Journal of Machine Learning Research 18 (2017) 1-49 Submitted 7/16; Revised 9/17; Published 11/17

Clustering with Hidden Markov Model on Variable Blocks

Lin Lin LLIN@PSU.EDU

Jia Li JIALI@PSU.EDU

Department of Statistics
Pennsylvanian State University
University Park, PA 16802, USA

Editor: Saharon Rosset

Abstract

Large-scale data containing multiple important rare clusters, even at moderately high dimensions, pose challenges for existing clustering methods. To address this issue, we propose a new mixture model called Hidden Markov Model on Variable Blocks (HMM-VB) and a new mode search algorithm called Modal Baum-Welch (MBW) for mode-association clustering. HMM-VB leverages prior information about chain-like dependence among groups of variables to achieve the effect of dimension reduction. In case such a dependence structure is unknown or assumed merely for the sake of parsimonious modeling, we develop a recursive search algorithm based on BIC to optimize the formation of ordered variable blocks. The MBW algorithm ensures the feasibility of clustering via mode association, achieving linear complexity in terms of the number of variable blocks despite the exponentially growing number of possible state sequences in HMM-VB. In addition, we provide theoretical investigations about the identifiability of HMM-VB as well as the consistency of our approach to search for the block partition of variables in a special case. Experiments on simulated and real data show that our proposed method outperforms other widely used methods.

Keywords: Gaussian mixture model, hidden Markov model, modal Baum-Welch algorithm, modal clustering

1. Introduction

Clustering is one of the most important topics in unsupervised learning, the goal of which is to discover structures from a collection of unlabeled data. Finite mixture modeling is a major statistical framework for clustering. Without attempting to review its expansive applications, we offer instead a few examples as a glimpse at the incredibly broad usage of mixture modeling (Escobar and West (1995); Allenby et al. (1998); McLachlan et al. (2002); Kasahara and Shimotsu (2009)).

One important advantage of the mixture model is that the goodness of fit to any data can be improved by increasing the number of mixture components. The simplest approach to clustering based on a mixture model is to assign each component to an individual cluster. However, there are several drawbacks to equate clusters with mixture components, among which is that the parametric distribution of a component is too restrictive for the potentially diverse shapes of clusters (see Li et al. (2007) for a thorough discussion). Various strategies have been proposed to merge multiple mixture components so that an individual cluster can be more properly modeled (Hennig, 2010; Li, 2005; Pyne et al., 2009; Finak et al.,

2009; Chan et al., 2010; Aghaeepour et al., 2011; Lin et al., 2016; Melnykov, 2016). In the statistical learning literature, a prominent method for merging multiple mixture components into one cluster is based on the modes of the mixture density, the so-called modal clustering by Li et al. (2007). We discuss this method in more detail in Section 2.

Despite their wide applications, existing mixture modeling approaches are severely challenged by high dimensional data encountered in certain research areas, for example, cell subset identification using data generated by the high-throughput single-cell technologies (Perfetto et al., 2004; Bandura et al., 2009; Maecker et al., 2012; Chattopadhyay et al., 2014; Spitzer and Nolan, 2016). These data sets contain a large number of highly unbalanced clusters. Furthermore, the most interesting clusters for scientific investigation are often of remarkably low occurrence. Even when the data dimension is not impressively large by today's standard, say in the order of tens, existing methods have much difficulty for detecting clusters of very low probabilities. Low probability mixture components tend to be "concealed" by large background clusters in the data. For the usual mixture modeling approach, in order to capture the rare clusters, we must increase the number of components dramatically. On the other hand, the curse of dimensionality prevents the use of many components; the growing computational intensity is also a concern.

Our new method is motivated by the popular manual gating analysis of single-cell cytometry data (details in Section 5.2), in which the variables are divided into groups based on prior information and examined sequentially. The key idea here is to exploit the chain-like dependence among groups of variables in the construction of a mixture model. Lin et al. (2013) used this idea to build a relatively primitive model. In this approach, the variables are partitioned into two groups, and the mixture model is estimated using the hierarchical Dirichlet process prior. This two-block model is substantially more efficient and accurate in rare clusters identification than the conventional mixture models are. The existing method, however, is unable to move beyond two variable blocks in practice due to the exponential computational complexity. The Markov chain Monte Carlo (MCMC) simulation for the two-block model is already highly intensive and is coupled by the long-standing issue of label switching when MCMC is applied to estimate mixture models (e.g., Richardson and Green (1997); Celeux et al. (2000); Stephens (2000)).

The aim of this paper is to design an effective and computationally accessible statistical model that can fit data robustly in both high and low probability regions and can identify clusters of non-Gaussian shapes. We propose a new model to exploit sequential dependence among variable groups. We also develop an algorithm to search for such a dependence structure when it is unknown. Our experiments show that even if the sequential dependence is not backed up by domain knowledge, it can still be useful as a mathematical mechanism for parsimonious modeling.

Our major contributions include: (1) We develop a *hidden Markov model on variable blocks* (*HMM-VB*) to leverage the sequential dependence structure among variable groups. To the best of our knowledge, this work is the first to exploit sequential dependence among variable groups by HMM. (2) We derive the Baum-Welch estimation algorithm for HMM-VB and the new *Modal Baum-Welch (MBW)* algorithm for finding modes of a HMM-VB, based on which clusters are formed. MBW achieves linear computational complexity without compromising optimality. (3) We develop a search algorithm to determine the grouping and ordering of the variables for a HMM-VB when variable groups are not pre-specified. (4) We

develop theorems on the identifiability of HMM-VB given the variable block structure and the identifiability of the variable blocks under certain conditions. We prove the consistency of the BIC criterion for finding the variable blocks in a special case.

The rest of the paper is organized as follows. In Section 2, we introduce notations and overview existing techniques most relevant to our proposed methods. In Section 3, we present HMM-VB and efficient algorithms for model fitting and modal clustering. In Section 4, we provide theoretical results on identifiability, consistency, and the mode search algorithm for HMM-VB. Proofs of the theorems appear in Appendix B∼D. In Section 5, experimental results are reported for both simulated and real data including mass cytometry, single-cell genomics, and image data. Comparisons are made with some competing models and popular methods. We conclude with discussions in Section 6.

2. Preliminaries

Given a random vector $X = (X_1, X_2, ..., X_d)' \in \mathcal{R}^d$, let $\mathbf{x}_i = (x_{i1}, ..., x_{id})' \in \mathcal{R}^d$ be the i-th sample of X, where $i = 1, ..., n$. Denote by $\mathbb{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) \in \mathcal{R}^{n\times d}$ the data matrix, and we let $\mathbb{X}_j$ be the j-th column of $\mathbb{X}$ which contains the values of X_j across all the sample points. For ease of notation, we let $\mathbf{x} \in \mathcal{R}^d$ be a realization of the random vector X.

The finite Gaussian mixture model (GMM) is commonly used for clustering. A GMM with M components has the density function: $f(\mathbf{x}|\theta) = \sum_{k=1}^{M} \pi_k \phi(\mathbf{x}|\theta_k)$, where π_k is the mixture component prior probability and $\phi(\cdot \mid \theta_k)$ is the multivariate normal density parameterized by $\theta_k = (\mu_k, \Sigma_k)$, μ_k being the d-dimensional mean vector and Σ_k the $d \times d$ covariance matrix. For model estimation, a latent indicator $Z \in \{1, 2, ..., M\}$ with $P(Z = k) = \pi_k$ is used. Specifically, conditioning on $Z = k$, X follows the k-th component distribution. Z is also called the component identity of X. To perform clustering, the usual approach is to compute the posterior probability $P(Z = k|X = \mathbf{x})$ and assign $\mathbf{x}$ to the cluster with the maximum posterior. However, this approach is inadequate to model clusters with arbitrary shapes and cannot ensure that the clusters are reasonably separated. One major idea explored in the literature is to merge multiple mixture components for a better and more flexible representation of an individual cluster. We refer to Melnykov and Maitra (2010) for a thorough review on clustering based on finite mixture models.

Banfield and Raftery (1993) and Celeux and Govaert (1995) propose to decompose the covariance matrix of the mixture components in a GMM into parts that control volume, shape, and orientation respectively and allow model constraints to be imposed in these aspects either individually or by combination. A model selection criterion is then used to choose a GMM in terms of not only the number of components but also the constraints on covariance matrices. A popular R package, namely `Mclust` (Fraley and Raftery, 2006), is implemented for this method, which we will use for comparison.

The Modal EM (MEM) algorithm developed by Li et al. (2007) performs efficient merging of mixture components. It resembles the expectation-maximization (EM) algorithm (Dempster et al., 1977), as reflected by the name "modal EM". However, the objective of MEM is to find an increasing path from any data point to a local maximum of a given density. Hence, the optimization objective of MEM is to find a local maximizer over $\mathbf{x}$ for $f(\mathbf{x}|\theta)$ under a given θ, while EM is to find local maxima over θ for $f(\mathbf{x}|\theta)$ given $\mathbf{x}$. Consider a general mixture density $f(\mathbf{x}) = \sum_{k=1}^{M} \pi_k f_k(\mathbf{x})$, where $f_k(\mathbf{x})$ is the density of the k-th mixture component.

Starting from any initial value denoted by $\mathbf{x}^{[0]}$, MEM solves a local maximum of the mixture density by the following two iterative steps: (1) At iteration r, let $p_k = \frac{\pi_k f_k(\mathbf{x}^{[r]})}{f(\mathbf{x}^{[r]})}$, $k = 1$, ..., M; (2) Update $\mathbf{x}^{[r+1]} = \text{argmax}_{\mathbf{x}} \sum_{k=1}^{M} p_k \log f_k(\mathbf{x})$. MEM stops when a pre-specified stopping criterion is met. Specifically for GMM with $f(\mathbf{x}) = \sum_{k=1}^{M} \pi_k \phi(\mathbf{x}|\mu_k, \Sigma_k)$, MEM becomes

1. E-step: Solve

$$p_k = \frac{\pi_k \phi_k(\mathbf{x}^{[r]} \mid \mu_k, \Sigma_k)}{f(\mathbf{x}^{[r]})}, \quad k = 1, \ ..., \ M. \tag{1}$$

2. M-step: Solve

$$\mathbf{x}^{[r+1]} = \left(\sum_{k=1}^{M} p_k \cdot \Sigma_k^{-1} \right)^{-1} \cdot \left(\sum_{k=1}^{M} p_k \cdot \Sigma_k^{-1} \mu_k \right). \tag{2}$$

The computational efficiency of MEM enabled the development of a new clustering approach by Li et al. (2007), referred to as *modal clustering*. In Li et al. (2007), a non-parametric Gaussian kernel density estimate is used, and MEM is applied to find the mode associated with every point. Data points associated with the same mode are assigned to the same cluster. In Lee and Li (2012), modal clustering based on the general finite GMM is studied. For computational efficiency, instead of applying MEM to every data point, it is applied to the means of the mixture components. Components with mean vectors associated with the same mode are merged into one cluster. Whether a point-wise mode association or a component-wise mode association is preferred depends on the nature of the application and the computational resources. In practice, the difference in the clustering results we have observed is quite small. We refer to the clustering method based on component-wise mode association as *modal GMM* and use it as a baseline for comparison with our new method.

Under the framework of modal clustering, the purpose of a mixture component is primarily for good density estimation. We no longer rely on a one-to-one correspondence between mixture components and clusters. Mode association also ensures that different clusters of data are well separated. See Li et al. (2007) for more detailed discussion on these advantages. The flexibility provided by modal clustering for fitting data is precisely what we need for the applications we consider. In the next section, we propose a HMM-type model which can be cast as a mixture model with an enormous number of components, even exceeding the data size. This complexity causes no difficulty in clustering via mode association.

3. Hidden Markov Model on Variable Blocks

As discussed in Section 1, in some specific application domains, a sequential dependence structure among groups of variables is available. This dependence prompts us to model a subset of variables conditioning on some other subset of variables. If we view the sequential ordering of the variable blocks as a "timeline", it seems natural to employ a HMM-type model, where each variable block follows a mixture distribution and the statistical dependence among the blocks is captured by the Markov process of the underlying states (that is, the

mixture component identities). The description of a conventional HMM is provided in Appendix A. We call our new model *Hidden Markov Model on Variable Blocks (HMM-VB)*.

Suppose the d-dimensional random vector X is partitioned into T blocks indexed by $t = 1, 2, ..., T$. Let the number of variables in block t be d_t, where $\sum_{t=1}^{T} d_t = d$. Assume that the d_1 variables in block 1 have indices before the d_2 variables in block 2, and so on. In general, obviously, such an ordering of variables may not hold. But this is only a matter of naming the variables and has no effect on our results. Let $X^{(t)}$ denote the t-th variable block. Without loss of generality, let $X^{(1)} = (X_1, X_2, ..., X_{d_1})'$ and $X^{(t)} = (X_{m_t+1}, X_{m_t+2}, ..., X_{m_t+d_t})'$, where $m_t = \sum_{\tau=1}^{t-1} d_\tau$, for $t = 2, ..., T$.

Denote the underlying state of $X^{(t)}$ as s_t, $t = 1, ..., T$. Let the index set of s_t be $\mathcal{S}_t = \{1, 2, ..., M_t\}$, where M_t is the number of mixture components for variable block $X^{(t)}$, $t = 1, ..., T$. Let the set of all possible sequences be $\hat{\mathcal{S}} = \mathcal{S}_1 \times \mathcal{S}_2 \cdots \times \mathcal{S}_T$. $|\hat{\mathcal{S}}| = \prod_{t=1}^{T} M_t$. We assume:

1. $\{s_1, s_2, ..., s_T\}$ follow a Markov chain. Let $\pi_k = P(s_1 = k)$, $k \in \mathcal{S}_1$. Let the transition probability matrix $A_t = (a_{k,l}^{(t)})$ between s_t and s_{t+1} be defined by $a_{k,l}^{(t)} = P(s_{t+1} = l | s_t = k)$, $k \in \mathcal{S}_t$, $l \in \mathcal{S}_{t+1}$.

2. Given s_t, $X^{(t)}$ is conditionally independent from other $s_{t'}$ and $X^{(t')}$, for all $t' \neq t$. We also assume that given $s_t = k$, the conditional density of $X^{(t)}$ is the Gaussian distribution $\phi(X^{(t)} \mid \mu_k^{(t)}, \Sigma_k^{(t)})$.

Let $\mathbf{s} = \{s_1, ..., s_T\}$. A realization of X is $\mathbf{x}$, and a realization of $X^{(t)}$ is $x^{(t)}$. To summarize, the density of HMM-VB is given by

$$f(\mathbf{x}) = \sum_{\mathbf{s} \in \hat{\mathcal{S}}} \left(\pi_{s_1} \prod_{t=1}^{T-1} a_{s_t, s_{t+1}}^{(t)} \right) \cdot \prod_{t=1}^{T} \phi(x^{(t)} | \mu_{s_t}^{(t)}, \Sigma_{s_t}^{(t)}). \tag{3}$$

Remark 1: Figure 1 illustrates two major differences between HMM-VB and the conventional HMM. (1) The variable blocks $X^{(t)}$'s are not from the same vector space. Hence, the parameters of the distribution of $X^{(t)}$ given $s_t = k$ depend not only on k but also on t; (2) The underlying Markov chain for $\{s_1, ..., s_T\}$ is not time invariant. In fact, the state space $\mathcal{S}_t$ varies with t.

