Probability in the Engineering and Informational Sciences, **21**, 2007, 451–475. Printed in the U.S.A.
DOI: 10.1017/S0269964807000083

ADJUSTED VITERBI TRAINING

A PROOF OF CONCEPT

JÜRI LEMBER
Tartu University
Tartu 50409, Estonia
E-mail: jyril@ut.ee

ALEXEY KOLOYDENKO
School of Mathematical Sciences
University of Nottingham
Nottingham, NG7 2RD, UK
E-mail: alexey.koloydenko@maths.nottingham.ac.uk

Viterbi training (VT) provides a fast but inconsistent estimator of hidden Markov models (HMM). The inconsistency is alleviated with a little extra computation when we enable VT to asymptotically fix the true values of the parameters. This relies on infinite Viterbi alignments and associated with them limiting probability distributions. First in a sequel, this article is a proof of concept; it focuses on mixture models, an important but special case of HMM where the limiting distributions can be calculated exactly. A simulated Gaussian mixture shows that our central algorithm (VA1) can significantly improve the accuracy of VT with little extra cost. Next in the sequel, we present elsewhere a theory of the adjusted VT for the general HMMs, where the limiting distributions are more challenging to find. Here, we also present another, more advanced correction to VT and verify its fast convergence and high accuracy; its computational feasibility requires additional investigation.

1. INTRODUCTION

Motivated by applications of the Viterbi training (VT) algorithm to estimate parameters of hidden Markov models (HMM) in speech recognition (Huang, Ariki, and Jack [11], Ney, Steinbiss, Haeb-Umbach, Tran, and Essen [26], Rabiner and Juang [31], Rabiner,

 0269-9648/07 $25.00

Wilpon, and Juang [32], Steinbiss et al. [36], Ström, Hetherington, Hazen, Sandness, and Glass [37]), natural language models (Ji and Bilmes [14], Och and Ney [27]), image analysis (Joshi, Li, and Wang [15], Li, Gray, and Olshen [22]), bioinformatics (Ehret et al. [8], Ohler, Niemann, Liao, and Rubin [28]), and gene discovery via unsupervized learning (Lomsadze, Ter-Hovhannisyan, Chernoff, and Borodovsky [24]), we propose a new principled way to improve the accuracy of the VT estimators while preserving the essential computational advantages of the baseline algorithm.

Let θ_l be the emission parameters of an HMM with states $l \in S = \{1, \ldots, K\}$. The central method for computing $\theta = (\theta_1, \ldots, \theta_K)$ (and, optionally, the parameters of the hidden chain) via likelihood maximization is the expectation-maximization (EM) algorithm that in the HMM context is also known as the *Baum–Welch* or *forward–backward algorithm* (Baum and Petrie [1], Bilmes [2], Huang et al. [11], Jelinek [13], Rabiner [30], Rabiner and Juang [31], Young [40]). Since expection-maximization (EM) can, in practice, be computationally expensive, it is commonly replaced by VT. Viterbi training effectively replaces the computationally costly expectation (E) step of EM by an appropriate maximization step that is computationally less intensive. An important example of successful and elaborate application of VT in industry is Philips speech recognition systems (Ney et al. [26]).

There are also variations of VT that use more than one best alignment, or several perturbations of the best alignment (Och and Ney [27]). The improvements that we explore are, however, of a different nature. Roughly, we increase the estimation accuracy by means of analytic calculations and do not require computing more than one optimal alignment.

The message of our work is as follows: *If an application relies on the computational efficiency of VT and, in particular, finds any of the efficient implementations of EM (e.g. Jank and Booth* [12], *Wei and Tanner* [39]) *still too intensive, such an application might still benefit from our adjustment since the proposed accuracy improvement requires no extra pointwise processing of the data.*

Let us recall that VT can be inferior to EM in terms of accuracy because the VT estimators need not be (local) maximum likelihood estimators (VT does not necessarily increase the likelihood), leading to bias and inconsistency (Section 2).

Given current parameter values, VT first finds a Viterbi *alignment* that is a sequence of hidden states maximizing the likelihood of the observed data. Observations assumed to have been emitted from state l are regarded as an independent and identically distributed (i.i.d.) sample from P_l, the corresponding emission distribution. These observations produce $\hat{P}_l^n$, the empirical version of P_l, and, ultimately, $\hat{\mu}_l$, a maximum likelihood estimate of θ_l. $\hat{\mu}$ is then used to find an alignment in the next step, and so forth. It can be shown that, in general, this procedure terminates in finitely many steps; moreover, it is usually much faster than EM.

In speech recognition, the same training procedure was already described by Rabiner and colleagues in [16,32] (see also [2,31]), who considered his procedure a variation of the *Lloyd algorithm* from vector quantization and referred to it as *segmential K-means training*. The analogy with vector quantization is especially pronounced when the underlying chain is a sequence of i.i.d. variables, in which

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

case the observations are simply an i.i.d. sample from a mixture distribution (Section 3). For mixture models, VT was also described by Gray and colleagues in (Chou, Lookbaugh, and Gray [6]), where the training algorithm was considered in the vector quantization context under the name *entropy constrained vector quantization* (ECVQ). (See also Gray, Linder, and Li [10] for more recent developments in this theory.) A better known name for VT in the mixture case is *Classification EM* (CEM) (Celeux and Govaert [5], Fraley and Raftery [9]), stressing that instead of the mixture likelihood, CEM maximizes the *Classification Likelihood* (Celeux and Govaert [5], Fraley and Raftery [9], McLachlan and Peel [25]). Also, for the uniform mixture of Gaussians with a common covariance matrix of the form $\sigma^2 I$ and unknown σ, VT, or CEM, is equivalent to the *k-means clustering* (Celeux and Govaert [5], Chou et al. [6], Fraley and Raftery [9], Sabine and Gray [34]).

Our ultimate goal is to alleviate the inconsistency of the VT estimators in the general HMM case while preserving the fast execution and computational feasibility of the baseline VT algorithm. First in a sequel, this article introduces the main ideas of our approach and provides an overall proof of their relevance to the goal. Thus, we begin by noticing that θ^*, the true emission parameters, are asymptotically a fixed point of EM but not of VT (Sections 2 and 3). That significance of this observation extends beyond its mere mathematics might be conjectured, for example, from that in the multivariate mixture models, EM typically produces improved partitions when *started with reasonable ones* (Fraley and Raftery [9]). This latter observation also leads us to expect the effect of the fixed-point property to be appreciable in the general HMM case, which is indeed verified via simulations in Koloydenko, Lember, and Käärik [19], the third part of the sequel. We therefore attempt to adjust VT in order to restore this property, and we do so by studying asymptotics of $\hat{P}_l^n$. Thus, we study the existence of Q_l, $l \in S$:

$$\hat{P}_l^n \Longrightarrow Q_l, \qquad l \in S \text{ a.s.}, \tag{1}$$

first in the general HMM context (Section 2), in much more detail in [21], and then in the special case of mixture models (Section 3). If such limiting measures exist, then under certain continuity assumptions, the estimators $\hat{\mu}_l$ will converge to μ_l [19], where

$$\mu_l = \arg\max_{\theta_l} \int \ln f_l(x;\theta_l) Q_l(dx) \quad \text{and} \quad f_l(x;\theta_l) \text{ is a p.d.f. of } P_l.$$

Taking into account the difference between μ_l and the true parameter, the appropriate adjustment of the Viterbi training can now be defined (Section 2).

However, the asymptotic behavior of $\hat{P}_l^n$ is not, in general, straightforward and its analysis requires an extension of the definition of Viterbi alignment, or path, *at infinitum* (Lember and Koloydenko [20]). Earlier attempts to consider convergence of Viterbi paths appear in [3,4] with a more general and more complete treatment of the problem to be found in [20,21], the second part of this sequel. Once the infinite alignment is properly defined, Lember and Koloydenko [20,21] prove the existence of the limiting measures Q_l (1), which is essential for the general definition of the adjusted VT.

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

To implement these ideas in practice, a closed form of Q_l (or $\hat{\mu}_l$) as a function of the true parameters is necessary. However, the measures Q_l depend on the transition as well as on the emission models, and computing Q_l can be very difficult. In the special case of mixture models (Section 3), on the other hand, the measures Q_l are easier to find. Although mixture models are not our goal, we are in part motivated by the continuing interest of others in computational efficiency and accuracy of parameter estimation in mixture models (Dias and Wedel [7], Lin, Chen, and Wu [23]). In Section 3, we describe the adjusted Viterbi training (VA1) for the mixture case, which we view, however, *only as a proof of concept*: VA1 *recovers the asymptotic fixed-point property*, and since its adjustment function *does not depend on data*, each iteration of VA1 enjoys the *same order of computational complexity (in terms of the sample size) as VT*. Moreover, for commonly used mixtures, such as mixtures of multivariate normal distributions with unknown means and known covariances (Example 3.1), the adjustment function is available in a *closed form* requiring integration with the mixture densities. Depending on the dimension of the emission variates, on the number of components, and on the available computational resources, one can vary the accuracy of the adjustment. We reiterate that, unlike the computations of the E step of EM, computations of the adjustment *do not involve evaluation and subsequent summation of the mixture density at individual data points*.

