Clustering Hidden Markov Models with Variational Bayesian Hierarchical EM

Hui Lan, Ziquan Liu, Janet H. Hsiao, Dan Yu, Antoni B. Chan

Abstract—The hidden Markov model (HMM) is a broadly applied generative model for representing time series data, and clustering HMMs attracts increased interests from machine learning researchers. However, the number of clusters ($K$) and the number of hidden states ($S$) for cluster centers are still difficult to determine. In this paper, we propose a novel HMM-based clustering algorithm, the variational Bayesian hierarchical EM algorithm, which clusters HMMs through their densities and priors, and simultaneously learns posteriors for the novel HMM-based clustering algorithm, the variational Bayesian hierarchical EM algorithm, which clusters HMMs through their densities and priors, and simultaneously learns posteriors for the novel HMM cluster centers that compactly represent the structure of each cluster. The numbers $K$ and $S$ are automatically determined in two ways. First, we place a prior on the pair $(K, S)$ and approximate their posterior probabilities, from which the values with the maximum posterior are selected. Second, some clusters and states are pruned out implicitly when no data samples are assigned to them, thereby leading to automatic selection of the model complexity. Experiments on synthetic and real data demonstrate that our algorithm performs better than using model selection techniques with maximum likelihood estimation.

Index Terms—Variational Bayesian, Hidden Markov mixture model, Clustering, Hierarchical EM

I. INTRODUCTION

THE hidden Markov model (HMM) [1] is an effective method for statistically representing time series data, assuming that each observation in a sequence is generated conditioned on a discrete state of a hidden Markov chain, i.e., a hidden state sequence. HMM has been popularly applied in many areas that need to analyze time series data, such as speech recognition [2, 3], cognitive science [4, 5], and bioinformatics [6, 7]. Although neural networks (NN) [8, 9] and reinforcement learning [10] are also works in these areas, they typically require large datasets to prevent over-fitting and learn models that are difficult to interpret. In contrast, as a generative probabilistic model, HMMs work well on smaller datasets with Bayesian estimation preventing over-fitting, while also being interpretable models.

Clustering HMMs to explore the hidden cluster structure can be an effective method for discovering commonalities and differences among HMMs, and the cluster center serves as a representation of the HMMs in each cluster. In particular, recent works represent an individual’s eye gaze pattern by observation sequences, with each group modeled by one HMM. For example, [22] proposed a Dirichlet process for learning an HMM mixture from music clips to represent a song, while [23] clusters sequences by forming a single HMM with a block-diagonal transition matrix and then training on all sequences with the Baum-Welch [24] algorithm. In contrast, clustering HMMs aims to form $K$ groups of $N$ HMMs, with each group represented by one HMM. Clustering HMMs is equivalent to building a large mixture model of HMMs, and then reducing it into a mixture model with fewer components and states, which can concisely represent the original HMMs. To this end, [25] proposes a variational hierarchical EM (VHEM) algorithm to cluster HMMs directly using their probability densities of the observation sequence, by minimizing the Kullback-Leibler divergence (KLD) [26] between the input HMMs and cluster center HMMs. In summary, clustering time series data with HMMs is a process mapping data to model, while clustering HMMs maps from model to model. Clustering HMMs is preferred in the above cognitive science works because it allows modeling and analysis of both individual

Hui Lan is with the School of Statistics and Data Science, Faculty of Science, Beijing University of Technology, Beijing, China.
Hui Lan, Ziquan Liu and Antoni Chan are with the Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China.
Janet H. Hsiao is with the Department of Psychology, The University of Hong Kong, Hong Kong SAR, China.
Dan Yu is with the Institute of Systems Science, Academy of Mathematics and System Science, Chinese Academy of Sciences, China.

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Our Contributions. In this paper, we propose a variational hierarchical EM algorithm that clusters HMMs within a Bayesian framework (VBHEM-H3M). VBHEM automatically determines the number of clusters and the number of hidden states, clusters HMMs directly, and estimates novel HMM cluster centers. Compared with VHEM, firstly, VBHEM automatically performs model selection (i.e., estimates the model hyperparameters, e.g., the number of clusters), while VHEM cannot. Automatic model selection is important for analyzing experimental data, since it removes the experimenter bias caused when manually selecting the model hyperparameters. Secondly, each input HMM of VBHEM is represented with a prior distribution over its parameters, which better incorporates the uncertainty of parameter estimation of the input HMMs as they are estimated from limited data samples. In contrast, this uncertainty information about the input HMMs is discarded by VHEM. In experiment, we demonstrate that using the input HMM uncertainty leads to better clustering performance; Thirdly, VBHEM computes a posterior distribution over the HMM cluster centers (e.g., a point-estimate obtained by VHEM), which gives a better characterization of the uncertainty in the estimated model. More detailed comparisons can be found in Sec. IV-D. Finally, we give complete derivations of Bayesian HEM algorithm for H3M with its prior distributions, which could be generalized to other mixture of mixture models.

The remainder of this paper is organized as follows. In Sec. II, we provide the related works. In Sec. III we review the necessary related knowledge. In Sec. IV we introduce a new objective function and derive the VBHEM algorithm. Sec. V presents experimental results obtained by applying VBHEM algorithm to synthetic data and real data. Finally, Sec. VI concludes this paper.

II. Related Work

Clustering HMMs. The existing approaches to cluster HMMs leverage different distances or similarity between two HMMs. [27] clustered HMMs by calculating a probability product kernel (PPK) similarity matrix between all HMMs, and then applying spectral clustering. [25] proposed the variational hierarchical EM (VHEM) algorithm to cluster HMMs directly using their probability densities of the observation sequence and estimate HMM cluster centers, via minimizing the KLD between input HMMs and cluster center HMMs. [28] modeled each gene sequence with an HMM and defined a distance matrix based on likelihood, and applied a hierarchical clustering algorithm to find the best clusters. [29] used a bagging method, where a large set of HMMs is computed from the data, the HMMs are grouped together based on KLD, and the cluster centers are found by averaging HMMs in the same group. For works [28, 29], data is used in the clustering process, while [25, 27] are only based on the input HMMs. [30] proposed a framework, Aggregated Wasserstein, for computing distances between two HMMs with state conditional distributions as Gaussians. However, clustering based on Wasserstein distance has not been studied for HMMs.

For the above HMM clustering methods, the numbers of clusters and states are set manually. Therefore, these methods need to resort to other model selection techniques, such as Akaike information criterion (AIC) [31], Bayesian information criterion (BIC) [32], Monte-Carlo cross-validation [33], and minimum description length (MDL) [34]. In our work, we propose a method that both clusters HMMs directly and automatically performs model selection via Bayesian formulation.

Bayesian Model Selection. The Bayesian view of model comparison involves the use of probabilities to represent uncertainty in the choice of model – the model structure is determined using the posterior distribution over the model structure, conditioned on the training data. Suppose we wish to compare a set of $M$ models $m_i$ where $i \in \{1, \ldots, M\}$. We then need to evaluate the posterior distribution $p(m_i|Y) \propto p(m_i)p(Y|m_i)$, where the uncertainty is expressed through a model prior probability distribution $p(m_i)$, and the model evidence $p(Y|m_i)$ expresses the preference shown by the data $Y$ for different models. Variational inference (VI) [35, 36] is widely used to approximate the model evidence for Bayesian models, while an alternative, but computationally expensive, strategy is Markov chain Monte Carlo (MCMC) sampling [37]. VI first posits a family of densities for the approximate posterior distribution, and then finds a member of that family that is closest to the target density, as measured by KLD.

The previous works using VI with Bayesian model selection focus on mixture of experts model [38], HMMs [39], and Dirichlet process mixture models [40]. In most cases, the model evidence $p(Y|m_i)$ is intractable, and thus the evidence lower bound (ELBO) is used as a model selection criterion. VI has been explored for mixture models [41, 42] and more generally [43]. [44] derived VI in the Bayesian framework for hidden Markov mixture models (H3M) [23], but [44] only considered learning H3M from data, not from the HMMs. [45] proposed a VB method for clustering Gaussians, i.e., learning a Gaussian mixture model from a set of input GMMs. However, a VB method for clustering HMMs, by learning an H3M from a set of input HMMs, has not been studied so far.

