The variational hierarchical EM algorithm for clustering hidden Markov models

Emanuele Coviello
ECE Dept., UC San Diego
ecoviell@ucsd.edu

Antoni B. Chan
CS Dept., CityU of Hong Kong
abchan@cityu.edu.hk

Gert R.G. Lanckriet
ECE Dept., UC San Diego
gert@ece.ucsd.edu

Abstract

In this paper, we derive a novel algorithm to cluster hidden Markov models (HMMs) according to their probability distributions. We propose a variational hierarchical EM algorithm that i) clusters a given collection of HMMs into groups of HMMs that are similar, in terms of the distributions they represent, and ii) characterizes each group by a "cluster center", i.e., a novel HMM that is representative for the group. We illustrate the benefits of the proposed algorithm on hierarchical clustering of motion capture sequences as well as on automatic music tagging.

1 Introduction

The hidden Markov model (HMM) [1] is a probabilistic model that assumes a signal is generated by a double embedded stochastic process. A discrete-time hidden state process, which evolves as a Markov chain, encodes the dynamics of the signal, and an observation process, at each time conditioned on the current state, encodes the appearance of the signal. HMMs have successfully served a variety of applications, including speech recognition [1], music analysis [2] and identification [3], and clustering of time series data [4, 5].

This paper is about clustering HMMs. More precisely, we are interested in an algorithm that, given a collection of HMMs, partitions them into K clusters of "similar" HMMs, while also learning a representative HMM "cluster center" that concisely and appropriately represents each cluster. This is similar to standard k-means clustering, except that the data points are HMMs now instead of vectors in $\mathbb{R}^d$. Various applications motivate the design of HMM clustering algorithms, ranging from hierarchical clustering of sequential data (e.g., speech or motion sequences modeled by HMMs [4]), over hierarchical indexing for fast retrieval, to reducing the computational complexity of estimating mixtures of HMMs from large datasets (e.g., semantic annotation models for music and video) — by clustering HMMs, efficiently estimated from many small subsets of the data, into a more compact mixture model of all data. However, there has been relatively little work on HMM clustering and, therefore, its applications.

Existing approaches to clustering HMMs operate directly on the HMM *parameter* space, by grouping HMMs according to a suitable pairwise distance defined in terms of the HMM parameters. However, as HMM parameters lie on a non-linear manifold, a simple application of the k-means algorithm will not succeed in the task, since it assumes real vectors in a Euclidean space. In addition, such an approach would have the additional complication that HMM parameters for a particular generative model are not unique, i.e., a permutation of the states leads to the same generative model. One solution, proposed in [4], first constructs an appropriate similarity matrix between all HMMs that are to be clustered (e.g., based on the Bhattacharyya affinity, which depends non-linearly on the HMM parameters [6]), and then applies spectral clustering. While this approach has proven successful to group HMMs into similar clusters [4], it does not allow to generate *novel* HMMs as cluster centers. Each cluster can still be represented by choosing one of the given HMMs, e.g., the HMM which the spectral clustering procedure maps the closest to each spectral clustering center. However, this may be suboptimal for various applications of HMM clustering, e.g., in hierarchical estimation

of HMM mixtures. Spectral clustering can be based on other affinity scores between HMMs distributions than Bhattacharyya affinity, such as KL divergence approximated with sampling [7].

Instead, in this paper we propose to cluster HMMs *directly* with respect to the *probability distributions* they represent. We derive a hierarchical expectation maximization (HEM) algorithm that, starting from a group of HMMs, estimates a smaller mixture model that concisely represents and clusters the input HMMs (i.e., the input HMM distributions guide the estimation of the output mixture distribution). Historically, the first HEM algorithm was designed to cluster *Gaussian* probability distributions [8]. This algorithm starts from a Gaussian mixture model (GMM) and reduces it to another GMM with fewer components, where each of the mixture components of the reduced GMM represents, i.e., *clusters*, a group of the original Gaussian mixture components. More recently, Chan et al. [9] derived an HEM algorithm to cluster *dynamic texture* (DT) models (i.e., linear dynamical systems, LDSs) through their probability distributions. HEM has been applied successfully to many machine learning tasks for images [10], video [9] and music [11, 12]. The HEM algorithm is similar in spirit to Bregman-clustering [13], which is based on assigning points to cluster centers using KL-divergence.

To extend the HEM framework for GMMs to hidden Markov mixture models (H3Ms), additional marginalization of the hidden-state processes is required, as for DTMs. However, while Gaussians and DTs allow tractable inference in the E-step of HEM, this is no longer the case for HMMs. Therefore, in this work, we derive a variational formulation of the HEM algorithm (VHEM), and then leverage a variational *approximation* derived in [14] (which has not been used in a learning context so far) to make the inference in the E-step tractable. The proposed VHEM algorithm for H3Ms (VHEM-H3M) allows to *cluster hidden Markov models*, while also learning *novel HMM centers* that are representative of each cluster, in a way that is consistent with the underlying generative model of the input HMMs. The resulting VHEM algorithm can be generalized to handle other classes of graphical models, for which exact computation of the E-step in standard HEM would be intractable, by leveraging similar variational approximations. The efficacy of the VHEM-H3M algorithm is demonstrated on hierarchical motion clustering and semantic music annotation and retrieval.

The remainder of the paper is organized as follows. We review the hidden Markov model (HMM) and the hidden Markov mixture model (H3M) in Section 2. We present the derivation of the VHEM-H3M algorithm in Section 3, discussion and an experimental evaluation in Section 4.

