} \ln\big(\nu_0\tau_0^2+\sum_{t\in S_k}(x_t-\mu_k)^2+\kappa_0(\mu_k-\xi_k)^2\big). \quad (3.4)$$

The first two terms in (3.4) tend to make the clusters unequal. This follows from the observation that under the constraint $\sum_k m_k = n$, the products

$$\prod_{k=1}^K \Gamma(\frac{\nu_0+m_k}{2}), \quad \prod_{k=1}^K (\kappa_0+m_k)^{-\frac{1}{2}}$$

are maximized when $m_k = n$ holds for some k. The smaller ν_0 and κ_0 are, the bigger is the influence of the first two terms. When $\kappa_0 \to \infty$, the problem in (3.4) reduces to minimizing

$$-\sum_k \ln\Gamma(\frac{\nu_0+m_k}{2})+\sum_k \frac{\nu_0+m_k}{2}\ln\big(\nu_0\tau_0^2+\sum_{t\in S_k}(x_t-\xi_k)^2\big).$$

The solution to this problem gives bigger clusters than obtained by minimizing $\sum_{k=1}^K \sum_{t\in S_k}(x_t-\xi_k)^2$. When $\tau_0^2 \to \infty$, then the last term in (3.4) disappears and the problem reduces to finding clusters that minimize the sum of the first two terms in (3.4). The solution here is one big cluster.

As we have exemplified, clustering under NIX setting is very dependent on hyperparameters. In order to keep optimal clusters of comparable size, the hyperparameters must be chosen very carefully.

3.5 Discussion

Similarity of algorithms. We have seen that out of five non-stochastic optimization methods (segmentation EM, segmentation MM, ICM, Bayesian EM and VB), ICM is clearly the most inadequate one, because it depends heavily on initial sequences and gets stuck into local optimums. The other four methods can be divided into two groups, which can be characterized as segmentation-based methods (segmentation EM and segmentation MM) and parameter-based methods (Bayesian EM and VB). We call VB a parameter-based method, because it updates the parameters iteratively and then, with final h_l and u_{lj}, the Viterbi algorithm is applied (see Subsection 4.1). The segmentation EM and MM methods apply the Viterbi algorithm at each iteration step. Our numerical examples demonstrate a clear advantage of the segmentation-based methods, which is also expected, because segmentation EM optimizes the objective function of interest and segmentation MM behaves very similarly.

We already observed the pairwise similarity of the segmentation-based methods and the parameter-based methods in Example 1 and 2. In the case emission distributions are known, the four algorithms

can be further summarized as follows. Comparing (3.1) and (4.8) shows that both the Bayesian EM and VB updates can be written as

$$\ln p_{lj}^{*(i+1)} = f_1\big(\xi^{(i)}(l,j) + Mq_{lj}\big) - f_2\big(\sum_j \xi^{(i)}(l,j) + M\big),$$

where p_{lj}^* is either p_{lj} (Bayesian EM) or u_{lj} (VB), and where $f_1(x) = \ln(x-1)$, $f_2(x) = \ln(x-K)$ for Bayesian EM and $f_1 = f_2 = \psi$ for VB. Similarly, the transition updates for segmentation MM (4.1) and segmentation EM (2.3) can be written as

$$\ln p_{lj}^{*(i+1)} = f_1\big(n_{lj}(y^{(i)}) + Mq_{lj}\big) - f_2\big(n_l(y^{(i)}) + M\big),$$

where $f_1(x) = \ln(x-1)$, $f_2(x) = \ln(x-K)$ for segmentation MM and $f_1 = f_2 = \psi$ for segmentation EM. Thus, the four methods can be characterized by two parameters: the *function* parameter (ln vs ψ) and the *counts* parameter (direct counts $n_{lj}(s)$ versus averaged counts $\xi(l,j)$):

Counts/Function	ln	ψ
Direct (n_{lj})	sMM	sEM
Averaged ($\xi(l,j)$)	B(EM)	VB

The results of Example 1 and 2 show that the difference in functions does not influence the algorithm that much as the difference in counts, because the methods behave similarly row-wise. The examples also show that in terms of maximizing the main study criterion, that is the posterior likelihood, the methods using direct counts outperform the methods that use averaged counts. We have noticed that the methods using ln-function give sometimes slightly bigger posterior probability than the ones using ψ, this is a matter of future research.

4 Appendix

4.1 Formulas for the studied segmentation methods

Due to our independence assumption, all emission and transition parameters can be estimated separately. In the formulas of this section we use the same notation for the random parameters $p_{l,j}$, μ_k and σ_k^2, $k,l,j \in \{1,\dots,K\}$, and the corresponding estimates. The exact meaning can be understood from the context.