Other model selection methods. Other clustering methods also focus on automatic selection of the number of cluster centers, but are not proposed for HMMs. [46] proposed an approach based on the idea that cluster centers are characterized by a higher density than their neighbors, and by a relatively large distance from points with higher densities. However, this method needs to draw a rectangle to manually select the cluster centers. [47] proposed a cluster center fast determination (CCFD) algorithm, which overcomes this problem and realizes automatic selection of the cluster centers. CCFD has been applied to image segmentation [48] and hybrid data stream clustering [49].

III. Preliminaries

A. Hidden Markov (Mixture) Model

We first briefly review hidden Markov models (HMMs) and the hidden Markov mixture model (H3M) [23], and define the notation used in the derivation (see summary in Table I, II).

An H3M models a set of observation sequences as samples from a group of $K$ hidden Markov models (HMMs), and is parameterized by $M = \{\omega_i, M_i\}_{i=1}^K$, where $M_i$ is the $i$-th HMM and $\omega_i$ is the corresponding mixture component
weight. An observation sequence with length \( \tau \) is denoted by \( y = (y_1, y_2, ..., y_\tau) \), and depends on a hidden state sequence \( x = (x_1, x_2, ..., x_\tau) \). The observation likelihood for \( y \sim \mathcal{M} \) is \( p(y|\mathcal{M}) = \sum_i \omega_i p(y_i|\mathcal{M}_i) \). The observation \( y \sim \mathcal{M} \) is stated is specified by parameters \( \mathcal{M}_i = \{\pi_i, A_i, \Theta_i\}_{i=1}^K \).

In detail, \( \pi_i = [\pi_{i1}, ..., \pi_{iK}] \) is the initial state probability, where \( \pi_{ij} = p(x_1 = \beta_i | \mathcal{M}_i) \). \( A_i = (a_{ij})_{K \times K} \) is the state transition matrix, where \( a_{ij} = p(\bar{x}_{i+1} = \beta_j | x_i = \beta_i, \mathcal{M}_i) \) is the transition probability from state \( \beta_i \) to \( \beta_j \). Thus the probability of a state sequence \( \beta = (\beta_1, ..., \beta_\tau) \) will be \( p(\beta | \mathcal{M}_i) = \pi_{i1} \prod_{t=2}^{\tau} a_{\beta_{t-1}, \beta_t} \). \( \Theta_i \) is the parameter set of emission density at state \( \beta_i \). Here, we assume a Gaussian distribution, \( p(y_i|x_i = \beta_i, \mathcal{M}_i) = N(y_i | \mu_{i\beta}, \Lambda_{i\beta}) \), with mean \( \mu_{i\beta} \) and precision matrix \( \Lambda_{i\beta} \). Thus, the probability of an observation \( y \) generated by \( \mathcal{M}_i \) will be \( p(y_i|\mathcal{M}_i) = N(y_i | \mu_{i\beta}, \Lambda_{i\beta}) \), with mean \( \mu_{i\beta} \) and precision matrix \( \Lambda_{i\beta} \).

Consider the H3M with unknown number of components \( K \) and number of states \( S \), and treating them as random variables, the observation likelihood will be \( p(y|\mathcal{M}) = \sum_{K,S} p(K,S)p(y|\mathcal{M}, K, S) \), where \( p(K,S) \) is a prior of pair \((K,S)\) and the summation is over all candidate number of components \( K \in [K_{\text{min}}, K_{\text{max}}] \) and states \( S \in [S_{\text{min}}, S_{\text{max}}] \), where we use shorthand \( [A,B] = \{A, B, \ldots\} \).

B. Variational Bayesian Inference

A central task in the application of probabilistic models is evaluating the posterior distribution \( p(H|Y) \) of the hidden (latent) variables \( H \) given the observed data \( Y \). In a fully Bayesian framework, any unknown model parameters are given prior distributions and are absorbed into the set of latent variables \( H \). When it is infeasible to evaluate the posterior distribution directly, e.g., it has a highly complex form, then variational inference can be used to approximate \( p(H|Y) \) with a variational distribution \( q(H) \). Furthermore, to consider different model structures, the number of mixture components and hidden states \((K,S)\) can be considered as latent variables with prior distributions. Hence, we introduce a variational distribution \( q(H, K, S) \) as an approximation of the true posterior distribution \( p(H, K, S|Y) \).

The VB framework for an H3M is formulated as follows. The marginal log-likelihood (i.e., model evidence) \( \log p(Y) \) is decomposed into a lower-bound and Kullback-Leibler divergence (KLD) term [26] (see derivation in Appendix A),

\[
\log p(Y) = \mathcal{L}(q) + \text{KL}(q||p)
\]

where we define

\[
\mathcal{L}(q) = \sum_{K,S} q(K,S) \left[ \mathcal{L}_{(K,S)}(q) + \log p(K,S) / q(K,S) \right],
\]

\[
\mathcal{L}_{(K,S)}(q) = \int q(H|K,S) \log \frac{p(Y, H|K,S)}{q(H|K,S)} dH,
\]

\[
\text{KL}(q||p) = \sum_{K,S} \int q(H, K, S) \log \frac{q(H, K, S)}{p(H, K, S|Y)} dH.
\]
mixture model $\mathcal{M}_r$ and automatically determine the number of components $K^{(r)}$ and states $S^{(r)}$ in $\mathcal{M}_r$. Rather than learning a single model $\mathcal{M}_r$ as in VHEM, VBHEM-H3M estimates a posterior distribution over the reduced model's parameters and structures. The reduced model is denoted by $\mathcal{M}_r = \{\omega^{(r)}, M^{(r)}\}_{r=1}^{K^{(r)}}$ with $K^{(r)}$ components and $S^{(r)}$ states, where $K^{(b)} > K^{(r)}$ and $S^{(b)} \geq S^{(r)}$. In the Bayesian framework, we also assume priors on all unknown parameters in $\mathcal{M}_r$, denoted by $p(K^{(r)}, S^{(r)}), p(\omega^{(r)}), p(M^{(r)}), p(\mu^{(r)}_0 | A^{(r)}_0), p(\Lambda^{(r)}_0, j), p(\pi^{(r)}_j)$, and $p(\alpha^{(r)}_j)$, where $\alpha^{(r)}_j$ is a row of $A^{(r)}_j$. Note that we will always use superscripts ($b$) and ($r$) to distinguish the parameters for base and reduced model, and $i$ and $j$ to index the mixture component in the base and reduced models, and $\beta$ and $\rho$ to index the hidden states in the base and reduced models, respectively. Table I and II summarize the notation used in the derivation, including the variable names, latent variables, model names, prior distributions and expectations and expected log-likelihood.

A. Framework

One possible solution to estimate $\mathcal{M}_r$ is to directly sample from $\mathcal{M}_b$ and then estimate $\mathcal{M}_r$ with any needed number of components and states by the EM algorithm. However, this would be inefficient when handling large-scale high-dimensional data. Also, the number of components and states must be set by hand, which introduces experimenter bias. Instead, we take our inspiration from VBmerge [45], which reduces a Gaussian mixture model (GMM) by directly clustering the Gaussian components in the Bayesian framework, and the number of components is simultaneously determined.

We define a set of $N$ sequence samples $Y = \{y_1, y_2, ..., y_N\}$, where $y_n = (y_{n,1}, y_{n,2}, ..., y_{n,T})$ is a sequence, and $y_{n,t} \in \mathbb{R}^d$ is the observation at time $t$. The generative process of the data set $Y$ is:

1) Sample a base model $\mathcal{M}_b \sim p(\mathcal{M}_b)$;

2) Sample (i.i.d.) data sequences $y_n \sim \mathcal{M}_b$, $n = [1, N]$.