2 The hidden Markov (mixture) model

A hidden Markov model (HMM) $\mathcal{M}$ assumes a sequence of τ observations $y_{1:\tau}$ is generated by a double embedded stochastic process. The hidden state process $x_{1:\tau}$ is a first order Markov chain on S states, with transition matrix A whose entries are $a_{\beta,\gamma} = P(x_{t+1} = \gamma | x_t = \beta)$, and initial state distribution $\pi = [\pi_1, \ldots, \pi_S]$, where $\pi_\beta = P(x_1 = \beta | \mathcal{M})$. Each state β generates observations according to an emission probability density function $p(y|x = \beta, \mathcal{M})$ which here we assume time-invariant and modeled as a Gaussian mixture with M components, i.e., $p(y|x = \beta, \mathcal{M}) = \sum_{m=1}^{M} c_{\beta,m} p(y|\zeta = m, \mathcal{M})$, where $\zeta \sim \text{multinomial}(c_{\beta,1}, \ldots, c_{\beta,M})$ is the hidden variable that selects the mixture component, $c_{\beta,m}$ the mixture weight of the mth Gaussian component, and $p(y|\zeta = m, \mathcal{M}) = \mathcal{N}(y; \mu_{\beta,m}, \Sigma_{\beta,m})$ is the probability density function of a multivariate Gaussian distribution with mean $\mu_{\beta,m}$ and covariance matrix $\Sigma_{\beta,m}$. The HMM is specified by the parameters $\mathcal{M} = \{\pi, A, \{\{c_{\beta,m}, \mu_{\beta,m}, \Sigma_{\beta,m}\}_{m=1}^{M}\}_{\beta=1}^{S}\}$ which can be efficiently learned from an observation sequence $y_{1:\tau}$ with the Baum-Welch algorithm [1].

A hidden Markov mixture model (H3M) models a set of observation sequences as samples from a group of K hidden Markov models, each associated to a specific sub-behavior [5]. For a given sequence, an assignment variable $z \sim \text{multinomial}(\omega_1, \cdots \omega_K)$ selects the parameters of one of the K HMMs. Each mixture component is parametrized by $\mathcal{M}_z = \{\pi^z, A^z, \{\{c^z_{\beta,m}, \mu^z_{\beta,m}, \Sigma^z_{\beta,m}\}_{m=1}^{M}\}_{\beta=1}^{S}\}$ and the H3M is parametrized by $\mathcal{M} = \{\omega_z, \mathcal{M}_z\}_{z=1}^{K}$. The likelihood of a random sequence $y_{1:\tau} \sim \mathcal{M}$ is

$$p(y_{1:\tau}|\mathcal{M}) = \sum_{i=1}^{K} \omega_i p(y_{1:\tau}|z = i, \mathcal{M}), \tag{1}$$

where $p(y_{1:\tau}|z = i, \mathcal{M})$ is the likelihood of $y_{1:\tau}$ under the ith HMM component. To reduce clutter, here we assume that all the HMMs have the same number S of hidden states and that all emission probabilities have M mixture components, though our derivation could be easily extended to the more general case, and in the remainder of the paper we use the notation in Table 1.

Table 1: Notation. (b) base model, (r) reduced model.

variables	(b)	(r)
index for HMM comp.	i	j
HMM states	β	ρ
HMM state sequence	$\beta_{1:\tau}=\{\beta_1\cdots\beta_\tau\}$	$\rho_{1:\tau}=\{\rho_1\cdots\rho_\tau\}$
index for comp. of GMM	m	ℓ
models		
H3M	$\mathcal{M}^{(b)}$	$\mathcal{M}^{(r)}$
HMM component	$\mathcal{M}_i^{(b)}$	$\mathcal{M}_j^{(r)}$
GMM emission	$\mathcal{M}_{i,\beta}^{(b)}$	$\mathcal{M}_{j,\rho}^{(r)}$
component of GMM	$\mathcal{M}_{i,\beta,m}^{(b)}$	$\mathcal{M}_{j,\rho,\ell}^{(r)}$

probability distributions	*notation*	*short-hand*
HMM state seq. (b)	$p(x_{1:\tau}{=}\beta_{1:\tau}\|z^{(b)}{=}i,\mathcal{M}^{(b)})$	$\pi_{\beta_{1:\tau}}^{(b),i}$
HMM state seq. (r)	$p(x_{1:\tau}{=}\rho_{1:\tau}\|z^{(r)}{=}j,\mathcal{M}^{(r)})$	$\pi_{\rho_{1:\tau}}^{(r),j}$
HMM obs. likelihood (r)	$p(y_{1:\tau}\|z^{(r)}=j,\mathcal{M}^{(r)})$	$p(y_{1:\tau}\|\mathcal{M}_j^{(r)})$
GMM emit likelihood (r)	$p(y_t\|x_t=\rho,\mathcal{M}_j^{(r)})$	$p(y_t\|\mathcal{M}_{j,\rho}^{(r)})$
Gaussian likelihood (r)	$p(y_t\|\zeta_t=\ell,x_t=\rho,\mathcal{M}_j^{(r)})$	$p(y_t\|\mathcal{M}_{j,\rho,\ell}^{(r)})$
expectations		
HMM obs. seq.	$\mathrm{E}_{y_{1:\tau}\|z^{(b)}=i,\mathcal{M}^{(b)}}[\cdot]$	$\mathrm{E}_{\mathcal{M}_i^{(b)}}[\cdot]$
GMM emission	$\mathrm{E}_{y_t\|x_t=\beta,\mathcal{M}_i^{(b)}}[\cdot]$	$\mathrm{E}_{\mathcal{M}_{i,\beta}^{(b)}}[\cdot]$
Gaussian component	$\mathrm{E}_{y_t\|\zeta_t=m,x_t=\beta,\mathcal{M}_i^{(b)}}[\cdot]$	$\mathrm{E}_{\mathcal{M}_{i,\beta,m}^{(b)}}[\cdot]$

3 Clustering hidden Markov models

We now derive the variational hierarchical EM algorithm for clustering HMMs (VHEM-H3M). Let $\mathcal{M}^{(b)}=\{\omega_i^{(b)},\mathcal{M}_i^{(b)}\}_{i=1}^{K^{(b)}}$ be a base hidden Markov mixture model (H3M) with $K^{(b)}$ components. The goal of the VHEM-H3M algorithm is to find a reduced hidden Markov mixture model $\mathcal{M}^{(r)}=\{\omega_j^{(r)},\mathcal{M}_j^{(r)}\}_{j=1}^{K^{(r)}}$ with fewer components (i.e., $K^{(r)}<K^{(b)}$), that represents $\mathcal{M}^{(b)}$ well. At a high level, the VHEM-H3M algorithm estimates the reduced H3M model $\mathcal{M}^{(r)}$ from virtual samples distributed according to the base H3M model $\mathcal{M}^{(b)}$. From this estimation procedure, the VHEM algorithm provides: (i) a (soft) clustering of the original $K^{(b)}$ HMMs into $K^{(r)}$ groups, encoded in assignment variables $\hat{z}_{i,j}$, and (ii) novel HMM cluster centers, i.e., the HMM components of $\mathcal{M}^{(r)}$, each of them representing a group of the original HMMs of $\mathcal{M}^{(b)}$. Finally, because we take the expectation over the virtual samples, the estimation is carried out in an efficient manner that requires only knowledge of the parameters of the base model without the need of generating actual virtual samples.