Segmentation MM. In the case of Dirichlet priors the matrix $\theta_{tr}^{(i+1)}$ can be found row-wise, the l-th row is the posterior mode:

$$p_{lj}^{(i+1)} = \frac{\alpha_{lj} + n_{lj}(y^{(i)}) - 1}{\alpha_l + n_l(y^{(i)}) - K}. \tag{4.1}$$

Emission parameters can be updated independently:

$$\theta_{em}^{k,(i+1)} = \arg\max_{\theta_{em}^k} p(\theta_{em}^k | x_{S_k}) = \arg\max_{\theta_{em}^k} \Big[\sum_{t:y_t^{(i)}=k} \ln f_k(x_t|\theta_{em}^k) + \ln \pi_{em}^k(\theta_{em}^k) \Big], \quad k = 1,\dots,K,$$

where x_{S_k} is the subsample of x corresponding to state k in $y^{(i)}$. Formally, for every sequence $y \in S^n$ define $S_k(y) = \{t \in \{1,\dots,n\} : y_t = k\}$, then $x_{S_k} = \{x_t : t \in S_k\}$.

Bayesian EM. The emission updates are given by

$$\theta_{em}^{k,(i+1)} = \arg\max_{\theta_{em}^k} \Big[\sum_t \ln f_k(x_t|\theta_{em}^k)\gamma_t^{(i)}(k) + \ln \pi_{em}^k(\theta_{em}^k) \Big], \quad k = 1, \dots, K, \tag{4.2}$$

where

$$\gamma_t^{(i)}(k) := P(Y_t = k|X = x, \theta^{(i)}) = \sum_{y: y_t = k} p(y|\theta^{(i)}, x). \tag{4.3}$$

In the case of Dirichlet priors the transition updates are given by

$$p_{lj}^{(i+1)} = \frac{\xi^{(i)}(l,j) + (\alpha_{lj} - 1)}{\sum_j \xi^{(i)}(l,j) + (\alpha_l - K)}, \quad \text{where} \quad \xi^{(i)}(l,j) := \sum_{t=1}^{n-1} P(Y_t = l, Y_{t+1} = j|x, \theta^{(i)}). \tag{4.4}$$

Since one of the studied methods (ICM) starts with an initial sequence, in order the comparison to be fair, we let all the other methods to start with a sequence as well. Therefore, for a given initial sequence $y^{(0)}$, define

$$\gamma_t^{(0)}(k) := I_k(y_t^{(0)}), \quad \xi^{(0)}(l,j) := n_{lj}(y^{(0)}). \tag{4.5}$$

Variational Bayes approach. Let us have a closer look at the measure $q_Y^{(i+1)}(y)$. We are going to show that there exists an HMM (Z, X) such that for every sequence y, $q_Y^{(i+1)}(y) = P(Z = y|X = x)$. By definition,

$$q_Y^{(i+1)}(y) \propto \exp\Big[\int \ln p(\theta, y|x) q_\theta^{(i+1)}(d\theta) \Big].$$

Apply the notation from (2.2) in the current case:

$$u_{lj}^{(i+1)} = \exp[\int \ln p_{lj}(\theta_{tr}) q_\theta^{(i+1)}(d\theta)], \quad h_k^{(i+1)}(x_t) = \exp[\int \ln f_k(x_t|\theta_{em}^k) q_\theta^{(i+1)}(d\theta)].$$

Since $\ln p(\theta, y|x) = \ln \pi(\theta) + \ln p(y, x|\theta) - \ln p(x)$, we obtain

$$\begin{aligned}
&\int \ln p(\theta, y|x) q_\theta^{(i+1)}(d\theta) = \int \ln \pi(\theta) q_\theta^{(i+1)}(d\theta) - \ln p(x) + \int \ln p(y, x|\theta) q_\theta^{(i+1)}(d\theta) \\
&= c(q_\theta^{(i+1)}, x) + \ln p_{0y_1} + \sum_{lj} n_{lj}(y) \ln u_{lj}^{(i+1)} + \sum_{k=1}^K \sum_{t: y_t = k} \ln h_k^{(i+1)}(x_t) \\
&= c(q_\theta^{(i+1)}, x) + \ln p_{0y_1} + \sum_{lj} n_{lj}(y) \ln \tilde{u}_{lj}^{(i+1)} + \sum_{k=1}^K \sum_{t: y_t = k} \ln \tilde{h}_k^{(i+1)}(x_t),
\end{aligned} \tag{4.6}$$

where $\tilde{u}_{lj}$ is the normalized quantity, $\tilde{u}_{lj} := \frac{u_{lj}}{\sum_j u_{lj}}$, and $\tilde{h}_k(x_t) := (\sum_j u_{kj}) h_k(x_t)$, if $t \le n-1$, $\tilde{h}_k(x_n) = h_k(x_n)$. Let now (Z, X) be an HMM, where Z is the underlying Markov chain with transition matrix $(\tilde{u}_{lj})$ and emission densities are given by $\tilde{h}_k$. From (4.6) it follows that $q_Y^{(i+1)}(y) \propto P(Z = y|X = x)$. Since $q_Y^{(i+1)}$ and $P(Z \in \cdot|X = x)$ are both probability measures, it follows that they are equal. To stress the dependence on iterations, we will denote $q_Y^{(i+1)}(y) = P^{(i+1)}(Z = y|X = x)$.