Thus, the marginal likelihood over the data according to the base model prior $p(\mathcal{M}_b)$ is

$$p'(Y) = \int p(Y | \mathcal{M}_b)p(\mathcal{M}_b)d\mathcal{M}_b$$

(6)

A similar generative process also exists for the reduced model, and thus the marginal likelihood of the data according to the reduced model prior $p(\mathcal{M}_r)$ is

$$p(Y) = \int p(Y | \mathcal{M}_r)p(\mathcal{M}_r)d\mathcal{M}_r$$

(7)

A typical VB method constructs a lower bound of the evidence $\log p(Y)$, under the model being learned, which in our case is the reduced model $\mathcal{M}_r$. However, in our scenario, we do not have direct access to the data $Y$, but instead have access to the model $\mathcal{M}_b$ that generates the data. Thus, our starting point is the expected evidence, where the “evidence” $Y \sim p'$ is generated from the input model $\mathcal{M}_b$, and evaluated according to the reduced model $\mathcal{M}_r$.

$$E_{Y \sim p'} \log p(Y)$$

$$= \int \int p(Y | \mathcal{M}_b)p(\mathcal{M}_b)d\mathcal{M}_b \log p(Y) dY$$

$$= \mathbb{E}_{\mathcal{M}_b} \mathbb{E}_{Y \sim \mathcal{M}_b} \log p(Y),$$

(8)
where the exchange of the integral is guaranteed by Fubini’s theorem. Substituting the lower bound in (1) of the marginal log-likelihood into (8), we have
\[
\mathbb{E}_{M(b)} \mathbb{E}_{Y|M(b)} \log p(Y) \geq \sum_{K(r)} \sum_{S(r)} q(K(r), S(r)),
\]
\[
\left[ \mathbb{E}_{Y} \mathbb{L}(K(r), S(r)) \right] + \log \frac{p(K(r), S(r))}{q(K(r), S(r))} = \mathbb{L}(q),
\]
(9)
where the inequality holds due to the non-negativity of probability density functions. Note that the expectation \( \mathbb{E}_{Y} \cdot [ \mathbb{L}(K(r), S(r)) \cdot q ] \) on the RHS of (9). The set of hidden variables is
\[
H = \{ Z, (\omega)_{r,j}, (\pi)_{r,j}, (\mu)_{r,j}, (A)_{r,j} \}_{j=1}^{K(r)}.
\]
Looking at each hidden variable \( H_i \) in \( H \) and using (3), the optimal solution for \( q^*_i(H_i | K(r), S(r)) \) is
\[
\log q^*_i(H_i | K(r), S(r)) \propto \mathbb{E}_{Y|\neq i} \mathbb{E}_{Y} \log p(Y, H | K(r), S(r)),
\]
and using (5), the optimal \( q^*(K(r), S(r)) \) is
\[
\log q^*(K(r), S(r)) \propto \log p(K(r), S(r)) + \mathbb{E}_{Y} \mathbb{L}(K(r), S(r)) \cdot q).
\]
(10)
From (10-11), our algorithm contains two steps:

**Step 1:** For each candidate pair \((K(r), S(r))\), calculate each optimal solution \( q^*_i(H_i | K(r), S(r)) \).

**Step 2:** Find the optimal model structure through
\[
(K(r), S(r)) = \arg \max_{K(r), S(r)} \log q^*(K(r), S(r)).
\]
However, the expectation in (10) cannot be calculated in the closed-form. We will show how to approximate that in the following sections.

### B. Priors

In this section, we introduce the conjugate prior distributions over the parameters of the H3M \( M \) with \( K \) components and \( S \) states (see Fig. 1),
\[
p(M | K, S) = p(\omega | K, S) \prod_{i=1}^{K} p(M_i | K, S),
\]
\[
= p(\omega | K, S) \prod_{i} p(\pi_{i} | K, S) p(\mu_{i} | K, S) p(\lambda_{i} | K, S).
\]
Here, we assume that \((K, S)\) are fixed and do not explicitly condition on them to remove clutter. The priors for \( M(b) \) are
\[
p(\omega(b)) = \text{Dir} (\omega(b) | \alpha_0),
\]
\[
p(\pi_{i}(b)) = \text{Dir} (\pi_{i}(b) | \gamma_{0,i}),
\]
\[
p(\mu_{i}(b)) = \prod_{\beta} \text{Dir} (\mu_{i}(b) | \epsilon_{\beta,0,i}),
\]
\[
p(\lambda_{i}(b)) = \prod_{\beta} \text{Dir} (\lambda_{i}(b) | \nu_{\beta,0,i}),
\]
where \( \text{Dir} (\cdot | \alpha) \) is a Dirichlet distribution with concentration vector \( \alpha \), \( \mathcal{N} (\cdot | \mu, \Sigma) \) is a Gaussian distribution with mean \( \mu \) and covariance \( \Sigma \), and \( \mathcal{W} (\cdot | W, \nu) \) is a Wishart distribution with scale matrix \( W \) and degrees-of-freedom \( \nu \) (see Appendix B for the details of each distribution). The hyperparameters of

### Algorithm 1: Optimizing the Variational Distribution

**Input:** hyperparameter sets \( \mathcal{P}(b) \) and \( \mathcal{P}(r) \), and the number of virtual samples \( N \), clusters \( K(r) \), and states \( S(r) \).

**Output:** variational distributions \( q^{*}(Z) \), \( q^{*}(\mathcal{W}) \), \( q^{*}(\mathcal{A}) \), \( q^{*}(\mathcal{M}) \), \( q^{*}(\mathcal{V}) \), \( j \in [1, K] \).

1: Pre-process base model using (16)-(21).
2: repeat
3: VBM E-step: compute responsibilities \( \hat{z}_{i,j} \) using (22).
4: VBM M-step: update variational parameters \( \alpha^{(r)} \), \( \eta^{(r)} \), \( m^{(r)} \), \( \lambda^{(r)} \), \( \mathcal{M}^{(r)} \), \( \mathcal{V}^{(r)} \) for each \( j \) using (24)-(25).
5: until convergence of \( \mathbb{E}_{Y} \mathbb{L}(K(r), S(r)) \).

The base model is summarized as the set \( \mathcal{P}(b) = \{ \alpha_0, \gamma_0, \mu_0, \lambda_0, W_0, \nu_0 \} \) and states \( \mathcal{P}(r) = \{ \epsilon_0, \gamma_0, \lambda_0, m_0, \nu_0 \} \). The priors for reduced model \( \mathcal{M}(r) \) share the same hyper-parameters as \( \mathcal{P}(b) \) but with simpler hyperparameters. We set \( \alpha_0, \gamma_0 = \alpha_0(b) \) (scalar), \( \mu_0, \gamma_0, \lambda_0 = \gamma_0(b) \), \( m_0, \nu_0 = \nu_0(b) \) which are common for all components \( j \) and states \( \rho \) in the reduced model. Similarly, we sum the hyperparameters as the set \( \mathcal{P}(r) = \{ \epsilon_0, \gamma_0, \lambda_0, m_0, \nu_0 \} \). For the priors over the number of components \( K(r) \) and states \( S(r) \), we assume a Poisson distribution on \( K(r) \) and a uniform distribution on \( S(r) \),
\[
p(K(r) = K, S(r) = S) = \frac{K^K e^{-\lambda_0}}{S_{\text{max}} - S_{\text{min}} + 1}.
\]
This prior allows us to express a preference for different models, through the hyperparameter \( \lambda_0 \).

### C. Optimizing the Variational Distribution

We next explain how to utilize the prior distributions over the parameters of base model \( M(b) \) and optimize the variational distribution. We assume grouped observations \( Y = \{ Y_1, ..., Y_K(b) \} \) as in [50]. The subset \( Y_i \) has size \( N_i = N_u(b) \), and consists of all \( y_r \) that are generated by \( M(b) \). Similarly the grouped assignments are \( Z = \{ z_1, ..., z_K(b) \} \), and \( z_i \) is a 1-of-\( K \) binary vector where each element is an indicator variable \( z_{ij} \), with \( z_{ij} = 1 \) if the observations \( Y_i \) are assigned to the \( j \)-th reduced model \( M^{(r)}(j) \), and \( z_{ij} = 0 \) otherwise.