3.1 Parameter estimation

We consider a set Y of N *virtual* samples distributed accordingly to the base model $\mathcal{M}^{(b)}$, such that the $N_i=N\omega_i^{(b)}$ samples $Y_i=\{y_{1:\tau}^{(i,m)}\}_{m=1}^{N_i}$ are from the ith component (i.e., $y_{1:\tau}^{(i,m)}\sim\mathcal{M}_i^{(b)}$). We denote the entire set of samples as $Y=\{Y_i\}_{i=1}^{K^{(b)}}$, and, in order to obtain a consistent clustering of the input HMMs $\mathcal{M}_i^{(b)}$, we assume the entirety of samples Y_i is assigned to the same component of the reduced model [8]. Note that, in this formulation, we are not using virtual samples $\{x_{1:\tau}^{(i,m)},y_{1:\tau}^{(i,m)}\}$ for each base component, according to its joint distribution $p(x_{1:\tau},y_{1:\tau}|\mathcal{M}_i^{(b)})$, but we treat $X_i=\{x_{1:\tau}^{(i,m)}\}_{m=1}^{N_i}$ as "missing" information, and estimate them in the E-step. The reason is that a basis mismatch between components of $\mathcal{M}_i^{(b)}$ will cause problems when the parameters of $\mathcal{M}_j^{(r)}$ are computed from virtual samples of the hidden states of $\{\mathcal{M}_i^{(b)}\}_{i=1}^{K^{(b)}}$.

The original formulation of HEM [8] maximizes log-likelihood of the virtual samples, i.e., $\log p(Y|\mathcal{M}^{(r)})=\sum_{i=1}^{K^{(b)}}\log p(Y_i|\mathcal{M}^{(r)})$, with respect to $\mathcal{M}^{(r)}$, and uses the law of large numbers to turn the virtual samples into an expectation over the base model components $\mathcal{M}_i^{(b)}$. In this paper, we will start with a slightly different objective function to derive the VHEM algorithm. To estimate $\mathcal{M}^{(r)}$, we will maximize the *expected* log-likelihood of the virtual samples,

$$\mathcal{J}(\mathcal{M}^{(r)})=\mathrm{E}_{\mathcal{M}^{(b)}}\left[\log p(Y|\mathcal{M}^{(r)})\right]=\sum_{i=1}^{K^{(b)}}\mathrm{E}_{\mathcal{M}_i^{(b)}}\left[\log p(Y_i|\mathcal{M}^{(r)})\right], \tag{2}$$

where the expectation is over the base model components $\mathcal{M}_i^{(b)}$.

A general framework for maximum likelihood estimation in the presence of hidden variables (which is the case for H3Ms) is the EM algorithm [15]. In this work, we take a variational perspective [16, 17, 18], which views both E- and M-step as a maximization step. The variational E-step first obtains a family of lower bounds to the log-likelihood (i.e., to equation 2), indexed by variational parameters, and then optimizes over the variational parameters to find the tightest bound. The corresponding M-step then maximizes the lower bound (with the variational parameters fixed) with respect to the

model parameters. One advantage of the variational formulation is that it allows to replace a difficult inference in the E-step with a variational approximation, by restricting the maximization to a smaller domain for which the lower bound is tractable.

3.1.1 Lower bound to an expected log-likelihood

Before proceeding with the derivation of VHEM for H3Ms, we first need to derive a lower-bound to an expected log-likelihood term (e.g., (2)). We will first consider the lower bound to a log-likelihood. In all generality, let $\{O, H\}$ be the observation and hidden variables of a probabilistic model, respectively, where $p(H)$ is the distribution of the hidden variables, $p(O|H)$ is the conditional likelihood of the observations, and $p(O) = \sum_H p(O|H)p(H)$ is the observation likelihood. We can define a *variational lower bound* to the observation log-likelihood [18, 19]:

$$\log p(O) \geq \log p(O) - D(q(H)||p(H|O)) = \sum_H q(H) \log \frac{p(H)p(O|H)}{q(H)} \tag{3}$$

where $p(H|O)$ is the posterior distribution of H given observation O, and $q(H)$ is the variational distribution (i.e., $\sum_H q(H) = 1$ and $q_i(H) \geq 0$) or approximate posterior distribution. $D(p\|q) = \int p(y) \log \frac{p(y)}{q(y)} dy$ is the Kullback-Leibler (KL) divergence between two distributions, p and q. When the variational distribution equals the true posterior, $q(H) = P(H|O)$, then the KL divergence is zero, and hence the lower-bound reaches $\log p(O)$. When the true posterior is not possible to calculate, then typically q is restricted to some set of approximate posterior distributions that are tractable, and the best lower-bound is obtained by maximizing over q,

$$\log p(O) \geq \max_{q \in Q} \sum_H q(H) \log \frac{p(H)p(O|H)}{q(H)} \tag{4}$$

Using the lower bound in (4), we can now derive a lower bound to an expected log-likelihood expression. Let $\mathrm{E}_b[\cdot]$ be the expectation of O with respect to a distribution $p_b(O)$. Since $p_b(O)$ is non-negative, taking the expectation on both sides of (4) yields,

$$\mathrm{E}_b\left[\log p(O)\right] \geq \mathrm{E}_b\left[\max_{q \in Q} \sum_H q(H) \log \frac{p(H)p(O|H)}{q(H)}\right] \geq \max_{q \in Q} \mathrm{E}_b\left[\sum_H q(H) \log \frac{p(H)p(O|H)}{q(H)}\right] \tag{5}$$

$$= \max_{q \in Q} \sum_H q(H) \left\{\log \frac{p(H)}{q(H)} + \mathrm{E}_b\left[\log p(O|H)\right]\right\}, \tag{6}$$

where (5) follows from Jensen's inequality (i.e., $f(\mathrm{E}[x]) \leq \mathrm{E}[f(x)]$ when f is convex), and the convexity of the $\max$ function.