Let us now calculate q_θ. Let $\gamma_t^{(i)}(k)$ denote the marginal of $q_Y^{(i)}(y)$,

$$\gamma_t^{(i)}(k) := P^{(i)}(Z_t = k|X = x) = \sum_{y: y_t = k} q_Y^{(i)}(y).$$

Observe that

$$\sum_y \ln p(y,x|\theta) q_Y^{(i)}(y) = C_1 + \sum_{l,j} \ln p_{lj}(\theta_{tr}) \Big(\sum_y n_{lj}(y) q_Y^{(i)}(y) \Big) + \sum_{t=1}^{n} \sum_{k=1}^{K} \ln f_k(x_t|\theta_{em}^k) \gamma_t^{(i)}(k),$$

where $C_1 := \sum_k (\ln p_{0k}) \gamma_1^{(i)}(k)$. The sum $\sum_y n_{lj}(y) q_Y^{(i)}(y)$ is the expected number of transitions from l to j, so that using the equality $q_Y^{(i)}(y) = P^{(i)}(Z = y|X = x)$, we have

$$\sum_y n_{lj}(y) q_Y^{(i)}(y) = \sum_{t=1}^{n-1} P^{(i)}(Z_t = l, Z_{t+1} = j|X = x) =: \xi^{(i)}(l,j).$$

Therefore,

$$\ln q_\theta^{(i+1)}(\theta) = C + \ln \pi_{tr}(\theta_{tr}) + \ln \pi_{em}(\theta_{em}) + \sum_{l,j} \xi^{(i)}(l,j) \ln p_{lj}(\theta_{tr}) + \sum_{t=1}^{n} \sum_{k=1}^{K} \ln f_k(x_t|\theta_{em}^k) \gamma_t^{(i)}(k). \quad (4.7)$$

From (4.7) we can see that under $q_\theta^{(i+1)}$ the parameters $\theta_{tr}, \theta_{em}^1, \ldots, \theta_{em}^K$ are still independent and can therefore be updated separately. In the case of Dirichlet transition priors the rows are independent as well. The transition update for the l-th row and the emission update for the k-th component are given by

$$q_\theta^{(i+1)}(p_{l1}, \ldots, p_{lK}) \propto \prod_{j=1}^{K} p_{lj}^{\alpha_{lj} - 1 + \xi^{(i)}(l,j)}, \quad q_\theta^{(i+1)}(\theta_{em}^k) \propto \pi(\theta_{em}^k) \prod_{t=1}^{n} \big(f_k(x_t|\theta_{em}^k) \big)^{\gamma_t^{(i)}(k)}.$$

The whole VB approach is applicable since $\xi^{(i)}(l,j)$ and $\gamma_t^{(i)}(k)$ can be found by the standard forward-backward formulas using $\tilde{u}_{lj}^{(i)}$ and $\tilde{h}_k^{(i)}$. Actually, it is not difficult to see that in these formulas the original $u_{lj}^{(i)}$ and $h_k^{(i)}$ can be used instead of the standardized ones. To summarize, in our setup the VB approach yields the following algorithm for calculating $\hat{v}_{\mathrm{VB}}$. For a given initial sequence $y^{(0)}$, find vector $\gamma_t^{(0)}$ and matrix $\xi^{(0)}$ as in (4.5). Given $\gamma_t^{(i)}$ and $\xi^{(i)}$, update $u_{lj}^{(i+1)}$ and $h_k^{(i+1)}$ as follows:

$$u_{lj}^{(i+1)} = \exp[\psi(\alpha_{lj} + \xi^{(i)}(l,j)) - \psi(\alpha_l + \xi^{(i)}(l))], \quad \text{where} \quad \xi^{(i)}(l) := \sum_j \xi^{(i)}(l,j), \quad (4.8)$$

$$h_k^{(i+1)}(x_t) = \exp[\int \ln f_k(x_t|\theta_{em}^k) q_\theta^{(i+1)}(d\theta)], \quad \text{where} \quad q_\theta^{(i+1)}(\theta_{em}^k) \propto \pi(\theta_{em}^k) \prod_{t=1}^{n} \big(f_k(x_t|\theta_{em}^k) \big)^{\gamma_t^{(i)}(k)}.$$

With these parameters, $\xi^{(i+1)}$ and $\gamma_t^{(i+1)}$ can be calculated with the usual forward-backward procedure for HMM. Then update $u_{lj}^{(i+2)}$ and $h_k^{(i+2)}$ and so on. After the convergence, say after m steps, apply the Viterbi algorithm with transitions $(u_{ij}^{(m)})$ and emission densities $h_k^{(m)}$. The obtained path maximizes $q_Y^{(m)}(y)$ over all the paths, so it is $\hat{v}_{\mathrm{VB}}$.