Remark 2: Although the density function of HMM-VB in Eq. (3) indicates block diagonal covariance matrices, there are important differences from a typical GMM with the same constraint on the covariance. First, the Gaussian mean vectors in Eq. (3) reside on a lattice in the Cartesian product space $\mathcal{R}^{d_1} \times \mathcal{R}^{d_2} \cdots \times \mathcal{R}^{d_T}$. Secondly, the number of components grows exponentially with T. In fact, it is often larger than the sample size. The enormous number of components cannot be handled by a typical covariance constrained GMM from either estimation or computational feasibility perspectives.

Remark 3: Since HMMs can be represented as singly connected directed acyclic graphs, HMM-VB is a special case of graphical models (e.g., Bishop (2006); Koller and Friedman (2009)). We named our model HMM-VB because we derived the estimation method based on the popular Baum-Welch algorithm for HMM. Moreover, HMM itself is a widely known model in machine learning as well as statistics.

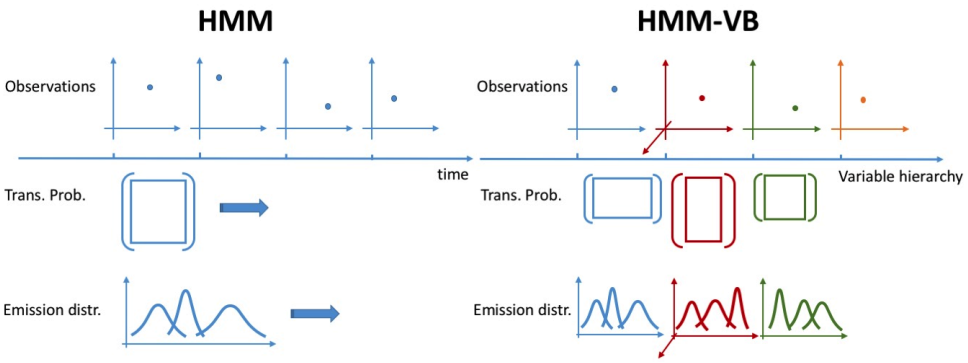


Figure 1: Comparison between HMM-VB and HMM models. The observations under HMM have to be of a fixed dimension, 2 in this illustration. Typically, only one transition probability matrix is applied through time, and the set of distributions conditioned on the states at any time spot is also fixed. HMM-VB models data across blocks of different variables, possibly of different dimensions. Both the transition probability matrix and the set of conditional distributions are defined individually at every "time" spot.

3.1 Maximum Likelihood Estimation

HMM is usually estimated by the EM algorithm. However, because the cardinality of $\bar{\mathcal{S}}$ grows exponentially with the sequence length, the computational complexity of a direct application of EM is of exponential complexity. This technical hurdle was overcome by the *Baum-Welch (BW) algorithm* which achieves complexity of linear order in the sequence length and quadratic in the number of states without compromising optimality. The BW algorithm, a special instance of EM, was developed in the 1960's before the general EM algorithm was developed in the 1970's. As a result, we still call the estimation algorithm Baum-Welch, following the convention of the literature on HMM. We present the BW algorithm for HMM model estimation in Appendix A. For a detailed exposure to HMM, we refer to Young et al. (1997).

We now present the corresponding BW algorithm for HMM-VB. It can be proved that the BW algorithm for HMM is an exact EM algorithm. The derivation of the BW algorithm for HMM-VB is similar to the derivation of BW for the usual HMM. We thus omit it here. Clearly, it is not meaningful to estimate HMM-VB using a single sequence, which is essentially a single data point in this case. HMM-VB is after all a model for $X \in \mathcal{R}^d$.

Denote by $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, ..., x_i^{(T)})'$ the ordered and grouped i-th sample according to the given variable blocks. We denote the original i-th sample before ordering by $\breve{\mathbf{x}}_i$. Consider estimation of HMM-VB based on a data set $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. We define $L_k(\mathbf{x}_i, t)$ and $H_{k,l}(\mathbf{x}_i, t)$ similarly as in Eq. (8) and Eq. (9) in Appendix A. For $i = 1, ..., n$,

$$L_k(\mathbf{x}_i, t) = P(s_{i,t} = k \mid \mathbf{x}_i)\,,\ k \in \mathcal{S}_t, \tag{4}$$

$$H_{k,l}(\mathbf{x}_i, t) = P(s_{i,t} = k, s_{i,t+1} = l \mid \mathbf{x}_i)\,,\ k \in \mathcal{S}_t, l \in \mathcal{S}_{t+1}. \tag{5}$$

The BW algorithm iterates the following two steps:

1. E-step: Under the current set of parameters, compute $L_k(\mathbf{x}_i, t)$, $i = 1, ..., n$, $k \in \mathcal{S}_t$, $t = 1, ..., T$, and $H_{k,l}(\mathbf{x}_i, t)$, $i = 1, ..., n$, $k \in \mathcal{S}_t$, $l \in \mathcal{S}_{t+1}$, $t = 1, ..., T-1$.

2. M-step: Update parameters by

$$\mu_k^{(t)} = \frac{\sum_{i=1}^n L_k(\mathbf{x}_i, t) x_i^{(t)}}{\sum_{i=1}^n L_k(\mathbf{x}_i, t)}, \quad k \in \mathcal{S}_t,\ t = 1, ..., T,$$

$$\Sigma_k^{(t)} = \frac{\sum_{i=1}^n L_k(\mathbf{x}_i, t) \left(x_i^{(t)} - \mu_k^{(t)}\right) \left(x_i^{(t)} - \mu_k^{(t)}\right)'}{\sum_{i=1}^n L_k(\mathbf{x}_i, t)}, \quad k \in \mathcal{S}_t,\ t = 1, ..., T,$$

$$a_{k,l}^{(t)} = \frac{\sum_{i=1}^n H_{k,l}(\mathbf{x}_i, t)}{\sum_{i=1}^n L_k(\mathbf{x}_i, t)}, \quad k \in \mathcal{S}_t,\ l \in \mathcal{S}_{t+1},\ t = 1, ..., T-1,$$

$$\pi_k \propto \sum_{i=1}^n L_k(\mathbf{x}_i, 1),\ k \in \mathcal{S}_1,\ s.t. \sum_{k \in \mathcal{S}_1} \pi_k = 1.$$

The above equations can be easily extended to the case of weighted sample points. It can occur in practice that each sample point is assigned with a weight. For instance, quantization is often used to reduce the data size significantly. Instead of using the original data, one may use the quantized points, each of which can represent a different number of original points and hence is assigned with a weight proportional to that number. Suppose weight w_i is assigned to sample $\mathbf{x}_i$. The E-step is not affected. In the M-step, we can simply multiply w_i in front of each summand appeared in the equations above.

The forward-backward algorithm for computing $L_k(\mathbf{x}_i, t)$ and $H_{k,l}(\mathbf{x}_i, t)$ is essentially the same as the forward-backward algorithm for the usual HMM. The fact that the variable blocks are not from the same vector space and the state spaces vary with t does not cause any intrinsic difference. Suppress the sample index i and consider one sample point $\mathbf{x}$.

Define the forward probability $\alpha_k(\mathbf{x}, t)$ as the joint probability of observing the first t variable blocks $x^{(\tau)}$, $\tau = 1, ..., t$, and being in state k at time t:

$$\alpha_k(\mathbf{x}, t) = P(x^{(1)}, x^{(2)}, ..., x^{(t)}, s_t = k), \quad k \in \mathcal{S}_t.$$

This probability can be evaluated by the following recursive formula:

$$\begin{aligned} \alpha_k(\mathbf{x}, 1) &= \pi_k \phi(x^{(1)} \mid \mu_k^{(1)}, \Sigma_k^{(1)}), \quad k \in \mathcal{S}_1, \\ \alpha_k(\mathbf{x}, t) &= \phi(x^{(t)} \mid \mu_k^{(t)}, \Sigma_k^{(t)}) \sum_{l \in \mathcal{S}_{t-1}} \alpha_l(\mathbf{x}, t-1) a_{l,k}^{(t-1)}, \quad 1 < t \le T,\ k \in \mathcal{S}_t. \end{aligned}$$

Define the backward probability $\beta_k(\mathbf{x}, t)$ as the conditional probability of observing the variable blocks after time t, $x^{(\tau)}$, $\tau = t+1, ..., T$, given the state at block t is k:

$$\begin{aligned} \beta_k(\mathbf{x}, t) &= P(x^{(t+1)}, ..., x^{(T)} \mid s_t = k), \quad 1 \le t \le T-1,\ k \in \mathcal{S}_t, \\ &\quad \text{Set } \beta_k(\mathbf{x}, T) = 1, \quad \text{for all } k \in \mathcal{S}_T. \end{aligned}$$

The backward probability can be evaluated using the following recursion:

$$\begin{aligned}\beta_k(\mathbf{x}, T) &= 1, \quad k \in \mathcal{S}_T, \\ \beta_k(\mathbf{x}, t) &= \sum_{l \in \mathcal{S}_{t+1}} a_{k,l}^{(t)} \phi(x^{(t+1)} \mid \mu_l^{(t+1)}, \Sigma_l^{(t+1)}) \beta_l(\mathbf{x}, t+1), \quad 1 \le t < T, \ \ k \in \mathcal{S}_t.\end{aligned}$$

The probabilities $L_k(\mathbf{x}, t)$ and $H_{k,l}(\mathbf{x}, t)$ are solved by

$$L_k(\mathbf{x}, t) = P(s_t = k \mid \mathbf{x}) = \frac{P(\mathbf{x}, s_t = k)}{P(\mathbf{x})} = \frac{\alpha_k(\mathbf{x}, t)\beta_k(\mathbf{x}, t)}{P(\mathbf{x})}, \quad k \in \mathcal{S}_t,$$

$$\begin{aligned}&H_{k,l}(\mathbf{x}, t) = P(s_t = k, s_{t+1} = l \mid \mathbf{x}) = \frac{P(\mathbf{x}, s_t = k, s_{t+1} = l)}{P(\mathbf{x})} \\ = \ &\frac{1}{P(\mathbf{x})} \alpha_k(\mathbf{x}, t) a_{k,l}^{(t)} \phi(x^{(t+1)} \mid \mu_l^{(t+1)}, \Sigma_l^{(t+1)}) \beta_l(\mathbf{x}, t+1), \quad k \in \mathcal{S}_t, \ l \in \mathcal{S}_{t+1}.\end{aligned}$$

The normalizing factor $P(\mathbf{x}) = \sum_{k \in \mathcal{S}_t} \alpha_k(\mathbf{x}, t)\beta_k(\mathbf{x}, t)$ holds for any t.

To initialize the model, we design several schemes. In our experiments, models from different initializations are estimated and the one with the maximum likelihood is chosen. In our baseline initialization scheme, k-means clustering is applied individually to each variable block using all the data instances. Based on the clustering result of k-means, we take every cluster as one mixture component and compute the sample mean and sample covariance matrix of data in that cluster. To reduce the sensitivity to the initial clustering result, we also compute the pooled common sample covariance matrix for the clusters. The initial covariance matrix of a component is then set as a convex combination of the cluster-specific sample covariance and the common sample covariance. The transition probabilities are always initialized to be uniform. Under the second initialization scheme, we randomly sample a subset from the whole data and apply the baseline initialization to the subset. Under the third initialization scheme, we randomly pick a subset from the data and treat points in this subset as the cluster centroids of the k-means. These centroids will induce a cluster partition of the whole data, based on which we initialize the component means and covariance matrices in the same way as the baseline method. Both the second and the third initialization schemes are repeated several times with different random starts.

3.2 Modal Baum-Welch Algorithm

HMM-VB can be viewed as a special case of a GMM where each component of the GMM corresponds to a particular sequence of states $\mathbf{s} = \{s_1, ..., s_T\}$, that is, a combination of states for all the variable blocks. We call this equivalent GMM the *GMM mapped from HMM-VB*. Each component is a Gaussian distribution with mean $\mu_{\mathbf{s}} = (\mu_{s_1}^{(1)}, \mu_{s_2}^{(2)}, ..., \mu_{s_T}^{(T)})$ (column-wise stack of vectors) and a covariance matrix, denoted by $\Sigma_{\mathbf{s}}$, that is block diagonal. The t-th diagonal block in $\Sigma_{\mathbf{s}}$ is $\Sigma_{s_t}^{(t)}$ with dimension $d_t \times d_t$. We can thus readily apply the modal clustering framework for GMM to data modeled by HMM-VB. However, the number of components in the mapped GMM is $M = \prod_{t=1}^{T} M_t$, which grows exponentially with T assuming similar M_t's. A direct application of the MEM algorithm (see Eq. (1) and (2)) is computationally infeasible. We discover that because of the block diagonal structure of the

covariance matrix of the GMM mapped from HMM-VB, we can in fact avoid computing the posterior of $\mathbf{x}$ belonging to each component (exponentially many of them!). Instead, we only need $L_k(\mathbf{x}, t)$ for all k and t when updating $\mathbf{x}$ in the M-step of MEM. Because the BW algorithm solves $L_k(\mathbf{x}, t)$ at a complexity linear in T, we can achieve linear complexity for solving the modes of HMM-VB as well. We call this new algorithm *Modal Baum-Welch (MBW)* Algorithm.

We denote by $x^{(t),r}$ the value of the t-th variable block at iteration r, and let $\mathbf{x}^{[r]} = (x^{(1),r}, x^{(2),r}, ..., x^{(T),r})$ be the concatenated and properly ordered full vector at iteration r. The equivalence of MBW and the Modal EM algorithm is ensured by Theorem 7 in Section 4.3, which is proved in Appendix D.

The MBW algorithm iterates the following two steps:

1. E-step: Compute $L_k(\mathbf{x}^{[r]}, t)$, for $k \in \mathcal{S}_t$, $t = 1, ..., T$.

2. M-step: For $t = 1, ..., T$,

$$x^{(t),r+1} = \left(\sum_{k \in \mathcal{S}_t} L_k(\mathbf{x}^{[r]}, t) \cdot \left(\Sigma_k^{(t)} \right)^{-1} \right)^{-1} \left(\sum_{k \in \mathcal{S}_t} L_k(\mathbf{x}^{[r]}, t) \cdot \left(\Sigma_k^{(t)} \right)^{-1} \cdot \mu_k^{(t)} \right).$$

The clustering method based on MBW is straightforward. We first find the state sequence $\mathbf{s}_i^{(*)}$ with maximum posterior given $\mathbf{x}_i$ by the Viterbi algorithm (Young et al., 1997):

$$\mathbf{s}_i^* = \arg\max_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}_i), \ i = 1, ..., n.$$

Since different $\mathbf{x}_i$'s may yield the same sequence, we then identify the collection of distinct $\mathbf{s}_i^*$. For each distinct sequence, say $\mathbf{s}^*$, find $\mu_{\mathbf{s}^*} = (\mu_{s_1^*}^{(1)}, \mu_{s_2^*}^{(2)}, ..., \mu_{s_T^*}^{(T)})$. Use $\mu_{\mathbf{s}^*}$ as an initialization for MBW to find the mode associated with it. If $\mu_{\mathbf{s}_i^*}$ and $\mu_{\mathbf{s}_j^*}$ are brought to the same mode by MBW, the corresponding data vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ are put into the same cluster. When $|\hat{\mathcal{S}}|$ is very large, the number of different $\mathbf{s}_i^*$'s can become close to the data size. Hence, the amount of computation we can save by seeking modes starting from $\mu_{\mathbf{s}_i^*}$'s instead of the original data diminishes. As a result, in such cases, we recommend seeking modes directly from the original data.