We first introduce these ideas for the case of known mixture weights (Section 3.1) and then extend them in Section 3.2 to the case of unknown weights. In terms of the general HMMs, the latter case corresponds to the transition matrix of the hidden chain being unknown.

To test our theory, in Section 5 we simulate a mixture of two univariate normal distributions with unit variance, unknown means, and unequal but comparable weights. The main goal of our simulations is to compare the performances of VT, VA1, and EM in terms of the accuracy, convergence, amount of computations per iteration, and the total amount of computations. The simulations are performed with different types of initialization, and with the weights assumed to be known (Section 5.1) and unknown (Section 5.2); the results (Section 5.3) are consistently in favor of VA1. Similar simulations have been performed for mixtures of multivariate Gaussians with known covariances, using stochastic approximations for the adjustment and leading to similar conclusions, but details (except for the discussion of Example 3.1) are omitted for conciseness.

In Section 4, we briefly introduce VA2, a more advanced correction to VT, presently merely as a mathematical complement of our adjustment idea; we verify its fast convergence and high accuracy on the simulated data in Section 5, but its computationally feasible implementations would require more investigation. A concluding summary is given in Section 6.

2. GENERAL HMMs

Let Y be a Markov chain with a finite state space S. We assume Y to be irreducible and aperiodic with the transition matrix $P = (p_{ij})$ and the initial distribution π that is also

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

the stationary distribution of Y. To every state $l \in S$ there corresponds an *emission distribution* P_l on $(\mathcal{X}, \mathcal{B})$, a separable metric space, and the corresponding Borel σ-algebra. Let f_l, the density of P_l with respect to some reference measure λ (for instance, the Lebesgue measure), be known up to the parametrization $f_l(x;\ \theta_l)$. When Y is in state l, an observation according to $P_l(\theta^*)$ and independent of everything else is emitted, with $\theta^* = (\theta_1^*, \ldots, \theta_K^*)$ being the unknown true parameters.

Thus, for any $y = y_1, y_2, \ldots$, a realization of Y, there corresponds a sequence of independent random variables $X_1, X_2, \ldots$, where X_n has distribution P_{y_n}. Note that we only observe $X = X_1, X_2, \ldots$ and the realization y is unknown (Y is hidden).

The distribution of X is completely determined by the chain parameters P and the emission distributions P_l, $l \in S$. The process X is also *mixing* and, therefore, ergodic. We now recall the notions of Viterbi alignment and training.

Let $x_1, \ldots, x_n$ be the first n observations on X. Let $\Lambda(q_1, \ldots, q_n; x_1, \ldots, x_n; \theta)$ be the (complete) likelihood function $\mathbf{P}(Y_i = q_i,\ i = 1, \ldots, n)\prod_{i=1}^n f_{q_i}(x_i;\ \theta_{q_i})$, $q_i \in S$.

The *Viterbi alignment* is any sequence of states $q_1, \ldots, q_n \in S$ that maximizes the likelihood of $x_1, \ldots, x_n$, θ being fixed. Thus, for a fixed θ, the Viterbi alignment is the maximum (conditional) likelihood estimator of *the realization of* $Y_1, \ldots, Y_n$, given $x_1, \ldots, x_n$. In the following, the Viterbi alignment will be referred to as the alignment. Since the alignment need not be unique, for each $n \geq 1$ let $\mathcal{V}$ denote the set of all state sequences resulting in the alignment

$$\begin{aligned}\mathcal{V}(x_1, \ldots, x_n; \theta) = \{v \in S^n\colon \forall w \in S^n \Lambda(v; x_1, \ldots, x_n; \theta) \\ \geq \Lambda(w; x_1, \ldots, x_n; \theta)\}.\end{aligned} \tag{2}$$

Any map $v\colon \mathcal{X}^n \mapsto \mathcal{V}(x_1, \ldots, x_n;\ \theta)$ will also be called an alignment. Further, unless explicitly specified, v_θ will denote an arbitrary element of $\mathcal{V}(x_1, \ldots, x_n;\ \theta)$.

Viterbi Training

1. Choose an initial value $\theta^0 = (\theta_1^0, \ldots, \theta_K^0)$.
2. Given θ^j $(j \geq 0)$, compute the alignment

$$v_{\theta^j}(x_1, \ldots, x_n) = (v_1, \ldots, v_n)$$

and partition $x_1, \ldots, x_n$ into (at most) K subsamples, with x_k going to the lth subsample if and only if $v_k = l$. Equivalently, define (at most) K empirical measures in accordance with (3):

$$\hat{P}_l^n(A; \theta^j) = \frac{\sum_{i=1}^n I_{A\times l}(x_i, v_i)}{\sum_{i=1}^n I_l(v_i)}, \qquad A \in \mathcal{B}, \quad l \in S, \tag{3}$$

where I_A stands for the indicator function of set A.

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

3. For every subsample, find the maximum likelihood estimate (MLE) given by

$$\hat{\mu}_l(\theta^j) = \arg\max_{\theta_l \in \Theta_l} \int \ln f_l(x; \theta_l) \hat{P}_l^n(dx; \theta^j) \tag{4}$$

and take $\theta_l^{j+1} = \hat{\mu}_l(\theta^j)$, $l \in S$. If for some $l \in S$, $v_i \neq l$ for any $i = 1, \ldots, n$ (lth subsample is empty), then the empirical measure $\hat{P}_l^n$ is formally undefined, in which case, we take $\theta_l^{j+1} = \theta_l^j$. We omit this exceptional case in the ensuing discussion.

Viterbi training can be interpreted as follows. Suppose that at step j, $\theta^j = \theta^*$ and, hence, v_{θ^j} is obtained using the true parameters. The training is then based on the assumption that the alignment $v(x_1, \ldots, x_n) = (v_1, \ldots, v_n)$ is correct (i.e., $v_i = Y_i$, $i = 1, \ldots, n$). If this assumption were true, the empirical measures $\hat{P}_l^n(\theta^j)$, $l \in S$, would be obtained from the i.i.d. sample generated from $P_l(\theta^*)$ and the MLE $\hat{\mu}_l(\theta^*)$ would be the natural estimator to use. Clearly, under this assumption (and passing from $x_1, x_2, \ldots$ to $X_1, X_2, \ldots$), $\hat{P}_l^n(\theta^*) \Rightarrow P_l(\theta^*)$ a.s. and, provided that $\{f_l(\cdot; \theta)\colon \theta \in \Theta_l\}$ is a P_l-Glivenko–Cantelli class and Θ_l is equipped with some suitable metric, $\lim_{n\to\infty} \hat{\mu}_l(\theta^*) = \theta_l^*$ a.s. Hence, if n is sufficiently large, then $\hat{P}_l^n \approx P_l$ and $\theta_l^{j+1} = \hat{\mu}_l(\theta^*) \approx \theta_l^* = \theta_l^j$, $\forall l$; that is, $\theta^j = \theta^*$ would be (approximately) a fixed point of the training algorithm.

A weak point of the previous argument is that the alignment in general is not correct even when the parameters used to find it are (i.e., generally $v_i \neq Y_i$). In particular, this implies that the empirical measures $\hat{P}_l^n(\theta^*)$ are not obtained from an i.i.d. sample taken from $P_l(\theta^*)$. Hence, we have no reason to believe that $\hat{P}_l^n(\theta^*) \Rightarrow P_l(\theta^*)$ a.s. and $\lim_{n\to\infty} \hat{\mu}_l(\theta^*) = \theta_l^*$ a.s. Moreover, we do not even know whether the sequences of empirical measures $\{\hat{P}_l^n(\theta^*)\}$ and MLE estimators $\{\hat{\mu}_l(\theta^*)\}$ converge (a.s.) at all.