Simulated annealing. Because of independence of the emission and transition parameters, it holds even for $\beta > 1$ that $p_\beta(\theta|y,x) = p_\beta(\theta_{tr}|y) p_\beta(\theta_{em}|y,x)$, thus the transition and emission parameters can be sampled separately. When the rows of a transition matrix have independent Dirichlet priors, the l-th row

can be generated from Dirichlet distribution with parameters $\beta(n_{lk}(s)+\alpha_{lk})+1-\beta$, $k=1,\dots,K$. For given θ, sampling from $p(y|\theta,x)$ can be performed in various ways, we use so-called *Markovian Backward Sampling* (Algorithm 6.1.1 in [5]). To sample from $p_\beta(y|\theta,x)$, note that

$$p(x,y|\theta)^\beta = \frac{p_{0y_1}^\beta}{\sum_j p_{0,j}^\beta}\prod_{t=2}^n \tilde{p}_{y_{t-1}y_t}\tilde{f}_{y_t}(x_t),$$

where $\tilde{p}_{ij} := p_{ij}^\beta/\sum_j p_{ij}^\beta$, and $\tilde{f}_k(x_t) := \big(\sum_j p_{ij}^\beta\big) f_k^\beta(x_t)$, $t=1,\dots,n-1$, $\tilde{f}_k(x_n) := \big(\sum_j p_{0j}^\beta\big) f_k^\beta(x_n)$. Although the functions $\tilde{f}_k$ are not densities, one can still use Markovian Backward Sampling.

4.2 Non-stochastic algorithms for the Dirichlet-NIX case

Suppose the emission distribution corresponding to state k is $\mathcal{N}(\mu_k,\sigma_k^2)$, where prior distributions for μ_k and σ_k^2 are given by a normal and scaled inverse-chi-square distribution, respectively:

$$\mu_k|\sigma_k^2 \sim \mathcal{N}\big(\xi_k, \frac{\sigma_k^2}{\kappa_0}\big), \quad \sigma_k^2 \sim \text{Inv}{-}\chi^2(\nu_0,\tau_0^2).$$

Here κ_0, ν_0 and τ_0^2 are hyperparameters that might depend on k, but in our example we assume they are equal. Recall the density of $\text{Inv}{-}\chi^2(\nu,\tau^2)$:

$$f(x;\nu,\tau^2) = \frac{(\tau^2\nu/2)^{\nu/2}}{\Gamma(\nu/2)}x^{-(1+\nu/2)}\exp[-\frac{\nu\tau^2}{2x}].$$

If $X\sim \text{Inv}{-}\chi^2(\nu,\tau^2)$, then

$$EX = \frac{\tau^2\nu}{\nu-2}, \quad \text{Var}(X) = \frac{2\tau^4\nu^2}{(\nu-2)^2(\nu-4)}, \quad E(\ln X) = \ln\big(\frac{\nu\tau^2}{2}\big) - \psi\big(\frac{\nu}{2}\big), \quad EX^{-1} = \tau^{-2},$$

and the mode of the distribution is given by $\nu\tau^2/(\nu+2)$. Therefore, if ν_0 and κ_0 are both very large, then $\sigma_k^2 \approx \tau_0^2$ and $\mu_k \approx \xi_k$, and we get back to the first example. If ν_0 is very large, then $\sigma_k^2 \approx \tau_0^2$, so that emission variances are τ_0^2, but the variance of the mean is approximately τ_0^2/κ_0.

Since emission and transition parameters are independent, the transition parameters can be updated as previously, that is as described in Section 4.1. Because the emission components $(\theta_{em}^1,\dots,\theta_{em}^K)$ are independent under prior and posterior, it holds that $p(\theta_{em}|y,x) = \prod_k p(\theta_{em}^k|x_{S_k})$, where x_{S_k} is the subsample of x along y corresponding to state k. Let $m_k(y)$ be the size of x_{S_k}. Let $\bar{x}_k$ and s_k^2 be the mean and variance of x_{S_k}. Since NIX-priors are conjugate, for any state k the posterior parameters κ_k, ν_k, μ_k and τ_k^2 can be calculated as follows:

$$\kappa_k = \kappa_0 + m_k\,, \qquad \nu_k = \nu_0 + m_k, \tag{4.9}$$

$$\mu_k = \frac{\kappa_0}{\kappa_0+m_k}\xi_k + \frac{m_k}{\kappa_0+m_k}\bar{x}_k, \tag{4.10}$$