Revisiting (10), the joint distribution of random variables \( Y \) and \( H \) condition on \( K(r) \) and \( S(r) \) is
\[
\log p(Y, H | K(r), S(r))
\]
\[
= \log p(Y | Z, \{ M^{(r)}(j) \}, K(r), S(r)) + \log p(\omega^{(r)} | K(r))
\]
\[
+ \log p(\mathcal{W} | \omega^{(r)}, K(r)) + \sum_{j} \log p(\mathcal{M}^{(r)} | K(r), S(r)).
\]
In the following, we consider a fixed pair \((K(r), S(r))\), and do not explicitly write the dependence to reduce clutter. Note that taking the expectation \( \mathbb{E}_{Y} \cdot [ \mathbb{L}(K(r), S(r)) \cdot q) \) w.r.t. (12) only affects the first term in the RHS of (12), i.e., (see Appendix C for the detailed derivation),
\[
\mathbb{E}_{Y} \mathbb{L}(K(r), S(r))
\]
\[
= \sum_{i,j} z_{ij} N \mathbb{E}_{\omega^{(b)}} \left[ \mathcal{L}_{(b)} \right] \mathbb{E}_{\mathcal{M}^{(b)}} \mathbb{E}_{Y | \mathcal{M}^{(b)}} \left[ \log p(y | \mathcal{M}^{(r)}(j)) \right].
\]
Then, we consider a variational distribution which factorizes between the latent variables and the parameters so that
\[
q(\mathbf{Z}, \omega^{(r)}, \pi^{(r)}, \Lambda^{(r)}, \mu^{(r)}, \Lambda^{(r)}) = q(\mathbf{Z})q(\omega^{(r)}, \pi^{(r)}, \Lambda^{(r)}, \mu^{(r)}, \Lambda^{(r)}).
\]

The corresponding sequential update equations for these factors can be derived from (10). In particular, the functional form of the factors \(q(\mathbf{Z})\) and \(q(\omega^{(r)}, \pi^{(r)}, \Lambda^{(r)}, \mu^{(r)}, \Lambda^{(r)})\) will be determined automatically by optimization of the variational distribution. The whole algorithm for optimizing the variational distribution is summarized in Alg. 1. We explain each step in the following sections.

1) Pre-processing the Input Prior over HMMs: In our algorithm, the input is the prior of base model, and in (13) the expectation w.r.t. the base model results in a new equivalent base model. In detail, for the expected log-likelihood, we have
\[
\mathbb{E}_{\mathcal{M}^{(b)}} \mathbb{E}_{\gamma|\mathcal{M}^{(b)}} \log p(y|\mathcal{M}^{(r)}) \geq \mathcal{L}^{ij}_{HMM},
\]
where
\[
\mathcal{L}^{ij}_{HMM} = \sum_{\beta} \mathbb{E}_{\pi^{(b)}} \mathbb{E}_{\theta^{(b)}} \mathbb{E}_{y|\theta^{(b)}} \mathbb{E}_{\pi^{(b)}} \log \mathcal{N}(y|\Theta^{(r)}).
\]

2) Variational E-Step: In the variational E-step, we derive the optimal variational distribution of the assignment variables \(Z\) (i.e., calculate the responsibilities). The conditional distribution of \(Z\), given the mixing coefficients \(\omega^{(r)}\), is
\[
p(Z|\omega^{(r)}) = \prod_{i,j} \omega^{(r)}_{ij} \tilde{z}_{ij}.
\]

Making use of the result (10), the optimized \(q(Z)\) is
\[
\log q^{*}(Z) \propto \mathbb{E}_{\omega^{(r)}, \{\mathcal{M}^{(r)}\}} \mathbb{E}_{Y} \log p(Y, Z, \omega^{(r)}, \{\mathcal{M}^{(r)}\}).
\]

After normalization, we have (see App. D for derivations):
\[
q^{*}(Z) = \prod_{i,j} \tilde{z}_{ij},
\]

where
\[
\tilde{z}_{ij} = \mathbb{E}[\tilde{z}_{ij}] = \frac{\tilde{z}_{ij}^{\omega^{(r)}} \tilde{N}_{i} \exp(\tilde{N}_{i} \tilde{z}_{ij} \mathcal{L}^{ij}_{HMM})}{\sum_{j} \tilde{N}_{i} \exp(\tilde{N}_{i} \tilde{z}_{ij} \mathcal{L}^{ij}_{HMM})},
\]

and \(\tilde{z}_{ij}\) are the responsibilities. Note that the optimal solution \(q^{*}(Z)\) depends on moments evaluated w.r.t the distributions of other variables, and thus the variational update equations are coupled and must be solved iteratively.

3) Variational M-step: In the variational M-step, we update the variational parameters \(\alpha^{(r)}, \eta^{(r)}, \lambda^{(r)}, \mu^{(r)}, \Lambda^{(r)}, \tilde{\lambda}^{(r)}, \tilde{\Lambda}^{(r)}, \omega^{(r)}, \tilde{\omega}^{(r)}\) for each component (i.e., compute the optimized distributions of parameters \(\omega^{(r)}, \pi^{(r)}, \Lambda^{(r)},\)
Algorithm 2 VBHEM-H3M

Input: hyperparameter sets \( \mathcal{P}(b) \) and \( \mathcal{P}(r) \), the number of virtual samples \( N \), candidates for number of clusters and states \( K(r) \in [K_{\text{min}}, K_{\text{max}}] \) and \( S(r) \in [S_{\text{min}}, S_{\text{max}}] \).

Output: reduced H3M \( \mathcal{M} = \{ \omega_j, M_j \}_{j=1}^K \).

1: for each pair \((K(r), S(r))\) do
2: repeat
3: Run Alg.1 and obtain \( \mathbb{E}_Y \mathcal{L}(K(r), S(r)) \langle q^* \rangle \).
4: Update hyperparameters \( \alpha_0(r), \eta_0(r), \epsilon_0(r), m_0(r), W_0(r), \gamma_0(r), \nu_0(r) \) using gradient ascent on \( \mathbb{E}_Y \mathcal{L}(K(r), S(r)) \langle q^* \rangle \).
5: until convergence
6: end for
7: Select the reduced H3M \( \mathcal{M}(r) = \{ \omega_j, M_j \}_{j=1}^{K(r)^*} \) with maximum log \( q^*(K(r), S(r)) \).
8: Prune out component \( j \) with low weight \( \mathbb{E}[\omega_j] \), and state \( \rho \) with low probability \( \mathbb{E}[\rho_{j,\rho}] \).

Finally, using the product rule, the variational posterior distribution \( q(\mu(r), \Lambda(r)) \) can be written as

\[
q(\mu(r), \Lambda(r)) = q(\mu(r), \Lambda(r)| q(\mu_0(r), \Lambda_0(r))) = N(\mu(r)| \mu_0(r), \Lambda_0(r)^{-1})W(\Lambda(r)| \mu_0(r), \Lambda_0(r)),
\]

where, \( \mu_0(r), \Lambda_0(r) \) are Gaussian-Wishart distribution, where we have defined \( \gamma_0(r) = \gamma_0^2 + N_\rho^2, \nu_0(r) = N_\rho^2 + N_{\rho_0}^2 + 1 \).

The sufficient synthetic statistics in (25) are defined as:

\[
\begin{align*}
\mu_{j,\rho} = & \frac{1}{\gamma_0(r)} \sum_\beta \hat{y}_j \hat{y}_j^T, \\
\nu_{j,\rho} = & \sum_\beta \hat{y}_j^T, \\
S_{j,\rho} = & \frac{1}{\gamma_0(r)} \sum_\beta \hat{y}_j \hat{y}_j^T, \\
C_{j,\rho} = & \frac{1}{\gamma_0(r)} \sum_\beta \hat{y}_j \hat{y}_j^T.
\end{align*}
\]

where \( \hat{y}_j \) has the same form as that in VHEM [25], and \( N_{\rho_0}^2 \) is the expected number of samples that have been assigned to \( M_j(r) \) with state \( \rho \) during the whole time.

For (25), as more samples are assigned to \( M_j(r) \) with state \( \rho \) (i.e., \( N_{\rho_0}^2 \) increases), the \( \nu_{j,\rho} \) will increase and the covariance of posterior of \( \mu_{j,\rho} \) will decrease; At the same time, the degree of freedom \( \nu_{j,\rho} \) will increase, which leads to increasing precision of the posterior of \( \mu_{j,\rho} \). The update equation of \( m_{j,\rho} \) is a mix between the prior and the soft sample mean \( \hat{y}_j \). Similarly, the update for \( W_{j,\rho} \) is the mix between the prior and the soft sample covariance \( S_{j,\rho} \) and mean base covariance \( C_{\rho} \). Thus the optimization of the variational posterior involves cycling between two stages analogous to the E and M steps of EM algorithm (see Alg. 1).

D. Comparison with VHEM-H3M

We compare our VBHEM-H3M algorithm with the VHEM-H3M algorithm of [25] in this section. Firstly, our method VHEM uses a Bayesian framework, compared to VHEM, which is not Bayesian. We provide priors for all the unknown parameters in the reduced model \( M(r) \) and estimate an approximate posterior distribution \( \hat{M}(r) \), rather than a point-estimate for each parameter as in VHEM.