In [20] we prove the existence of limiting probability measures $Q_l(\theta, \theta^*)$, $l \in S$, that depend on θ, the parameters used to find the alignment $v_\theta(x_1, \ldots, x_n)$, and on θ^*, the true parameters with which the random samples are emitted; namely Q_l, $l \in S$, are such that for every l,

$$\hat{P}_l^n(\theta_l^*) \Longrightarrow Q_l(\theta^*, \theta^*) \quad \text{a.s.} \tag{5}$$

Suppose also that the parameter space Θ_l is equipped with some metric. Then, under certain consistency assumptions on classes $\mathcal{F}_l = \{f_l(\cdot; \theta_l)\colon \theta_l \in \Theta_l\}$, the convergence

$$\lim_{n\to\infty} \hat{\mu}_l(\theta^*) = \mu_l(\theta^*, \theta^*) \quad \text{a.s.} \tag{6}$$

can be deduced from (5), where

$$\mu_l(\theta, \theta^*) \stackrel{\text{def}}{=} \arg\max_{\theta_l' \in \Theta_l} \int \ln f_l(x; \theta_l') Q_l(dx; \theta, \theta^*). \tag{7}$$

We also show that, in general, for the baseline VT $Q_l(\theta^*, \theta^*) \neq P_l(\theta^*)$, implying $\mu_l(\theta^*, \theta^*) \neq \theta_l^*$. In an attempt to reduce the bias $\theta_l^* - \mu_l(\theta^*, \theta^*)$, we next propose the *adjusted Viterbi training*. Suppose (5) and (6) hold. Based on (7), we now consider the mapping

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

$$\mu_l(\theta) = \mu_l(\theta, \theta), \qquad l = 1, \ldots, K. \tag{8}$$

Since this function is independent of the sample, we can define the following correction for the bias:

$$\Delta_l(\theta) = \theta_l - \mu_l(\theta), \qquad l = 1, \ldots, K. \tag{9}$$

VA1: Adjusted Viterbi Training

1. Choose an initial value $\theta^0 = (\theta_1^0, \ldots, \theta_K^0)$.
2. Given θ^j, perform the alignment and define K empirical measures $\hat{P}_l^n(\theta^j)$ as in (3).
3. For every $\hat{P}_l^n$, find $\hat{\mu}_l(\theta^j)$ as in (4), and for each l, define $\theta_l^{j+1} = \hat{\mu}_l(\theta^j) + \Delta_l(\theta^j)$, where Δ_l is defined $\forall l \in S$ in (9).

Note that, as desired, for n sufficiently large, the adjusted training algorithm has θ^* as its (approximately) fixed point: Indeed, suppose $\theta^j = \theta^*$. From (6), $\hat{\mu}_l(\theta^j) = \hat{\mu}_l(\theta^*) \approx \mu_l(\theta^*) = \mu_l(\theta^j)$, for all $l \in S$. Hence,

$$\theta_l^{j+1} = \hat{\mu}_l(\theta^*) + \Delta_l(\theta^*) \approx \mu_l(\theta^*, \theta^*) + \Delta_l(\theta^*) = \theta_l^* = \theta^j, \qquad l \in S. \tag{10}$$

3. MIXTURE MODELS

3.1. Known Weights

In general, no closed form for the distribution $Q_l(\theta^*, \theta^*)$ in (5) is available. Therefore, the mapping (8) might be impossible to determine exactly and approximations of Q_l should be used for the adjustments of Viterbi training (Section 2). However, in the case of the mixture models, the distributions Q_l are straightforward to find and the adjusted Viterbi training can therefore be immediately given. In this model, Y, the underlying Markov chain, is a sequence of i.i.d. discrete random variables with the state space $S = \{1, \ldots, K\}$ of *mixture components*. Thus, the transition probabilities are $p_{ij} = p_j$, $i, j \in S$, where p_j are mixture weights. To each component $l \in S$ there corresponds a probability distribution $P_l(\theta^*)$ with density $f_l = f_l(\cdot; \theta_l^*)$, where θ_i^* are the true parameters. Unless explicitly stated otherwise, the mixture weights p_l will be assumed to be known. Such a model produces observations $x_1, \ldots, x_n$ that are regarded as an i.i.d. sample from the mixture distribution $P(\theta^*)$ with density

$$\sum_{i=1}^{K} p_i f_i = \sum_{i=1}^{K} p_i f_i(\cdot; \theta_i^*) = f(\cdot; \theta^*) = f. \tag{11}$$

For any set of parameters $\theta = (\theta_1, \ldots, \theta_K)$, the alignment v_θ can be obtained via a *Voronoi partition* $\mathcal{S}(\theta) = \{S_1(\theta), \ldots, S_K(\theta)\}$, where

$$S_1(\theta) = \{x : p_1 f_1(x; \theta_1) \geq p_j f_j(x; \theta_j), \quad \forall j \in S\}, \tag{12}$$

$$S_l(\theta) = \{x : p_l f_l(x; \theta_l) \geq p_j f_j(x; \theta_j), \quad \forall j \in S\} \backslash (S_1 \cup \cdots \cup S_{l-1}), \qquad l = 2, \ldots, K. \tag{13}$$

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

Now, the alignment can be defined as follows: $v_\theta(x) = l$ if and only if $x \in S_l(\theta)$. In particular, given the Voronoi partition $\mathcal{S}(\theta) = \{S_1, \ldots, S_l\}$, the empirical measures $\hat{P}_l^n$ (3) are

$$\hat{P}_l^n(A; \theta) = \frac{\sum_{i=1}^n I_{S_l(\theta) \cap A}(x_i)}{\sum_{i=1}^n I_{S_l(\theta)}(x_i)}, \qquad A \in \mathcal{B},\ l \in S. \tag{14}$$

Thus, given the same partition, $\hat{\mu}_l(\theta)$ (4), the subsample MLE for component l becomes

$$\hat{\mu}_l(\theta) = \arg\max_{\theta_l' \in \Theta_l} \int_{S_l(\theta)} \ln f_l(x; \theta_l') \hat{P}_n(dx), \tag{15}$$

where $\hat{P}_n$ is the ordinary empirical measure associated with the given random sample. The convergence (5) then follows immediately from (14). Indeed, for any θ, by virtue of the strong law of large numbers, we have

$$\begin{aligned}\lim_{n\to\infty} \hat{P}_l^n(A; \theta) &\overset{\text{a.s}}{=} \frac{P(A \cap S_l(\theta); \theta^*)}{P(S_l(\theta); \theta^*)} \\ &= \frac{\int_{S_l(\theta)\cap A} f(x; \theta^*)\, d\lambda(x)}{\int_{S_l(\theta)} f(x; \theta^*)\, d\lambda(x)} \\ &= \frac{\sum_i p_i \int_{S_l(\theta)\cap A} f_i(x; \theta_i^*)\, d\lambda(x)}{\sum_i p_i \int_{S_l(\theta)} f_i(x; \theta_i^*)\, d\lambda(x)}.\end{aligned}$$

Since $\mathcal{X}$ is separable, it follows that $\hat{P}_l^n \Rightarrow Q_l$ a.s., where

$$q_l(x; \theta, \theta^*) \propto f(x; \theta^*) I_{S_l(\theta)} = \left(\sum_i p_i f_i(x; \theta^*)\right) I_{S_l(\theta)}, \qquad l = 1, \ldots, K,$$

are the densities of respective $Q_l(\theta, \theta^*)$s.

Now, it is clear that even when the partition $\mathcal{S}(\theta^*)$ is obtained using the true parameters θ^*, $Q_l(\theta^*, \theta^*)$, the limiting distribution (with density $q_l(x; \theta^*, \theta^*)$), can be different from $P_l(\theta^*)$, the desired distribution (with density $f_l(x; \theta^*)$). Likewise, $\mu_l(\theta^*)$ (8) can be different from

$$\theta_l^* = \arg\max_{\theta_l' \in \Theta_l} \int \ln f_l(x; \theta_l') f_l(x; \theta_l^*)\, d\lambda(x).$$

In order to see this, note that (7) and (8) in the context of the mixture model specialize to

$$\mu_l(\theta, \theta^*) = \arg\max_{\theta_l' \in \Theta_l} \int_{S_l(\theta)} \ln f_l(x; \theta_l') f(x; \theta^*)\, d\lambda(x), \tag{16}$$

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

$$\mu_l(\theta) = \arg\max_{\theta_l' \in \Theta_l} \int_{S_l(\theta)} \ln f_l(x; \theta_l') \left(\sum_i p_i f_i(x; \theta_i) \right) d\lambda(x), \tag{17}$$

respectively. We also emphasize that Δ can be significant, which justifies the adjustment.