3.3 Computational Complexity

For both the BW estimator of HMM-VB and the MBW algorithm, the vast majority of the computation is on obtaining $L_k(\mathbf{x}_i, t)$ and $H_{k,l}(\mathbf{x}_i, t)$, $i = 1, ..., n$, $t = 1, ..., T$, $k \in \mathcal{S}_t$, $l \in \mathcal{S}_{t+1}$. For clarity of discussion, suppose the number of states in each block, $|\mathcal{S}_t|$, is a constant. Then the complexity for computing the two quantities is $O(nT|\mathcal{S}_t|^2)$.

The two quantities $L_k(\mathbf{x}_i, t)$ and $H_{k,l}(\mathbf{x}_i, t)$ can be computed separately for each sample $\mathbf{x}_i$. Thus the E-step in both the BW and the MBW algorithms are easily parallelizable by simply dividing the data among multiple processors. For the MBW algorithm, since we search for the mode separately starting from each sample point, the M-step is also naturally parallelizable. For the BW estimation algorithm, in the M-step, some quantities need to be

transmitted to a central processor to update the parameters, which will then be broadcast to the distributed processors. However, an inspection of the M-step shows that we do not need to communicate $L_k(\mathbf{x}_i, t)$ and $H_{k,l}(\mathbf{x}_i, t)$ for every sample point $\mathbf{x}_i$ to the central processor. Suppose the first processor treats a data segment containing points 1, 2, ..., n_1. It only needs to send $\sum_{i=1}^{n_1} L_k(\mathbf{x}_i, t)$, $\sum_{i=1}^{n_1} L_k(\mathbf{x}_i, t) x_i^{(t)}$, $\sum_{i=1}^{n_1} L_k(\mathbf{x}_i, t) x_i^{(t)} x_i^{(t)'}$, $\sum_{i=1}^{n_1} H_{k,l}(\mathbf{x}_i, t)$, $k \in \mathcal{S}_t$, $l \in \mathcal{S}_{t+1}$, $t = 1, ..., T$, to the central processor. Thus the communication load from a processor treating one data segment to the central processor does not depend on the data size and is nearly as low as the number of parameters in the model (the negligible relative difference shrinks quickly when $|\mathcal{S}_t|$ grows). The central processor can then update the parameters precisely according to the formula in the M-step. Since the design of the parallel algorithm of BW is simple, we hereby skip the details for brevity.

3.4 Constructing Variable Dependence Structure

We refer to the grouping and ordering of the variable blocks, that is, the formation of $X^{(t)}$, $t = 1, ..., T$, as the dependence structure or variable block structure. We have so far assumed that the dependence structure of HMM-VB is given. To complete our new framework for clustering, we now address how to determine such a dependence structure when it is not pre-specified. We propose to seek a dependence structure that yields the HMM-VB with the minimum BIC for the data. We use the following definition of BIC:

$$BIC = -2\log(\hat{L}) + k\log(n),$$

where $\hat{L}$ is the maximum value of the likelihood function, n is the sample size, and k is the number of free parameters to be estimated. The challenge lies in the computational complexity of the combinatorial optimization problem. An exhaustive search is intractable even for moderate dimensions (the number of possible groupings and orderings is much larger than that of variable permutations). To achieve computational feasibility, we design a greedy local search scheme.

We first generate an ordering of the variables based on some prior knowledge or on a random permutation, which we call raw ordering hereafter. The raw ordering is not necessarily the final ordering of the variables after the grouping is decided, although the former indeed strongly influences the latter. We may generate several random raw orderings of the variables. Given any raw ordering, the variables are grouped by a step-wise optimization procedure. Based on the likelihoods of the corresponding HMM-VBs, the dependence structure will be chosen under each raw ordering. We then compare structures found from all the raw orderings and select the best from them.

Denote the raw ordering of the variables by Q, with $Q(j) \in \{1, ..., d\}$, for $j = 1, ..., d$. Note that $Q(j)$ is a bijection between $\{1, ..., d\}$ and itself. Denote by G the grouping for the original variable vector. For example, if $Q = (3, 2, 1)$, with $d = 3$, then the specified ordering of the input data for HMM-VB is $(X_{Q(1)}, X_{Q(2)}, X_{Q(3)}) = (X_3, X_2, X_1)$. Suppose that there are two variable blocks, where X_2 and X_3 form the first variable block and X_1 belongs to the second block. Then $G(Q(1)) = 1$, $G(Q(2)) = 1$ and $G(Q(3)) = 2$, or equivalently $G(1) = 2$, $G(2) = 1$, and $G(3) = 1$.

Next, we determine how the variables are grouped based on a step-wise selection process. The variables are treated one by one in the order given by Q. The algorithm ensures that

$G(Q(1))$, ..., $G(Q(j-1))$ have been determined when solving $G(Q(j))$. Specifically, suppose $G(Q(1)), ..., G(Q(j-1)) \in \{1, 2, ..., g\}$. That is, the first $j-1$ ordered variables have been put into g non-empty groups. Note that $g \leq j-1$. The possible value for $G(Q(j))$ is 1, ..., g, or $g+1$. If the $Q(j)$-th variable is put in any existing group, then $G(Q(j)) \leq g$; otherwise it forms by itself a new group with identity number $g+1$. In order to determine $G(Q(j))$, we compare exhaustively the structures with $G(Q(j)) = 1, 2, ..., g, g+1$, which means experimenting with putting the $Q(j)$-th variable in each existing group as well as forming a new group by the $Q(j)$-th variable alone. We use each structure to estimate a HMM-VB for the first j ordered variables: $X_{Q(1)}, X_{Q(2)}, ..., X_{Q(j)}$ (note that this is not the full dimensional data). These HMM-VBs are compared by BIC using the j-dimensional data, and the one with the optimal BIC is chosen, which in turn determines $G(Q(j))$. The process is repeated to sweep through all the variables until the full dimension $j = d$.

Note that we need to specify the number of components for each variable block. After extensive numerical experiments, we set $M_t = 10$ if $d_t \leq 5$, $M_t = 15$ if $d_t \in [6, 10]$, otherwise, $M_t = d_t + 10$ for t being any variable block. In addition, the greedy local search algorithm is quite robust to the change in the number of mixture components.

We denote the BIC of the estimated HMM-VB under raw ordering Q and grouping $G(Q(1))$, ..., $G(Q(j))$ for the partial data containing the first j ordered variables by $\mathcal{L}_{BIC}(\mathbb{X}_{Q(1),...,Q(j)}, G(Q(1)), ..., G(Q(j)), \theta)$, where θ denotes the parameters of HMM-VB. Recall that $\mathbb{X}_{Q(j)}$ denotes the $Q(j)$-th column of the data matrix $\mathbb{X}$. Our step-wise selection algorithm under a given raw ordering Q is as follows:

1. Input data matrix $\mathbb{X}$ and the ordering structure $\{Q(1), ..., Q(d)\}$.

2. Set $j = 1$, $g = 1$, $G(Q(1)) = 1$.

3. For $j = 2$, ..., d

   (a) For each $k = 1$, ..., g, $g+1$, obtain the maximum likelihood estimation of HMM-VB for partial data composed of $\mathbb{X}_{Q(1)}, \mathbb{X}_{Q(2)}, ..., \mathbb{X}_{Q(j)}$ under structure Q and $(G(Q(1)), ..., G(Q(j-1)), G(Q(j)) = k)$. Let the estimated parameter at k be $\theta^{*,k}$.

   (b) Compute

$$G^*(Q(j)) = \underset{k \in \{1,...,g,g+1\}}{\operatorname{argmin}} \mathcal{L}_{BIC}(\mathbb{X}_{Q(1),...,Q(j)}, G(Q(1)), ..., G(Q(j)) = k, \theta^{*,k}).$$

   (c) Set $G(Q(j)) \longleftarrow G^*(Q(j))$.

   (d) If $G(Q(j)) = g+1$, set $g \longleftarrow g+1$.

Note that if the prior knowledge of the ordering structure is unknown, we randomly generate the raw ordering $\{Q(j)\}_{j=1}^d$ multiple times and adopt the one achieving the optimal BIC. We find that this approach performs well empirically as illustrated in the experiment section. It is also not difficult to see that this algorithm runs on the order of $O(d^2)$, which is efficient in practice.

4. Theoretical Properties

We study the identifiability of HMM-VB, prove the consistency of using BIC for model selection in a special case, and prove the optimality of MBW algorithm for HMM-VB.

4.1 Identifiability

Since HMM-VB can be viewed as a special case of a GMM, we need to ensure model identifiability, which is a necessary condition for estimating the parameters of a mixture model consistently. Specifically, we need to make sure that no two essentially different mixture parameters define the same distribution. We now introduce a type of GMM that includes HMM-VB as a special case and prove some results that help establish the identifiability of HMM-VB. A list of new definitions and notations is provided first.

The variable blocks $X^{(t)}$, $t = 1, ..., T$ are mutually exclusive and collectively exhaustive (that is, their union is the set of all the variables). For brevity, with a slight abuse of notation, using $X^{(t)}$ to mean both a subvector of X as well as a set of variables, but ensure that the specific meaning is clear from context. Denote a *variable partition* by $\mathcal{P} = \{X^{(1)}, X^{(2)}, ..., X^{(T)}\}$.

1. *Lattice GMM*: Denote the Gaussian parameter of a mixture component for variable block $X^{(t)}$ by $\theta_{i_t}^{(t)} = (\mu_{i_t}^{(t)}, \Sigma_{i_t}^{(t)})$, $i_t = 1, ..., M_t$. We define a *lattice GMM* on a variable partition $\mathcal{P}$ as a GMM that bears the form

$$f(\mathbf{x}) = \sum_{i_1=1}^{M_1} \sum_{i_2=1}^{M_2} \cdots \sum_{i_T=1}^{M_T} \pi(i_1, ..., i_T) \cdot \prod_{t=1}^{T} \phi(x^{(t)} \mid \theta_{i_t}^{(t)}). \tag{6}$$

   For a lattice GMM, a component is indexed by a T-tuple $(i_1, ..., i_T)$. Denote the number of components for each variable block collectively by $\mathbb{M} = \{M_1, ..., M_t\}$. We introduce latent state variables s_t for each block $X^{(t)}$, $s_t \in \{1, ..., M_t\}$, $t = 1, ..., T$. The joint pmf of s_t's is given by $\pi(i_1, ..., i_T)$. We use $\Pi(s_1, ..., s_T)$ to denote the joint pmf of $s_1, ..., s_T$ and $\Pi(s_{t_1}, s_{t_2}, ..., s_{t_k})$ to denote the marginal pmf of any subset of the latent states.

   $\Theta^{(t)} = \{\theta_{i_t}^{(t)}, i_t = 1, ..., M_t\}$ is the *grid* of parameters for variable block $X^{(t)}$. Clearly, a lattice-GMM is a mixture of components whose parameters are points from the Cartesian product of the grid of each variable block: $\Theta = \Theta^{(1)} \times \Theta^{(2)} \cdots \times \Theta^{(T)}$. We call Θ the *lattice* of parameters for the full dimensional vector X.

   We say $\theta_{i_t}^{(t)} \in \Theta^{(t)}$ *exists* if $P(s_t = i_t) > 0$, or equivalently there is at least one set of i_1, ..., i_{t-1}, i_{t+1}, ..., i_T such that $\pi(i_1, i_2, ..., i_T) > 0$. We say that the grid $\Theta^{(t)}$ is *distinct* if $\theta_{i_t}^{(t)} \neq \theta_{i_t'}^{(t)}$ for any $i_t \neq i_t'$. The grid $\Theta^{(t)}$ is *non-redundant* if it is distinct and every $\theta_{i_t}^{(t)} \in \Theta^{(t)}$, $i_t = 1, ..., M_t$, exists. If $\Theta^{(t)}$ is not non-redundant, then we can shrink the grid $\Theta^{(t)}$ by eliminating some θ_{i_t}'s and/or merging identical θ_{i_t}'s in the set. Lattice Θ is *non-redundant* if all $\Theta^{(t)}$, $t = 1, ..., T$ are non-redundant.

2. *Tight variable partition*: $\mathcal{P}$ is a *tight variable partition* (or simply tight partition) for a lattice GMM if by Eq. (6), the component prior $\pi(i_1, ..., i_T) > 0$ for all the T-tuples $(i_1, ..., i_T)$.

3. *Maximum variable partition*: A variable partition $\mathcal{P}_1$ is *nested* in partition $\mathcal{P}_2$, denoted by $\mathcal{P}_1 \succ \mathcal{P}_2$ or $\mathcal{P}_2 \prec \mathcal{P}_1$, if every variable block of $\mathcal{P}_1$ is a subset of a variable block of $\mathcal{P}_2$. That is, the partition $\mathcal{P}_1$ can be obtained from $\mathcal{P}_2$ by further dividing a variable block into smaller blocks. $\mathcal{P}$ is a *maximum variable partition* (or simply maximum partition) for a lattice GMM if $\mathcal{P}$ is a tight variable partition and there exists no other tight partition $\mathcal{P}'$ for the GMM such that $\mathcal{P}' \succ \mathcal{P}$.

We let $\mathcal{M}$ to denote the collection of parameters that specify a lattice GMM on the variable partition $\mathcal{P}$. $\mathcal{M} = \{\mathbb{M}, \Theta, \Pi(s_1, ..., s_T)\}$. We also use $\mathcal{M}_{X^{(t)}}$ to denote the marginal GMM for $X^{(t)}$ derived from $\mathcal{M}$. For instance, $\mathcal{M}_{X^{(t)}} = \{M_t, \Theta^{(t)}, \Pi(s_t)\}$. Similarly, we can have marginal GMM for multiple variable blocks, e.g., $\mathcal{M}_{X^{(t)},X^{(t+1)}}$.

Suppose $\mathcal{O}$ is a permutation function from one index set to another and Γ is a set of indexed parameters. We denote the permuted parameters by $\mathcal{O}(\Gamma)$. For instance, if $\Gamma = \{\mu_1, \mu_2, \mu_3\}$ and $\mathcal{O}$ permutes $\{1, 2, 3\}$ to $\{3, 2, 1\}$, then $\mathcal{O}(\Gamma) = \{\mu_3, \mu_2, \mu_1\}$. Consider an index given by a T-tuple $(i_1, ..., i_T)$ and $\mathcal{O}_t$ is a permutation function on the tth position i_t. We use $\mathcal{O}_{1:T} = \mathcal{O}_1 \times \mathcal{O}_2 \cdots \times \mathcal{O}_T$ to denote the permutation on the T-tuple: $(i_1, i_2, ..., i_T) \to (\mathcal{O}_1(i_1), \mathcal{O}_2(i_2), ..., \mathcal{O}_T(i_T))$.

Lemma 1 *The identifiability of GMM gives that* $\sum_{k=1}^{M} \pi_k \phi(X|\mu_k, \Sigma_k) = \sum_{l=1}^{M^*} \pi_l^* \phi(X|\mu_l^*, \Sigma_l^*)$ *with distinct* (μ_k, Σ_k)*'s, distinct* (μ_l^*, Σ_l^*)*'s, and all the priors* $\pi_k > 0$ *and* $\pi_l^* > 0$, $k = 1, ..., M$, $l = 1, ..., M^*$ *implies* $M = M^*$ *and up to a permutation of mixture components,* $\pi_k = \pi_k^*$, $\mu_k = \mu_k^*$ *and* $\Sigma_k = \Sigma_k^*$.