$$\nu_k\tau_k^2 = \nu_0\tau_0^2 + (m_k-1)s_k^2 + \frac{\kappa_0 m_k}{\kappa_0+m_k}(\bar{x}_k-\xi_k)^2, \quad s_k^2 = \frac{1}{m_k-1}\sum_{t\in S_k}(x_t-\bar{x}_k)^2, \tag{4.11}$$

see [25]. We also need to calculate for every path y the joint probability $p(x,y) = p(y)p(x|y)$. Due to the independence of transition and emission parameters, $p(y)$ is still as in (1.6) and $p(x|y)$ depends on

emission parameters, only. According to the formula for the marginal likelihood (see, e.g. [25]) we obtain

$$p(x|y) = \prod_{k=1}^{K} \int \prod_{t \in S_k} f_k(x_t|\theta_{em}^k)\pi(\theta_{em}^k)d\theta_{em}^k = \prod_{k=1}^{K} \frac{\Gamma(\frac{\nu_k}{2})}{\Gamma(\frac{\nu_0}{2})} \sqrt{\frac{\kappa_0}{\kappa_k}} \frac{(\nu_0\tau_0^2)^{\frac{\nu_0}{2}}}{(\nu_k\tau_k^2)^{\frac{\nu_k}{2}}} \pi^{-\frac{m_k}{2}}. \tag{4.12}$$

We will now give a more detailed description of the non-stochastic algorithms for Example 2.

Bayesian EM. Start with initial state sequence $y^{(0)}$. With this sequence, find for any state k the parameters κ_k, μ_k, ν_k, τ_k^2 as defined in (4.9), (4.10), (4.11) and calculate the posterior modes, that is update

$$p_{lj}^{(1)} = \frac{n_{lj}(y^{(0)}) + (\alpha_{lj} - 1)}{\sum_j n_{lj}(y^{(0)}) + (\alpha_l - K)}, \quad \mu_k^{(1)} = \mu_k, \quad (\sigma_k^2)^{(1)} = \frac{\nu_k\tau_k^2}{\nu_k + 2}, \quad k = 1 \ldots, K.$$

With these parameters calculate the vectors $\gamma_t^{(1)}$ and matrix $(\xi^{(1)}(l,j))$ as in (4.3) and (4.4) using the forward-backward formulas. Given $\gamma_t^{(i)}$ and $\xi^{(i)}(l,j)$, the transition parameters are updated according to (4.4). The emission updates are given by (4.2). Let us calculate $\theta_{em}^{k,(i+1)}$ for the NIX-model. Suppress k from the notation and observe that $\theta_{em}^{(i+1)} = (\mu^{(i+1)}, (\sigma^2)^{(i+1)})$ maximizes the following function over μ and σ^2:

$$\sum_t \ln f(x_t|\mu,\sigma^2)\gamma_t^{(i)} + \ln \pi(\mu|\sigma^2) + \ln \pi(\sigma^2) =$$
$$\text{const} - \frac{1}{2}\Big[(\ln\sigma^2)\big(\sum_t \gamma_t^{(i)} + (\nu_0 + 3)\big) + \frac{1}{\sigma^2}\big(\sum_t (x_t - \mu)^2\gamma_t^{(i)} + (\mu - \xi)^2\kappa_0 + \nu_0\tau_0^2\big)\Big].$$

The solutions $\mu_k^{(i+1)}$ and $(\sigma_k^2)^{(i+1)}$ are given by:

$$\mu_k^{(i+1)} = \frac{\sum_t x_t\gamma_t^{(i)}(k) + \xi_k\kappa_0}{\sum_t \gamma_t^{(i)}(k) + \kappa_0}, \quad (\sigma_k^2)^{(i+1)} = \frac{\nu_0\tau_0^2 + \sum_t (x_t - \mu_k^{(i+1)})^2\gamma_t^{(i)}(k) + (\mu_k^{(i+1)} - \xi_k)^2\kappa_0}{\sum_t \gamma_t^{(i)}(k) + \nu_0 + 3}.$$

With $\kappa_0 \to 0$ (non-informative prior), $\mu_k^{(i+1)}$ is the same as in the standard EM algorithm. Using the updated parameters, calculate $\gamma_t^{(i+1)}$ and $\xi^{(i+1)}(l,j)$. Keep updating until the change in the log-likelihood is below the stopping criterion.