Secondly, in addition to the prior on the reduced model \( M(r) \), which is common for traditional Bayesian inference, we also assume priors on the input base model \( M(b) \), while VHEM uses the point-estimate of the parameters of the input HMMs. When we assume a delta function prior for \( M(b) \), then VBHEM is a Bayesian version of VHEM, where the point-estimate base models are inputs and Bayesian priors are placed on the reduced models. In this case, equality holds in (15) and...
 Thus, the model selection can be done by: 

\[ c_{j}^{(b),i} = 1, \] and thus the expectations in (17 - 20) yield back the point-estimates.

Thirdly, a coefficient \( c_{j}^{(b),i} \) is introduced in VBHEM in (20). It is worth emphasizing that \( c_{j}^{(b),i} > 1 \), which makes \( C_{j}^{(b)} \) in (26) “larger” (in the positive definite sense), i.e., variance \( (A_{j}^{(r),i})^{-1} \sim (W_{j}^{(r),i})^{-1} \), increases, compared to VHEM where \( c_{j}^{(b),i} = 1 \). Note that this will help to mitigate the variance that variation inference generally underestimates the variance of the posterior density [35].

Fourthly, VBHEM enlarges (reduces) the responsibilities of the components with large (small) weight. The assignment variable \( \tilde{z}_{ij} \) in VHEM is

\[ \tilde{z}_{ij} = \frac{\omega_{j}^{(r)} \exp (N_{i} t_{H,j}^{M})}{\sum_{j'} \omega_{j'}^{(r)} \exp (N_{i} t_{H,j'}^{M})}, \]

in contrast to (22) for VBHEM. For VBHEM, the power of \( \tilde{\omega}_{j}^{(r)} \) in (22) increases the gap among the weights, e.g., the ratio between the maximum and minimum of \( \tilde{\omega}_{j}^{(r)} \), \( b = \frac{\max_{j} \tilde{\omega}_{j}^{(r)}}{\min_{j} \tilde{\omega}_{j}^{(r)}} > 1 \), is smaller than \( b^{N_{i}} \), thus the probability of component with largest \( \tilde{\omega}_{j}^{(r)} \) will increase, and vice versa for the smallest \( \tilde{\omega}_{j}^{(r)} \).

Fifthly, as discussed in the next section, VBHEM can simultaneously perform model selection, while VHEM cannot.

### E. Optimizing the Hyperparameters

Given a pair of \((K^{(r)}, S^{(r)})\), our model contains hyperparameters \( \mathcal{P}^{(r)} = \{ q_{0}^{(r)}, \eta_{0}^{(r)}, \epsilon_{0}^{(r)}, m_{0}^{(r)}, W_{0}^{(r)}, \gamma_{0}^{(r)}, \nu_{0}^{(r)} \} \) (now assumed \( \lambda_{0} \) is known and determined according to data). One approach for estimating the hyperparameters is to maximize the marginal log-likelihood of the data (i.e., empirical Bayes, type-II maximum likelihood), or a lower bound when the marginal log-likelihood is intractable. Applying this to our model, we maximize the expected lower bound \( \hat{\mathcal{L}}(q) \), and proceed by firstly maximizing \( \mathbb{E}_{\mathcal{Y}}[\mathcal{L}(K^{(r)}, S^{(r)})(q^{*})] \) under a fixed model structure \( (K^{(r)}, S^{(r)}) \) (see Appendix E for details of \( \mathbb{E}_{\mathcal{Y}}[\mathcal{L}(K^{(r)}, S^{(r)})(q^{*})] \)). For the continuous parameters (e.g., \( \alpha_{0} \)), \( \mathbb{E}_{\mathcal{Y}}[\mathcal{L}(K^{(r)}, S^{(r)})(q^{*})] \) is maximized using gradient ascent.

For the discrete parameters \( K^{(r)} \) and \( S^{(r)} \), we train on a range of possible \( K^{(r)} \) and \( S^{(r)} \), and select the pair that yields the highest \( \hat{\mathcal{L}}(q) \). Recall from (11), we have

\[
\log q^{*}(K^{(r)}, S^{(r)}) \propto \log p(K^{(r)}) + \log p(S^{(r)}|K^{(r)}) + \mathbb{E}_{\mathcal{Y}}[\mathcal{L}(K^{(r)}, S^{(r)})(q^{*})].
\]

Thus, the model selection can be done by:

1) For each candidate \( K^{(r)} \), select the optimal \( S^{(r)*}(K^{(r)}) = \arg \max_{S^{(r)}} \log p(S^{(r)}|K^{(r)}) + \mathbb{E}_{\mathcal{Y}}[\mathcal{L}(K^{(r)}, S^{(r)})(q^{*})] \)

(27)

2) Select the optimal \( K^{(r)*} = \arg \max_{K^{(r)}} q^{*}(K^{(r)}, S^{(r)*}(K^{(r)})). \) (28)

The entire VBHEM algorithm is summarized in Alg. 2.

### V. Experiments

We present experiment results on synthetic data and real data to demonstrate that our proposed VBHEM-H3M can be effectively applied in several domains. The experiments on the synthetic data show the performance of VBHEM in various aspects, including estimating the parameters, automatically choosing the number of clusters \( K \) and the number of states \( S \), sensitivity analysis of VBHEM to various parameter settings, and the number of clusters \( K \). In the experiments, we remove components/statistics with weights lower than \( 10^{-3} \) (for all the cluster centers from all the compared methods). Finally, we compare our VBHEM-H3M using marginal likelihood (denoted as VBHEM), with VBHEM using DIC [54] (denoted as DIC), VHEM [25] with AIC [31] and BIC [32] (denoted as V-H-AIC and V-H-BIC), PP-K-SC [27] with AIC and BIC (denoted as SC+AIC, and SC+BIC) and CCFD algorithm [47] (see Appendix G-A for details of each compared method).

#### A. Synthetic Data

1) **Experiment 1**: We first consider a 2-dimensional case of a deceptively simple “toy” problem, which is also considered in [23, 55]. The ground truth is a 2D H3M with 2 components (HMMs), and each with 2 states. The transition matrices are

\[
A^{(1)} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}, \quad A^{(2)} = \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}.
\]

(29)

2) **Experiment 2**: The two HMMs share the same 2D Gaussian emission densities \( \mathcal{N}(\mu^{(s)}, \Sigma^{(s)}) \), \( s=1, 2 \), \( i=1, 2 \), where means \( \mu^{(1)} = [0, 3]^{T} \), and \( \mu^{(2)} = [3, 0]^{T} \), and variances \( \Sigma^{(1)} = \Sigma^{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \).

The weights and initial probabilities are all uniform.

The synthetic experiment proceeds as follows: (1) generate 20 sample sets from each HMM, where each sample set contains 25 sequences with length \( \tau = 50 \) [56]; (2) add noise \( e \sim \mathcal{N}(0, 0.1 \cdot I_{2}) \) to each observation; (3) estimate the posteriors over HMM parameters for each noisy sample set via the EMHMM toolbox [4]4, resulting in 20 HMMs

\[ \text{1) source code is released at https://doi.org/10.5281/zenodo.4468501} \]

\[ \text{2) http://mocap.cs.cmu.edu/} \]

\[ \text{3) https://archive.ics.uci.edu/ml/datasets/Character+Trajectories} \]

\[ \text{4) available at http://visal.cs.cityu.hk/research/emhmm/} \]
TABLE III

<table>
<thead>
<tr>
<th></th>
<th>Ri ↑</th>
<th>Purity ↑</th>
<th>Acc % ↑</th>
<th>Over-est % ↓</th>
<th>Under-est % ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>K</td>
<td>S</td>
<td>K</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>VHEM</td>
<td></td>
<td></td>
<td>65.9(0.02)</td>
<td>20.0(0.00)</td>
<td>0.06(0.01)</td>
</tr>
<tr>
<td>VH+AIC</td>
<td>1.000(0.00)</td>
<td>1.000(0.00)</td>
<td>100.0(0.00)</td>
<td>90.5(0.29)</td>
<td>0.00(0.00)</td>
</tr>
<tr>
<td>VH+BIC</td>
<td>1.000(0.00)</td>
<td>1.000(0.00)</td>
<td>100.0(0.00)</td>
<td>98.0(0.14)</td>
<td>0.00(0.00)</td>
</tr>
<tr>
<td>SC+AIC</td>
<td>0.508(10)</td>
<td>0.520(10)</td>
<td>4.0(0.20)</td>
<td>100.0(0.00)</td>
<td>0.00(0.00)</td>
</tr>
<tr>
<td>SC+BIC</td>
<td>0.487(0.00)</td>
<td>0.500(0.00)</td>
<td>0.00(0.00)</td>
<td>100.0(0.00)</td>
<td>0.00(0.00)</td>
</tr>
<tr>
<td>DIC</td>
<td>0.999(0.02)</td>
<td>1.000(0.00)</td>
<td>99.5(0.07)</td>
<td>99.5(0.07)</td>
<td>0.5(0.07)</td>
</tr>
<tr>
<td>CCFD</td>
<td>0.990(0.10)</td>
<td>0.990(0.10)</td>
<td>99.0(0.10)</td>
<td>-</td>
<td>0.00(0.00)</td>
</tr>
<tr>
<td>VHEM (ours)</td>
<td>1.000(0.00)</td>
<td>1.000(0.00)</td>
<td>100.0(0.00)</td>
<td>100.0(0.00)</td>
<td>0.00(0.00)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(ρ(1), ρ(2) → ∞)</td>
<td>0.487(0.00)</td>
<td>0.500(0.00)</td>
</tr>
</tbody>
</table>