See for instance Yakowitz and Spragins (1968); Titterington et al. (1985).

Theorem 2 *Let* $\mathcal{M}$ *and* $\mathcal{M}'$ *be two sets of parameters for a lattice GMM on the same variable partition* $\mathcal{P}$*. Assume that* $\mathcal{M}$ *and* $\mathcal{M}'$ *specify the same density function and their lattices* Θ *and* Θ' *are both non-redundant. Then the number of components for each variable block* $M_t = M_t'$, $t = 1, ..., T$*. There exists a unique permutation* $\mathcal{O}_t : i_t \to i_t'$ *for each variable block* $X^{(t)}$ *such that* $\mathcal{M}'_{X^{(t)}} = \mathcal{O}_t(\mathcal{M}_{X^{(t)}})$ *and* $\mathcal{M}' = \mathcal{O}_{1:T}(\mathcal{M})$.

Remark: $M_t = M_t'$ and $\mathcal{M}'_{X^{(t)}} = \mathcal{O}_t(\mathcal{M}_{X^{(t)}})$ are simple results of Lemma 1. The assumption that Θ and Θ' are non-redundant does not imply every prior in Eq. (6) is positive. Hence Lemma 1 cannot be directly applied to prove $\mathcal{M}$ and $\mathcal{M}'$ are identical up to permutation. We provide the proof for Theorem 2 in Appendix B.

The generic HMM-VB density in Eq. (3), given a pre-determined variable block structure can be re-written as

$$f(\mathbf{x}) = \sum_{i_1=1}^{M_1} \sum_{i_2=1}^{M_2} \cdots \sum_{i_T=1}^{M_T} \left(\pi_{i_1} a_{i_1,i_2}^{(1)} a_{i_2,i_3}^{(2)} \cdots a_{i_{T-1},i_T}^{(T-1)} \right) \cdot \prod_{t=1}^{T} \phi(x^{(t)}|\mu_{i_t}^{(t)}, \Sigma_{i_t}^{(t)}). \tag{7}$$

It is clear that HMM-VB is a lattice GMM on partition $\mathcal{P}$. Specifically, the parameter set $\mathcal{M}$ for a HMM-VB is $\mathcal{M} = \{\mathbb{M}, \Theta, \pi_{i_1}, a_{i_{t-1},i_t}^{(t-1)}, i_1 = 1, ..., M_1, i_t = 1, ..., M_t, t = 2, ..., T\}$.

We prove the following lemma in Appendix B which specifies the conditions under which a HMM-VB has non-redundant lattice.

Lemma 3 *The HMM-VB in Eq.* (7) *has non-redundant lattice* Θ *if and only if* Θ *is distinct and* $\pi_{i_1} > 0$*, for* $\forall i_1 \in \{1, ..., M_1\}$ *and for* $\forall t = 2, ..., T$ *and* $i_t \in \{1, ..., M_t\}$*, there exists at least one* $i_{t-1} \in \{1, ..., M_{t-1}\}$ *such that* $a^{(t-1)}_{i_{t-1}, i_t} > 0$*.*

Corollary 4 *For a given variable block structure, if two HMM-VBs,* $\mathcal{M}$ *and* $\mathcal{M}'$*, both have non-redundant lattices* Θ *and* Θ'*, and define the same density function, then there exists a unique permutation* $\mathcal{O}_t$ *for the mixture components of every variable block* $X^{(t)}$*,* $t = 1, ..., T$*, such that* $\mathcal{M}' = \mathcal{O}_{1:T}(\mathcal{M})$*.*

Remark: Corollary 4 establishes the identifiability of HMM-VB under a given variable block structure. By Theorem 2, it is obvious that $\mathbb{M}' = \mathcal{O}_{1:T}(\mathbb{M})$, $\Theta' = \mathcal{O}_{1:T}(\Theta)$, and $\Pi'(s_1, ..., s_T) = \mathcal{O}_{1:T}(\Pi(s_1, ..., s_T))$. We only need to show that the last equation implies that the transition probabilities are identical up to permutation $\mathcal{O}_{1:T}$. We prove this in Appendix B.

We have so far assumed that the partition $\mathcal{P}$ for the lattice GMM is given. A natural question is whether $\mathcal{P}$ is identifiable. To answer the question, we assume $\mathcal{P}$ is a tight variable partition. Without this constraint, we can express any GMM with M components as a lattice GMM with every block containing a single variable and every block being assigned with M components. The number of components for the lattice GMM will be M^d with only M components assigned with non-zero priors. If we restrict to tight variable partitions, the maximum partition will not be trivial. Furthermore, we can prove that the the maximum partition always exists and is unique. Thus the identifiability of a GMM (according to Lemma 1) ensures that the maximum partition is identifiable. That is, the variable block structure, in its most refined partition, is identifiable.

Theorem 5 *The maximum variable partition of a GMM exists and is unique.*

The proof is given in Appendix B.

Suppose $\mathcal{M}$ is a HMM-VB on a tight partition $\mathcal{P}$. It is obvious that $\mathcal{P}$ is a tight partition for a HMM-VB if and only if $\pi_{i_1} > 0$, $\forall i_1 \in \{1, ..., M_1\}$, and all the transition probabilities are positive. If $\mathcal{P}$ is a maximum partition for the equivalent lattice GMM of $\mathcal{M}$, $\mathcal{P}$ is identifiable once $\mathcal{M}$ is given (by Theorem 5). However, because the latent states s_1, ..., s_T of the HMM-VB follow a Markov chain (an extra constraint), even if $\mathcal{P}$ cannot be further refined for the HMM-VB, it is not necessarily the maximum partition for the corresponding lattice GMM. In other words, a lattice GMM that is not a HMM-VB on its maximum partition can be an HMM-VB on a coarser partition. Another subtlety with HMM-VB is that even when $\mathcal{P}$ is identifiable, the order of the variable blocks is not for the simple reason that a reversed Markov chain is still Markov. The fact that the order of the variable blocks is not identifiable is also clear from the extreme case when the latent states s_1, ..., s_T are independent and therefore any order of them is valid. On the other hand, because our mode-based clustering method for HMM-VB searches for the modes of the joint density, the modes are not affected as long as the equivalent lattice GMM is the same. Hence regardless of the order of the variable blocks (e.g., forward chain or backward chain), if the HMM-VB is a correct model, the modes will not change, and neither will the clustering result.

4.2 Consistency of BIC for Model Selection

In this section, we prove a special case that the probability of selecting the true variable block structure by minimizing BIC approaches 1 as $n \to \infty$ under some assumptions and the conditions that the true variable structure is among the candidates and the number of components for each variable block is known.

Let the number of variable blocks be T and its true value be T^0. Denote the true number of components in each block by M_t^0, $t = 1, ..., T^0$. Also let $\mathbb{M}^{0^{T^0}} = \{M_1^0, ..., M_{T^0}^0\}$. Denote the variable index sets of the true ordered blocks by $\mathbf{C}^{\mathbf{0}^{T^0}} = (C_1^0, C_2^0, ..., C_{T^0}^0)$. For example, C_1^0 contains the indices of the variables in the first block. For a particular sequence of variable blocks $\mathbf{C}^T = (C_1, C_2, ..., C_T)$, we use the notation $X^{(C_t)}$ to denote the tth block of variables according to $\mathbf{C}^T$. Denote the parameters of a model collectively by $\gamma_{(\mathbf{C}^T, \mathbb{M}^T)} \in \Gamma_{(\mathbf{C}^T, \mathbb{M}^T)}$, where $\Gamma_{(\mathbf{C}^T, \mathbb{M}^T)}$ is the space of the parameters.

Denote by g the true density function of X. Let $D_{KL}(g||f) = \int g(x) \log(g(x)/f(x))dx$ be the Kullback-Leibler divergence from density f to g. We define the following two notations:

$$\begin{aligned}
\gamma^*_{(\mathbf{C}^T, \mathbf{M}^T)} &= \underset{\gamma_{(\mathbf{C}^T, \mathbf{M}^T)} \in \Gamma_{(\mathbf{C}^T, \mathbf{M}^T)}}{\operatorname{argmin}} D_{KL}(g||f(.|\gamma_{(\mathbf{C}^T, \mathbf{M}^T)})) \\
&= \underset{\gamma_{(\mathbf{C}^T, \mathbf{M}^T)} \in \Gamma_{(\mathbf{C}^T, \mathbf{M}^T)}}{\operatorname{argmax}} E_X\{\log f(X|\gamma_{(\mathbf{C}^T, \mathbf{M}^T)})\}, \\
\hat{\gamma}_{(\mathbf{C}^T, \mathbf{M}^T)} &= \underset{\gamma_{(\mathbf{C}^T, \mathbf{M}^T)} \in \Gamma_{(\mathbf{C}^T, \mathbf{M}^T)}}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \log\{f(\mathbf{x}_i|\gamma_{(\mathbf{C}^T, \mathbf{M}^T)})\}.
\end{aligned}$$

Consider the case where the true model contains finite T^0 variable blocks and all the models in consideration are restricted to have the same number of variable blocks ($T = T^0$). To simplify the notation, all the dependence over T^0 and T is omitted below. We make two assumptions:

A1 There exists a unique $(\mathbf{C}^0, \mathbb{M}^0)$ such that $g = f(.|\gamma^*_{(\mathbf{C}^0, \mathbb{M}^0)})$ for some parameter value γ^*.

As discussed previously, when the order of the variable blocks is reversed, we obtain a HMM-VB that yields the same density function for X although parameterized differently. Hence, the order is not identifiable. Thus, the uniqueness in assumption **A1** is implicitly up to a reverse ordering. To further simplify the notation, the dependency over $\mathbb{M}^0$ is omitted below.

A2 $\gamma^*_{(\mathbf{C})}$ and $\hat{\gamma}_{(\mathbf{C})}$ are assumed to belong to a compact subspace $\Gamma'_{(\mathbf{C})}$:

$$\Gamma'_{(\mathbf{C})} = (\Lambda \times \mathcal{A} \times \mathcal{B}(\eta, |C_1|)^{M_1^0} \times \mathcal{D}_{|C_1|}^{M_1^0} \times \mathcal{B}(\eta, |C_2|)^{M_2^0} \times \mathcal{D}_{|C_2|}^{M_2^0} \times \cdots \times \mathcal{D}_{|C_T|}^{M_T^0} \times \mathcal{B}(\eta, |C_T|)^{M_T^0}) \cap \Gamma_{(\mathbf{C})},$$

where

[1] $\Lambda = \{(\pi_1, ..., \pi_{M_1^0}) \in [0,1]^{M_1^0}; \sum_{k=1}^{M_1^0} \pi_k = 1\}$ denotes the set of possible proportions,

[2] $$\mathcal{A} = \left\{ \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,M_{t+1}^0} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,M_{t+1}^0} \\ \vdots & \vdots & \cdots & \vdots \\ a_{M_t^0,1} & a_{M_t^0,2} & \cdots & a_{M_t^0,M_{t+1}^0} \end{pmatrix} \in [0,1]^{M_t^0 \times M_{t+1}^0}; \forall j \in 1 : M_t^0, \textstyle\sum_{k=1}^{M_{t+1}^0} a_{j,k} = 1 \right\}_{t=1}^{T-1},$$

[3] $\mathcal{B}(\eta, d) = \{\mathbf{x} \in \mathcal{R}^d, ||\mathbf{x}|| \leq \eta\}$, where $\forall \mathbf{x} \in \mathcal{R}^d$, $||\mathbf{x}|| = \sqrt{\sum_{i=1}^d x_i^2}$,

[4] $|C|$ denotes the cardinality of the set C,

[5] $\mathcal{D}_d$ is the set of $d \times d$ positive definite matrices with eigenvalues in $[a, b]$ with $0 < a < b$.

Theorem 6 *Under the special case that $T = T^0 < \infty$, and assumptions* **A1**, **A2**, *the ordered variable block structure $\hat{\mathbf{C}}$ that minimizes BIC under a given $\mathbb{M}^0$ is consistent in the sense that $P(\hat{\mathbf{C}} = \mathbf{C}^0) \to 1$ as $n \to \infty$.*

The proof is given in Appendix C.

Remark: We point out that the proved consistency in the asymptotic setting as stated above holds for marginal likelihood approaches as well. We observe empirically, however, that BIC performs better for model selection when the sample size is only modestly large. This is expected as BIC addresses overfitting, which has been discussed extensively in literature (e.g., Burnham and Anderson (2003)). Our simulation study in Section 5.1.1 further confirms the advantage of BIC when the sample size is modestly large.

4.3 Modal Baum-Welch Algorithm and Its Optimality

Recall the definition for probability $L_k(\mathbf{x}, t) = P(s_t = k \mid \mathbf{x})$, $k \in \mathcal{S}_t$, $t = 1, ..., T$.

Theorem 7 *For a HMM-VB, suppose the solution of the M-step in the MEM algorithm provided by Eq. (2) is divided into blocks $\mathbf{x}^{[r+1]} = (x^{(1),r+1}, ..., x^{(T),r+1})$. Then*

$$x^{(t),r+1} = \left(\sum_{k \in \mathcal{S}_t} L_k(\mathbf{x}^{[r]}, t) \cdot \left(\Sigma_k^{(t)} \right)^{-1} \right)^{-1} \left(\sum_{k \in \mathcal{S}_t} L_k(\mathbf{x}^{[r]}, t) \cdot \left(\Sigma_k^{(t)} \right)^{-1} \cdot \mu_k^{(t)} \right), \ t = 1, ..., T.$$

This theorem ensures that the MBW algorithm is the exact special case of the MEM algorithm when the GMM is a HMM-VB. The proof is provided in Appendix D.

5. Experiments

In this section, we present experimental results on several simulated data sets (Section 5.1), one mass cytometry data (Section 5.2), and two data sets with very high dimensions (Section 5.3). For each data set, the BW algorithm was run repeatedly starting from multiple initial models. In Section 3.1, the different ways of initialization are described. Among the final models, we choose the one yielding the maximum likelihood.

5.1 Simulation with Various Types of Variable Block Structures

We first conduct simulation studies to examine the effectiveness of HMM-VB at capturing small clusters, the overall clustering performance, and its robustness against different parameter settings.

5.1.1 Two variable blocks

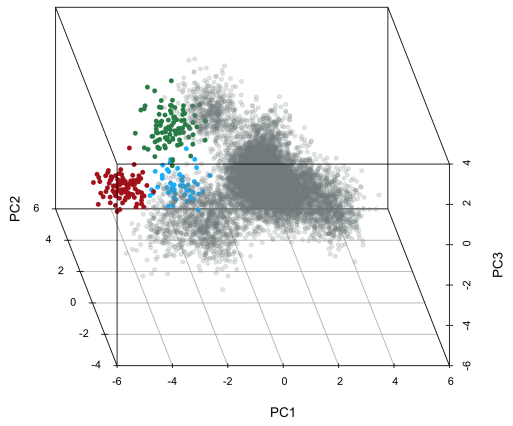
Using a similar set-up as in Lin et al. (2013), a sample of size $10,000$ with dimension $d = 8$ is drawn from a hierarchical mixture model. There are two variable blocks. Following the notations in the previous section, $\mathbf{x}_i$ is divided into two variable blocks, also called subvectors, $x_i^{(1)}$ and $x_i^{(2)}$. The first subvector contains the first 5 dimensions: $x_i^{(1)} = (x_{i,1}, ..., x_{i,5})$, with $d_1 = 5$. The second subvector contains the last 3 dimensions: $x_i^{(2)} = (x_{i,6}, x_{i,7}, x_{i,8})$, with $d_2 = 3$. In particular, $x_i^{(1)}$'s are generated from a mixture of 7 normal distributions such that the last two normal distributions (assigned with component priors 0.01 and 0.02 respectively) have high mean values for the second and third dimensions. The other normal components have very different proportions and mean vectors. The second subvector, $x_i^{(2)}$'s, are generated from a mixture of 10 normal distributions, where only two of them have high mean values across all three dimensions. The component proportions of $x_i^{(2)}$ vary according to which normal component $x_i^{(1)}$ was generated from, as in the assumption of HMM-VB. A detailed description of the data generation mechanism is in Appendix E. The data is designed to have at least one distinct cluster after standardization (subtract mean and divided by the standard error). In particular, the standardized data have a well-separated region that the five dimensions x_2, x_3, x_6, x_7, x_8 are of high positive values, and the rest are negative. The particular designed data region contains 100 data points, which account for only 1% of the data.

