Variational Bayes Variants of the Viterbi Algorithm

Viet Hung Tran [†] **and Anthony Quinn**[*] [1]

Department of Electronic and Electrical Engineering
Trinity College Dublin

E-mail: [†]`tranv@tcd.ie` [*]`aquinn@tcd.ie`

Abstract — **The Viterbi algorithm (VA) is important in digital decoding because it greatly reduces the computational complexity of maximum a posteriori (MAP) estimation of an M-state, length-n, hidden Markov chain from $\mathcal{O}(M^n n)$ to $\mathcal{O}(M^2 n)$. In contrast, the Bayesian forward-backward algorithm computes the exact posterior state probabilities for each symbol, providing measures of decoding uncertainty. However, it has a greater complexity, at $\mathcal{O}(2M^2 n)$. In this paper, the Bayesian setting for digital decoding is reviewed, and the variational Bayes (VB) algorithm is used to compute approximate posterior state probabilities for the symbols. The resulting iterative algorithm is then specialized to the MAP case, in analogy to VA. With appropriate initialization, this novel VB-based decoding algorithm achieves decoding performance comparable to VA, but with a much lower complexity, at $\mathcal{O}(Mn)$.**

Keywords — **Viterbi algorithm, Forward-Backward, Variational Bayes, Hidden Markov model, MAP state estimator**

I Introduction

Any unknown distribution $f(x)$ can be modeled as a mixture of known components $f_l(x) \equiv f(x|l)$, i.e. $f(x) = \sum_l p_l f_l(x)$, where p_l is probability of discrete label l pointing to one of M finite states $k = \{1, \ldots, M\}$. The problem is to estimate the state k to which the observation x belongs, via the posterior inference $f(l|x)$ of label variable.

Given a batch of observations $\mathbf{x}_n = [x_1, \ldots, x_n]'$, the simplest case is independent labels l_i with known prior probability p_{l_i}, $i = \{1, \ldots, n\}$. By Bayes' rule, the inference problem is feasible to compute $f(l_i|\mathbf{x}_n, p) = f(l_i|x_i, p) \propto p_{l_i} f_{l_i}(x_i)$.

In first-order Hidden Markov model (HMM), both filtering $f(l_i|\mathbf{x}_i)$ and smoothing $f(l_i|\mathbf{x}_n)$ distributions of each label l_i are recursively tractable via step-wise update of Forward and Forward-Backward (FB) algorithm [1], respectively. Their recursive marginalizations, however, are still slow and become a serious problem in applications requiring a fast estimate method.

Hence the point-estimate Viterbi algorithm (VA) [2] was designed to directly evaluate the true maximum-a-posteriori (MAP) of joint trajectory: $\widehat{L_n} \equiv \underset{L_n}{\arg\max} f(L_n|\mathbf{x}_n)$.

By replacing marginalization with maximization, VA can be computed much faster than individual label's inference method. Owing to that efficiently recursive estimation, the application of VA is vast [3], e.g. speech recognition [4] and Turbo decoder [5]. Nevertheless, the performance of a joint trajectory estimate, which is generally different from a sequence of individual label's estimate, has not been well-studied in literature.

In this paper, the difference between those two sequences will be investigated. Because the label's MAP estimate via time-consuming Forward and FB algorithm is not much practical, we will use the Variational Bayes (VB) algorithm [6] to produce approximate VB-marginals $\widetilde{f}(l_i|\mathbf{x}_i)$ and $\widetilde{f}(l_i|\mathbf{x}_n)$ of filtering $f(l_i|\mathbf{x}_i)$ and smoothing $f(l_i|\mathbf{x}_n)$ distribution. In order to efficiently reduce the computational load, a functionally constrained VB (FCVB) will be adapted to produce two sequences of state's estimate, based on VB-filtering $\widehat{l_i} = \arg\max_{l_i} \widetilde{f}(l_i|\mathbf{x}_i)$ and VB-smoothing $\widehat{l_i} = \arg\max_{l_i} \widetilde{f}(l_i|\mathbf{x}_n)$. We will show that the estima-

[1] This research was supported by Science Foundation Ireland, grant 08/RFP/MTH1710.

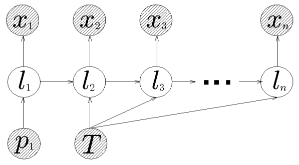

Figure 1: Directed acyclic graph (DAG) for HMM

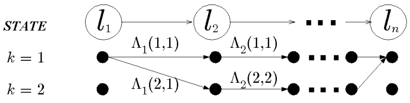

Figure 2: Trellis diagram. $\Lambda_i(j,k)$ is the element (j,k) of updated transition matrix Λ_i at time i

tion in FCVB can be evaluated faster than VA, with a slightly better accuracy in simulation.