Segmentation EM. Given sequence $y^{(i)}$, calculate for every state k the parameters $\kappa_k^{(i)}$, $\mu_k^{(i)}$, $\nu_k^{(i)}$ and $(\tau_k^2)^{(i)}$ using formulas (4.9), (4.10) and (4.11). With these parameters, calculate $h_k^{(i+1)}(x_t)$ as follows:

$$\ln h_k^{(i+1)}(x_t) = -\frac{1}{2}\ln\left(2\pi(\tau_k^2)^{(i)}\right) - \frac{1}{2}\left[\ln\left(\frac{\nu_k^{(i)}}{2}\right) - \psi\left(\frac{\nu_k^{(i)}}{2}\right)\right] - \frac{x_t^2}{2\left(\tau_k^2\right)^{(i)}} + x_t\frac{\mu_k^{(i)}}{\left(\tau_k^2\right)^{(i)}} - \frac{1}{2}\left[\frac{1}{\kappa_k^{(i)}} + \left(\frac{\mu_k^{(i)}}{\tau_k^{(i)}}\right)^2\right]. \tag{4.13}$$

Compute the matrix $(u_{lj}^{(i+1)})$, where $\ln u_{lj}^{(i+1)} = \psi(\alpha_{lj} + n_{lj}(y^{(i)})) - \psi(\alpha_l + n_l(y^{(i)}))$. To find $y^{(i+1)}$, apply the Viterbi algorithm with $u_{lj}^{(i+1)}$ and $h_k^{(i+1)}(x_t)$. Keep doing so until no changes occur in the path estimate.

Segmentation MM. Given $y^{(i)}$, calculate $\mu_k^{(i)}$, $\nu_k^{(i)}$ and $(\tau_k^2)^{(i)}$ using formulas (4.9), (4.10) and (4.11) and update the posterior modes as follows:

$$p_{lj}^{(i+1)} = \frac{n_{lj}(y^{(i)}) + (\alpha_{lj} - 1)}{\sum_j n_{lj}(y^{(i)}) + (\alpha_l - K)}, \quad \mu_k^{(i+1)} = \mu_k^{(i)}, \quad (\sigma_k^2)^{(i+1)} = \frac{\nu_k^{(i)}}{\nu_k^{(i)} + 2}(\tau_k^2)^{(i)}.$$

With these parameters find $y^{(i+1)}$ by the Viterbi algorithm. Keep doing so until no changes occur in the estimated state path.

VB algorithm. Given initial state sequence $y^{(0)}$, find $h_k^{(1)}(x_t)$ and $u_{lj}^{(1)}$ like in the segmentation EM algorithm. With these parameters, calculate $\gamma_t^{(1)}$ and $\xi^{(1)}(l,j)$ using the forward-backward formulas. Given the matrix $\big(\xi^{(i)}(l,j)\big)$, update the matrix $(u_{lj}^{(i+1)})$ according to (4.8). Given $\gamma_t^{(i)}(k)$, the parameters $\kappa_k^{(i)}$, $\mu_k^{(i)}$, $\nu_k^{(i)}$ and $(\tau_k^2)^{(i)}$ can be calculated by (see, e.g., [23])

$$\kappa_k^{(i)} = \kappa_0 + g_k^{(i)}, \quad \nu_k^{(i)} = \nu_0 + g_k^{(i)}, \quad g_k^{(i)} = \sum_{t=1}^n \gamma_t^{(i)}(k),$$

$$\mu_k^{(i)} = \frac{\kappa_0}{\kappa_0 + g_k^{(i)}}\xi_k + \frac{g_k^{(i)}}{\kappa_0 + g_k^{(i)}}\tilde{x}_k^{(i)}, \quad \tilde{x}_k^{(i)} = \frac{1}{g_k^{(i)}}\sum_{t=1}^n \gamma_t^{(i)}(k)x_t,$$

$$\nu_k^{(i)}(\tau_k^2)^{(i)} = \nu_0\tau_0^2 + \sum_{t=1}^n (x_t - \tilde{x}_k^{(i)})^2\gamma_t^{(i)}(k) + \frac{\kappa_0 g_k^{(i)}}{\kappa_0 + g_k^{(i)}}(\tilde{x}_k^{(i)} - \xi_k)^2.$$

Compute then $h_k^{(i+1)}(x_t)$ as in (4.13). With help of $h_k^{(i+1)}(x_t)$ and $u_{lj}^{(i+1)}$, find $\gamma_t^{(i+1)}$ and $\xi^{(i+1)}(l,j)$ using the forward-backward formulas. After that update $h_k^{(i+2)}(x_t)$ and $u_{lj}^{(i+2)}$ and so on. When the VB algorithm has converged, say after m steps, apply the Viterbi algorithm with $u_{lj}^{(m)}$ as transitions and with $h_k^{(m)}(x_t)$ as emission values.