We also run the MBW algorithm to find modes of the true density (see Appendix E). The result serves two purposes: 1) to validate the effectiveness of MBW for finding small clusters when there is no model estimation error; 2) to be used as a ground truth for comparison with our HMM-VB method. MBW identifies perfectly that particular cluster of size 100. In addition, it finds in total 16 modes (clusters). Among them, there are two additional small clusters of size 54 and 98 respectively. Figure 5.1.1 (Left) shows the three smallest clusters.

To fit HMM-VB given the dependence structure among the 8 variables, we only need to specify M_1 and M_2, the numbers of mixture components for the two variable blocks. If casted as a GMM, the HMM-VB has $M_1 \times M_2$ components for the full dimensional data. Model selection by BIC is conducted to select the optimal M_1 and M_2. Summaries on various model specifications are listed in Figure 5.1.1 (Right). The model with $M_1 = 7$ and $M_2 = 10$ has the lowest BIC and is thus chosen.

Recall that we search modes by MBW starting from the mean of every component that has been chosen by a data vector according to the maximum a posteriori rule. We call the components that have not been chosen by any data point the empty mixture components, while the others are nonempty. Figure 5.1.1 (Right) shows that even though the total number of nonempty mixture components increases with M_1 and M_2, the number of clusters after modal clustering remains relatively stable.

In Table 1, we compare clustering results of HMM-VB with four baseline methods: K-means, agglomerative hierarchical clustering on a set of dissimilarities, Mclust, and modal GMM. We also tried other methods, such as spectral clustering using R `kernlab` package, pdfCluster (Azzalini and Menardi, 2014) using R `pdfCluster` package, and the mean-shift algorithm (Fukunaga and Hostetler, 1975; Cheng, 1995) using R `MeanShift` package. R `pdfCluster` program was aborted by the computer system when applied to the data. R `kernlab` and `MeanShift` packages failed to obtain results after several hours, while the



(M_1, M_2)	BIC	# nonempty components	# clusters
$(3, 3)$	$183,687.4$	8	2
$(3, 5)$	$176,134.6$	9	7
$(7, 10)$	$163,486.1$	35	16
$(8, 10)$	$163,724.0$	37	16
$(10, 10)$	$164,245.1$	45	16
$(10, 15)$	$164,822.9$	70	17
$(15, 10)$	$165,338.6$	56	16
$(15, 15)$	$166,313.4$	85	18
$(20, 15)$	$167,761.7$	121	17
$(20, 20)$	$206,478.6$	158	19

Figure 2: (Left): Visualization of the simulated data in Section 5.1.1 based on the first three principal components (PC). The smallest three clusters are plotted in blue (54 observations), green (98), red (100), and the rest are in grey. (Right): Comparisons of BIC, total number of nonempty components and clusters under various model specifications (M_1 and M_2).

methods in Table 1 take less than one minute. For K-means, hierarchical clustering, and Mclust, we used the three R functions `kmeans` (with 20 starting points), `hclust` and `Mclust`. In this analysis, we treat the 16 clusters found by MBW applied to the true mixture density as the ground truth. Hence, for K-means and hierarchical clustering, we manually set 16 as the number of clusters. The number of normal mixture components and the covariance structure are determined by BIC in the `Mclust` package (the package version is 5.2.3). More specifically, we let `Mclust` search a maximum of 20 mixture components and then used the BIC to choose among all the default covariance matrix models, as computational errors were generated when searching over all the models. For this data set, the optimal number of clusters chosen by Mclust is 15. The modal GMM analysis is performed using the same C codes for fitting the HMM-VB model. We simply specified a single variable block containing all the 8 variables. The HMM-VB model is then reduced to a usual GMM. In addition, we let the total number of mixture components be $7 \times 10 = 70$.

The first column of Table 1 shows the 16 true clusters and their cluster sizes. Let us call the true clusters E_1^*, ..., E_{16}^*. Results about each true cluster occupy one row in the table, while results obtained from every clustering method are reported in one column. Take the first row for cluster E_1^* as an example. E_1^* has $5,098$ data points. A cluster generated by any method is described by a pair of numbers, the one outside the parenthesis being the size of the cluster and the one inside being the number of points shared with the true cluster. For instance, HMM-VB generates a cluster containing $5,096$ points, among which $5,088$ are shared with E_1^*. This cluster is the best match with E_1^*. In comparison, modal GMM yields a cluster of size $4,695$ overlapping E_1^* by $4,467$. By K-means, hierarchical clustering or Mclust, E_1^* is split roughly into three clusters. Thus three pairs of cluster sizes are recorded in the first row under each method. For clarity of presentation, if the overlap between a

cluster and any E_i^* is too small, the cluster is not listed in the ith row. Reversely, a cluster generated by a certain method may also cover a substantial portion of points from multiple true clusters. For instance, hierarchical clustering yields one small cluster of size 257, which contains both part of E_1^* and E_2^*. A cluster of size 154 generated by K-means contains E_5^* and E_8^*. Therefore, a single cluster generated by a certain method may be reported in multiple rows. To indicate clearly such a case, the same color (except for black) is used to mark out a single cluster in any column. If a cluster is reported only once in any column, the result is shown in black (in other words, black is not a color code).

True cluster	HMM-VB	K-means	HC	Mclust	Modal GMM
$1:5,098$	$5,096(5,088)$	$1,700(1,693)$ $1,452(1,447)$ $1,477(1,474)$	$1,192(1,140)$ $3,852(3,757)$ $257(116)$	$2{,}240(2{,}239)$ $2,815(2,815)$ $211(28)$	$4,695(4,467)$
$2:184$	$187(184)$	$638(173)$	$257(132)$	$211(182)$	$4,695(140)$
$3:483$	$483(483)$	$485(481)$	$400(394)$ $81(81)$	$482(482)$	$299(297)$ $179(179)$
$4:258$	$263(254)$	$254(249)$	$257(247)$	$260(256)$	$214(18)$
$5:100$	$100(100)$	$154(100)$	$154(100)$	$154(100)$	$152(100)$
$6:375$	$375(373)$	$380(371)$	$1,242(362)$	$378(375)$	$374(362)$
$7:345$	$346(345)$	$510(338)$	$500(338)$	$513(345)$	$377(338)$
$8:54$	$54(54)$	$154(54)$	$154(54)$	$154(54)$	$152(52)$
$9:161$	$161(161)$	$510(156)$	$500(155)$	$513(161)$	$145(143)$
$10:502$	$505(502)$	$499(499)$	$552(499)$	$506(502)$	$348(348)$ $155(153)$
$11:201$	$195(191)$	$201(193)$	$198(188)$	$198(196)$	$214(196)$
$12:98$	$98(98)$	$98(98)$	$98(98)$	$98(98)$	$98(98)$
$13:384$	$384(384)$	$384(383)$	$383(382)$	$384(384)$	$383(383)$
$14:915$	$914(908)$	$929(907)$	$1,242(841)$ $2(1)$	$919(913)$	$852(848)$ $4,695(60)$
$15:707$	$704(704)$	$704(704)$	$697(697)$ $2(1)$	$707(707)$	$556(556)$ $149(149)$
$16:135$	$135(135)$	$135(135)$	$135(135)$	$135(135)$	$299(1)$
Time	4.087s	1.106s	4.245s	1.457s	48.169s

Table 1: Comparisons of clustering performance among HMM-VB, K-means, hierarchical clustering (HC), Mclust, and modal GMM for the simulation data in Section 5.1.1.

Table 1 demonstrates that HMM-VB outperforms clearly the other methods in terms of matching the true clusters, especially for the rare clusters. K-means, hierarchical clustering, and Mclust tend to yield similar-sized clusters. All of them split the largest true cluster E_1^* into 3 smaller ones. Except for HMM-VB, all the other four methods fail to correctly identify all three smallest clusters, E_5^*, E_8^* and E_{12}^*. Specifically, according to the four methods, E_5^* and E_8^* are in the same cluster. The CPU time per model fitting (based on the best model specification) on an iMac with Intel Core i7 3.0GHz/8GB memory is recorded, given in the last row of Table 1. The unit of time is second. Under the same specification of

the number of clusters, K-means is faster than hierarchical clustering. Mclust is performed on 15 mixture components, while both HMM-VB and modal GMM are fitted using 70 mixture components. The fact that HMM-VB is much faster than modal GMM shows the computational advantage to regularize a general GMM by a variable block structure.

In addition, we use this simulation data to test the search algorithm for the variable block structure, which is described in Section 3.4. We run the algorithm with 6 different initial random permutations. The algorithm successfully selects the correct variable blocks according to the true distribution. Furthermore, to empirically demonstrate the efficiency of BIC for model selection, we also incorporate the comparison with (log) marginal likelihood approach in Table 2. Both approaches select the same model under the larger sample size.

Random permutation	Log marginal likelihood	BIC
	$n = 10,000$	
$3,1,8,5,7,2,6,4$	$(3,5,2,4) * (1,8,7,6) : -80887.22$	$(3,1,8,5,7,2,6,4) : 167105.9$
$6,4,7,3,5,2,8,1$	$(6,7,8) * (4,3,5) * (2) * (1) : -80330.64^{\dagger}$	$(6,4,7,3,5,2,8,1) : 167113.3$
$4,6,8,5,7,2,3,1$	$(4) * (6,8,5,7) * (2,3) * (1) : -82102.33$	$(4,6,8,5,7) * (2,3,1) : 166259$
$4,6,5,8,1,3,7,2$	$(4,5,3,2) * (6,8,1,7) : -80924.62$	$(4,6,5,8,1,3,7,2) : 167593.6$
$5,3,2,6,8,7,1,4$	$(5,1,4) * (3,2) * (6,7) * (8) : -80693.58$	$(5,3,2,1,4) * (6,8,7) : 164213.2^{*}$
$7,4,5,3,1,6,8,2$	$(7,6,8) * (4,3,1) * (5) * (2) : -80343.08$	$(7,4,5,3,1,6,8,2) : 167468.1$
	$n = 50,000$	
$3,1,8,5,7,2,6,4$	$(3,8,7,2,6,4) * (1,5) : -406073.13$	$(3,1,2,4) * (8,5,6) * (7) : 817840.6$
$6,4,7,3,5,2,8,1$	$(6,4,7,3,2,8,1) * (5) : -403308.14$	$(6,4,7,3,2,8,1) * (5) : 814125.2$
$4,6,8,5,7,2,3,1$	$(4,6,8,7) * (5,2,3) * (1) : -402800.94$	$(4,6,8,7) * (5,2,3,1) : 809706.6$
$4,6,5,8,1,3,7,2$	$(4,6,8,7) * (5,1) * (3,2) : -402785.41$	$(4,6,8,7) * (5,1) * (3,2) : 810212.5$
$5,3,2,6,8,7,1,4$	$(5,3,2,1,4) * (6,8,7) : -402450.86^{\dagger}$	$(5,3,2,1,4) * (6,8,7) : 809110.6^{*}$
$7,4,5,3,1,6,8,2$	$(7,6,8) * (4,3,1) * (5,2) : -402624.38$	$(7,4,5,6,8,2) * (3,1) : 822285.2$

Table 2: Comparison of BIC and log marginal likelihood for model selection using 6 initial random permutations under two different sample sizes. We mark the model selected by BIC with * and the model selected by log marginal likelihood with †. We use parentheses to indicate a particular variable block, and $*$ to separate the different variable blocks.

5.1.2 No Information on Variable Blocks

We now study the clustering performance of HMM-VB when there are no clear-cut variable block structures. A sample of size $10,000$ with $d = 5$ dimensions is drawn from a $10-$component GMM. We ensure that all the 10 components correspond one-to-one with 10 distinct clusters by setting mean vectors far apart. Hence, each cluster can be described by a single Gaussian component. The detailed description of the data generation mechanism is in Appendix E. Figure 3 shows the scatter plots of the generated data. Colors indicate the cluster membership. Several clusters overlap tremendously within two-dimensional space. As in the previous experiment (Section 5.1.1), we run MBW on the true model density, which identifies in total 10 clusters with cluster memberships matching precisely the true latent component identities.

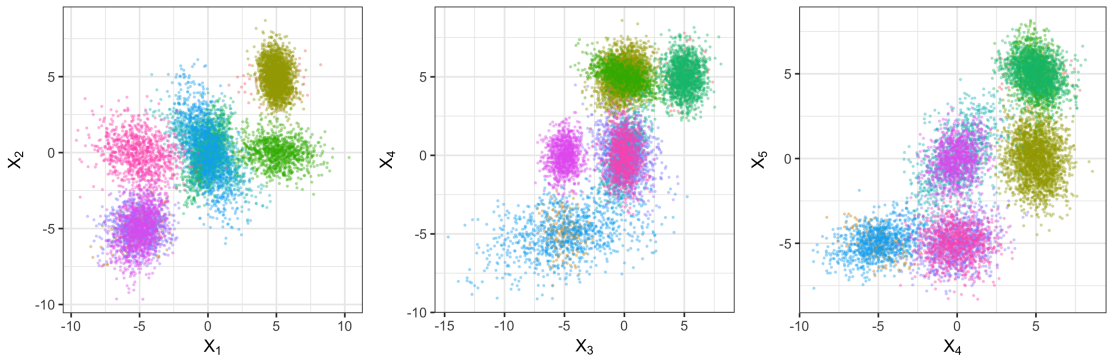


Figure 3: Pairwise scatter plots of simulated data from Section 5.1.2. Colors indicate the cluster membership.

True cluster 1 : 96	HMM-VB $\mathcal{M}_{1,2};\mathcal{M}_3$ 94(94); 96(96)	K-means 1582(71) 2327(24)	HC 766(95)	Mclust 96(96)	Modal GMM 96(96)
2 : 190	190(190); 182(180)	191(190)	538(190)	190(190)	190(190)
3 : 2304	2302(2302); 2300(2300)	2327(2303)	2133(2133) 817(171)	2304(2304)	2304(2304)
4 : 993	991(987); 985(980)	1010(992)	817(321) 766(671)	993(993)	993(993)
5 : 1509	1512(1508); 1507(1506)	1582(1508)	1517(1509)	1509(1509)	1509(1509)
6 : 1012	1002(983); 997(976) 936(22); 966(28)	908(903) 942(29)	817(325) 593(591) 1967(89)	1009(1008)	1010(1009)
7 : 982	971(965); 911(906) 46(46)	517(517) 528(463)	538(348) 1967(28) 521(520) 84(84)	986(982)	985(982)
8 : 972	979(962); 942(938) 966(20)	971(955)	1967(919) 1064(53)	971(971)	972(971)
9 : 1011	1015(1009); 1017(1008)	1024(1009)	1064(1011)	1011(1011)	1011(1011)
10 : 931	936(903); 966(910)	942(909) 971(16)	1967(931)	931(930)	930(929)
Time	32.823s	0.298s	2.636s	0.902s	1.108s

Table 3: Comparison of clustering performance among three HMM-VB models, K-means, hierarchical clustering (HC), Mclust, and modal GMM for simulation data in Section 5.1.2. The first two HMM-VB models give the same results for major clusters.