II Hidden Markov Chain

Let us consider an M states HMM of known components in Fig. 1. The HMM's transition matrix $\mathbf{T}$ and initial distribution p_1 are also assumed known. At each time point $i = \{1, \ldots n\}$, we define a label vector $l_i \equiv [l_{1,i}, \ldots, l_{M,i}]' \in \{\epsilon(1), \ldots, \epsilon(M)\}$ pointing to a state $k = \{1, \ldots, M\}$, where $\epsilon(k) = [\delta\,(k-1)\,, \ldots, \delta\,(k-M)]'$.

We also collect the labels into an $M \times n$ matrix $L_n = [l_1, \ldots, l_n]$ and the whole observations into a vector $\mathbf{x}_n = [x_1, \ldots, x_n]' \in \mathbb{R}^n$, with $x_i = x[i] \in \mathbb{R}$ at discrete time point$i$. Because the components are known, we denote their value at time i as $f_k(x_i) = f(x_i|l_{k,i})$ for each state k and collect them into an $M \times 1$ vector $\boldsymbol{f}_{x_i} = [f_1(x_i), \ldots, f_M(x_i)]'$. Then the joint distribution in Fig. 1 is:

$$f(\mathbf{x}_n, L_n|\mathbf{T}, p_1) = f(\mathbf{x}_n|L_n)f(L_n|\mathbf{T}, p_1) \quad (1)$$
$$= \prod_{i=2}^{n} f(x_i, l_i|l_{i-1}, \mathbf{T})f(x_1, l_1|p_1)$$

where each transition probability is assigned by an element $t_{j,k}$ of the left stochastic matrix $\mathbf{T}$, i.e. $f(l_{j,i}|l_{k,i-1}, \mathbf{T}) = t_{j,k}$, $\sum_{j=1}^{M} t_{j,k} = 1$. Hence the observation model at each time belongs to Exponential Family (EF), as follows:

$$f(x_i, l_i|l_{i-1}, \mathbf{T}) = f(x_i|l_i)f(l_i|l_{i-1}, \mathbf{T}) \quad (2)$$
$$= \exp(-l_i'\Lambda_i l_{i-1}), \ i = 2, \ldots, n$$

where the sufficient statistics $\Lambda_i(x_i)$ are collected into $M \times M$ matrices of positive elements:

$$\Lambda_i = -(\log(\boldsymbol{f}_{x_i}\mathbf{1}_{1\times M}) + \log\mathbf{T}) \quad (3)$$

The conjugate prior for (2) is a Multinomial distribution $f(l_1|p_1) = Mu_{l_i}(p_1)$. Then we have:

$$f(x_1, l_1|p_1) = f(x_1|l_1)f(l_1|p_1) \quad (4)$$
$$= (l_1'\boldsymbol{f}_{x_1})Mu_{l_i}(p_1)$$
$$= Mu_{l_i}(\exp(-\Gamma_1))$$

in which $\Gamma_1 = -\log(\boldsymbol{f}_{x_1} \circ p_1)$ is a positive $M \times 1$ sufficient statistics vector and $\circ$ is Hadamard product. Then the joint distribution (1) can be reparameterized feasibly by the sum of sufficient statistics:

$$f(\mathbf{x}_n, L_n|\Lambda_i, \Gamma_1) = \exp(-(\sum_{i=2}^{n} l_i'\Lambda_i l_{i-1} + l_1'\Gamma_1)) \quad (5)$$

In literature, owing to the addictivity in (3, 5), a term "metric length" is often assigned to the positive elements of matrix Λ_i [2]. In a trellis diagram (Fig. 2), a metric length is often considered as an updated transition probability in (3), rather than its statistic role of EF in (2).

a) Viterbi algorithm (VA)

The purpose of VA is to compute efficiently the MAP estimate of joint state trajectory. By Bayes' rule, the MAP of trajectory can be defined as $\widehat{L}_n \equiv \arg\max_{L_n} f(L_n|\mathbf{x}_n, \mathbf{T}, p_1) = \arg\max_{L_n} f(\mathbf{x}_n, L_n|\mathbf{T}, p_1)$. Notice that, because the joint distribution (5) has a domain of M^n trajectories at time n, a direct computation of the MAP $\widehat{L}_n$ becomes impractical. Hence the VA was proposed to evaluate $\widehat{L}_n$ in a recursive procedure, such that only M potentially maximal trajectories are kept at each time.