4.3 NIX clustering formulas

From (4.12) it follows that for any sequence y', the likelihood ratio is given by

$$\frac{p(x|y)}{p(x|y')} = \prod_{k=1}^K \frac{\Gamma(\frac{\nu_0+m_k}{2})}{\Gamma(\frac{\nu_0+m'_k}{2})} \prod_{k=1}^K \sqrt{\frac{\kappa_0 + m'_k}{\kappa_0 + m_k}} \prod_{k=1}^K \frac{(\nu'_k\tau_k'^2)^{\frac{\nu_0}{2}}}{(\nu_k\tau_k^2)^{\frac{\nu_0}{2}}} \prod_{k=1}^K \frac{(\nu'_k\tau_k'^2)^{\frac{m'_k}{2}}}{(\nu_k\tau_k^2)^{\frac{m_k}{2}}}. \tag{4.14}$$

When $\nu_0 \to \infty$ and $\tau_0^2 > 0$, then due to $\sum_k m_k = \sum_k m'_k = n$ we have

$$\lim_{\nu_0\to\infty} \prod_{k=1}^K \frac{\Gamma(\frac{\nu_0+m_k}{2})}{\Gamma(\frac{\nu_0+m'_k}{2})} = 1, \quad \lim_{\nu_0\to\infty} \prod_{k=1}^K \frac{(\nu'_k\tau_k'^2)^{\frac{m'_k}{2}}}{(\nu_k\tau_k^2)^{\frac{m_k}{2}}} = 1.$$

Write $\nu_k\tau_k^2$ as

$$\nu_k\tau_k^2 = \nu_0\tau_0^2 + \sum_{t\in S_k}(x_t - \bar{x}_k)^2 + \frac{\kappa_0 m_k}{\kappa_0 + m_k}(\bar{x}_k - \xi_k)^2 = \nu_0\tau_0^2 + A_k = \nu_0\tau_0^2\left(1 + \frac{2A_k}{2\nu_0\tau_0^2}\right).$$

Then

$$\frac{(\nu'_k\tau'^2_k)^{\frac{\nu_0}{2}}}{(\nu_k\tau_k^2)^{\frac{\nu_0}{2}}} = \frac{\left(1+\frac{2A'_k}{2\nu_0\tau_0^2}\right)^{\frac{\nu_0}{2}}}{\left(1+\frac{2A_k}{2\nu_0\tau_0^2}\right)^{\frac{\nu_0}{2}}} \to \exp\left[\frac{A'_k - A_k}{2\tau_0^2}\right].$$

Therefore, when $\nu_0 \to \infty$, then the likelihood ratio in (4.14) converges to

$$\prod_{k=1}^{K}\sqrt{\frac{\kappa_0+m'_k}{\kappa_0+m_k}}\exp\left[\frac{\sum_k A'_k - \sum_k A_k}{2\tau_0^2}\right].$$

Thus, maximizing $p(x|y)$ corresponds to the following clustering problem: find clusters $S_1,\dots,S_k$ that minimize

$$\sum_{k=1}^{K}\sum_{t\in S_k}(x_t-\bar{x}_k)^2 + \kappa_0\sum_{k=1}^{K}\frac{m_k}{\kappa_0+m_k}(\bar{x}_k-\xi_k)^2 + \tau_0^2\sum_{k=1}^{K}\ln(\kappa_0+m_k). \tag{4.15}$$

Formula (4.15) is the right hand side of (3.2). Given cluster S_k, it is easy to see that

$$\arg\min_{\mu\in\mathbb{R}}\Big[\sum_{t\in S_k}(x_t-\mu)^2 + \kappa_0(\mu-\xi_k)^2\Big] = \frac{m_k\bar{x}_k+\kappa_0\xi_k}{\kappa_0+m_k} =: \mu_k. \tag{4.16}$$

Since

$$\sum_{t\in S_k}(x_t-\mu_k)^2 + \kappa_0(\mu_k-\xi_k)^2 = \sum_{t\in S_k}(x_t-\bar{x}_k)^2 + \kappa_0\frac{m_k}{\kappa_0+m_k}(\bar{x}_k-\xi_k)^2, \tag{4.17}$$

we obtain the left hand side of (3.2).

To understand the behavior of the segmentation EM and segmentation MM algorithms in that case, recall the segmentation EM iteration formula from (4.13). When $\nu_0 \to \infty$, then $\ln(\nu_k^{(i)}/2) - \psi(\nu_k^{(i)}/2) \to 0$ and $(\tau_k^2)^{(i)} \to \tau_0^2$. Thus, leaving the superscript (i) out of the notation, we get

$$\ln h_k(x_t) \to -\frac{1}{2}\ln\left(2\pi(\tau_0^2)\right) - \frac{1}{2\tau_0^2}\big(x_t-\mu_k\big)^2 - \frac{1}{2(\kappa_0+m_k)},$$

where μ_k is as in (4.16). The Viterbi alignment is now obtained as

$$y_t = \arg\min_{k=1,\dots,K}\left[\big(x_t-\mu_k\big)^2 + \frac{\tau_0^2}{m_k+\kappa_0}\right].$$

Acknowledgments

This work is supported by Estonian institutional research funding IUT34-5.

References

[1] M. Beal. *Variational algorithms for approximate Bayesian inference.* PhD thesis, Gatsby Computational Neurosience unit, University College London, 2003.

[2] M. Beal and Z. Ghahramani. The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. In *Bayesian statistics, 7 (Tenerife, 2002)*, pages 453–463. Oxford Univ. Press, New York, 2003.