TABLE IV

<table>
<thead>
<tr>
<th></th>
<th>(K, S)</th>
<th>Ri ↑</th>
<th>Purity ↑</th>
<th>Acc % ↑</th>
<th>K</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>VHEM</td>
<td>(3,3)</td>
<td>0.999(0.00)</td>
<td>0.899(0.00)</td>
<td>79.5(0.01)</td>
<td>17.4(0.03)</td>
<td></td>
</tr>
<tr>
<td>VH+AIC</td>
<td>(3,5)</td>
<td>0.897(0.01)</td>
<td>0.899(0.00)</td>
<td>51.5(0.20)</td>
<td>14.5(0.02)</td>
<td></td>
</tr>
<tr>
<td>VH+BIC</td>
<td>(5,3)</td>
<td>0.846(0.01)</td>
<td>0.784(0.01)</td>
<td>27.0(0.14)</td>
<td>13.2(0.01)</td>
<td></td>
</tr>
<tr>
<td>SC+AIC</td>
<td>(3,3)</td>
<td>0.977(0.07)</td>
<td>0.967(0.10)</td>
<td>90.5(0.29)</td>
<td>81.7(0.37)</td>
<td></td>
</tr>
<tr>
<td>SC+BIC</td>
<td>(3,5)</td>
<td>0.964(0.03)</td>
<td>1.000(0.00)</td>
<td>30.2(0.46)</td>
<td>41.3(0.30)</td>
<td></td>
</tr>
<tr>
<td>DIC</td>
<td>(5,3)</td>
<td>0.976(0.02)</td>
<td>0.999(0.00)</td>
<td>15.9(0.37)</td>
<td>56.9(0.41)</td>
<td></td>
</tr>
<tr>
<td>CCFD</td>
<td>(3,5)</td>
<td>0.976(0.02)</td>
<td>0.998(0.01)</td>
<td>35.4(0.48)</td>
<td>58.2(0.31)</td>
<td></td>
</tr>
<tr>
<td>VHEM (ours)</td>
<td>(3,5)</td>
<td>0.970(0.03)</td>
<td>0.983(0.06)</td>
<td>15.9(0.37)</td>
<td>62.6(0.39)</td>
<td></td>
</tr>
</tbody>
</table>

*Fig. 3. The decision graph in CCFD algorithm for (a) Synthetic data (K = 2), (b) MoCap data set (K = 10), (c) Hand-writing data set (K = 10), and (d) Music data set (K = 32).*

III. Our V佰HEM-H3M is the most consistent (100% accuracy) in selecting the correct number of components K = 2 and number of states S = 2, as compared to the other methods. Furthermore, V佰HEM obtains perfect Rand-index (Ri) of 1 and perfect Purity of 1, and thus finds the correct clustering consistently. For VHEM, note that we also pruned out the components and states with no samples assigned. V佰HEM has 65.9% accuracy in selecting the true number of components K. V佰HEM overestimates the number of states S in 60.0% of the trials, i.e., the maximum-likelihood-based V佰HEM is likely to over-fit the emission model. The over-fitting problem is mitigated using Bayesian methods, i.e., V佰HEM, or by adding complexity penalization terms, such as AIC and BIC. Using complexity terms, V佰HEM and VH+BIC are slightly less accurate than V佰HEM in estimating S.

Spectral clustering (SC+AIC, SC+BIC) performed worst in selecting K, resulting in lower Rand-index and Purity, but was perfect at selecting S. Since SC clusters the HMMs learned with different states numbers, it can perfectly select S possibly because the input HMMs under that S are learned well. Thus, SC+AIC and SC+BIC cannot obtain the true K are because the AIC penalty is too heavy for SC to select K, and the BIC penalty is even worse. DIC performs well at selecting the number of components and states, but slightly worse than V佰HEM. Note that DIC takes advantage of the good posterior estimate of the reduced model from V佰HEM. CCFD also works well when selecting K (99% accuracy), and shows good clustering performance with Rand-index 0.990 and Purity 0.990. The decision graph shows CCFD can successfully find two cluster centers (see Fig. 3a).
To show the effectiveness of the prior distributions on the input, we also train VBHEM where the priors of the mean and precision matrices are collapsed into delta function priors by setting \( \gamma \to \infty \) and \( \nu \to 0 \). This is equivalent to using point-wise estimates of the mean and precision. The results are in the last row of Table III, and always underestimate \( K \) and \( S \), i.e., the HMMs could not be separated. Thus, using prior distributions on the input HMMs can better handle the uncertainty and leads to better clustering results.

Fig. 2a shows the model selection criteria for varying \( K^{(r)} \) and fixed \( S^{(r)} \), while Fig. 2b shows the criteria for fixed \( K^{(r)} \) and varying \( S^{(r)} \). In Fig. 2a, VH+AIC, VH+BIC, DIC and VBHEM all successfully find the true number of components, while SC+AIC and SC+BIC underestimate the number of components. In Fig. 2b, the six methods all successfully select the true number of states. Finally, Appendix G-B shows an example result.

b) Experiment 2: We next test the robustness of VBHEM for different settings of true number of clusters and true number of states. We generate the synthetic data for different \( K \) and \( S \) states; (2) use \textit{EMHMM toolbox} [4] to learn 20 HMMs \( \{K^{(b)} = 20 \times K\} \) for each given true HMM. Here we test three pairs of \( (K,S) \in \{(3,3),(3,5),(5,3)\} \). For each \( (K,S) \) pair, we run VBHEM with \( K^{(r)} \in [1,10] \) and \( S^{(r)} \in [1,10] \) for 1000 times with different random initializations. For comparison we also run VHEM, PPK-SC, and CCFD for 1000 trials. Note that the number of states can be different for each HMM component in the reduced H3M \( \mathcal{M}^{(r)} \); e.g., for the selected \( S^{(r)} = 5 \), the final number of states can be less than 5, since some states may be pruned out in Step 8 of Alg. 2.

Table IV shows the average performance over 1000 trials, showing that our VBHEM-H3M method outperforms other methods. The Rand-index, Purity and Accuracy do not change significantly with different \( (K,S) \), and thus VBHEM is robust to changes in the number of clusters/states. DIC performs well for setting \( (3,3) \), but the accuracy of \( S \) decrease sharply as the model complexity in \( S \) increases, and likewise for \( K \). VHEM, VH+AIC and VH+BIC have similar Rand-index and Purity, and perform better when \( (K,S) \) are small, but are not robust when they increase. Likewise SC+AIC and SC+BIC are not robust when \( K \) and \( S \) are increased. CCFD also obtains good Rand-index and Purity, but the accuracy decreases as \( K \) increases, which shows that it is less robust than our method.