The maximization of (1) can be expanded in the time-update and data-update context, as follows:

$$\max_{L_n} f(\mathbf{x}_n, L_n|\mathbf{T}, p_1) = \max_{l_n} f(x_n|l_n) \max_{L_{n-1}} f(\mathbf{x}_{n-1}, L_n|\mathbf{T}, p_1)$$
$$\propto \max_{l_n} f_p(l_n|\mathbf{x}_n) \quad (6)$$

in which we defined n profile distributions f_p as:

$$f_p(l_1|x_1) \equiv f(l_1|x_1, p_1) \propto f(x_1, l_1|p_1) \quad (7)$$

$$f_p(l_i|\mathbf{x}_i) \propto \max_{L_{i-1}} f(\mathbf{x}_i, L_i|\mathbf{T}, p_1), \ i = 2, \ldots, n \quad (8)$$

Because of conjugacy, expanding (8) yields a chain rule of maximization:

$$f_p(l_i|\mathbf{x}_i) \propto f(x_i|l_i) \max_{l_{i-1}} f(l_i|l_{i-1}, \mathbf{T})f_p(l_{i-1}|\mathbf{x}_{i-1})$$
$$\propto (l_i'\boldsymbol{f}_{x_i}) \max_{l_{i-1}} Mu_{l_i}(\mathbf{T}l_{i-1})Mu_{l_{i-1}}(\exp(-\Gamma_{i-1}))$$
$$= Mu_{l_i}(\exp(-\Gamma_i)), \ i = 2, \ldots, n \quad (9)$$

in which Γ_i are shaping parameters. From (4, 7, 9), we have:

$$\Gamma_1 = -\log(\boldsymbol{f}_{x_i} \circ p_1) \tag{10}$$

$$\Gamma_i = \min_{l_{i-1}}(\Lambda_i l_{i-1} + \Gamma_{i-1}) \tag{11}$$

By recursive maximization in (9), the VA reduces M^2 trajectories down to M potentially maximal trajectories. Moreover, because of minimization and summation in (11), each one of M positive elements of Γ_i is often called a "length" of one of those M "survivor" trajectories in literature [2], rather than its role of updated shaping parameters for profile distribution $f_p(l_i|\mathbf{x}_i)$ at each time.

From (6, 9), the MAP trajectory $\widehat{L_n}$ can be evaluated via a fast backward recursion. The last label's estimate is found first:

$$\widehat{l_n} = \arg\max_{l_n} f_p(l_n|\mathbf{x}_n) = \epsilon(\widehat{k_n}) \tag{12}$$

where $\widehat{k_n} = \arg\min_k(\Gamma_{k,n})$. Afterward, the previous labels leading to $\widehat{l_n}$ will be traced back, as following:

$$\widehat{l_{i-1}}(l_i) = \arg\max_{l_{i-1}} f(l_i|l_{i-1}, \mathbf{T}) f_p(l_{i-1}|\mathbf{x}_{i-1}) \tag{13}$$

with $i = n, \ldots, 2$.

The above VA, originally proposed by Viterbi [7] for decoding problem, was firstly formalized in trellis diagram of HMM in [2]. Because at time $i = 1$, an initial state $k = 1$ is often assigned to trellis diagram, i.e. $p_1 = \epsilon(1)$, the VA is often considered as Maximum Likelihood estimate, e.g. [5], instead of MAP estimate.

Notice that, the MAP trajectory may change entirely based on the last observation x_n, the VA is, therefore, an offline (batch-based) algorithm.

Viterbi algorithm (the same convention as [2])
Storage: $n-1$ vectors τ_i, $\tau_{k,i} = \{1, \ldots, M\}$.
Γ_i: one vector of shaping parameters at time i.
Initialization: initialize (10)
Recursion: For $i = 2, \ldots, n$:
evaluate (11), then from (13), evaluate:
$\tau_{j,i} = \arg\min_k(\Lambda_i(j,k) + \Gamma_{k,i-1})$
where $\Lambda_i(j,k)$ is the element at jth row and kth column of matrix Λ_i (Fig. 2)
Termination: Evaluate (12)
then trace the previous states: $\widehat{k_{i-1}} = \tau_{\widehat{k_i},i}$, $i = n, \ldots, 2$. Report the MAP of trajectory as: $\widehat{L_n} = [\epsilon(\widehat{k_1}), \ldots, \epsilon(\widehat{k_n})]$.
Complexity: $\mathcal{O}(M^2 n)$ of computations and $\mathcal{O}(Mn)$ of memory, owing to minimization on $M \times M$ matrices in (11) and backtracking (13).

b) Bayesian inference of label

In comparison with trajectory MAP of VA, we will evaluate the sequence of each label's estimate $\widehat{l_i}$, $i = 1, \ldots, n$. The estimate can be the mode of either filtering distribution $\widehat{l_i} \equiv \arg\max_{l_i} f(l_i|\mathbf{x}_i, \mathbf{T}, p_1)$ or smoothing distribution $\widehat{l_i} \equiv \arg\max_{l_i} f(l_i|\mathbf{x}_n, \mathbf{T}, p_1)$.