To demonstrate the robustness of HMM-VB under different variable block structures, we fit the data using different model specifications. We let model $\mathcal{M}_1$ be a HMM-VB with 5 variable blocks each containing one variable and ordered by the given indices of the

variables. The number of components for each block chosen by BIC is $M_t = 7$, $t = 1, ..., 5$. Model $\mathcal{M}_2$ is defined similarly as $\mathcal{M}_1$, but we reverse the ordering of the variables such that $x = (x_5, x_4, ..., x_1)$. Lastly, we let $\mathcal{M}_3$ be a HMM-VB with 5 variable blocks ordered by a random permutation $x = (x_1, x_3, x_4, x_2, x_5)$. Table 3 shows the clustering results for the three HMM-VB models and the other four baseline methods. The format of the results in this table is the same as that for Table 1 in the previous section. Again we pre-set the number of clusters for K-means, hierarchical clustering, and modal GMM as the true value. The model specification for Mclust is similar to the previous experiment, but we allow `Mclust` to search for up to 12 Gaussian components. For clarity of presentation, clusters that share fewer than 5 points with any true cluster are not reported in Table 3. As expected $\mathcal{M}_1$ and $\mathcal{M}_2$ yield very similar results. As discussed at the end of Section 4.1, when the ordering of the variable blocks is reversed, the GMM casted from the HMM-VB is essentially the same. Small differences may arise due to some random factors caused by initialization. We simply report results from one of the two models. Table 3 suggests that HMM-VB is quite robust to the ordering of the variables. Both K-means and hierarchical clustering fail to extract the cluster structures. As expected, both Mclust and modal GMM can accurately recover all the clusters because the data is generated according to the generic GMM model.

The search scheme for selecting the optimal variable block structure determines that only one variable block is needed (that is, the usual GMM). This is consistent with the true model. However, Table 3 shows that although the HMM-VB with five blocks is not the right model, it can still perform reasonably well for clustering. This finding suggests that HMM-VB can be a new strategy to regularize a GMM, the complexity of which is normally only controlled by the number of components or constraints on the covariance matrices.

5.1.3 Large Data Size

Since HMM-VB is motivated by single-cell data analysis, where the sample size can be up to several millions, we now study the performance of HMM-VB for large data sets with moderately high dimensions. We let dimension $d = 40$, and sample size n ranges from $100,000$, $1,000,000$ and $5,000,000$. The first 10 dimensions are generated from a 3-component GMM. The remaining 30 dimensions are generated from a 5-component GMM, where the mixture component proportions vary according to which normal component the first 10 dimensions are generated from. In addition, the covariance matrices of the last 30 dimensions are block diagonal, containing two blocks of sizes 10×10 and 20×20. Furthermore, by specifically designing the mean vectors and the transition probability matrix from the first 10 dimensions to the rest, the data generated contain 5 distinct normal components which correspond to 5 distinct clusters. The detailed description for the data generation is in Appendix E. The relative proportions for the 5 clusters are about 0.005, 0.045, 0.07, 0.18, and 0.7.

In this particular study, Mclust encounters some numerical errors when performing model selection. Because K-means and hierarchical clustering perform poorly in the previous two studies, we hereby focus on the comparison between HMM-VB and modal GMM. To fit HMM-VB, we divide the 40 dimensions into 3 blocks: $x_i^{(1)} = (x_{i,1}, ..., x_{i,10})$, $x_i^{(2)} = (x_{i,11}, ..., x_{i,20})$, $x_i^{(3)} = (x_{i,21}, ..., x_{i,40})$. According to the model specification, we let $M_1 = 3$, $M_2 = 5$, and $M_3 = 5$. To fit modal GMM, we let the total number of mixture components to be $3 \times 5 = 15$.

True cluster	HMM-VB	Modal GMM
	$n = 100,000$	
$1:510$	$510(510)$	$4,940(510)$
$2:4,430$	$4,430(4,430)$	$4,940(4,430)$
$3:7,006$	$7,006(7,006)$	$7,006(7,006)$
$4:17,995$	$17,995(17,995)$	$17,995(17,995)$
$5:70,059$	$70,059(70,059)$	$70,059(70,059)$
Time	4.1 mins	9.9 mins
	$n = 1,000,000$	
$1:5,045$	$5,045(5,045)$	$50,279(5,045)$
$2:45,234$	$45,234(45,234)$	$50,279(45,234)$
$3:69,684$	$69,684(69,684)$	$69,684(69,684)$
$4:180,789$	$180,789(180,789)$	$180,789(180,789)$
$5:699,248$	$699,248(699,248)$	$699,248(699,248)$
Time	40.59 mins	64.55 mins
	$n = 5,000,000$	
$1:25,119$	$25,119(25,119)$	$250,142(25,119)$
$2:225,023$	$225,023(225,023)$	$250,142(225,023)$
$3:351,202$	$351,202(351,202)$	$351,202(351,202)$
$4:899,498$	$899,498(899,498)$	$899,498(899,498)$
$5:3,499,158$	$3,499,158(3,499,158)$	$3,499,158(3,499,158)$
Time	56.56 mins	322.25 mins

Table 4: Comparison of clustering performance between HMM-VB and modal GMM for simulation study in Section 5.1.3.

Table 4 shows the clustering results for the two models for simulated data of three different sample sizes. HMM-VB correctly identifies all the clusters for the three data sets. However, modal GMM cannot separate the two smallest clusters. In addition, HMM-VB requires much shorter computation time than modal GMM, even though the equivalent GMM of the HMM-VB has $3 \times 5 \times 5 = 75$ components. The difference in computation time is even more dramatic when the sample size is large. In order to detect more accurately the rare clusters by modal GMM, we could increase the number of mixture components. However, the increase may have to be large causing much slower computation.

5.1.4 Clustering Variation Study

To investigate the variation of clustering results for the above three simulations, we in addition randomly generate 100 replicates from the two variable blocks model in Section 5.1.1, 100 replicates from the 10-component GMMs with randomly generated covariance matrices described in Section 5.1.2, and only 10 replicates for the model in Section 5.1.3 due to the extensive computational time caused by large data size. We use the adjusted Rand index (ARI) (Rand, 1971) to assess the similarity between the clustering result by each method and the true cluster assignments. Summaries based on the ARI for the first two simulation studies are shown in Figure 4, and that for the last simulation is provided in Table 5.

For the simulation in Section 5.1.1, the variables have a two block structure. The average ARI achieved by HMM-VB is substantially better than the other methods, and the variation of the HMM-VB results is much smaller than those by Mclust or modal GMM. For the simulation in Section 5.1.2, Model 1 is used for HMM-VB. The variable structure assumed

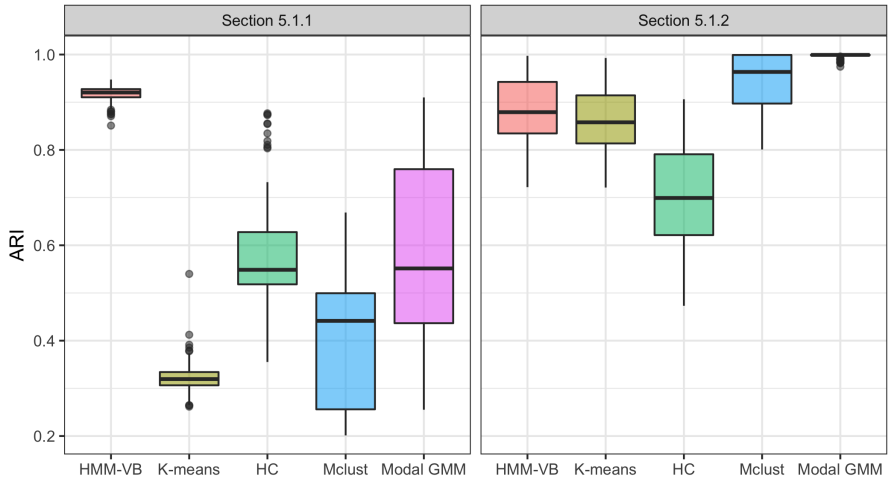


Figure 4: Boxplots of ARI based on 100 replicates generated by simulations in Section 5.1.1 and 5.1.2.

by HMM-VB is not right, while Mclust and modal GMM are based on the correct model. Nevertheless, the boxed range of ARIs by HMM-VB overlaps with that by Mclust although the average by HMM-VB is lower. The best result is obtained by modal GMM. For the third simulation in Section 5.1.3, although both HMM-VB and modal GMM yield high ARIs, the average number of clusters obtained by HMM-VB is always the same or nearly the same as the true value, while that by modal GMM is consistently lower by one. Finally, we note that ARI measures overall similarity between clustering results. Thus small clusters matter less than large clusters for ARI. If finding rare clusters is important, we should not rely only on ARI to evaluate the methods. Results shown in Table 1, 3, and 4 can be more pertinent.

Sample size	mean ARI	median ARI	sd ARI	$\bar{C}$
		HMM-VB		
$100,000$	1	1	0	5
$1,000,000$	1	1	0	5
$5,000,000$	0.9999	1	2.88×10^{-4}	4.9
		Modal GMM		
$100,000$	0.9991	0.9991	5.74×10^{-5}	4
$1,000,000$	0.9992	0.9991	2.88×10^{-4}	4.1
$5,000,000$	0.9513	0.9991	0.15	4.1

Table 5: Summaries of ARI and the mean number of clusters found ($\bar{C}$) based on 10 replicates generated by the simulation in Section 5.1.3.

5.2 Study of CyTOF Data

As a motivating example, current high-throughput flow cytometry experiments routinely measure $10 \sim 20$ parameters (cell markers/variables) on a large number of single cells. The current mass cytometry (CyTOF) can measure up to 50 parameters at a single cell level. A key first step to analyze this wealth of data is to partition the data (cells) from a blood or tissue sample into clusters based on the measured cell markers. The identified clusters are usually referred to as (cell) subsets.

Many studies adopt clustering based on mixture models to identify cell subsets objectively and automatically for single-cell data analysis (e.g., Boedigheimer and Ferbas (2008); Lo et al. (2008); Chan et al. (2008); Pyne et al. (2009); Aghaeepour et al. (2013); Lin et al. (2016)). One challenge encountered is that cell subsets of interest are typically of low frequencies (e.g., $\sim 0.01\%$ of total cells), while it is important to detect cell heterogeneity, especially very low frequency cell subsets for subsequent analysis, e.g., to understand the association between cellular heterogeneity and disease progression (Darrah et al., 2007; Lin et al., 2015a; Seshadri et al., 2015; Corey et al., 2015).

One important structure of the CyTOF data is a natural chain-like dependence among groups of variables (e.g., Roederer et al. (2004); Perfetto et al. (2004)). Biologists utilize such inherent property of the data to manually identify cell subsets by sequentially visualizing data on the groups of variables, known as manual gating, which has been extensively used in real data analysis. Specifically, the manual gating is a manual sequential process that visually demarcates cells in bounded regions (called gates) on histogram or 2-D scatter plot projections. Interestingly, the modal clustering technique has been applied to assist manual gating for cytometric data by Ray and Pyne (2012). Figure 5 provides a simplified illustration of the manual gating analysis on one flow cytometry data. Red lines are the gates, and cells within the region defined by the gates are identified as a specific cell subset. For example, to discriminate CD4+ T cells, which is one major cell subset, a sequence of subsetting procedures is performed. Two physical markers, Forward (FSC-A) and side (SSC-A) light scatter, are first used to distinguish lymphocytes from all the live cells. Lymphocytes can then be further partitioned based on 3 fluorescence parameters: CD3, CD4, and CD8 cell-surface markers. CD4+ T cells are the subclass of lymphocytes having high values of CD3 and CD4 but low value of CD8. Within CD4+ T-cell populations, additional functional markers such as intracellular makers (IL2 and IFNg) can further distinguish many functionally different CD4+ T-cell subsets. The sequence of groups of markers to use is called *gating hierarchy*, which is determined by expert knowledge. The shape and location of the gates are manually drawn. The gates are typically used to dichotomize the continuous marker expressions into binary value: positive and negative.

In manual gating analysis, the variables are grouped and ordered based on expert knowledge. By using the well-established gating hierarchy that projects cells on lower dimensions, each subplot provides a finer resolution of the cellular heterogeneity. In the sequential visualization process, any move one step further means a new group of variables are examined. Despite its wide usage, the manual gating analysis is highly subjective, time consuming, and hard to reproduce.

In this section, we study the performance of HMM-VB on a data set obtained from CyTOF experiment (Becher et al., 2014). The particular data set that we analyze here is

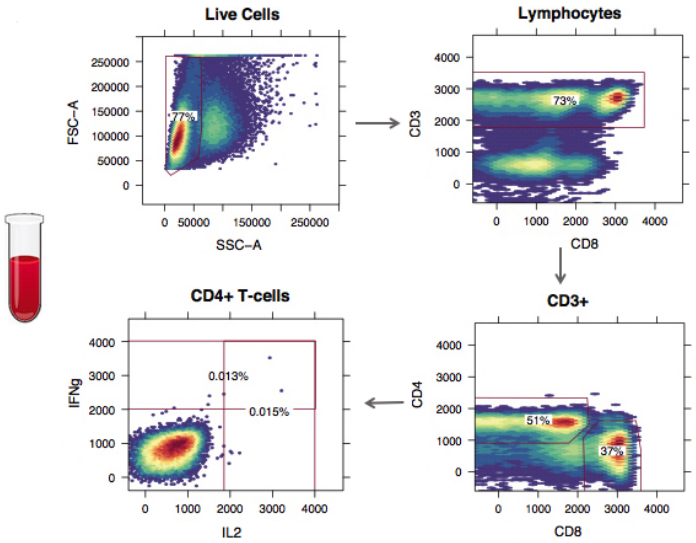


Figure 5: A simplified example of cell subsets identification by manual gating analysis. The flow cytometry measurements on single cells from a blood sample are shown using 4 heat maps of 2-D scatter plots projected on different dimensions (markers). The red lines on each subplots are called gates. Cells within the red lines are the subset of interest, which is subsetted and projected on the next subplot in the sequence. The percentages are the frequencies of the identified cell subsets relative to the total number of cells.

from mouse lung sample obtained from three C57BI6 wild-type mice and three Csf2rb$^{-/-}$ mice, which in total contains $46,204$ single cells with 39 measured cell markers. According to the gating hierarchy provided in Becher et al. (2014), it defines roughly 11 variable blocks, with maximum block size 8 and minimum block size 1. Becher et al. (2014) performed automated clustering on a projected (latent) $2-$dimensional space by first using nonlinear dimension reduction technique on the original $39-$dimensional space. However, it has been studied, e.g. Lin et al. (2015b), that dimension reduction generates a "cluttered" display, which can prevent density estimation from accurately representing low probability regions.