3.1.2 Variational lower bound

We now derive the lower bound of the expected log-likelihood cost function in (2). The derivation proceeds by successively applying the lower bound from (6) on each arising expected log-likelihood term, which results in a set of nested lower bounds. We first define the following three lower bounds:

$$\mathrm{E}_{\mathcal{M}_i^{(b)}}\left[\log p(Y_i|\mathcal{M}^{(r)})\right] \geq \mathcal{L}_{H3M}^{i}, \tag{7}$$

$$\mathrm{E}_{\mathcal{M}_i^{(b)}}\left[\log p(y_{1:\tau}|\mathcal{M}_j^{(r)})\right] \geq \mathcal{L}_{HMM}^{i,j}, \tag{8}$$

$$\mathrm{E}_{\mathcal{M}_{i,\beta_t}^{(b)}}\left[\log p(y_t|\mathcal{M}_{j,\rho_t}^{(r)})\right] \geq \mathcal{L}_{GMM}^{(i,\beta_t),(j,\rho_t)}. \tag{9}$$

The first lower bound, $\mathcal{L}_{H3M}^{i}$, is on the expected log-likelihood between an HMM and an H3M. The second lower bound, $\mathcal{L}_{HMM}^{i,j}$, is on the expected log-likelihood of an HMM $\mathcal{M}_j^{(r)}$, marginalized over observation sequences from a *different* HMM $\mathcal{M}_i^{(b)}$. Although the data log-likelihood $\log p(y_{1:\tau}|\mathcal{M}_j^{(r)})$ can be computed exactly using the forward algorithm [1], calculating its expectation is not analytically tractable since $y_{1:\tau} \sim \mathcal{M}_j^{(r)}$ is essentially an observation from a mixture with $O(S^\tau)$ components. The third lower bound is between GMM emission densities $\mathcal{M}_{i,\beta_t}^{(b)}$ and $\mathcal{M}_{j,\rho_t}^{(r)}$.

H3M lower bound - Looking at an individual term in (2), $p(Y_i|\mathcal{M}^{(r)})$ is a mixture of HMMs, and thus the observation variable is Y_i and the hidden variable is z_i (the assignment of Y_i to a component $\mathcal{M}_j^{(r)}$). Hence, introducing the variational distribution $q_i(z_i)$ and applying (6), we have

$$\begin{aligned}\mathrm{E}_{\mathcal{M}_i^{(b)}}\left[\log p(Y_i|\mathcal{M}^{(r)})\right] &\geq \max_{q_i}\sum_j q_i(z_i=j)\left\{\log\frac{p(z_i=j)}{q_i(z_i=j)}+N_i\mathrm{E}_{\mathcal{M}_i^{(b)}}[\log p(y_{1:\tau}|\mathcal{M}_j^{(r)})]\right\}\\ &\geq \max_{q_i}\sum_j q_i(z_i=j)\left\{\log\frac{p(z_i=j)}{q_i(z_i=j)}+N_i\mathcal{L}_{HMM}^{i,j}\right\}\triangleq\mathcal{L}_{H3M}^{i}. \qquad (10)\end{aligned}$$

where we use the fact that Y_i is a set of N_i i.i.d. samples, and we use the lower bound (8) for the expectation of $\log p(y_{1:\tau}|\mathcal{M}_j^{(r)})$, which is the observation log-likelihood of an HMM and hence its expectation cannot be calculated directly. To compute $\mathcal{L}_{H3M}^{i}$, we will restrict the variational distributions to the form $q_i(z_i=j)=z_{ij}$ for all i, where $\sum_{j=1}^{K^{(r)}} z_{ij}=1$, and $z_{ij}\geq 0\,\forall j$.

HMM lower bound - For the HMM likelihood $p(y_{1:\tau}|\mathcal{M}_j^{(r)})$, the observation variable is $y_{1:\tau}$ and the hidden variable is its state sequence $\rho_{1:\tau}$. Hence, for the lower bound $\mathcal{L}_{HMM}^{i,j}$ we get

$$\mathrm{E}_{\mathcal{M}_i^{(b)}}[\log p(y_{1:\tau}|\mathcal{M}_j^{(r)})]=\sum_{\beta_{1:\tau}}\pi_{\beta_{1:\tau}}^{(b),i}\mathrm{E}_{\mathcal{M}_i^{(b)}|\beta_{1:\tau}}[\log p(y_{1:\tau}|\mathcal{M}_j^{(r)})] \qquad (11)$$

$$\geq\sum_{\beta_{1:\tau}}\pi_{\beta_{1:\tau}}^{(b),i}\max_{q^{i,j}}\sum_{\rho_{1:\tau}}q^{i,j}(\rho_{1:\tau}|\beta_{1:\tau})\left\{\log\frac{p(\rho_{1:\tau}|\mathcal{M}_j^{(r)})}{q^{i,j}(\rho_{1:\tau}|\beta_{1:\tau})}+\sum_t\mathrm{E}_{\mathcal{M}_{i,\beta_t}^{(b)}}[\log p(y_t|\mathcal{M}_{j,\rho_t}^{(r)})]\right\} \qquad (12)$$

$$\geq\sum_{\beta_{1:\tau}}\pi_{\beta_{1:\tau}}^{(b),i}\max_{q^{i,j}}\sum_{\rho_{1:\tau}}q^{i,j}(\rho_{1:\tau}|\beta_{1:\tau})\left\{\log\frac{p(\rho_{1:\tau}|\mathcal{M}_j^{(r)})}{q^{i,j}(\rho_{1:\tau}|\beta_{1:\tau})}+\sum_t\mathcal{L}_{GMM}^{(i,\beta_t),(j,\rho_t)}\right\}\triangleq\mathcal{L}_{HMM}^{i,j} \qquad (13)$$

where in (11) we first rewrite the expectation $\mathrm{E}_{\mathcal{M}_i^{(b)}}$ to explicitly marginalize over the HMM state sequence $\beta_{1:\tau}$ from $\mathcal{M}_i^{(b)}$, in (12) we introduce a variational distribution $q_{\beta_{1:\tau}}^{i,j}(\rho_{1:\tau})$ on the state sequence $\rho_{1:\tau}$, which depends on the particular sequence $\beta_{1:\tau}$, and apply (6) , and in the last line we use the lower bound, defined in (9), on each expectation.