Owing to the conjugacy in (5), the posterior inference of each label is completely tractable, i.e. $f(l_i|\mathbf{x}_n, \mathbf{T}, p_1) \propto \sum_{L_{/i}} f(\mathbf{x}_n, L_n|\mathbf{T}, p_1)$ where $L_{/i}$ is the complement of l_i in L_n. A direct marginalization of (5) would, however, yield an exponential expansion of computational complexity $\mathcal{O}(M^n n)$ for all n label's smoothing distributions. For efficient computation, the label's inference will be evaluated recursively. The smoothing observation model can be factorized as follows:

$f(\mathbf{x}_n, l_1|\mathbf{T}, p_1) = f(\mathbf{x}_{2:n}|l_1, \mathbf{T}) f(x_1|l_1) f(l_1|p_1)$
$f(\mathbf{x}_n, l_i|\mathbf{T}, p_1) = f(\mathbf{x}_{i+1:n}|l_i, \mathbf{T}) f(x_i|l_i) f(l_i|\mathbf{x}_{i-1}, \mathbf{T}, p_1)$
$f(\mathbf{x}_n, l_n|\mathbf{T}, p_1) = f(x_n|l_n) f(l_n|\mathbf{x}_{n-1}, \mathbf{T}, p_1)$

with $i = 2, \ldots, n-1$. Owing to the conjugacy, we can derive feasibly two distinguished recursions, one for forward filtering:

$$f(l_1|x_1, p_1) \propto f(x_1, l_1|p_1) = Mu_{l_1}(\alpha_1) \tag{14}$$

$$\begin{aligned} f(l_i|\mathbf{x}_i, \mathbf{T}, p_1) &\propto f(x_i|l_i) \sum_{l_{i-1}} f(l_i|l_{i-1}, \mathbf{T}) f(l_{i-1}|\mathbf{x}_{i-1}, \mathbf{T}, p_1) \\ &= Mu_{l_i}(\alpha_i), \ i = 2, \ldots, n \end{aligned} \tag{15}$$

and one for backward observation:

$$\begin{aligned} f(\mathbf{x}_{i+1:n}|l_i, \mathbf{T}) &= \sum_{l_{i+1}} f(\mathbf{x}_{i+1:n}|l_{i+1}, \mathbf{T})(l_{i+1}|l_i, \mathbf{T}) \\ &= \beta_i' l_i, \ i = n-1, \ldots, 1 \end{aligned} \tag{16}$$

Given the initialization $\alpha_1 \propto \boldsymbol{f}_{x_1} \circ p_1$ and $\beta_n = \mathbf{1}_{M \times 1}$, the parameter's updates are:

$$\alpha_i \propto \exp(-\Lambda_i)\alpha_{i-1}, \ i = \{2, \ldots, n\} \tag{17}$$

$$\beta_{i-1} \propto \exp(-\Lambda_i')\beta_i, \ i = \{n, \ldots, 2\} \tag{18}$$

Substituting the results of forward and backward recursions (14-16) to $f(\mathbf{x}_n, l_i|\mathbf{T}, p_1)$, we have:

$f(l_i|\mathbf{x}_n, \mathbf{T}, p_1) = Mu_{l_i}(\alpha_i \circ \beta_i), \ i = 1, \ldots, n$

The above forward filtering is called Forward algorithm, which recursively evaluates the filtering distributions $f(l_i|\mathbf{x}_i, \mathbf{T}, p_1)$ and their mode $\widehat{l_i} = \arg\max_{l_i}(l_i'\alpha_i)$.

For smoothing distribution $f(l_i|\mathbf{x}_n, \mathbf{T}, p_1)$, the Forward-Backward (FB) algorithm, proposed by Baum et al. [1], consists of both forward filtering and backward observation steps. The MAP estimate can be evaluated by $\widehat{l_i} = \arg\max_{l_i}(l_i'(\alpha_i \circ \beta_i))$.

Because of marginalization for $M \times M$ matrices in (17, 18), the Forward and FB algorithm need $O(M^2 n)$, $O(2M^2 n)$ of computations and $\mathcal{O}(Mn)$, $\mathcal{O}(2Mn)$ of memory, respectively.

III Variational Bayesian approximation

a) Iterative VB (IVB) algorithm

Given a multivariate posterior distribution $f(\Theta|\mathbf{x})$ and a binary partition $\Theta = [\theta_i, \Theta_{/i}]$, where $\Theta_{/i}$ is complement of θ_i in Θ. The VB approximation is to seek $\tilde{f}(\Theta|\mathbf{x})$ in independent distribution class $\mathbb{F}_c$: $\breve{f}(\Theta|\mathbf{x}) = \breve{f}(\theta_i|\mathbf{x})\breve{f}(\Theta_{/i}|\mathbf{x})$ for which the Kullback-Leibler divergence $KLD(\tilde{f}(\Theta|\mathbf{x})||f(\Theta|\mathbf{x})) = E_{\tilde{f}(\Theta|\mathbf{x})} \log(\tilde{f}(\Theta|\mathbf{x})/f(\Theta|\mathbf{x}))$ is minimized. Given an arbitrarily initialized distribution $\widetilde{f}^{[0]}(\Theta_{/i}|\mathbf{x})$, the IVB algorithm [6] was proposed to update VB-marginals $\tilde{f}^{[\nu]}(\theta_i|\mathbf{x})$ in iterative cycles until converged, i.e. for $\nu = 1, 2 \ldots,$:

$$\widetilde{f}^{[\nu]}(\theta_i|\mathbf{x}) \propto \exp(E_{\widetilde{f}^{[\nu-1]}(\Theta_{/i}|\mathbf{x})} \log f(\Theta|\mathbf{x}))$$
$$\widetilde{f}^{[\nu]}(\Theta_{/i}|\mathbf{x}) \propto \exp(E_{\widetilde{f}^{[\nu]}(\theta_i|\mathbf{x})} \log f(\Theta|\mathbf{x}))$$

b) Functionally constrained VB (FVCB)

If VB-marginals $\widetilde{f}^{[\nu-1]}(\Theta_{/i}|\mathbf{x})$, $\widetilde{f}^{[\nu]}(\theta_i|\mathbf{x})$ are projected into their tractably constrained class $\widetilde{f}_\delta^{[\nu-1]}(\Theta_{/i}|\mathbf{x})$, $\widetilde{f}_\delta^{[\nu]}(\theta_i|\mathbf{x})$, respectively, before they are applied to expectation steps in IVB cycle, we have another approximation called FCVB approximation [6]. The well-known Expectation-Maximization (EM) algorithm is a special case of FCVB, where $\widetilde{f}^{[\nu]}(\theta_i|\mathbf{x})$ is projected to its certainty equivalence space via Dirac-δ function $\widetilde{f}_\delta^{[\nu]}(\theta_i|\mathbf{x}) = \delta(\theta_i - \widehat{\theta}_i)$, with $\widehat{\theta}_i = \arg\max_{\theta_i} \widetilde{f}(\theta_i|\mathbf{x})$ and $\widetilde{f}^{[\nu-1]}(\Theta_{/i}|\mathbf{x})$ is kept unchanged.

Similarly, an adaptation of FCVB will be proposed for state's estimate in HMM such that both VB-marginals $\widetilde{f}^{[\nu-1]}(\Theta_{/i}|\mathbf{x})$, $\widetilde{f}^{[\nu]}(\theta_i|\mathbf{x})$ will be projected into the space of their modes $\delta(\Theta_{/i} - \widehat{\Theta_{/i}}^{[\nu-1]})$, $\delta(\theta_i - \widehat{\theta}_i^{[\nu]})$, respectively. Via sifting property of $\delta(\cdot)$, the VB-marginals can be updated feasibly, as follows:

$$\widetilde{f}^{[\nu]}(\theta_i|\mathbf{x}) \propto f(\theta_i|\widehat{\Theta_{/i}}^{[\nu-1]}, \mathbf{x}) \tag{19}$$

$$\widetilde{f}^{[\nu]}(\Theta_{/i}|\mathbf{x}) \propto f(\Theta_{/i}|\widehat{\theta}_i^{[\nu]}, \mathbf{x}) \tag{20}$$

Hence, we only need two efficiently iterative maximization steps in each IVB cycle:

$$\widehat{\theta_{/i}}^{[\nu]} = \arg\max_{\theta_{/i}} f(\theta_i|\widehat{\Theta_{/i}}^{[\nu-1]}, \mathbf{x}) \tag{21}$$

$$\widehat{\Theta_{/i}}^{[\nu]} = \arg\max_{\Theta_{/i}} f(\Theta_{/i}|\widehat{\theta}_i^{[\nu]}, \mathbf{x}) \tag{22}$$

which yield a local MAP estimate of original distribution $f(\Theta|\mathbf{x})$ at the convergence.

c) Traditional FCVB for HMM

Let us define an independent class of n variables for label field $\mathbb{F}_C$: $\breve{f}(L_n|\mathbf{x}_n) = \prod_{i=1}^n \breve{f}(l_i|\mathbf{x}_n)$. The smoothing VB-marginals $\widetilde{f}(l_i|\mathbf{x}_n)$ is projected into their certainty equivalence space: $\widetilde{f}_\delta(l_i|\mathbf{x}_n) = \delta(l_i - \widehat{l}_i)$, where $\widehat{l}_i = \arg\max_{l_i} \widetilde{f}(l_i|\mathbf{x}_n)$ and $\delta(\cdot)$ is discrete Dirac-δ function. The smoothing VB-marginals in FCVB can be evaluated feasibly:

$$\begin{aligned}\widetilde{f}(l_i|\mathbf{x}_n) &\propto \exp(E_{\widetilde{f}_\delta(L_{/i}|\mathbf{x}_n)} \log f(\mathbf{x}_n, L_n|\mathbf{T}, p_1)) \\ &\propto f(\mathbf{x}_n, l_i|\widehat{L_{/i}}, \mathbf{T}, p_1)\end{aligned} \tag{23}$$