We first standardized the data to compensate for the nearly singular covariance matrix when a single Gaussian is fit despite the moderate dimension. This factor prevents the direct use of GMM. We note that HMM-VB encounters no difficulty in fitting the original data because of the smaller dimensions of the individual variable blocks. In order to compare modal GMM and HMM-VB, we work on the standardized data through the rest of the section.

We first fit the data using HMM-VB. There are 11 variable blocks. For the ith variable blocks with dimension lower than 4, we set the corresponding number of mixture components $M_i = 5$. For the jth variable blocks with dimension between 5 and 7, we set the number of components $M_j = 10$. For the other variable blocks, we set the number of components to 15. After modal clustering, this model results in 194 clusters of size greater than 5. The average

CPU time per model fitting is 11.4 min. We then fit the data using GMM with $M = 500$. On the same iMac, the model fitting takes 284.2 min, 25 times longer than does HMM-VB. Modal clustering results in 232 clusters of size greater than 5.

We compare the performance of the two models for clustering a relatively low probability region, which is shown in Figure 6. Specifically, Figure 6 compares the finer cellular compositions of one well separated region, which is visualized on two latent dimensions obtained from a dimension reduction technique applied to the original 39 dimensions, as provided by the result of Becher et al. (2014). HMM-VB (right subplot) uses 18 clusters to define this particular region. GMM (left subplot) uses 40 clusters to define the same region. However, some of the clusters include cells that are visually far away from the particular region. This result indicates that GMM has difficulty in estimating the structure of this moderately high-dimensional data.

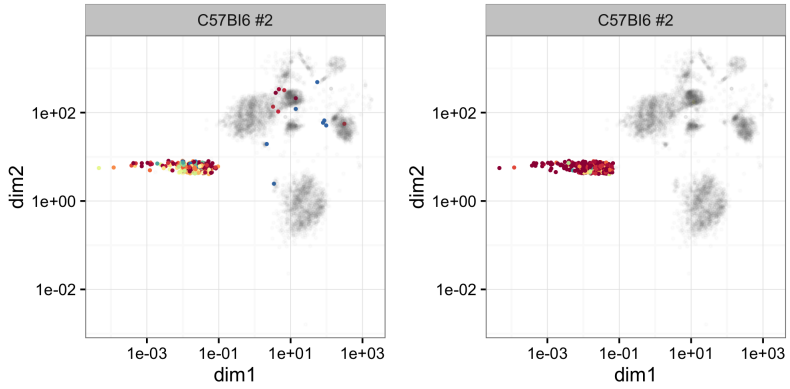


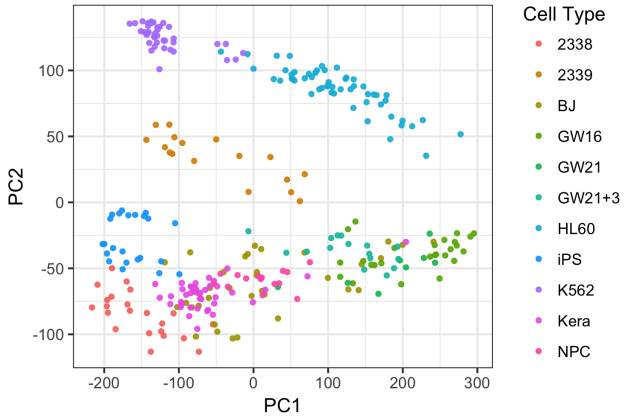
Figure 6: Left: GMM clustering analysis for a particular data region of one selected mouse. Right: The corresponding HMM-VB clustering analysis. The data is shown in grey. Clustering results are in different colors with one color defines one cluster.

5.3 Examples of Very High Dimensional Data

We apply HMM-VB to two data sets with dimension $d \gg n$.

5.3.1 Single-Cell Genomics Data

Current single-cell genomics technologies can routinely generate transcriptomics data at the single-cell level. Similar to the cytometry data, a key first analysis step is to identify cellular heterogeneity among single cells. The challenge is to cluster high dimensional data with substantially more variables (genes) than objects (cells). Existing clustering methods fall roughly into two categories. The first category is based largely on pairwise similarity measures among single cells (e.g., Guo et al. (2015); Wang et al. (2017)). The second category applies clustering, e.g., K-means and mixture models, in a reduced dimension space (typically 2 or 3 dimensions) obtained by principal component analysis or other nonlinear dimension reduction techniques (e.g., Satija et al. (2015); Žurauskienė and Yau (2016); Kiselev et al.



Cell type: size	HMM-VB
2338: 22	22(22)
2339: 17	17(17)
HL60: 54	54(54)
iPS: 23	23(23)
K562: 42	42(42)
NPC: 15	15(15)
BJ: 37	29(29); 9(8)
GW16: 26	28(15); 22(11)
GW21: 8	28(5); 22(3)
GW21+3: 16	28(8); 22(8)
Kera: 40	39(39); 9(1)

Figure 7: (Left): Visualization of the single cell RNA-seq data based on the first two principal components. Colors indicate the 11 cell types. (Right): Clustering performance for HMM-VB compared with the 11 cell types.

(2017)). Here, we study the performance of HMM-VB on a single-cell RNA-seq data set for 300 cells whose transcriptional measurements were taken across 8,686 genes. The data set is obtained from the Github repository: `https://github.com/JustinaZ/pcaReduce` provided by Žurauskienė and Yau (2016). The data were derived from 11 cell types: K562 – myeloid (chronic leukemia); HL60 – myeloid (acute leukemia); CRL-2339 – lymphoblastoid; iPS – pluripotent; CRL-2338 – epithelial; BJ – fibroblast (from human foreskin); Kera – foreskin keratinocyte; NPC – neural progenitor cells; GW(16, 21, 21+3) – fetal cortex at gestational week (16,21, 21+3 weeks). Figure 7 (Left) shows the data projected on the first two principal components for visualization. It is clear that the 11 cell types in the two-dimensional principal component space cannot be separated into distinct groups. Žurauskienė and Yau (2016) perform clustering analysis in the subspace spanned by the first 30 principal components, and demonstrate the superiority of pcaReduce in terms of clustering accuracy over some competing methods. However, pcaReduce can only correctly identify 4 cell types which are HL60, CRL-2339, CRL-2338, and NPC.

We first identified the top 500 highly variable genes based on their sample variances after applying log transformation. We then fit this reduced data of size 300×500 using HMM-VB. The genes are divided into 5 variable blocks of equal dimension, and each block is set to have 40 mixture components. We also tried other model specifications, and they yielded similar clustering results. HMM-VB identifies 12 cell clusters. More importantly, HMM-VB not only correctly identifies the 4 cell types that were also discovered by pcaReduce, it also finds correctly 2 other cell types: iPS and K562. Moreover, pcaReduce groups the three cell types GW(16, 21, 21+3) into one cluster. HMM-VB partitions these three cell types into two clusters, which could be visually supported by Figure 7 (Left). Figure 7 (Right) provides the details of the clustering performance.

Because the dimension is higher than the data size for this data set, we constrained the covariance matrix for each variable block to be diagonal. This constraint can be offset by having more components for a good fit of data. We found that when the dimension

of the variable block is at similar order of the data size, without any constraint, the estimated covariance matrix is sometimes singular or ill-conditioned even if a single Gaussian distribution is assumed. We also found that the number of modes in a HMM-VB can become quite large if the number of variable blocks is large, while the number of modes is much more stable with the increase of components in each variable block. In summary, as practical guidance on the usage of HMM-VB for very high dimensional data ($d \gg n$), we recommend the use of diagonal covariance and the avoidance of too many variable blocks.

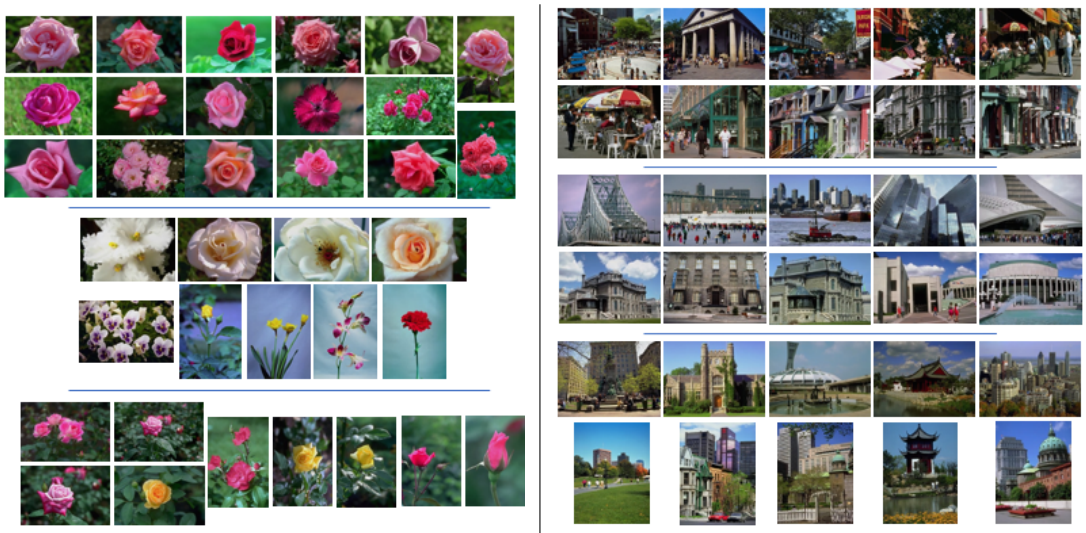
5.3.2 Image Data Analysis

We take a collection of general-purpose photograph images and represent each image by a high dimensional vector. First, the Red, Blue, and Green values of each pixel in an image are converted to the LUV color space. Second, each image is divided into 50×50 blocks and the average LUV values of all the pixels in each block are used to characterize this block. The whole image is represented by a stacked vector of the LUV values of the 2500 blocks. Thus the dimension of the data is $d = 7500$. We put the LUV values of every image block into one variable block. There are thus 2500 variable blocks each of dimension 3. These variable blocks are ordered according to a zig-zag row by row scan, that is, to scan from left to right in the first row, then right to left in the second, so on so forth. Clearly, because each variable block corresponds to a particular location in the image plane, the clustering result based on HMM-VB will be sensitive to the position of objects in an image. For many applications in multimedia retrieval or computer vision, such sensitivity is undesirable. However, our purpose here is not to promote our experiment as a standard for clustering images, but to show the viability of our method when $d \gg n$. We acknowledge that for any particular application, it may be necessary to perform certain pre-processing, such as extracting sub-images based on segmentation, etc. An in-depth evaluation from an image analysis perspective is beyond the scope of this paper.

We estimated the HMM-VB on the aforementioned variable block structure, assuming 5 mixture components for each block. Since the dimension of the variable blocks is small, we did not constrain the covariance matrices to be diagonal in this analysis. We estimated a model for a set of 200 photos of flowers, a set of 200 photos of city scenes, and a set of 2400 photos of various themes. On a Mac with Intel Core i5 3.5GHz/16GB Memory, the time for training is respectively 263, 326, 2118 seconds for the three data sets, and the time for solving the modes for all the images in a data set is respectively 118, 141, 1930 seconds. We found that due to the very high dimension, if we apply the same stringent criterion of identical modes, then nearly every image becomes a single cluster. We obtain larger clusters when the criterion is relaxed. We show in Figure 8 example clusters generated.

6. Discussion

We have developed a novel mixture model, namely, HMM-VB, with the goal of automated clustering of large-scale and high-dimensional data that contain rare clusters. One key feature of HMM-VB is its ability to leverage sequential dependence among groups of variables for more effective clustering, especially in identifying rare clusters that are almost undetectable by existing mixture modeling approaches. Technically, our clustering method integrates two new algorithms, one for estimating HMM-VB and the other for performing modal



(a) Flower (b) City scenes

Figure 8: Example images in the clusters generated for two data sets. The clusters are separated by the horizontal bars.

clustering, both necessary for making the new model a practical tool in large-scale data analysis. Because the number of mixture components grows exponentially with the number of variable groups, existing EM and MEM algorithms for estimation and mode identification have intractable exponential complexity. We have derived and implemented algorithms with linear complexity in the number of variable blocks for both tasks.

We have so far declared clusters based on association with identical modes. Due to the huge number of mixture components in the GMM casted from HMM-VB, many tiny clusters can be generated. The total number of points in those clusters can be a small fraction of the entire data. In this situation, more sophisticated methods that measure the separability of these tiny clusters and other larger ones can be used for clustering (e.g., Lee and Li (2012)). As another direction of future work, we can test the capability of the variable block search algorithm to reveal hidden dependence structure among the variables or to help validate existing gating hierarchy.

Acknowledgments

We would like to acknowledge support for this work from the National Science Foundation under grant ECCS-1462230 and DMS-1521092. We also thank the reviewers and the AE wholeheartedly for many detailed and constructive suggestions.

Appendix A

We present here the conventional HMM and the Baum-Welch (BW) algorithm for estimation. Consider sequential data $\mathbf{x} = \{x_1, x_2, ..., x_t, ..., x_T\}$, with $x_t \in \mathcal{R}^d$. Same as in the mixture model, assume there is an underlying state s_t associated with every x_t, for $t = 1, ..., T$. The underlying state is the counterpart of the mixture component identity in GMM. The state $s_t \in \mathcal{S} = \{1, 2, ..., M\}$, where M is the number of states. Let the set of all possible state sequences be $\bar{\mathcal{S}}$, that is, the set of T-tuples on $\mathcal{S}$. $|\bar{\mathcal{S}}| = M^T$. Denote $\mathbf{s} = \{s_1, ..., s_T\} \in \bar{\mathcal{S}}$. The basic assumptions of a HMM are:

1. The underlying states $\{s_1, s_2, ..., s_T\}$ follow a Markov chain. The Markov chain is usually time invariant with transition probability matrix $A = (a_{k,l})$, where $a_{k,l} = P(s_{t+1} = l \mid s_t = k)$, $k, l \in \mathcal{S}$. Denote the initial probabilities of states by $\pi_k = P(s_1 = k)$, $k \in \mathcal{S}$.

2. Given the hidden state s_t, the observation x_t is conditionally independent from $s_{t'}$ and $x_{t'}$ for any $t' \neq t$, and the distribution of x_t given s_t depends on s_t, but not on t. Denote the conditional density of $P(x_t|s_t = k)$ by $\varphi_k(x_t)$. In particular, for Gaussian HMM, $\varphi_k(x_t) = \phi(x_t|\mu_k, \Sigma_k)$.

In summary,

$$P(\mathbf{x}, \mathbf{s}) = P(\mathbf{s})P(\mathbf{x} \mid \mathbf{s}) = \pi_{s_1}\varphi_{s_1}(x_1)a_{s_1,s_2}\varphi_{s_2}(x_2)\cdots a_{s_{T-1},s_T}\varphi_{s_T}(x_T),$$
$$P(\mathbf{x}) = \sum_{\mathbf{s}\in\bar{\mathcal{S}}} P(\mathbf{s})P(\mathbf{x} \mid \mathbf{s}) = \sum_{\mathbf{s}\in\bar{\mathcal{S}}} \pi_{s_1}\varphi_{s_1}(x_1)a_{s_1,s_2}\varphi_{s_2}(x_2)\cdots a_{s_{T-1},s_T}\varphi_{s_T}(x_T).$$

The parameters to be estimated in a HMM are the transition probabilities: $a_{k,l}$, $k, l = 1, ..., M$, the initial probabilities: π_k, $k = 1, ..., M$, and μ_k, Σ_k for each state $k = 1, ..., M$.