To compute $\mathcal{L}_{HMM}^{i,j}$ we will restrict the variational distributions to the form of a Markov chain [14],

$$q^{i,j}(\rho_{1:\tau}|\beta_{1:\tau})=\phi^{i,j}(\rho_{1:\tau}|\beta_{1:\tau})=\phi^{i,j}(\rho_1|\beta_1)\prod_{t=2}^{\tau}\phi_{\beta_t}^{i,j}(\rho_t|\rho_{t-1}), \qquad (14)$$

where $\sum_{\rho_1=1}^{S}\phi_{\beta_1}^{i,j}(\rho_1)=1$ for each value of β_1, and $\sum_{\rho_t=1}^{S}\phi_{\beta_t}^{i,j}(\rho_t|\rho_{t-1})=1$ for each value of β_t and ρ_{t-1}. The variational distribution $q_{\beta_{1:\tau}}^{i,j}(\rho_{1:\tau})$ assigns state sequences $\beta_{1:\tau}\sim\mathcal{M}_i^{(b)}$ to state sequences $\rho_{1:\tau}\sim\mathcal{M}_j^{(r)}$, based on how well (in expectation) the state sequence $\rho_{1:\tau}\sim\mathcal{M}_j^{(r)}$ can explain an *observation* sequence generated by HMM $\mathcal{M}_i^{(b)}$ evolving through state sequence $\beta_{1:\tau}\sim\mathcal{M}_i^{(b)}$, i.e., by $p(y_{1:\tau}|\mathcal{M}_i^{(b)},\beta_{1:\tau})$.

GMM lower bound - In [20] we derive the lower bound (9), by marginalizing $\mathrm{E}_{\mathcal{M}_{i,\beta_t}^{(b)}}$ over GMM assignment m, introducing the variational distributions $q_{\beta,\rho}^{i,j}(\zeta=l|m)$, and applying (6). We will restrict the variational distributions to $q_{\beta,\rho}^{i,j}(\zeta=l|m)=\eta_{\ell|m}^{(i,\beta),(j,\rho)}$, where $\sum_{\ell=1}^{M}\eta_{\ell|m}^{(i,\beta_t),(j,\rho_t)}=1\,\forall m$, and $\eta_{\ell|m}^{(i,\beta_t),(j,\rho_t)}\geq 0\,\forall\ell,m$. Intuitively, $\boldsymbol{\eta}^{(i,\beta_t),(j,\rho_t)}$ is the responsibility matrix between Gaussian observation components for state β_t in $\mathcal{M}_i^{(b)}$ and state ρ_t in $\mathcal{M}_j^{(r)}$, where $\eta_{\ell|m}^{(i,\beta_t),(j,\rho_t)}$ is the probability that an observation from component m of $\mathcal{M}_{i,\beta_t}^{(b)}$ corresponds to component ℓ of $\mathcal{M}_{j,\rho_t}^{(r)}$.

3.2 Variational HEM algorithm

Finally, the variational lower bound of the expected log-likelihood of the virtual samples in (2) is

$$\mathcal{J}(\mathcal{M}^{(r)})=\mathrm{E}_{\mathcal{M}^{(b)}}\left[\log p(Y|\mathcal{M}^{(r)})\right]\geq\sum_{i=1}^{K^{(b)}}\mathcal{L}_{H3M}^{i}, \qquad (15)$$

which is composed of three nested lower bounds, corresponding to different model elements (the H3M, the component HMMs, and the emission GMMs). The VHEM algorithm for HMMs consists in coordinate ascent on the right hand side of (15).

E-step - The variational E-step (see [20] for details) calculates the variational parameters z_{ij}, $\phi^{i,j}(\rho_{1:\tau}|\beta_{1:\tau}) = \phi^{i,j}_{\beta_1}(\rho_1)\prod_{t=2}^{\tau}\phi^{i,j}_{\beta_t}(\rho_t|\rho_{t-1})$, and $\boldsymbol{\eta}^{(i,\beta),(j,\rho)}$ for the lower bounds in (9) (13) (10). In particular, given the nesting of the lower bounds, we proceed by first maximizing the GMM lower bound $\mathcal{L}^{(i,\beta_t),(j,\rho_t)}_{GMM}$ for each (i,j,β_t,ρ_t). Next, the HMM lower bound $\mathcal{L}^{i,j}_{HMM}$ is maximized for each (i,j), which is followed by maximizing $\mathcal{L}^{i}_{H3M}$ for each i. The latter gives $\hat{z}_{ij} \propto w_j^{(r)}\exp(N_i\mathcal{L}^{i,j}_{HMM})$, which is similar to the formula derived in [8, 9], but the expectation is now replaced with its lower bound. We then collect the summary statistics: $\nu_1^{i,j}(\rho_1,\beta_1) = \pi^{(b),i}_{\rho_1}\hat{\phi}_1^{i,j}(\rho_1|\beta_1)$, $\xi_t^{i,j}(\rho_{t-1},\rho_t,\beta_t) = \left(\sum_{\beta_{t-1}=1}^{S}\nu_{t-1}^{i,j}(\rho_{t-1},\beta_{t-1})a^{(b),i}_{\beta_{t-1},\gamma_{t-1}}\right)\hat{\phi}_t^{i,j}(\rho_t|\rho_{t-1},\beta_t)$, and $\nu_t^{i,j}(\rho_t,\beta_t) = \sum_{\rho_{t-1}=1}^{S}\xi_t^{i,j}(\rho_{t-1},\rho_t,\beta_t)$, the last two for $t = 2,\ldots,\tau$, and their aggregates which are necessary for the M-step:

$$\hat{\nu}_1^{i,j}(\sigma) = \sum_{\beta=1}^{S}\nu_1^{i,j}(\sigma,\beta), \quad \hat{\nu}^{i,j}(\sigma,\beta) = \sum_{t=1}^{\tau}\nu_t^{i,j}(\sigma,\beta), \quad \hat{\xi}^{i,j}(\rho,\rho') = \sum_{t=2}^{\tau}\sum_{\beta=1}^{S}\xi_t^{i,j}(\rho,\rho',\beta). \tag{16}$$