where $L_{/i}$ is complement of l_i in L_n and $\widetilde{f}_\delta(L_{/i}|\mathbf{x}_n) = \prod_{j \neq i} \widetilde{f}_\delta(l_j|\mathbf{x}_n) = \prod_{j \neq i} \delta(l_j - \widehat{l}_j)$, $j = \{1, \ldots, n\}$. Similar to smoothing's factorization in FB algorithm, each FCVB cycle (19, 20) consists of n updates:

$$\begin{aligned}\widetilde{f}^{[\nu]}(l_1|\mathbf{x}_n) &\propto f(x_2, \widehat{l}_2^{[\nu-1]}|l_1, \mathbf{T}) f(x_1|l_1) f(l_1|p_1) \\ &= Mu_{l_1}(\kappa_1^{[\nu]})\end{aligned} \tag{24}$$

$$\begin{aligned}\widetilde{f}^{[\nu]}(l_i|\mathbf{x}_n) \propto& f(x_{i+1}, \widehat{l_{i+1}}^{[\nu-1]}|l_i, \mathbf{T}) \\ &\times f(x_i|l_i) f(l_i|\widehat{l_{i-1}}^{[\nu]}, \mathbf{x}_{i-1}, \mathbf{T}, p_1) \\ =& Mu_{l_i}(\kappa_i^{[\nu]}),\ i = 2, \ldots, n-1\end{aligned} \tag{25}$$

$$\begin{aligned}\widetilde{f}^{[\nu]}(l_n|\mathbf{x}_n) &\propto f(x_n, l_n|\widehat{l_{n-1}}^{[\nu]}, \mathbf{x}_{n-1}, \mathbf{T}) \\ &= Mu_{l_n}(\kappa_n^{[\nu]})\end{aligned} \tag{26}$$

where $\widehat{l}_i^{[\nu]} = \arg\max_{l_i} \widetilde{f}^{[\nu]}(l_i|\mathbf{x}_n) = \arg\max_{l_i}(l_i' \kappa_i^{[\nu]})$, $i = 1, \ldots, n$. Similar to (21, 22), the shaping parameters $\kappa_i^{[\nu]}$ of $M \times 1$ dimension are updated via n maximization steps:

$$\kappa_1^{[\nu]} \propto (t'_{k_2^{[\nu-1]},:}) \circ \boldsymbol{f}_{x_1} \circ p_1 \tag{27}$$
$$k_1^{[\nu]} = \arg\max_k(\kappa_{k,1}^{[\nu]}) \tag{28}$$

$$\kappa_i^{[\nu]} \propto (t'_{k_{i+1}^{[\nu-1]},:}) \circ \boldsymbol{f}_{x_i} \circ (t_{:,k_{i-1}^{[\nu]}}) \tag{29}$$
$$k_i^{[\nu]} = \arg\max_k(\kappa_{k,i}^{[\nu]}),\ i = 2, \ldots n-1. \tag{30}$$

$$\kappa_n^{[\nu]} \propto \boldsymbol{f}_{x_n} \circ (t_{:,k_{n-1}^{[\nu]}}) \tag{31}$$
$$k_n^{[\nu]} = \arg\max_k(\kappa_{k,n}^{[\nu]}), \tag{32}$$

where $\circ$ is Hadamard product, $t_{k,:}$, $t_{:,k}$ are kth row, column of $\mathbf{T}$, respectively. Notice that the VB-moments, i.e. certainty equivalence $\widehat{l_i}^{[\nu]} = \epsilon(k_i^{[\nu]})$ in this case, are updated via propagation of smoothing's certainty equivalence in previous IVB cycle $\widehat{l_{i+1}}^{[\nu-1]}$ and filtering's certainty equivalence in current IVB cycle $\widehat{l_{i-1}}^{[\nu]}$. Hence the converged VB-marginal $\widetilde{f}^{[\nu_c]}(l_i|\mathbf{x}_n)$ at $\nu = \nu_c$ will become an approximation of smoothing distribution $f(l_i|\mathbf{x}_n, \mathbf{T}, p_1)$. This traditional FCVB algorithm will be called "$FCVB$ 1" in this paper.