\[
\text{TABLE V} \\
\text{EXPERIMENT RESULTS FROM CLUSTERING MoCap DATA SET, AVERAGED OVER 20 TRIALS.} \\
\begin{tabular}{ l c c c c c }
\hline
 & \text{Ri ↑} & \text{Purity↑} & \text{Acc↑} & \text{Over-est↑} & \text{Under-est↓} \\
\hline
VHEM & 0.823(0.00) & 0.534(0.01) & 14(0.01) & 43(0.00) & 43(0.01) \\
VH+AIC & 0.827(0.01) & 0.502(0.04) & 0(0.00) & 100(0.00) & 0(0.00) \\
VH+BIC & 0.827(0.01) & 0.502(0.04) & 0(0.00) & 100(0.00) & 0(0.00) \\
VH/r+AIC & 0.827(0.01) & 0.502(0.04) & 0(0.00) & 100(0.00) & 0(0.00) \\
VH/r+BIC & 0.827(0.01) & 0.502(0.04) & 0(0.00) & 100(0.00) & 0(0.00) \\
SC+AIC & 0.795(0.01) & 0.349(0.03) & 0(0.00) & 100(0.00) & 0(0.00) \\
SC+BIC & 0.794(0.01) & 0.348(0.03) & 0(0.00) & 100(0.00) & 0(0.00) \\
SC/τ+AIC & 0.794(0.01) & 0.347(0.03) & 0(0.00) & 100(0.00) & 0(0.00) \\
SC/τ+BIC & 0.794(0.01) & 0.347(0.03) & 0(0.00) & 100(0.00) & 0(0.00) \\
DIC & 0.993(0.01) & 0.99(0.03) & 0(0.00) & 100(0.00) & 0(0.00) \\
DIC/r & 0.898(0.04) & 0.688(0.08) & 0(0.00) & 100(0.00) & 0(0.00) \\
CCFD & 0.129(0.00) & 0.143(0.00) & 0(0.00) & 100(0.00) & 0(0.00) \\
VBHEM (ours) & 0.994(0.01) & 1.000(0.00) & 90(0.00) & 10(0.00) & 0(0.00) \\
\hline
\end{tabular}
\]

B. Motion Capture Data

This experiment uses Motion Capture data (MoCap), which are time series representing human locomotions and actions. We use 63 motion examples spanning 7 different classes (sit, run, jump, yoga, swim, dance, and baseball). Each example is a sequence of 123-dimensional vectors representing the \((x, y, z)\)-coordinates of 41 body markers tracked spatially through time. For each example, we learn the HMM posteriors over parameters using \textit{EMHMM toolbox}. This HMM summarizes the appearance (Gaussian emission) and dynamics (state prior and transition) of the particular motion sequence it represents. We then use these posteriors as input to our algorithm to find the true number of motion classes. For running VBHEM, we set \( K^{(b)} \in [4, 10] \) and \( S^{(r)} \in [3, 8] \), \( N = 10K^{(b)}, \tau = 10 \), and \( \lambda_0 = 1 \). This experiment is repeated 20 times with different random initialization, and the average results are reported in Table V.

Our VBHEM obtains the true number of motion classes with 90% accuracy and perfect Purity 1, and outperforms other methods. Fig. 2c plots the model selection curve, indicating that VBHEM has a peak at \( K = 7 \), leading to the correct choice of number of clusters. VHEM has 14% accuracy in selecting the correct \( K = 7 \), while overestimating and underestimating \( K \) equally 43% of the time, which is close to random chance. Fig. 2c shows the model selection criteria for the various methods versus \( K \). VH+AIC, VH+BIC, SC+AIC and SC+BIC have a tendency to overestimate \( K \), as their curves always increase as \( K \) increases, resulting in selection of \( K = 10 \). This is because the log-likelihood approximation used by VHEM and PPK-SC increase as the model complexity increases, and the increase cannot be effectively penalized by the AIC or BIC terms. Moreover, it demonstrates that as the data becomes noisier (as in real-world data) and the dimension increases, our method performs better than BIC and AIC. DIC achieves an Accuracy of 40% in determining \( K \), which performs better than other methods, but is still inferior to our VBHEM. Although DIC has a 55% probability of overestimating \( K \), it obtains high Rand-index and Purity, which means DIC is still forming consistent groups of data. CCFD performs the worst in this experiment and always underestimates \( K \). The main reason is that the cut-off \( d_c \) is too small; \( d_c = d_{min} + (d_{max} - d_{min}) \times p \), where \( d_{min} \) and \( d_{max} \) are respectively the minimum and maximum among all distances of two HMMs, and \( p \) is searched in \([1\%, 20\%]\). However, in this experiment \( d_{min} \) is about 560, \( d_{max} \) is about 730, and the most of distances are centered on 700 (see Fig. 3b). Thus \( d_c \) is still small even when \( p = 20\% \) (about 600), and thus most of the densities are \( \rho_1 = 0 \), which causes CCFD to fail to find the cluster centers.
The analysis of the results show that VH+AI C, VH+BI C, SC+AI C, SC+BI C and DIC might have failed because the log-likelihood approximation term is too large compared to the penalty terms. To alleviate this problem, we modify these methods by dividing this term by the length of the sequence \( \tau \). The results are shown in Table V, denoted as VH/\( \tau \)+BI C, VH/\( \tau \)+AI C, SC/\( \tau \)+BI C, SC/\( \tau \)+AI C and DIC/\( \tau \). The clustering results do not change or even get worse, which shows this process does not help.

Finally, Fig. 4 presents the selected \( S \) for each method. VBHEM has a high probability to select \( S = 3 \) or \( S = 6 \) in the reduced H3M. VHEM is not effective at selecting \( S \), and thus the percentages are mostly uniform. When using the model selection methods, VHEM tends to select the largest candidate \( S \), while PPK-SC tends to select the smallest candidate \( S \). DIC is more likely to select \( S = 7 \) or \( S = 8 \), while DIC/\( \tau \) mostly selects 3 states. Among the methods with decent Rand-index, our method selects a more parsimonious model while also having high Rand-index (0.998). While there is no ground-truth value for the number of states, a more parsimonious model may be preferred since it more succinctly summarizes the data, and also is easier to interpret compared to a model with many states.

C. Eye Movement Data

In this experiment, we use the Eye Movement dataset from [5], which is a collection of eye fixation trajectories from 68 participants (34 older adults and 34 young adults) while performing a recognition experiment on face images. An eye movement data is a sequence of 2-dimensional vectors representing the \((x, y)\)-coordinates of the location of an eye fixation on the face image over time. Previous work [59] models the regions of interest (ROIs) as GMMs and study the correlation between GMMs and cognitive abilities. [4, 5, 60] also consider the temporal dynamics, where each participant’s eye movements are modeled with an HMM, including both person-specific ROIs and transitions among the ROIs. Individual HMMs are then clustered using VHEM into two groups [4, 5] or three groups [60] in order to discover common eye gaze strategies among the participants, but requires setting the number of group and number of states by hand.

1) Clustering results: We use VBHEM to automatically choose the number of clusters and states to discover common patterns among individuals. We set \( K(\tau) \in \{1, 5 \} \) and \( S(\tau) \in \{1, 5 \} \), \( N = 10K(\tau) \), and \( \lambda_0 = 1 \) for running VBHEM and VH+BI C. VBHEM automatically selects an H3M with 2 components and 2 states, and the estimated group HMMs are shown in Fig. 5. Fig. 6a plots the model selection curve for VBHEM. The number of selected states is inversely proportional to the number of components. Given \( K = 1 \), then the best selection is \( S = 3 \); increasing \( K \) will decrease the best selection of \( S \). The reason is that when using an H3M to model the given data, more components means less data for each component to model, and thus less hidden states are needed within each component. The pair \((K, S) = (2, 2)\) has the maximum \( \log q^*(K, S) \) among all candidate model structures, and thus is the overall selected model structure. Compared with BIC (see Fig. 6b), \( S = 2 \) is selected when \( K \in \{1, 2, 3 \} \), and \( S = 1 \) is selected when \( K \in \{4, 5 \} \). The final selection is \((5, 1)\); BIC cannot effectively penalize the growth of the log-likelihood caused by increasing \( K \), which leads to the selection of the largest \( K \).

2) Results analysis: The HMMs clustering results are displayed in Fig. 5. The pattern in Fig. 5 (left) resembles a holistic pattern, a scan path typically started at the nose/mouth region, and then staying around the same region. In contrast, the pattern in Fig. 5 (right) resembles an analytic pattern, a scan path typically started around the face center, and then transitioned to the eye region. (i.e., more frequent fixation transitions between the eyes; [4]). The differences can also be seen in the corresponding fixation heatmap shown in Fig. 5. The two representative HMMs significantly differ from each other based on the KL divergence test [4]; using data from holistic HMMs, \( t(38) = 7.10, p < 0.001 \); using data from analytic HMMs, \( t(28) = 6.10, p < 0.001 \).