Example 3.1: Let

$$f(x; \theta^*) = \frac{1}{K} \sum_{l=1}^{K} \phi(x; \theta_l^*),$$

where $\phi(x; \theta_l^*)$ is the density of the d-variate normal distribution with identity covariance matrix and vector of means $\theta_l^* \in \mathbb{R}^d = \Theta_l$ for $l = 1, 2, \ldots, K$. In this case, for each K-tuple of parameters $\theta = (\theta_1, \ldots, \theta_K)$, the decision rule for the alignment is essentially as follows (disregarding possible ties): $v_\theta(x) = i$ if and only if $||x - \theta_i|| \leq \min_j ||x - \theta_j||$. Thus, the decision regions in this case correspond to the Voronoi partition in its original sense, justifying our generalization of this term. Now, it can be easily seen that for all $m = 1, \ldots, d$,

$$(\mu_l(\theta))_m = \frac{\sum_{i=1}^{K} \int_{S_l(\theta)} x_m \phi(x; \theta_i)\, dx_1 \cdots dx_d}{\sum_{i=1}^{K} \int_{S_l(\theta)} \phi(x; \theta_i)\, dx_1 \cdots dx_d}. \tag{18}$$

Although the functions μ_l are data independent, the exact integration in (18) can require intensive computations when d and K are large. If this becomes an issue, one might be interested in approximations of (18). Even when approximated, the adjustment can still asymptotically reduce the bias provided, of course, that the approximation error is smaller than Δ_l. In the context of the above example, one might think of the following directions of approximating $\Delta_l(\theta) = \theta_l - \mu_l(\theta)$:

1. Approximate $\left(\sum_i \phi(x; \theta_i)\right) I_{S_l(\theta)}$ in (18) by $\phi(x; \theta_l) I_{S_l(\theta)}$, so

$$(\mu_l(\theta))_m \approx \frac{\int_{S_l(\theta)} x_m \phi(x; \theta_l)\, dx_1 \cdots dx_d}{\int_{S_l(\theta)} \phi(x; \theta_l)\, dx_1 \cdots dx_d}. \tag{19}$$

   This approximation is motivated by the limiting case when the components are "infinitely" far from each other.

2. If $K > d$, then some components are fully surrounded by others, and the partition cells corresponding to such internal components are bounded (Fig. 1). It is then conceivable that Δ_ls that correspond to the bounded cells are less significant than the others, in which case one might correct only the estimators of the outer components. This approach seems to be particularly appealing for speech recognition. In speech recognition, a phoneme is often modeled by a mixture of Gaussians or Laplace densities (Ney et al. [26]). One significant difficulty in the acoustic–phonetic modeling is determining

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

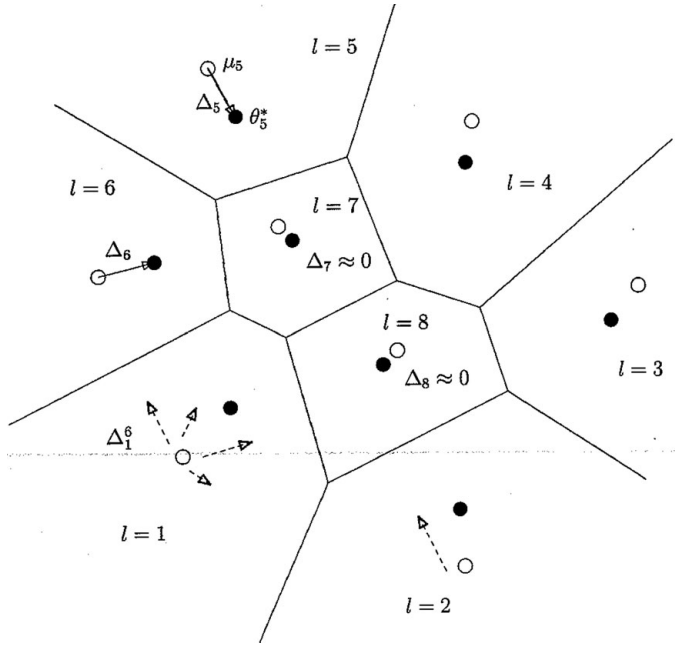

Figure 1. An eight-region Voronoi partition. True parameters θ^* and (hypothetical) $\mu(\theta^*)$ are marked with solid and open dots, respectively. For $l = 1$, $\Delta_l^j(\theta^*)$, hypothetical individual correction components are indicated to illustrate the ideas of Approximations 3 and 4. Similarly, for $l = 2$, a "significant" component of the correction is indicated. Neglecting the corrections for the estimators corresponding to the bounded Voronoi regions appears reasonable, as discussed in Approximation 2.

the boundaries of the phonemes (in the appropriate feature space). The boundaries depend mostly on the outer components. If the mixture parameters are estimated by VT, then the external components tend to be too far from their means (see Fig. 1), resulting in less accurate boundaries and an overall imprecision of the model estimation. Thus, correcting only the outer components might improve the entire acoustic–phonetic model.

3. Note that $\Delta_l = \sum_j \Delta_l^j$, where

$$\Delta_l^j(\theta) = \frac{\int_{S_l(\theta)} (\theta_l - x)\phi(x; \theta_j)\, dx_1 \cdots dx_d}{\sum_{i=1}^{K} \int_{S_l(\theta)} \phi(x; \theta_i)\, dx_1 \cdots dx_d}.$$

It might be reasonable for each l to replace Δ_l by its "leading component" (i.e., Δ_l^j with largest $||\Delta_l^j||$). Alternatively, instead of choosing the leading component, a single random component can be taken.

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

4. There are several motivations to approximate the denominator of (18) by 1. First, as in Approximation 1, this is reasonable when all of the centers are very far apart. Also, note that when $K = 2$, the denominator of (18) is equal to 1 exactly. Now, note that every Voronoi cell is determined by several hyperplanes. Suppose cell l is determined by hyperplanes H_l^q for q in some $I^l \subset \{1, \ldots, K\}$, namely it is the intersection of half-planes HP_l^q, $q \in I^l$. Since integrating over $S_l(\theta)$ is the same as integrating over the entire $\mathbb{R}^d$ and subtracting the integral over $S_l^C(\theta)$, the complement of $S_l(\theta)$, we note that $\Delta_l \approx \theta_l - (\theta_l - \sum_{i=1}^K \int_{S_l^C(\theta)} x\phi(x;\ \theta_i)\ dx) = \sum_{i=1}^K \int_{S_l^C(\theta)} x\phi(x;\ \theta_i)\ dx$. Suppose the defining hyperplanes are somehow ordered $I^l = \{q_1, \ldots, q_{|I^l|}\}$ and note that $S_l^C(\theta)$ is the union of the half-planes $O_1, \ldots, O_{|I^l|}$ opposite $HP_l^{q_1}, \ldots, HP_l^{q_{|I^l|}}$, respectively. Let us make this union a disjoint one as follows: $S_l^C(\theta) = \bigcup_{j=1}^{|I^l|} A_j$, where $A_1 = O_1$, $A_2 = O_2 \backslash O_1, \ldots, A_{|I^l|} = O_{|I^l|} \backslash \bigcup_{j=1}^{|I^l|-1} O_j$. Therefore, $\Delta_l \approx \sum \Delta_l^j$, where $\Delta_l^j = \sum_{i=1}^K \int_{A_j} x\phi(x;\ \theta_i)\ dx$. It then can be sensible to replace Δ_l by a single component Δ_l^j of significant contribution.
5. The integrals (18) are very easy to compute by Monte Carlo (or quasi Monte Carlo) methods (Ripley [33], Sobol [35]). This leads to the *stochastically adjusted Viterbi training* (SAV1) that modifies Step 3 of VA1 as follows:
   (a) Generate a sample $x_1, \ldots, x_n$ from $\sum_{l=1}^K p_l f_l(\cdot;\ \theta_l^j)$.
   (b) Based on the sample and Voronoi partition $\mathcal{S}(\theta^j)$, approximate $\mu_l(\theta^j)$ (18) by $\hat{\mu}_l^{MC}(\theta^j)$, an appropriate Monte Carlo estimate.
   (c) Use the following estimate for the correction:

$$\hat{\Delta}_l^{MC}(\theta^j) = \theta_l^j - \hat{\mu}_l^{MC}(\theta^j), \qquad l = 1, \ldots, K.$$

   The additional sampling step in SAV1 obviously jeopardizes the computational attractiveness of VA1. However, there are many ways to control the Monte Carlo integration in order to keep the overall complexity of SAV1 lower than that of EM. A great advantage of SAV1 is that *it is easy to implement even in very complex settings (including that of HMM)*. In [17], SAV1 is implemented for two-dimensional Gaussian mixtures. The simulations showed that in terms of precision, SAV1 is comparable with VA1 and EM and strongly outperforms VT. Moreover, in this two-dimensional setting, SAV1 and VA1 outperform VT even in terms of the number of iterations.

Remark 3.2: In Example 3.1, the decision regions correspond to the Voronoi partition in its original sense. Moreover, it is easy to see that in this particular case, VT is none other than the well-known (generalized) Lloyd algorithm designed for finding vector quantizers, which in this case are also called K-means (see, e.g., Sabine and Gray [34]). In this case, the estimators obtained by VT are empirical K-means. These latter estimators enjoy certain desirable properties, and in particular, they are consistent with respect to the population K-means [29]. However, they

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

need not be consistent with respect to θ^*, our parameters of interest. In the mixture case, VT can always be regarded as the (generalized) Lloyd algorithm, and the estimators obtained by VT can be regarded as (generalized) empirical K-means [6]. This observation links the study of VT and related algorithms to the theory of vector quantization.

3.2. Unknown Weights

We consider the case when the mixture weights p_l are unknown, which corresponds to the case of the unknown transition parameters P in the general HMM context.