"FCVB 1" algorithm

Storage: n scalars of state's value k_i

Initialization: initialize all $k_i^{[0]} \in \{1, \ldots, M\}$

Recursion: For $\nu = 1, \ldots,$ evaluate (27-32)

Termination: At convergence $\nu = \nu_c$, the label's VB-smoothing estimate is $\widehat{l_i} = \epsilon(k_i^{[\nu_c]})$, $i = 1, \ldots, n$

Complexity: Because of maximization on $M \times 1$ vector $\kappa_i^{[\nu]}$, we need $O(\nu M n)$ of computation for ν IVB cycles and a small amount $O(n)$ of memory.

d) *Filtering initialization for FCVB*

One way of increasing the quality of FCVB approximation is to design a proper initialization $\widetilde{f}_{\delta}^{[0]}(L_n|\mathbf{x}_n)$. In an HMM model, the sequence of label is conditionally generated from the previous label via transition matrix, i.e. $f(l_i|l_{i-1}, \mathbf{T})$. Hence, an arbitrary initialization $\widetilde{f}_{\delta}^{[0]}(l_i|\mathbf{x}_n)$, $i = 2, \ldots, n$, of "$FCVB$ 1" probably yields an unrealistic smoothing VB-moments $f(x_{i+1}, \widehat{l_{i+1}}^{[0]}|l_i, \mathbf{T})$ in (24-26).

A reasonable solution is to eliminate those initialized smoothing VB-moments. By this way, n steps of "$FCVB$ 1" algorithm (24-26), in the first IVB cycle $\nu = 1$, will become an FCVB approximation for filtering distribution $f(l_i|\mathbf{x}_i, \mathbf{T}, p_1)$, as following:

$$\widetilde{f}^{[1]}(l_1|x_1) \propto f(x_1, l_1|p_1) = Mu_{l_1}(\kappa_1^{[1]}) \tag{33}$$

$$\begin{aligned} \widetilde{f}^{[1]}(l_i|\mathbf{x}_i) &\propto \exp(E_{\widetilde{f}_{\delta}^{[1]}(L_{1:i-1}|\mathbf{x}_i)} \log f(\mathbf{x}_i, L_i|\mathbf{T}, p_1)) \\ &\propto f(x_i|l_i) f(l_i|\widehat{l_{i-1}}^{[1]}, \mathbf{x}_{i-1}, \mathbf{T}, p_1) \qquad (34) \\ &= Mu_{l_i}(\kappa_i^{[1]}), \ i = 2, \ldots, n-1 \end{aligned}$$

where $\widetilde{f}_{\delta}^{[1]}(L_{1:i-1}|\mathbf{x}_i) = \prod_{j=1}^{i-1} \widetilde{f}_{\delta}^{[1]}(l_j|x_j)$ are found in previous steps. The updated VB shaping parameters are:

$$\kappa_1^{[1]} = \boldsymbol{f}_{x_1} \circ p_1 \tag{35}$$

$$k_1^{[1]} = \arg\max_k(\kappa_{k,1}^{[1]}) \tag{36}$$

$$\kappa_i^{[1]} = \boldsymbol{f}_{x_i} \circ (t_{:,k_{i-1}^{[1]}}) \tag{37}$$

$$k_i^{[1]} = \arg\max_k(\kappa_{k,i}^{[1]}), \ i = 2, \ldots n. \tag{38}$$

with the same conventions in "$FCVB$ 1" scheme. From the second IVB cycle onward, the approximate $\widetilde{f}^{[1]}(l_i|\mathbf{x}_i)$ (33, 34) of filtering distribution $f(l_i|\mathbf{x}_i, \mathbf{T}, p_1)$ will be fed to FCVB algorithm (24-26) of "$FCVB$ 1". Hence this objective initialization is expected to yield a better FCVB approximation. This scheme of FCVB will be called "$FCVB$ 2" in this paper.

"FCVB 2" algorithm

Storage: The same as "$FCVB$ 1".

Initialization: For $\nu = 1$, evaluate (35-38)

Recursion: For $\nu = 2, \ldots$: same as "$FCVB$ 1"

Termination:

Report label's VB-filtering estimate: $\widehat{l_i} = \epsilon(k_i^{[1]})$, $i = 1, \ldots, n$ after first cycle.

Report label's VB-smoothing estimate: $\widehat{l_i} = \epsilon(k_i^{[\nu_c]})$, $i = 1, \ldots, n$ at convergence.

Complexity: The same as "$FCVB$ 1".

IV Simulation

We set up an HMM model of 3 Gaussian components $\mathcal{N}_{x_i}(-d, 1)$, $\mathcal{N}_{x_i}(0, 1)$ and $\mathcal{N}_{x_i}(d, 1)$, where d is the distance between their mean's value. The elements $t_{j,k}$ of 3×3 transition matrix $\mathbf{T}$ were generated by uniform distributions $U_{t_{j,k}}(0, 1)$, then the columns were marginalized. We chose $n = 256$, together with $p_1 = \frac{1}{3}\mathbf{1}_{3\times 1}$. For Forward and FB algorithm, we reported the sequences of MAP of filtering distributions $f(l_i|\mathbf{x}_i, \mathbf{T}, p_1)$ and smoothing distribution $f(l_i|\mathbf{x}_n, \mathbf{T}, p_1)$, respectively. For FCVB approximation, we set up the initialization $k_i^{[0]} = 1$, $i = 2, \ldots, n$ for "$FCVB$ 1". The initialization for "$FCVB$ 2" was chosen automatically in the first IVB cycle.