The statistic $\hat{\nu}_1^{i,j}(\rho)$ is the expected number of times that the HMM $\mathcal{M}_j^{(r)}$ starts from state ρ, when modeling sequences generated by $\mathcal{M}_i^{(b)}$. The quantity $\hat{\nu}^{i,j}(\rho,\beta)$ is the expected number of times that the HMM $\mathcal{M}_j^{(r)}$ is in state ρ when the HMM $\mathcal{M}_i^{(b)}$ is in state β, when both are modeling sequences generated by $\mathcal{M}_i^{(b)}$. Similarly, the quantity $\hat{\xi}^{i,j}(\rho,\rho')$ is the expected number of transitions from state ρ to state ρ' of $\mathcal{M}_j^{(r)}$, when modeling sequences generated by $\mathcal{M}_i^{(b)}$.

M-step - The lower bound (15) is maximized with respect to the parameters $\mathcal{M}^{(r)}$. Defined a weighted sum operator $\Omega_{j,\rho,\ell}(x(i,\beta,m)) = \sum_{i=1}^{K^{(b)}}\hat{z}_{i,j}\omega_i^{(b)}\sum_{\beta=1}^{S}\hat{\nu}^{i,j}(\rho,\beta)\sum_{m=1}^{M}c^{(b),i}_{\beta,m}\,x(i,\beta,m)$, the parameters $\mathcal{M}^{(r)}$ are updated according to (derivation in [20]):

$$\omega_j^{(r)*} = \frac{\sum_{i=1}^{K^{(b)}}\hat{z}_{i,j}}{K^{(b)}},\ \pi_\rho^{(r),j*} = \frac{\sum_{i=1}^{K^{(b)}}\hat{z}_{i,j}\omega_i^{(b)}\hat{\nu}_1^{i,j}(\rho)}{\sum_{\rho'=1}^{S}\sum_{i=1}^{K^{(b)}}\hat{z}_{i,j}\omega_i^{(b)}\hat{\nu}_1^{i,j}(\rho'))},\ a^{(r),j*}_{\rho,\rho'} = \frac{\sum_{i=1}^{K^{(b)}}\hat{z}_{i,j}\omega_i^{(b)}\hat{\xi}^{i,j}(\rho,\rho')}{\sum_{\sigma=1}^{S}\sum_{i=1}^{K^{(b)}}\hat{z}_{i,j}\omega_i^{(b)}\hat{\xi}^{i,j}(\rho,\sigma)},$$

$$c^{(r),j*}_{\rho,\ell} = \frac{\Omega_{j,\rho,\ell}\left(\hat{\eta}^{(i,\beta),(j,\rho)}_{\ell|m}\right)}{\sum_{\ell'=1}^{M}\Omega_{j,\rho,\ell'}\left(\hat{\eta}^{(i,\beta),(j,\rho)}_{\ell'|m}\right)},\quad \mu^{(r),j*}_{\rho,\ell} = \frac{\Omega_{j,\rho,\ell}\left(\eta^{(i,\beta),(j,\rho)}_{\ell|m}\ \mu^{(b),i}_{\beta,m}\right)}{\Omega_{j,\rho,\ell}\left(\hat{\eta}^{(i,\beta),(j,\rho)}_{\ell|m}\right)}, \tag{17}$$

$$\Sigma^{(r),j*}_{\rho,\ell} = \Omega_{j,\rho,\ell}\left(\hat{\eta}^{(i,\beta),(j,\rho)}_{\ell|m}\left[\Sigma^{(b),i}_{\beta,m} + (\mu^{(b),i}_{\beta,m} - \mu^{(r),j}_{\rho,\ell})\ (\mu^{(b),i}_{\beta,m} - \mu^{(r),j}_{\rho,\ell})^t\right]\right)/\Omega_{j,\rho,\ell}\left(\hat{\eta}^{(i,\beta),(j,\rho)}_{\ell|m}\right). \tag{18}$$

Equations (17) and (18) are all weighted averages over all base models, model states, and Gaussian components. The covariance matrices of the reduced models (18) are never smaller in magnitude than the covariance matrices of the base models, due to the outer-product term. This regularization effect derives from the E-step, which averages all possible observations from the base model.

4 Discussion, Experiments and Conclusions

Jebara et al. [4] cluster a collection of HMMs by applying spectral clustering to a probability product kernel (PPK) matrix between HMMs. While this has been proven successful in grouping HMMs into similar clusters, it cannot learn novel HMM cluster centers and therefore is suboptimal for hierarchical estimation of mixture models (see Section 4.2). A second limitation is that the cost of building the PPK matrix is quadratic in the number $K^{(b)}$ of input HMMs. Note that we extended the algorithm in [4] to support GMM observations instead of only Gaussians.

The VHEM-H3M algorithm clusters a collection of HMMs directly through the distributions they represent, by estimating a smaller mixture of *novel* HMMs that concisely models the distribution represented by the input HMMs. This is achieved by maximizing the log-likelihood of "virtual" samples generated from the input HMMs. As a result, the VHEM cluster centers are consistent with the underlying generative probabilistic framework. As a first advantage, since VHEM-H3M estimates novel HMM cluster centers, we expect the learned cluster centers to retain more information on the clusters' structure and VHEM-H3M to produce better hierarchical clusterings than [4], which suffers out-of-sample limitations. A second advantage is that VHEM does not build a kernel embedding as in [4], an is therefore expected to be more efficient, especially for large $K^{(b)}$.