The Voronoi partition depends on the weight vector $p = (p_1, \dots, p_K)$ as well as on θ. Hence, $\mathcal{S}(\theta, p)$ and the vector p should be reestimated at each step along with θ. Given a Voronoi partition $\mathcal{S} = \{S_1, \dots, S_K\}$, the simplest way to estimate the weights p_l is to take $p_l = \hat{P}_n(S_l)$, the empirical measure of S_l. Hence, all of the algorithms considered so far can be modified accordingly to include the weight estimation as in (20):

$$p_l^{j+1} = \hat{P}_n(S_l(\theta^j, p^j)), \qquad l = 1, \dots, K. \tag{20}$$

Taking into account the asymptotics, it is easy to correct the estimators p^{j+1} as well. Indeed, suppose $\theta^j = \theta^*$ and $p^j = p$ [i.e., $\mathcal{S}(\theta^j, p^j) = \mathcal{S}(\theta^*, p) = \mathcal{S}^*$]. If $n \to \infty$, then

$$\begin{aligned} \hat{P}_n(S_l(\theta^*, p)) \xrightarrow{\text{a.s}} P(S_l(\theta^*, p)) &= \int_{S_l(\theta^*, p)} f(x; \theta^*)\, d\lambda \\ &= \sum_i p_i \int_{S_l(\theta^*, p)} f_i(x; \theta_i^*)\, d\lambda. \end{aligned} \tag{21}$$

In general, the latter differs from p_l. The difference is $p_l - P(S_l(\theta^*, p))$. Hence, by analogy with (9), we can define the weight correction $D(\theta, p) = (D_1(\theta, p), \dots, D_K(\theta, p))$ as follows:

$$D_l(\theta, p) = p_l - \sum_i p_i \int_{S_l(\theta, p)} f_i(x; \theta_i)\, d\lambda, \tag{22}$$

which is also *data independent*. We now summarize the above by giving a formal definition of the adjusted VT with the weight correction. Viterbi training with p unknown can be defined similarly.

VA1 with the Weight Correction

1. Choose $\theta^0 = (\theta_1^0, \dots, \theta_K^0)$ and $p^0 = (p_1^0, \dots, p_K^0)$.
2. Given $\theta^j = (\theta_1^j, \dots, \theta_K^j)$ and $p^j = (p_1^j, \dots, p_K^j)$, define the Voronoi partition $\mathcal{S}(\theta^j, p^j) = \{S_1, \dots, S_K\}$ as in (12) and (13) and the empirical measures $\hat{P}_l^n(\theta^j, p^j)$ as in (14).
3. Put $\theta^{j+1} = \hat{\mu}^j(\theta^j) + \Delta(\theta^j)$, where $\hat{\mu}^j$ is defined in (17).
4. Put $p^{j+1} = \hat{P}_n(S_l(\theta^j, p^j)) + D(\theta^j, p^j)$.

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

4. VA2: A MORE ADVANCED ADJUSTMENT

The adjusted Viterbi training is designed to asymptotically fix the true parameter θ^*, returning approximately the correct solution, given this solution as the initial guess and given an infinitely large data sample: $\mathrm{VA1}(\theta^*) \approx \theta^*$. VA2 goes further and attempts to maximally expand $\{\theta\colon \mathrm{VA1}(\theta) \approx \theta^*\}$, the set of parameter values that are asymptotically mapped to the true ones, to $\{\theta\colon \mathrm{VA2}(\theta) \approx \theta^*\}$. Specifically, if the algorithm ever arrives at $\mathcal{S}(\theta^*)$, the Voronoi partition corresponding to the true parameters θ^*, then we would like to coerce the adjusted estimates to return θ^*. Let us explain these ideas in more detail.

Let $\mathcal{S}^*$ stand for $\mathcal{S}(\theta^*)$, the true Voronoi partition (that also coincides with the Bayes decision boundary). The mapping $\theta \mapsto \mathcal{S}(\theta)$ is generally many-to-one; hence, the set $\Theta(\mathcal{S}^*) = \{\theta\colon \mathcal{S}(\theta) = \mathcal{S}^*\}$ generally contains more than one element. (This also means that guessing $\mathcal{S}^*$ [i.e., guessing any element from $\Theta(\mathcal{S}^*)$], is generally easier than guessing θ^*.) We now introduce VA2.

Note first that $\mu_l(\theta, \theta^*)$ in (16), as well as the estimate $\hat{\mu}_l(\theta)$ in (15), depends on θ through $\mathcal{S}(\theta)$ only. However, the correction $\Delta_l(\theta) = \theta_l - \mu_l(\theta, \theta)$ does depend on θ fully and, hence, would not generally work (in the sense of (10)) for an arbitrary $\theta^j \in \Theta(\mathcal{S}^*)$ unless $\theta^j = \theta^*$. We now attempt to improve the first type of adjustment that is based on adding $\Delta(\theta^j)$ to $\hat{\mu}(\theta^j)$. Namely we propose the following iterative update for $l = 1, \ldots, K$. First, define $\mu_{l,\Theta(\mathcal{S}(\theta^0))}(\theta)$ (as function of θ only) to be the restriction of $\mu_l(\theta^0, \theta)$ to $\Theta(\mathcal{S}(\theta^0))$ and write $\mu_{l,\theta^0}(\theta)$ in place of the more cumbersome $\mu_{l,\Theta(\mathcal{S}(\theta^0))}(\theta)$. Let

$$\theta_l^{j+1} = \begin{cases} \mu_{l,\theta^j}^{-1}(\hat{\mu}_l(\theta^j)) & \text{if a unique } \mu_{l,\theta^j}^{-1}(\hat{\mu}_l(\theta^j)) \text{ exists} \\ \hat{\mu}_l(\theta^j) + \Delta_l(\theta^j) & \text{otherwise.} \end{cases} \tag{23}$$

For any θ^j and θ^*, the event that $\hat{\mu}(\theta^j, \theta^*)$ belongs to the range of $\mu(\theta^j, \theta)$ as a function of $\theta \in \Theta(\mathcal{S}(\theta^j))$ is of zero probability, as Example 4.1 illustrates. Hence, the introduction of the individual inverses μ_{l,θ^j}^{-1}, $l = 1, \ldots, K$, is essential, although still not always effective. Indeed, in some mixture models (a mixture of normal distributions with unequal weights is one such example), for a fixed l, the event that $\hat{\mu}_l(\theta^j)$ belongs to the range of $\mu_l(\theta^j, \theta)$ (as a function of $\theta \in \Theta(\mathcal{S}(\theta^j))$) need not occur with probability 1 for all θ^j and θ^*. This, and the fact that the inverses in general need not have a closed form, or might require intensive computations, might reduce the attractiveness of the suggested method. Further discussion of the computational issues related to this method is outside the scope of this article, except for mentioning the possibility of various (e.g., linear or quadratic approximations of the above functions $\mu_{l,\theta}^{-1}$).

In order to better understand the meaning of the new adjustment, imagine that $\theta^j \in \Theta(\mathcal{S}^*)$. We would then expect that for $l = 1, \ldots, K$,

$$\theta_l^{j+1} = \mu_{l,\theta^j}^{-1}(\hat{\mu}_l(\theta^j)) = \mu_{l,\theta^*}^{-1}(\hat{\mu}_l(\theta^*)) \approx \mu_{l,\theta^*}^{-1}(\mu_l(\theta^*, \theta^*)) = \theta_l^*.$$

The above argument, of course, also depends on the regularity of the above inverses at $\mu_l(\theta^*, \theta^*)$, $l = 1, \ldots, K$, and in this regard, our experiments in Section 5 provide

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

encouraging results for an important model similar to the model in the following example.