Table VI shows the number of young and older participants belonging to holistic and analytic pattern, and compare with the VHEM results from [5]. There are 39 adults assigned to the holistic pattern and 29 adults assigned to the analytic pattern. Comparing with the results from [5] using VHEM, the
The main difference is that VBHEM assigns 6 additional young adults into the holistic group. This difference may be due to our analytic HMM focusing more on the regions around the two eyes than the analytic HMM in [5] (our transition matrix shows a larger probability to stay in the eye regions, while the transition matrix in [5] is more uniform). Thus, those participants that do not show significant focus on the eyes are assigned to holistic pattern by our algorithm.

As we may be interested in individual differences in eye movement patterns, we also quantify the degree to which a subject’s eye-movement pattern resembles the representative holistic and analytic HMMs using the H-A scale [5]. For each subject, the H-A scale measures a normalized difference between the log-likelihoods of a subject’s eye movement data being generated by the representative holistic and analytic HMM models,

\[ H-A \text{ Scale} = \frac{\log p(Y|\mathcal{M}_h) - \log p(Y|\mathcal{M}_a)}{\log p(Y|\mathcal{M}_h) + \log p(Y|\mathcal{M}_a)} \]

where \( Y \) is the fixation data for the individual, and \( \mathcal{M}_h \) and \( \mathcal{M}_a \) are the representative holistic and analytic HMMs learned by VBHEM. A positive H-A value indicates that the subject’s gaze pattern is more similar to a holistic pattern, while a negative value indicates similarity to an analytic pattern. According to the H-A values, older adults tended to exhibit holistic patterns (M=0.0089), and younger adults tended to exhibit analytic patterns (M=0.0029), and this difference was statistically significant according to a two-sample t-test, \( t(66)=2.139, p=0.036 \). The same comparison between older/young adults using the H-A scale built using VHEM (i.e. from [5]) yielded a marginal difference (older adults M=0.0032, young adults M=0.0010), \( t(66)=1.916, p=0.060 \). We also performed correlation analysis to see how eye gaze patterns (as quantified by H-A scale) are correlated with the subjects’ behavioral data, as in [5]. The subjects’ recognition accuracy was negatively correlated with H-A scale, \( r(66)=-0.428, p=0.011 \). The lower the recognition accuracy, the more holistic the eye-gaze pattern (see Fig. 7a). In addition, the participants’ recognition performance \( d' \) was also negatively correlated with H-A scale, \( r(66)=-0.430, p=0.011 \) (see Fig 7b). Finally, the MoCA scores\(^5\) for the older adults was negatively correlated with the H-A scale, \( r(32)=-0.437, p=0.009 \). In other words, the lower the MoCA score (the more cognitive impairment), the more holistic the pattern (see Fig.7c). These results from VBHEM are consistent with the previous study [4, 5] - two strategies of eye movements (holistic and analytic patterns) in face recognition tasks are discovered by clustering HMMs, and the H-A scales are negatively correlated with recognition performance and MoCA. However, here we use VBHEM to automatically determine the number of clusters and the number of states, whereas [5] set these values by hand.

D. On-Line Hand-Writing Data Set

In this experiment, we evaluate VBHEM for clustering characters from the Character Trajectories Data Set, which consists of 2858 examples for 20 characters from the same writer. Each example is the trajectory of one character that corresponds to a single pen-down segment. The data was captured using a Wacom tablet at 200Hz, and consists of \((x, y)\)-coordinates and pen tip force, and the data has been numerically differentiated and Gaussian smoothed [61].

In consideration of the aim for VBHEM is to cluster data while automatically choosing the number of clusters. We simplify the experiment by only selecting 10 character types. For each character, we randomly select 25 examples to learn an HMM, and repeat 10 times, resulting in a base H3M with \( K^{(b)} = 100 \) components. We then perform clustering using VBHEM, VHEM, PPK-SC and CCFD. For VBHEM, we set \( K^{(r)} \in [6, 14], S^{(r)} = 6, N = 10K^{(b)}, \tau = 100, \) and \( \lambda_0 = 1. \) The experiment is repeated 10 times with different initializations, and the average results are in Table VII.

VBHEM and DIC obtain the same best performance among the compared methods. Indeed the model selection curves for VBHEM and DIC (Fig. 2d) are coincident. Comparatively, our method of model selection is more straightforward than DIC, which requires further approximation on top of the VB framework.

\(^5\)Montreal Cognitive Assessment (MoCA) is a valid brief assessment tool for screening of people with mild cognitive impairment, and 22 points or more (out of 30) is a normal score.
TABLE VIII

**Experiment Results from Clustering Music Genre Dataset, Averaged over 10 Trials.**

<table>
<thead>
<tr>
<th>Method</th>
<th>$\tau$</th>
<th>Purity ↑</th>
<th>Over-est ↓</th>
<th>Under-est ↓</th>
<th>K %</th>
</tr>
</thead>
<tbody>
<tr>
<td>VH/AIC</td>
<td>0.641(0.02)</td>
<td>0.435(0.03)</td>
<td>100.00</td>
<td>50.00</td>
<td>40.00</td>
</tr>
<tr>
<td>VH+BIC</td>
<td>0.735(0.03)</td>
<td>0.502(0.05)</td>
<td>100.00</td>
<td>50.00</td>
<td>0.00</td>
</tr>
<tr>
<td>VH/τ+AIC</td>
<td>0.735(0.03)</td>
<td>0.502(0.05)</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>VH/τ+BIC</td>
<td>0.827(0.01)</td>
<td>0.502(0.04)</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SC/τ+AIC</td>
<td>0.742(0.01)</td>
<td>0.424(0.04)</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SC/τ+BIC</td>
<td>0.679(0.08)</td>
<td>0.384(0.06)</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SC/τ+AIC</td>
<td>0.432(0.12)</td>
<td>0.260(0.04)</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SC/τ+BIC</td>
<td>0.470(0.10)</td>
<td>0.276(0.04)</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>DIC</td>
<td>0.757(0.02)</td>
<td>0.554(0.05)</td>
<td>20.42</td>
<td>60.52</td>
<td>20.42</td>
</tr>
<tr>
<td>DIC/τ</td>
<td>0.197(0.00)</td>
<td>0.200(0.00)</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
</tr>
<tr>
<td>CCFD</td>
<td>0.427(0.30)</td>
<td>0.385(0.24)</td>
<td>0.00</td>
<td>40.52</td>
<td>60.52</td>
</tr>
<tr>
<td>VBHEM (ours)</td>
<td>0.870(0.03)</td>
<td>0.591(0.07)</td>
<td>60.52</td>
<td>10.32</td>
<td>0.00</td>
</tr>
</tbody>
</table>

VH/AIC and VH+BIC always overestimate the number of clusters. VH/τ+AIC and VH/τ+BIC only slightly increases the Rand-index. Although SC/τ+AIC selects the true number of clusters in one trial, the 70% probabilities of underestimating $K$ still leading to worse Rand-index and Purity than SC+AIC. Moreover, as the penalty increase, from SC+AIC to SC/τ+BIC, the performances get worse, which implies that the penalty is not appropriate. DIC has slightly lower Purity than our method and normalizing with the length of sequence also does not help. CCFD could not find the true number of genres $K$. In 5 trials, $K$ was overestimated, e.g., the decision graph of one trial where CCFD selected $K = 32$ is in Fig. 3d. In the remaining 6 trials, CCFD failed to separate the music HMMs and formed only one cluster. Fig. 2e shows the model selection curve for VBHEM, VH+BIC, SC+BIC and DIC. VBHEM and DIC have peaks at $K = 5$, while VH+BIC and SC+BIC overestimate the number of clusters.

VI. CONCLUSIONS

We have derived the VBHEM algorithm for clustering HMMs, which automatically determines the number of clusters and the number of states. We show the efficacy of VBHEM on both synthetic dataset and real-world datasets, including motion capture, eye fixation sequences, character trajectories, and music. For the synthetic datasets considered, our VBHEM recovers the correct number of components/states in the H3M model, and finds good posterior estimates of the component HMMs. For the real datasets, we obtained results for clustering and model selection that are better or comparable to other methods. For future work, we now use the same value of $S_j^\tau$ for all the reduced HMMs, and we can consider using different values of $S_j^\tau$ for each component $j$, but this requires a more efficient search process over $K$ and $S$ to make it scalable.

ACKNOWLEDGMENT

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (GRF 17609117), and by a Strategic Research Grant from City University of Hong Kong (Project No. 7005218).

REFERENCES