In addition, VHEM-H3M allows for efficient estimation of HMM mixtures from large datasets using a *hierarchical estimation procedure*. In particular, in a first stage intermediate HMM mixtures are estimated in *parallel* by running standard EM on small independent portions of the dataset, and the final model is estimated from the intermediate models using the VHEM algorithm. Relative to direct EM estimation on the entire data, VHEM-H3M is more time- and memory-efficient. First, it does not need to evaluate the likelihood of all the samples at each iteration, and converges to effective estimates in shorter times. Second, it no longer requires storing in memory the entire data set during parameter estimation. Another advantage is that the intermediate models implicitly provide more "samples" (virtual variations of each time-series) to the final VHEM stage. This acts as a form of *regularization* that prevents over-fitting and improves robustness of the learned models. Therefore, we expect models learned using the hierarchical estimation procedure to perform better than those learned with EM directly on the entire data. Note that in the second stage we could use the spectral clustering algorithm in [4] instead of VHEM — run spectral clustering over intermediate models pooled together, and form the final H3M with the HMMs mapped the closest to the K cluster centers. VHEM, however, is expected to do better since it learns *novel* cluster centers. As an alternative to VHEM, we tested a version of HEM that, instead of marginalizing over virtual samples, uses actual sampling and the EM algorithm [5] to learn the reduced H3M. Despite its simplicity, the algorithm requires a large number of samples for learning accurate models, and has longer learning times (since it evaluates the likelihood of all samples at each iteration).

4.1 Experiment on hierarchical motion clustering

Table 2: Hierarchical clustering on Motion Capture data, using various algorithms. The Rand-index is the probability that any pair of motion sequences are correctly clustered with respect to each other. Results are averages of 10 trials.

	Rand-index			log-likelihood ($\times 10^6$)			time (s)
Level (#samples)	2	3	4	2	3	4	
VHEM-H3M	0.937	0.811	0.518	-5.361	-5.682	-5.866	30.97
PPK-SC	0.956	0.740	0.393	-5.399	-5.845	-6.068	37.69
SHEM-H3M (560)	0.714	0.359	0.234	-13.632	-69.746	-275.650	843.89
SHEM-H3M (2800)	0.782	0.685	0.480	-14.645	-30.086	-52.227	3849.72
EM-H3M	0.831	0.430	0.340	-5.713	-202.55	-168.90	667.97
HEM-DTM	0.897	0.661	0.412	-7.125	-8.163	-8.532	121.32

Table 3: Annotation and retrieval on CAL500, for VHEM-H3M, PPK-SC, EM-H3M, HEM-DTM and HEM-GMM, averaged over the 97 tags with at least 30 examples in CAL500, and result of 5 fold-cross validation.

	annotation			**retrieval**		
	P	R	F	MAP	P@10	time (h)
VHEM-H3M	0.446	0.211	0.260	0.440	0.451	678
EM-H3M	0.415	0.214	0.248	0.423	0.422	1860
PPK-SC	0.299	0.159	0.151	0.347	0.340	1033
HEM-DTM	0.430	0.202	0.252	0.439	0.453	426
HEM-GMM	0.374	0.205	0.213	0.417	0.425	5

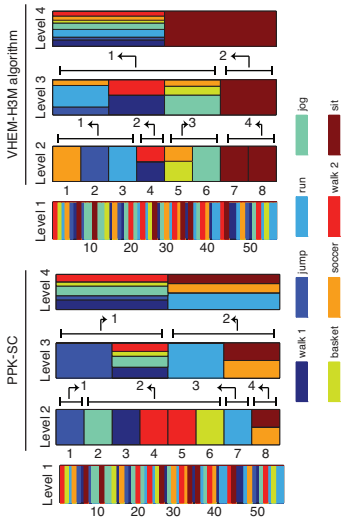

Figure 1: Hierarchical clustering of Motion Capture data (qualitative). *Best in color.*

We tested the VHEM algorithm on hierarchical motion clustering, where each of the input HMMs to be clustered is estimated on a sequence of motion capture data from the Motion Capture dataset (*http://mocap.cs.cmu.edu/*). In particular, we start from $K_1 = 56$ motion examples from 8 different classes ("jump", "run", 'jog'", "walk 1" and "walk 2" which are from two different subjects, "basket", "soccer", "sit"), and learn a HMM for each of them, forming the first level of the hierarchy. A tree-structure is formed by successively clustering HMMs with the VHEM algorithm, and using the learned cluster centers as the representative HMMs at the new level. Level 2, 3, and 4 of the hierarchy correspond to $K_2 = 8$, $K_3 = 4$ and $K_4 = 2$ clusters.

The hierarchical clustering obtained with VHEM is illustrated in Figure 1 (top). In the first level, each vertical bar represents a motion sequence, and different colors indicate different ground-truth classes. At Level 2, the 8 HMM clusters are shown with vertical bars, with the colors indicating the proportions of the motion classes in the cluster. At Level 2, VHEM produces clusters with examples from a single motion class (e.g., "run", "jog", "jump"), but mixes some "soccer" examples with "basket", possibly because both actions consists in a sequence of movement-shot-pause. Moving up the hierarchy, VHEM clusters similar motions classes together (as indicated by the arrows), and at Level 4 it creates a dichotomy between "sit" and the other (more dynamic) motion classes. On the

bottom, in Figure 1, the same experiment is repeated using spectral clustering in tandem with PPK similarity (PPK-SC). PPK-SC clusters motion sequences properly, however, at Level 2 it incorrectly aggregates "sit" and "soccer" that have quite different dynamics, and Level 4 is not as interpretable as the one by VHEM. Table 2 provides a *quantitative* comparison. While VHEM has lower Rand-index than PPK-SC at Level 2 (0.937 vs. 0.956), it has higher Rand-index at Level 3 (0.811 vs. 0.740) and Level 4 (0.518 vs. 0.393). In addition, VHEM-H3M has higher data log-likelihood than PPK-SC at each level, and is more efficient. This suggests that the novel HMM cluster centers learned by VHEM-H3M retain more information on the clusters' structure than the spectral cluster centers, which is increasingly visible moving up the hierarchy. Finally, VHEM-H3M performs better and is more efficient than the HEM version based on actual sampling (SHEM-H3M), the EM applied directly on the motion sequences, and the HEM-DTM algorithm [9].