Example 4.1: Let $f(x;\ \theta^*) = (1/2)\phi(x - \theta_1^*) + (1/2)\phi(x - \theta_2^*)$, where ϕ is the density of the standard normal distribution. In this case, any Voronoi partition is specified by a single parameter $t = 0.5(\theta_1 + \theta_2)$ solving $\phi(t - \theta_1) = \phi(t - \theta_2)$ (ties are evidently inessential in this context). The true Voronoi partition corresponds to $t^* = 0.5(\theta_1^* + \theta_2^*)$. Given a Voronoi partition $\mathcal{S}(t(\theta))$, $\Theta(t) = \{(t - a), (t + a)\colon a \in \mathbb{R}^+\}$. Hence, restricted to $\Theta(t)$, the function $\mu_{\mathcal{S}(t)}(\theta) = (\mu_{1,\mathcal{S}(t)}(\theta),\ \mu_{2,\mathcal{S}(t)}(\theta))$ depends on one parameter only. Let a be this parameter and define $\mu_{\mathcal{S}(t)}(\theta(a)) = (\mu_1(a),\ \mu_2(a))$ as follows: $\mu_1(a) = -a(1 - 2\Phi(-a)) - 2\phi(-a) + t$, and $\mu_2(a) = 2t - \mu_1(a)$, where Φ is the distribution function of the standard normal distribution. After calculating $\hat{\mu}_1 < \hat{\mu}_2$ from the data, the inversion equations of (23) become

$$t - [a(1 - 2\Phi(-a)) + 2\phi(a)] = \hat{\mu}_1, \quad t + [a(1 - 2\Phi(-a)) + 2\phi(a)] = \hat{\mu}_2. \tag{24}$$

Obviously, (24) has a (unique) solution if and only if $\hat{\mu}_1$ and $\hat{\mu}_2$ are symmetric with respect to t and the probability of this latter event is clearly zero under the model. Thus, as suggested in (23), we consider the equations separately:

$$a(1 - 2\Phi(-a)) + 2\phi(a) = t - \hat{\mu}_1, \tag{25}$$

$$a(1 - 2\Phi(-a)) + 2\phi(a) = \hat{\mu}_2 - t. \tag{26}$$

It can be shown that (25) and (26) have unique solutions; let us denote the latter by a_1 and a_2, respectively. The points $t - a_1$ and $t + a_2$ will now be taken as the estimators of θ_1^* and θ_2^* for the next step of iterations.

VA2

1. Choose $\theta^0 = (\theta_1^0, \ldots, \theta_K^0)$.
2. Given θ^j, find $\mathcal{S}(\theta^j)$ and define empirical measures $\hat{P}_l^n(\theta^j)$ as in (14).
3. For every $\hat{P}_l^n$, find $\hat{\mu}_l(\theta^j, \theta^*)$ as in (15).
4. Update θ^{j+1} in accordance with (23).

VA2 with p unknown can be defined by analogy with Section 3.2.

5. SIMULATION STUDIES

In order to support our theory of adjusted Viterbi training, we simulate 1000 i.i.d. random samples of size 1000 according to the following mixture:

$$\frac{1}{\sqrt{2\pi}}\left(pe^{-(x-\theta_1)^2/2} + (1-p)e^{-(x-\theta_2)^2/2}\right).$$

The true parameters in our experiments are $\theta^* = (-2.5, 0)$ and $(p, 1 - p) = (0.7, 0.3)$. The corresponding density is plotted in Figure 2. Note that for all such mixtures with $p > 0.5$ and $\theta_1 < \theta_2$, $\theta_2 - \theta_1 < \sqrt{2p/(1-p)}$ (=2.1602 in our case) implies that both means fall on one side of the decision boundary, which makes detection

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

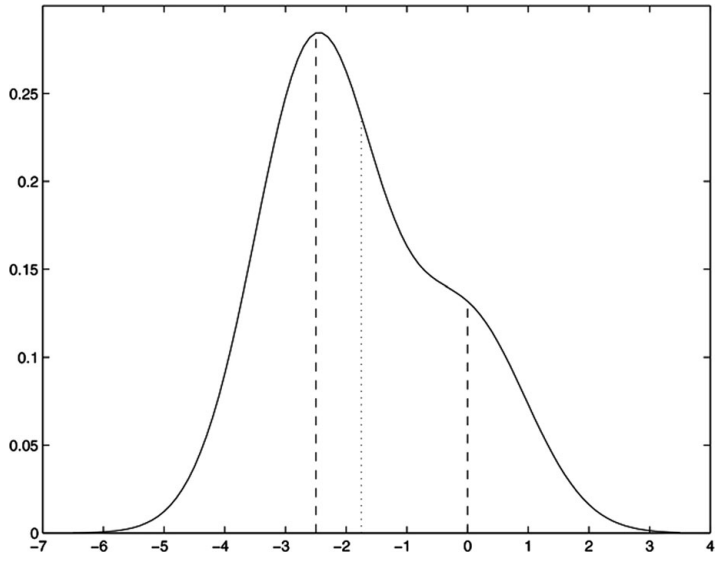

FIGURE 2. $(1/\sqrt{2\pi})(0.7e^{-(x+2.5)^2/2}+0.3e^{-x^2/2})$. The dashed vertical lines indicate the means of the individual components; the dotted line marks the mean of the mixture.

of the second component particularly difficult, as is already becoming the case in our setting with $\theta_2^* - \theta_1^* = 2.5$.

Our main goal is to compare the performances of VT, VA1, and EM in terms of the accuracy, convergence, amount of computations per iteration, and the total amount of computations. We implement these algorithms in Matlab [38], providing a fair comparison of their computational intensities based on their execution times. Our code is available for the reader's perusal [18] and is fully optimized for speed in the case of VT and EM. Consequently, our simulations possibly only overestimate the execution times for VA1.

Additionally, we compare VA2 with the above algorithms by the accuracy and convergence. We use a numerical solver to compute the adjustment function of VA2 and presently make no effort to replace this by a computationally efficient approximation. Hence, we do not discuss the computational intensity of VA2 in this work.

In our experiments, the algorithms are instructed to terminate as soon as the L_2 distance between consecutive θ updates falls below 0.001. We also provide a high precision MLE computed with a built-in Matlab optimization function. The cases of known (Section 3.1) and unknown weights (Section 3.2) are considered in Sections 5.1 and 5.2, respectively. We report the following statistics for each of the algorithms in the form average $\pm$ one standard deviation:

- $\theta = (\theta_1, \theta_2)$: the estimates of the means
- p: the estimate of the weight of the first component
- $||\theta - \theta^*||_{1,2}$: L_1- and L_2-normed distances between θ and the true parameters

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

- n: number of steps used by the algorithm
- T: total time in milliseconds to execute the entire algorithm
- t: time in milliseconds to execute one iteration of the algorithm

5.1. Known Weights

It is often the case in practice (e.g., speech recognition models) that the weights are assumed known; hence, we start with this case (Section 3.1). First, consider $(-1, 2)$ as an "arbitrary" initial guess for θ. Table 1 presents the performance statistics based on the 1000 samples. The baseline Viterbi method terminates quickly (on average, in 9.04 steps), outperformed only by VA2, but is the least accurate among the considered methods. As expected, VT also requires the least amount of computations: 0.2 ms per iteration and 1.85 ms total. Ranked from low to high, the accuracies of VA1, VA2, and EM appear similar and are about three times superior to that of VT. In units of the VT execution time, EM compares to VA1 as $16.85 : 6.7$ per iteration and as $20.43 : 7.59$ by the total execution times. In order to illustrate the asymptotic fixed-point property, we intialize the algorithms to $(-2.5, 0)$, the true value of the parameters; see Table 2. In this case, as expected, both VA1 and VA2 take noticeably fewer steps than VT and EM, are comparable in accuracy to EM, and are about three times more accurate than VT. Unlike VA1, VA2, or EM, the baseline algorithm, as predicted, disturbs the correct initial guess, resulting in an appreciable bias. The times per iteration of VA1 and EM are similar as earlier, and their total times are (in units of the VT time) 5.71 and 16.03, respectively.

In order to illustrate the idea of the second type of adjustment, we now initialize the algorithms to $(-3.1229, 0.8771)$, which produces the same decision boundary $t = -0.9111$ as $\theta^* = (-2.5, 0)$, the true values. Table 3 collects these results. Note that since VT and VA2 depend on the initial guess only via the decision boundary, they produce in this case exactly the same results (disregarding a small rounding error) as in the case of the correct initial guess (Table 2). As expected, VA2 now terminates significantly faster than its competitors, and accuracywise, it is only slightly superior to VA1 and slightly inferior to EM. The times per iteration of VA1 and EM are similar as earlier, and their total times are 7.84 and 20.83, respectively.

5.2. Unknown Weights

Assume now that the weights are unknown (Section 3.2) and need to be estimated along with the means. We use the same data and the same three types of condition as in the case of known weights. Namely, these are: arbitrary initialization to $(-1, 2)$ (Table 4), initialization to the correct values $(-2.5, 0)$ (Table 5), and initialization to $(-3.1229, 0.8771)$, the arbitrary point giving rise to the correct intercomponent boundary (Table 6). The initial weights are equal (i.e., $p = 0.5$) for all of the experiments. VT and the adjusted algorithms VA1 and VA2 in this case are implemented with the asymptotic correction (22). (The maximization in the high-precision MLE is now performed in the three variables.)