[3] J. Besag. An introduction to Markov chain Monte Carlo methods. In *Mathematical foundations of speech and language processing*, volume 138 of *IMA Vol. Math. Appl.*, pages 247–270. Springer, New York, 2004.

[4] C. Bishop. *Pattern recognition and machine learning*. Information Science and Statistics. Springer, New York, 2006.

[5] O. Cappé, E. Moulines, and T. Rydén. *Inference in hidden Markov models*. Springer, New York, 2005.

[6] L. Carvalho and C. Lawrence. Centroid estimation in discrete high-dimensional spaces with applications in biology. *Proceedings of the National Academy of Sciences*, 105(9):3209–3214, 2008.

[7] C. Christodoulopoulos, S. Goldwater, and M. Steedman. Two decades of unsupervised POS induction: How far have we come? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2010.

[8] J. Corander, J. Xiong, Y. Cui, and T Koski. Optimal Viterbi Bayesian predictive classification for data from finite alphabets. *J. Statist. Plann. Inference*, 143(2):261–275, 2013.

[9] C. Fox and S. Roberts. A tutorial on variational Bayesian inference. *Artificial intelligence review*, 38(2):85–95, 2012.

[10] K. Ganchev, B. Taskar, P. Fernando, and G. Joao. Posterior vs parameter sparsity in latent variable models. In *Advances in Neural Information Processing Systems 22*. 2009.

[11] J. Gao and M. Johnson. A comparison of Bayesian estimators for unsupervised hidden Markov model pos taggers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 344–352, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[12] S. Goldwater and T. Griffiths. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 2007.

[13] N.L. Hjort, C. Holmes, P. Müller, and S.G. Walker, editors. *Bayesian Nonparametrics*. Cambridge University Press, New York, 2010.

[14] M. Johnson. Why doesn't EM find good HMM POS-taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296–305, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[15] M. Jordan, Z. Gharhami, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

[16] A. Koloydenko, M. Käärik, and J. Lember. On adjusted Viterbi training. *Acta Appl. Math.*, 96(1-3):309–326, 2007.

[17] A. Koloydenko and J. Lember. Bridging Viterbi and posterior decoding: A generalized risk approach to hidden path inference based on hidden Markov models. *Journal of Machine Learning Research*, 15:1–58, 2014.

[18] T. Koski. *Hidden Markov Models for Bioinformatics*, volume 2 of *Computational Biology Series*. Kluwer Academic Publishers, Dordrecht, 2001.

[19] J. Lember and A. Koloydenko. Adjusted Viterbi training: A proof of concept. *Probab. Eng. Inf. Sci.*, 21(3):451 – 475, 2007.

[20] J. Lember and A. Koloydenko. The Adjusted Viterbi training for hidden Markov models. *Bernoulli*, 14(1):180–206, 2008.

[21] J. Lember, K. Kuljus, and A. Koloydenko. Theory of segmentation. In P. Dymarsky, editor, *Hidden Markov Models, Theory and Applications*, pages 51–84. InTech, 2011.

[22] J. Marin and C. Robert. *Bayesian core: a practical approach to computational Bayesian statistics*. Springer Texts in Statistics. Springer, New York, 2007.

[23] C. A. McGrory and D. M. Titterington. Variational Bayesian analysis for hidden Markov models. *Aust. N. Z. J. Stat.*, 51(2):227–244, 2009.

[24] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley, Hoboken, NJ, 2008.

[25] K. Murphy. Conjugate Bayesian analysis of the Gaussian distribution. Technical report, 2007.

[26] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[27] T. Rydén. EM versus Markov chain Monte Carlo for estimation of hidden Markov models: a computational perspective. *Bayesian Anal.*, 3(4):659–688, 2008.

[28] A. Samé, C. Ambroise, and G. Govaert. A classification EM algorithm for binned data. *Comput. Statist. Data Anal.*, 51(2):466–480, 2006.

[29] A. Samé, C. Ambroise, and G. Govaert. An online classification EM algorithm based on the mixture model. *Stat. Comput.*, 17(3):209–218, 2007.

[30] S. Scott. Bayesian methods for hidden Markov models: recursive computing in the 21st century. *J. Amer. Statist. Assoc.*, 97(457):337–351, 2002.

[31] V. Smidl and A. Quinn. *The Variational Bayes Method in Signal Processing*. Springer, 2006.

[32] S. Watanabe. *Algebraic geometry and statistical learning theory*, volume 25 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2009.

[33] G. Winkler. *Image analysis, random fields and Markov chain Monte Carlo methods*, volume 27 of *Applications of Mathematics*. Springer-Verlag, Berlin, 2003.

[34] C. Yau and C. Holmes. A decision-theoretic approach for segmental classification. *Ann. Appl. Stat.*, 7(3):1814–1835, 2013.