The sequence of label's estimate from each algorithm was compared with true generated label sequence, in which the error rate's criterion is Hamming distance. 10^4 runs of Monte Carlo was also implemented. The smaller is d, the closer are Gaussian components. Hence, informally, the signal-to-noise (SNR) ratio is decreased in that case. The simulation's results are shown in Fig.3. The mean value of converged IVB cycle ν_c, given all of Monte Carlo runs for all cases of d, were 3.5 and 3.4 for "$FCVB$ 1" and "$FCVB$ 2", respectively.

The best performance was Forward algorithm of filtering distribution, while the FB algorithm of smoothing distribution was the second best. The "$FCVB$ 1" with unsuitable initialization yielded the worst result. In "$FCVB$ 2" scheme, the estimate from approximate smoothing distribution

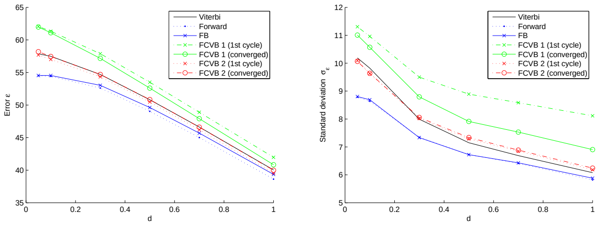

Figure 3: Expectation (left) and standard deviation (right) of normalized Hamming distance (%) versus mean distance between observations

(24-26) seems to be worse than the one from approximate filtering distribution (33, 34). Hence "*FCVB* 2" can achieve an estimate's accuracy comparable with VA right after the first IVB cycle, with a low complexity of computation $O(Mn)$ and memory $\mathcal{O}(n)$, compared with $O(M^2n)$ and $O(Mn)$ of VA, respectively. In this simulation, because the convergence number ν_c of FCVB is about 3.5 and very close to $M = 3$, the computational loads $O(\nu_c Mn)$ of both FCVB schemes are still close to the one of VA.

V Discussion

VA is usually applied to estimation of a finite-state HMM (i.e. a hidden Markov chain), because of the chain rule of maximization on which it depends (9). In the continuous-state case, the VA can only be used for state estimation in simple cases, such as the Kalman linear Gaussian context [8]. In contrast, the FCVB algorithm (24-26), which propagates VB-moments, is still applicable in the continuous state case.

VA and FB are inherently off-line algorithms. If applied online, computations must be restarted to account for each new symbol, incurring major computational overheads. Furthermore, the marginal MAP estimate of a subset of symbols is unavailable under VA. In contrast, the initialized filtering approximations, $\widetilde{f}(l_i|\mathbf{x}_i)$, of *FCVB* 2 yield online estimates as a natural output, and the converged smoothing distribution, $\widetilde{f}(l_i|\mathbf{x}_n)$, can be used to compute a MAP estimate of a part of the trajectory.

The soft-output Viterbi algorithm has been proposed in the literature [9] as a mean of increasing the reliability of VA-based state estimation, but its computational load is greater. The FCVB algorithm for state inference proposed in this paper computes soft outputs as intermediate calculations, via the approximate filtering and smoothing distributions, $\widetilde{f}(l_i|\mathbf{x}_i)$ and $\widetilde{f}(l_i|\mathbf{x}_n)$, respectively. It does so without any extra computational overheads.

VI Conclusion

In this paper, the industry-standard Viterbi (VA) and forward-backward (FB) algorithms for digital decoding have been reinterpreted in a Bayesian context. Although the inference of the hidden Markov chain is tractable under both algorithms (state estimates (VA) and state probabilities (FB), respectively), we proposed the variational Bayes (VB) algorithm as a means of reducing the computational load of these algorithms. Its specialization, as FCVB, yielded a novel computational flow for sequential state estimation. The new FCVB algorithm provided approximate symbol probabilities (soft information) at a computational cost far lower than FB. In terms of the comparative performance of VA and the new FCVB algorithm for MAP-based symbol decoding, FCVB yielded commensurate performance but with far lower computational and memory requirements.

References

[1] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[2] G. J. Forney, "The viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268 – 278, 1973.

[3] A. J. Viterbi, "A personal history of the viterbi algorithm," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 120–142, 2006.

[4] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," pp. 257–286, 1989.

[5] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of error-control coding," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2531–2560, 1998.

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

[7] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[8] C. Pavel and R. Yaacov, "On the viterbi process with continuous state space," *to apprear in Bernoulli*, 2009.

[9] J. Hagenauer and P. Hoeher, "A viterbi algorithm with soft-decision outputs and its applications," *GLOBECOM '89, IEEE*, vol. 3, pp. 1680–1686, 1989.