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

TABLE 1. Arbitrary Initial Guess

	VT	VA1	VA2	EM	MLE
θ_1	-2.4869 ± 0.0497	-2.4952 ± 0.0500	-2.4959 ± 0.0498	-2.4970 ± 0.0456	-2.4973 ± 0.0456
θ_2	0.2880 ± 0.0732	0.0099 ± 0.0917	0.0082 ± 0.0916	0.0030 ± 0.0757	0.0024 ± 0.0757
$\|\|\theta - \theta^*\|\|_1$	0.3291 ± 0.0844	0.1138 ± 0.0681	0.1133 ± 0.0678	0.0958 ± 0.0562	0.0958 ± 0.0562
$\|\|\theta - \theta^*\|\|_2$	0.2927 ± 0.0727	0.0902 ± 0.0537	0.0899 ± 0.0536	0.0761 ± 0.0451	0.0761 ± 0.0451
n	9.04 ± 1.55	10.49 ± 1.61	7.84 ± 1.59	11.20 ± 0.42	N/A
t	0.20 ± 0.05	1.34 ± 0.19	39.24 ± 1.56	3.37 ± 0.07	N/A
T	1.85 ± 0.55	14.04 ± 2.95	308.57 ± 68.63	37.79 ± 1.56	N/A

Table 2. Correct Initial Guess

	VT	VA1	VA2	EM	MLE
θ_1	-2.4904 ± 0.0495	-2.4973 ± 0.0488	-2.4973 ± 0.0490	-2.4973 ± 0.0455	-2.4973 ± 0.0456
θ_2	0.2820 ± 0.0729	0.0051 ± 0.0880	0.0052 ± 0.0892	0.0024 ± 0.0753	0.0024 ± 0.0757
$\|\|\theta - \theta^*\|\|_1$	0.3223 ± 0.0829	0.1087 ± 0.0661	0.1102 ± 0.0664	0.0953 ± 0.0561	0.0958 ± 0.0562
$\|\|\theta - \theta^*\|\|_2$	0.2867 ± 0.0721	0.0861 ± 0.0523	0.0874 ± 0.0525	0.0756 ± 0.0450	0.0761 ± 0.0451
n	5.56 ± 1.72	5.06 ± 1.57	4.73 ± 1.55	5.69 ± 1.29	N/A
t	0.22 ± 0.02	1.37 ± 0.05	42.23 ± 0.94	3.42 ± 0.08	N/A
T	1.21 ± 0.31	6.91 ± 2.05	199.52 ± 65.67	19.39 ± 4.28	N/A

Table 3. Correct Decision Boundary

	VT	VA1	VA2	EM	MLE
θ_1	-2.4904 ± 0.0495	-2.4954 ± 0.0497	-2.4973 ± 0.0490	-2.4971 ± 0.0456	-2.4973 ± 0.0456
θ_2	0.2820 ± 0.0729	0.0094 ± 0.0909	0.0052 ± 0.0892	0.0030 ± 0.0757	0.0024 ± 0.0757
$\|\|\theta - \theta^*\|\|_1$	0.3223 ± 0.0829	0.1131 ± 0.0668	0.1102 ± 0.0664	0.0958 ± 0.0562	0.0958 ± 0.0562
$\|\|\theta - \theta^*\|\|_2$	0.2867 ± 0.0721	0.0897 ± 0.0528	0.0874 ± 0.0525	0.0761 ± 0.0450	0.0761 ± 0.0451
n	5.56 ± 1.72	7.09 ± 1.38	4.72 ± 1.56	7.44 ± 0.94	N/A
t	0.22 ± 0.03	1.35 ± 0.05	42.37 ± 1.12	3.42 ± 0.08	N/A
T	1.22 ± 0.31	9.56 ± 1.81	200.24 ± 66.30	25.41 ± 3.19	N/A

Table 4. Unknown Weights; "Arbitrary" Guess

	VT	VA1	VA2	EM	MLE
p	0.747 ± 0.031	0.703 ± 0.028	0.702 ± 0.028	0.700 ± 0.024	0.699 ± 0.024
θ_1	-2.4299 ± 0.0753	-2.4919 ± 0.0596	-2.4930 ± 0.0594	-2.4976 ± 0.0531	-2.4992 ± 0.0532
θ_2	0.3944 ± 0.1178	0.0194 ± 0.1099	0.0173 ± 0.1094	0.0070 ± 0.0944	0.0039 ± 0.0947
$\|\|\theta - \theta^*\|\|_1$	0.4775 ± 0.1653	0.1382 ± 0.0851	0.1372 ± 0.0846	0.1179 ± 0.0708	0.1179 ± 0.0710
$\|\|\theta - \theta^*\|\|_2$	0.4058 ± 0.1237	0.1084 ± 0.0657	0.1076 ± 0.0653	0.0931 ± 0.0558	0.0931 ± 0.0560
n	14.16 ± 3.60	13.85 ± 3.25	12.23 ± 2.86	24.90 ± 2.60	N/A
t	0.72 ± 0.15	1.32 ± 0.09	39.01 ± 1.26	3.52 ± 0.35	N/A
T	10.13 ± 3.01	18.25 ± 4.27	478.19 ± 117.67	87.67 ± 12.98	N/A

TABLE 5. Unknown Weights; Correct Guess

	VT	VA1	VA2	EM	MLE
p	0.737 ± 0.030	0.699 ± 0.026	0.699 ± 0.026	0.699 ± 0.023	0.699 ± 0.024
θ_1	-2.4526 ± 0.0700	-2.4987 ± 0.0555	-2.4987 ± 0.0557	-2.4991 ± 0.0522	-2.4992 ± 0.0532
θ_2	0.3537 ± 0.1114	0.0058 ± 0.1007	0.0060 ± 0.1021	0.0038 ± 0.0925	0.0039 ± 0.0947
$\|\|\theta - \theta^*\|\|_1$	0.4212 ± 0.1467	0.1244 ± 0.0782	0.1263 ± 0.0782	0.1149 ± 0.0701	0.1179 ± 0.0710
$\|\|\theta - \theta^*\|\|_2$	0.3626 ± 0.1146	0.0978 ± 0.0607	0.0994 ± 0.0607	0.0907 ± 0.0553	0.0931 ± 0.0560
n	8.53 ± 3.47	6.01 ± 2.44	6.27 ± 2.40	11.89 ± 4.24	N/A
t	0.74 ± 0.04	1.36 ± 0.05	41.01 ± 1.24	3.53 ± 0.06	N/A
T	6.27 ± 2.42	8.13 ± 3.19	257.35 ± 99.70	41.84 ± 14.81	N/A

Table 6. Unknown Weights; Correct Boundary

	VT	VA1	VA2	EM	MLE
p	0.737 ± 0.029	0.702 ± 0.026	0.700 ± 0.026	0.700 ± 0.023	0.699 ± 0.024
θ_1	-2.4517 ± 0.0689	-2.4941 ± 0.0573	-2.4972 ± 0.0556	-2.4981 ± 0.0526	-2.4992 ± 0.0532
θ_2	0.3549 ± 0.1096	0.0148 ± 0.1050	0.0087 ± 0.1024	0.0059 ± 0.0930	0.0039 ± 0.0947
$\|\|\theta - \theta^*\|\|_1$	0.4218 ± 0.1459	0.1327 ± 0.0779	0.1271 ± 0.0780	0.1164 ± 0.0694	0.1179 ± 0.0710
$\|\|\theta - \theta^*\|\|_2$	0.3637 ± 0.1132	0.1043 ± 0.0606	0.0999 ± 0.0606	0.0919 ± 0.0548	0.0931 ± 0.0560
n	8.16 ± 3.40	7.74 ± 2.58	6.54 ± 2.22	12.98 ± 4.22	N/A
t	0.74 ± 0.04	1.34 ± 0.04	41.12 ± 1.74	3.52 ± 0.05	N/A
T	5.97 ± 2.36	10.32 ± 3.35	268.96 ± 91.44	45.59 ± 14.69	N/A

The adjusted algorithms now take 1.7 (VA1) and 2 (VA2) fewer steps than EM, and, what is more remarkable, VA1 and VA2 require fewer steps than VT. The per-iteration times of VA1 and EM compare as approximately 1.8 : 4.8 for all of the initializations, and the total times as 1.8 : 8.7 (arbitrary guess), 1.3 : 6.67 (true values), and 1.73 : 7.64 (true boundary); all are in units of the VT time. VA1 and VA2 are again at least three times more accurate than VT in θ estimation and about one standard deviation more accurate than VT in the weight estimation. They are also comparable in accuracy to EM.

5.3. Summary of the Results

VA1 is consistently close in accuracy to EM, which is always superior to VT. Specifically, in estimating the means, the gain in accuracy is about threefold, as measured by L_1- and L_2-distances, and in estimating the weights, it is about one standard deviation.

VA1 always converges almost as fast as VT and noticeably (by 30% in the case of unknown weights) faster than EM.

When the weights are known, an iteration of VA1 is about six times longer than that of VT and is more than twice as fast as that of EM. By total execution, VA1 is at most eight times slower than VT and is more than two and a half times faster than EM.

When the weights are unknown, VA1 is at most twice slower than VT and more than two and a half times faster than EM, per iteration. It is also about 50% slower than VT and more than four times faster than EM in total times.

Accuracy of VA2 is consistently between those of VA1 and EM, and VA2 converges faster than VA1.