4.2 Experiment on automatic music tagging

We evaluated VHEM-H3M on content-based music auto-tagging on the CAL500 [11], a collection of 502 songs annotated with respect to a vocabulary $\mathcal{V}$ of 149 tags. For each song, we extract a time series $\mathcal{Y} = \{y_1, \ldots, y_T\}$ of 13 Mel frequency cepstral coefficients (MFCC) [1] over half-overlapping windows of 46ms, with first and second instantaneous derivatives. We formulate music auto-tagging as supervised multi-class labeling [10], where each class is a tag from $\mathcal{V}$ and is modeled as a H3M probability distribution estimated from audio-sequences (of $T = 125$ audio features, i.e., approximately $3s$ of audio) extracted from the relevant songs in the database, using the VHEM-H3M algorithm. First, for each song the EM algorithm is used to learn a H3Ms with $K^{(s)} = 6$ components (as many as the structural parts of most pop songs). Then, for each tag, the relevant song-level H3Ms are pooled together and the VHEM-H3M algorithm is used to learn the final H3M tag model with $K = 3$ components.

We compare the proposed VHEM-H3M algorithm to PPK-SC,[1] direct EM-estimation (EM-H3M) [5] from the relevant songs' audio sequences, HEM-DTM [12] and HEM-GMM [11]. The last two use an efficient HEM algorithm for learning, and are state-of-the-art baselines for music tagging. We were not able to successfully estimate tag-H3Ms with the sampling version of HEM-H3M. Annotation (precision P, recall R, and f-score F) and retrieval (mean average precision MAP, and top-10 precision P@10) are reported in Table 3. VHEM-H3M is the most efficient algorithm for learning H3Ms as it requires only 36% of the time of EM-H3M, and 65% of the time of PPK-SC. VHEM-H3M capitalizes on the song-level H3Ms learned in the first stage (about one third of the total time), by efficiently using them to learn the final tag models. The gain in computational efficiency does not negatively affect the quality of the resulting models. On the contrary, VHEM-H3M achieves better performance than EM-H3M (differences are statistically significant based on a paired t-test with 95% confidence), since it has the benefit of regularization, and outperforms PPK-SC. Designed for clustering HMMs, PPK-SC does not produce accurate annotation models, since it discards information on the clusters' structure by approximating it with one of the original HMMs. Instead, VHEM-H3M generates novel HMM cluster centers that effectively summarizes each cluster. VHEM-H3M outperforms HEM-GMM, which does not model temporal information in the audio signal. Finally, HEM-DTM, based on LDSs (a continuous-state model), can model only stationary time-series in a linear subspace. In contrast, VHEM-H3M uses HMMs with discrete states and GMM emissions, and can also adapt to non-stationary time-series on a non-linear manifold. Hence, VHEM-H3M outperforms HEM-DTM on the human MoCap data (see Table (2)), which has non-linear dynamics, while the two perform similarly on the music data (difference were statistically significant only on annotation P), where the audio features are stationary over short time frames.

4.3 Conclusion

We presented a variational HEM algorithm for clustering HMMs through their distributions and generates novel HMM cluster centers. The efficacy of the algorithm was demonstrated on hierarchical motion clustering and automatic music tagging, with improvement over current methods.

Acknowledgments

The authors acknowledge support from Google, Inc. E.C. and G.R.G.L. acknowledge support from Qualcomm, Inc., Yahoo! Inc., and the National Science Foundation (grants CCF-083053, IIS-1054960 and EIA-0303622). A.B.C. acknowledges support from the Research Grants Council of the Hong Kong SAR, China (CityU 110610). G.R.G.L. acknowledges support from the Alfred P. Sloan Foundation.

[1] It was necessary to implement PPK-SC with song-level H3Ms with $K^{(s)}$=1. $K^{(s)}$=2 took about quadruple the time with no improvement in performance. Larger $K^{(s)}$ would determine impractical learning times.

References

[1] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Upper Saddle River (NJ, USA), 1993.

[2] Y. Qi, J.W. Paisley, and L. Carin. Music analysis using hidden markov mixture models. *Signal Processing, IEEE Transactions on*, 55(11):5209–5224, 2007.

[3] E. Batlle, J. Masip, and E. Guaus. Automatic song identification in noisy broadcast audio. In *IASTED International Conference on Signal and Image Processing*. Citeseer, 2002.

[4] T. Jebara, Y. Song, and K. Thadani. Spectral clustering and embedding with hidden markov models. *Machine Learning: ECML 2007*, pages 164–175, 2007.

[5] P. Smyth. Clustering sequences with hidden markov models. In *Advances in neural information processing systems*, 1997.

[6] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004.

[7] B. H. Juang and L. R. Rabiner. A probabilistic distance measure for hidden Markov models. *AT&T Technical Journal*, 64(2):391–408, February 1985.

[8] N. Vasconcelos and A. Lippman. Learning mixture hierarchies. In *Advances in Neural Information Processing Systems*, 1998.

[9] A.B. Chan, E. Coviello, and G.R.G. Lanckriet. Clustering dynamic textures with the hierarchical em algorithm. In *Intl. Conference on Computer Vision and Pattern Recognition*, 2010.

[10] G. Carneiro, A.B. Chan, P.J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):394–410, 2007.

[11] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):467–476, February 2008.

[12] E. Coviello, A. Chan, and G. Lanckriet. Time series models for semantic music annotation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 5(19):1343–1359, 2011.

[13] A. Banerjee, S. Merugu, I.S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *The Journal of Machine Learning Research*, 6:1705–1749, 2005.

[14] J.R. Hershey, P.A. Olsen, and S.J. Rennie. Variational Kullback-Leibler divergence for hidden Markov models. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 323–328. IEEE, 2008.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[16] R.M. Neal and G.E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *NATO ASI SERIES D BEHAVIOURAL AND SOCIAL SCIENCES*, 89:355–370, 1998.

[17] I. Csisz, G. Tusnády, et al. Information geometry and alternating minimization procedures. *Statistics and decisions*, 1984.

[18] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[19] Tommi S. Jaakkola. Tutorial on Variational Approximation Methods. In *In Advanced Mean Field Methods: Theory and Practice*, pages 129–159. MIT Press, 2000.

[20] Anonymous. Derivation of the Variational HEM Algorithm for Hidden Markov Mixture Models. Technical report, Anonymous, 2012.