6. Conclusion

We have considered the problem of the parameter estimation of the emission distribution in hidden Markov models using the two most relevant estimation principles: VT and MLE. We have identified the sources of bias, or inconsistency, in the VT algorithm, contrasting this with the EM algorithm that is generally used to compute MLE: Trading the EM's accuracy for the VT's ease of computations, one, in particular, loses the asymptotic fixed-point property; namely VT no longer fixes the true parameter values, not even asymptotically. We have proposed to restore this property and, consequently, increase the accuracy of VT. Specifically, we have proposed two types of analytic adjustment to the baseline VT algorithm, neither requiring additional pointwise processing of the data. In particular, *our correction functions are independent of the data size*. Our first adjustment, VA1, simply restores the asymptotic fixed-point property, whereas the second one, VA2, additionally ensures that asymptotically the true parameters are returned as soon as the algorithm finds the true alignment (i.e., Voronoi partition). To our knowledge, these kinds of consistency corrections for VT have not been proposed elsewhere in the literature.

This article has also shown that in the case of mixture models (a special and important case of HMM), the VA1 correction is always available, either in a closed form or via integration that can be suitably approximated. We have also explained why providing the VA1 correction in the general HMM case is more challenging, and we present our general theory [19, 21] elsewhere.

https://doi.org/10.1017/S0269964807000083 Published online by Cambridge University Press

This work has also presented evidence that, at least in the case of mixture models, the actual amount of extra computations of VA1 relative to VT can be very reasonable. For this special case, we have provided simulation studies based on 1000 large random samples that illustrate the key features of the adjusted algorithms in contrast with baseline VT and EM. In our simulations, VA1 demonstrates a significant increase of accuracy (threefold and one standard deviation in estimating the mixture means and weights, respectively) relative to VT. In fact, the accuracy of VA1 is already comparable to that of EM. In terms of computation, VA1 in our studies is still several factors faster than EM.

Due to the more sophisticated nature of the VA2 correction, its computationally feasible implementations require more work.

Certainly, the final decision as to which algorithm to use remains application dependent.

Acknowledgment

The first author was supported by the Estonian Science Foundation Grant 5694.

References

1. Baum, L. & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics* 37: 1554–1563.
2. Bilmes, J. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report 97–021, International Computer Science Institute, Berkeley, CA.
3. Caliebe, A. (2006). Properties of the maximum *a posteriori* path estimator in hidden Markov models. *IEEE Transactions on Information Theory* 52(1): 41–51.
4. Caliebe, A. & Rösler, U. (2002). Convergence of the maximum *a posteriori* path estimator in hidden Markov models. *IEEE Transactions on Information Theory* 48(7): 1750–1758.
5. Celeux, G. & Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis* 14(3): 315–332.
6. Chou, P., Lookbaugh, T., & Gray, R. (1989). Entropy-constrained vector quantization. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37(1): 31–42.
7. Dias, J. & Wedel, M. (2004). An empirical comparison of EM, SEM and MCMC performance for problematic Gaussian mixture likelihoods. *Statistics and Computing* 14: 323–332.
8. Ehret, G., Reichenbach, P., Schindler, U., Horvath, C., Fritz, S., Nabholz, M., & Bucher, P. (2001). DNA binding specificity of different STAT proteins. *Journal of Biological Chemistry* 276(9): 6675–6688.
9. Fraley, C. & Raftery, A. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association* 97(458): 611–631.
10. Gray, R., Linder, T., & Li, J. (2002). A Lagrangian formulation of Zador's entropy-constrained quantization theorem. *IEEE Transactions on Information Theory* 48(3): 695–707.
11. Huang, X., Ariki, Y., & Jack, M. (1990). Hidden Markov models for speech recognition. Edinburgh: Edinburgh University Press.
12. Jank, W. & Booth, J. (2003). Efficiency of Monte Carlo EM and simulated maximum likelihood in two-stage hierarchical models. *Journal of Computational and Graphical Statistics* 12(1): 214–229.
13. Jelinek, F. (2001). *Statistical methods for speech recognition*. Cambridge, MA: MIT Press.
14. Ji, G. & Bilmes, J. (2006). Backoff model training using partially observed data: Application to dialog act tagging. In *Proceedings of the Human Language Technology of the NAACL, Main Conference*. New York: Association for Computational Linguistics, pp. 280–287.

15. Joshi, D., Li, J., & Wang, J. (2006). A computationally efficient approach to the estimation of two- and three-dimensional hidden Markov models. *IEEE Transactions on Image Processing* 15(7): 1871–1886.
16. Juang, B.H. & Rabiner, L. (1990). The segmental K-means algorithm for estimating parameters of hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38(9): 1639–1641.
17. Kolde, R. (2005). Estimating of mixture density parameters with adjusted Viterbi training. Bachelor thesis, Tartu University, Estonia (in Estonian).
18. Koloydenko, A. & Lember, J. (2003). Matlab code for simulation studies of adjusted Viterbi training. Available from http://www.maths.nottingham.ac.uk/personal/pmzaak/VA.
19. Koloydenko, A., Käärik, M., & Lember, J. (2006). On adjusted Viterbi training. *Acta Applicandae Mathematicae* 96(1–3): 309–326.
20. Lember, J. & Koloydenko, A. (2007). Adjusted Viterbi training for hidden Markov models. Available from http://www.maths.nottingham.ac.uk/personal/pmzaak/VA/AVT2.pdf.
21. Lember, J. & Koloydenko, A. (2006). Adjusted Viterbi training for hidden Markov models. Available from http://www.maths.nottingham.ac.uk/personal/pmzaak/VA/AVT2.pdf.
22. Li, J., Gray, R., & Olshen, R. (2000). Multiresolution image classification by hierarchical modeling with two dimensional hidden Markov models. *IEEE Transactions on Information Theory* 46(5): 1826–1841.
23. Lin, C., Chen, C., & Wu, W. (2004). Fuzzy clustering algorithm for latent class model. *Statistics and Computing* 14: 299–310.
24. Lomsadze, A., Ter-Hovhannisyan, V., Chernoff, V., & Borodovsky, M. (2005). Gene identification in novel eukaryotic genomes by self-training algorithm. *Nucleic Acids Research* 33(20): 6494–6506.
25. McLachlan, G. & Peel, D. (2000). Finite mixture models. *Probability and Statistics*. New York: Wiley.
26. Ney, H., Steinbiss, V., Haeb-Umbach, R., Tran, B., & Essen, U. (1994). An overview of the Philips research system for large vocabulary continuous speech recognition. *International Journal of Pattern Recognition and Artificial Intelligence* 8(1): 33–70.
27. Och, F. & Ney, H. (2000). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics. A digital archive of research papers in computational linguistics.* Available from http://acl.ldc.upenn.edu/P/P00/P00-1056.pdf.
28. Ohler, U., Niemann, H., Liao, G., & Rubin, G. (2001). Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition. *Bioinformatics* 17(Suppl. 1): S199–S206.
29. Pollard, D. (1981). Strong consistency of *k*-means clustering. *Annals of Statistics* 9(1): 135–140.
30. Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2): 257–286.
31. Rabiner, L. & Juang, B. (1993). *Fundamentals of speech recognition*. Upper Saddle River, NJ: Prentice-Hall.
32. Rabiner, L., Wilpon, J., & Juang, B. (1986). A segmental K-means training procedure for connected word recognition. *AT&T Technical Journal* 64(3): 21–40.
33. Ripley, B.D. (1987). *Stochastic simulation*. New York: Wiley.
34. Sabine, M. & Gray, R. (1986). Global convergence and empirical consistency of the generalized Lloyd algorithm. *IEEE Transactions on Information Theory* 32(2): 148–155.
35. Sobol, I.M. (1973). *Chislennye metody Monte-Karlo*. Moscow: Nauka.
36. Steinbiss, V., Ney, H., Aubert, X., Besling, S., Dugast, C., Essen, U., Geller, D., Haeb-Umbach, R., Kneser, R., Meyer, H., Oerder, M., & Tran, B. (1995). The Philips research system for continuous-speech recognition. *Philips Journal of Research* 49: 317–352.
37. Ström, N., Hetherington, L., Hazen, T., Sandness, E., & Glass, J. (1999). Acoustic modeling improvements in a segment-based speech recognizer. In *Proceedings of the IEEE ASRU Workshop*, Keystone, CO. MIT Computer Science and AI Laboratory Spoken Language Systems. Available from http://www.sls.lcs.mit.edu/sls/publications/1999/asru99-strom.pdf.
38. The MathWorks (2004). *Getting started with Matlab*. Natick, MA: The MathWorks, Inc. http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml.
39. Wei, G. & Tanner, M. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association* 85: 699–704.
40. Young, S. (2003). The hidden vector state language model. Technical Report CUED/F-INFENG/TR.467, Cambridge University, Cambridge.