Under a set of parameters, let $L_k(t)$ be the conditional probability of being in state k at position t given the entire observed sequence $\mathbf{x} = \{x_1, x_2, ..., x_T\}$. Let $I(\cdot)$ be the indicator function that equals 1 when the argument is true and 0 otherwise. Then

$$L_k(t) = P(s_t = k|\mathbf{x}) = \sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{x})I(s_t = k), \quad k \in \mathcal{S}. \tag{8}$$

Let $H_{k,l}(t)$ be the conditional probability of being in state k at position t and being in state l at position $t+1$, i.e., seeing a transition from k to l at t, given the entire observed sequence $\mathbf{x}$. Then

$$\begin{aligned} H_{k,l}(t) &= P(s_t = k, s_{t+1} = l|\mathbf{x}) \\ &= \sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{x})I(s_t = k)I(s_{t+1} = l), \quad k, l \in \mathcal{S}. \end{aligned} \tag{9}$$

Note that $L_k(t) = \sum_{l=1}^{M} H_{k,l}(t)$, $\sum_{k=1}^{M} L_k(t) = 1$. Since $\mathbf{s} \in \bar{\mathcal{S}}$ and the $|\bar{\mathcal{S}}| = M^T$, it is infeasible to compute $L_k(t)$ and $H_{k,l}(t)$ by the above equations directly. As part of the BW algorithm, the *forward-backward* algorithm is used to compute $L_k(t)$ and $H_{k,l}(t)$ efficiently. The amount of computation needed is at the order of M^2T; and the memory required is at the order of MT.

Define the forward probability $\alpha_k(t)$ as the joint probability of observing the first t vectors x_τ, $\tau = 1, ..., t$, and being in state k at time t:

$$\alpha_k(t) = P(x_1, x_2, ..., x_t, s_t = k).$$

This probability can be evaluated by the following recursive formula:

$$\begin{aligned} \alpha_k(1) &= \pi_k \varphi_k(x_1), \quad 1 \le k \le M, \\ \alpha_k(t) &= \varphi_k(x_t) \sum_{l=1}^{M} \alpha_l(t-1) a_{l,k}, \quad 1 < t \le T,\ 1 \le k \le M. \end{aligned}$$

Define the backward probability $\beta_k(t)$ as the conditional probability of observing the vectors after time t, x_τ, $\tau = t+1, ..., T$, given the state at time t is k.

$$\begin{aligned} \beta_k(t) &= P(x_{t+1}, ..., x_T \mid s_t = k), \quad 1 \le t \le T-1, \\ \beta_k(T) &= 1, \quad \text{for all } k. \end{aligned}$$

As with the forward probability, the backward probability can be evaluated using the following recursion:

$$\begin{aligned} \beta_k(T) &= 1, \\ \beta_k(t) &= \sum_{l=1}^{M} a_{k,l} \varphi_l(x_{t+1}) \beta_l(t+1), \quad 1 \le t < T. \end{aligned}$$

The probabilities $L_k(t)$ and $H_{k,l}(t)$ are solved by

$$\begin{aligned} L_k(t) &= P(s_t = k \mid \mathbf{x}) = \frac{P(\mathbf{x}, s_t = k)}{P(\mathbf{x})} = \frac{1}{P(\mathbf{x})} \alpha_k(t) \beta_k(t), \\ H_{k,l}(t) &= P(s_t = k, s_{t+1} = l \mid \mathbf{x}) = \frac{P(\mathbf{x}, s_t = k, s_{t+1} = l)}{P(\mathbf{x})} \\ &= \frac{1}{P(\mathbf{x})} \alpha_k(t) a_{k,l} \varphi_l(x_{t+1}) \beta_l(t+1), \end{aligned}$$

where $P(\mathbf{x}) = \sum_{k=1}^{M} \alpha_k(t) \beta_k(t)$.

For notational brevity, we assume all the sequences are of length T. The extension to sequences of different lengths is trivial. Denote the ith sequence by $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, ..., x_{i,T}\}$, $i = 1, ..., n$. In each iteration, we compute the forward and backward probabilities for each sequence separately as previously described. We also compute $L_k(t)$ and $H_{k,l}(t)$ separately for each sequence. As a general pattern of notations, we put a superscript (i) to indicate the quantities for the ith sequence.

1. Compute the forward and backward probabilities $\alpha_k^{(i)}(t)$, $\beta_k^{(i)}(t)$, $k = 1, ..., M$, $t = 1, ..., T$, $i = 1, ..., n$, under the current set of parameters.

$$\begin{aligned}
\alpha_k^{(i)}(1) &= \pi_k \varphi_k(x_{i,1}), \quad 1 \le k \le M,\ 1 \le i \le n, \\
\alpha_k^{(i)}(t) &= \varphi_k(x_{i,t}) \sum_{l=1}^{M} \alpha_l^{(i)}(t-1) a_{l,k}, \quad 1 < t \le T,\ 1 \le k \le M,\ 1 \le i \le n, \\
\beta_k^{(i)}(T) &= 1, \quad 1 \le k \le M,\ 1 \le i \le n, \\
\beta_k^{(i)}(t) &= \sum_{l=1}^{M} a_{k,l} \varphi_l(x_{i,t+1}) \beta_l^{(i)}(t+1), \quad 1 \le t < T,\ 1 \le k \le M,\ 1 \le i \le n.
\end{aligned}$$

2. Compute $L_k^{(i)}(t)$, $H_{k,l}^{(i)}(t)$ using $\alpha_k^{(i)}(t)$, $\beta_k^{(i)}(t)$. Let $P(\mathbf{x}_i) = \sum_{k=1}^{M} \alpha_k^{(i)}(1)\beta_k^{(i)}(1)$.

$$\begin{aligned}
L_k^{(i)}(t) &= \frac{1}{P(\mathbf{x}_i)} \alpha_k^{(i)}(t) \beta_k^{(i)}(t), \\
H_{k,l}^{(i)}(t) &= \frac{1}{P(\mathbf{x}_i)} \alpha_k^{(i)}(t) a_{k,l} \varphi_l(x_{i,t+1}) \beta_l^{(i)}(t+1).
\end{aligned}$$

3. Update the parameters using $L_k^{(i)}(t)$, $H_{k,l}^{(i)}(t)$.

$$\begin{aligned}
\mu_k &= \frac{\sum_{i=1}^{n} \sum_{t=1}^{T} L_k^{(i)}(t) x_{i,t}}{\sum_{i=1}^{n} \sum_{t=1}^{T} L_k^{(i)}(t)}, \\
\Sigma_k &= \frac{\sum_{i=1}^{n} \sum_{t=1}^{T} L_k^{(i)}(t)(x_{i,t} - \mu_k)(x_{i,t} - \mu_k)'}{\sum_{i=1}^{n} \sum_{t=1}^{T} L_k^{(i)}(t)}, \\
a_{k,l} &= \frac{\sum_{i=1}^{n} \sum_{t=1}^{T-1} H_{k,l}^{(i)}(t)}{\sum_{i=1}^{n} \sum_{t=1}^{T-1} L_k^{(i)}(t)}.
\end{aligned}$$

Appendix B

We prove the results in Section 4.1.

The proof for Theorem 2

Proof Because $\mathcal{M}$ and $\mathcal{M}'$ define the same density function, their marginal densities on any block $X^{(t)}$ are also identical. Since $\Theta^{(t)}$ is non-redundant for any t, by Lemma 1, we have $\mathcal{M}'_{X^{(t)}} = \mathcal{O}_t(\mathcal{M}_{X^{(t)}})$, where $\mathcal{O}_t$ is the permutation on the mixture components of $X^{(t)}$, for $t = 1, ..., T$. The permutation $\mathcal{O}_t$ is unique because $\Theta^{(t)}$ is distinct.

Again by Lemma 1, if we remove all the components in $\mathcal{M}$ and $\mathcal{M}'$ with zero priors, that is, $P(s_1 = i_1, s_2 = i_2, ..., s_t = i_T) = 0$ and $P(s'_1 = i'_1, s'_2 = i'_2, ..., s'_T = i'_T) = 0$, the two models have the same number of components, and the Gaussian component parameters as well as the positive priors in $\mathcal{M}$ and $\mathcal{M}'$ are identical up to permutation. Let the permutation

that matches the two models be $\mathcal{O}^*$. Note that $\mathcal{O}^*$ is only defined on T-tuples $(i_1, ..., i_T)$ such that $P(s_1 = i_1, s_2 = i_2, ..., s_t = i_T) > 0$.

Let $\mathcal{O}_{1:T} = \mathcal{O}_1 \times \mathcal{O}_2 \cdots \mathcal{O}_T$. It is obvious that $\Theta' = \mathcal{O}_{1:T}(\Theta)$. Consider a T-tuple with $P(s_1 = i_1, ..., s_T = i_T) > 0$. Let $(i_1^*, ..., i_T^*) = \mathcal{O}^*(i_1, ..., i_T)$. By Lemma 1,

$$\mu_{i_t}^{(t)} = \mu'^{(t)}_{i_t^*}, \quad \Sigma_{i_t}^{(t)} = \Sigma'^{(t)}_{i_t^*},$$

$$\mu_{i_t}^{(t)} = \mu'^{(t)}_{\mathcal{O}_t(i_t)}, \quad \Sigma_{i_t}^{(t)} = \Sigma'^{(t)}_{\mathcal{O}_t(i_t)}.$$

Because $\Theta'^{(t)}$ is distinct, $i_t^* = \mathcal{O}_t(i_t)$, for $t = 1, ..., T$. Thus we have shown that if $P(s_1 = i_1, ..., s_T = i_T) > 0$, $\mathcal{O}^*(i_1, ..., i_T) = \mathcal{O}_{1:T}(i_1, ..., i_T)$. Because $\mathcal{M}$ and $\mathcal{M}'$ have the same number of components with positive priors, if $P(i_1, ..., i_T) = 0$, we can extend the definition of $\mathcal{O}^*$ to these T-tuples by simply setting it to $\mathcal{O}_{1:T}$. Thus we have shown $\mathcal{M}' = \mathcal{O}_{1:T}(\mathcal{M})$. ■

The proof for Lemma 3

Proof Because $\Theta^{(t)}$ is non-redundant for $t = 1, ..., T$, and $\pi_{i_1} = P(s_1 = i_1)$, we have $\pi_{i_1} > 0$, $\forall i_1 \in \{1, ..., M_1\}$. We also have $P(s_t = i_t) > 0$, $\forall i_t \in \{1, ..., M_t\}$, $t = 2, ..., T$. Since $P(s_t = i_t) = \sum_{i_{t-1}=1}^{M_{t-1}} P(s_{t-1} = i_{t-1}) a_{i_{t-1}, i_t}^{(t-1)}$ and $P(s_t = i_t) > 0$, there exist at least one i_{t-1} such that $a_{i_{t-1}, i_t}^{(t-1)} > 0$.

The reverse can be proved by induction. Because $\pi_{i_1} = P(s_1 = i_1)$ and $\pi_{i_1} > 0$, we have $P(s_1 = i_1) > 0$, $\forall i_1 \in \{1, ..., M_1\}$. Assume $P(s_{t-1} = i_{t-1}) > 0$, $\forall i_{t-1} \in \{1, ..., M_{t-1}\}$, then $P(s_t = i_t) = \sum_{i_{t-1}=1}^{M_{t-1}} P(s_{t-1} = i_{t-1}) a_{i_{t-1}, i_t}^{(t-1)} > 0$ because at least for one i_{t-1}, the summand $P(s_{t-1} = i_{t-1}) a_{i_{t-1}, i_t}^{(t-1)} > 0$. ■

The proof for Corollary 4

Proof Because HMM-VB on partition $\mathcal{P}$ is a lattice GMM on the same partition, by Theorem 2, there exists a unique permutation $\mathcal{O}_t$ for each variable block $X^{(t)}$, $t = 1, ..., T$ such that $\mathbb{M}' = \mathcal{O}_{1:T}(\mathbb{M})$ and $\Theta' = \mathcal{O}_{1:T}(\Theta)$.

We only need to show that under permutation $\mathcal{O}_{1:T}$, π_{i_1}, $a_{i_{t-1}, i_t}^{(t-1)}$, $i_1 = 1, ..., M_1$, $i_t = 1, ..., M_t$, $t = 2, ..., T$ are also identical. By Theorem 2, $\Pi(s_1, ..., s_T)$ and $\Pi'(s'_1, ..., s'_T)$ are identical up to permutation $\mathcal{O}_{1:T}$. The same is true for any marginal $\Pi(s_t)$ or $\Pi(s_{t-1}, s_t)$. That is

$$P(s_t = i_t) = P(s'_t = \mathcal{O}_t(i_t)), \ \ \forall t = 1, ..., T, \ \forall i_t = 1, ..., M_t.$$

Since π_{i_1}, $i_1 = 1, ..., M_1$, is the marginal $\Pi(s_1)$, it is identical to π'_{i_1} up to permutation $\mathcal{O}_1$.

Under permutation $\mathcal{O}_{t-1:t}$, we have

$$P(s'_{t-1} = \mathcal{O}_{t-1}(i_{t-1}), s'_t = \mathcal{O}_t(i_t)) = P(s_{t-1} = i_{t-1}, s_t = i_t).$$

Also,

$$P(s'_{t-1} = \mathcal{O}_{t-1}(i_{t-1}), s'_t = \mathcal{O}_t(i_t)) = P(s'_{t-1} = \mathcal{O}_{t-1}(i_{t-1})) a'^{(t-1)}_{\mathcal{O}_{t-1}(i_{t-1}), \mathcal{O}_t(i_t)},$$

$$P(s_{t-1}=i_{t-1}, s_t=i_t) = P(s_{t-1}=i_{t-1}) a^{(t-1)}_{i_{t-1},i_t}.$$

Because Θ and Θ' are non-redundant, $P(s_{t-1}=i_{t-1})>0$, $\forall i_{t-1}$, and similarly $P(s'_{t-1}=i'_{t-1})>0$, $\forall i'_{t-1}$, we thus have

$$a'^{(t-1)}_{\mathcal{O}_{t-1}(i_{t-1}),\mathcal{O}_t(i_t)} = a^{(t-1)}(i_{t-1}, i_t),$$
$$\forall t = 2, ..., T,\ \forall i_{t-1} = 1, ..., M_{t-1},\ \forall i_t = 1, ..., M_t.$$

■

The proof for Theorem 5

Proof If all the variables are put in one block, we can remove any component with zero prior. Hence, the partition containing only one block is always a tight partition. Thus a tight partition always exists. Because there are only finitely many partitions, the maximum partition must exist. We have proved the existence of a maximum partition for any GMM.

We now prove the uniqueness. Suppose $\mathcal{P}_1$ and $\mathcal{P}_2$ are both maximum partitions and $\mathcal{P}_1 \neq \mathcal{P}_2$. Let $\mathcal{P}_i = \{X^{(i,1)}, X^{(i,2)}, ..., X^{(i,T_i)}\}$, $i = 1, 2$. Let the kth Gaussian component parameter for block $X^{(i,j)}$ in partition $\mathcal{P}_i$ be $\theta_k^{(i,j)}$. For any sub-vector of $X^{(i,j)}$ that contains some of the variables in $X^{(i,j)}$, which for instance is denoted by $X^{(i,j,l)}$, we denote the projection of $\theta_k^{(i,j)}$ on the sub-vector by $\theta_k^{(i,j)}(X^{(i,j,l)})$. Specifically, the mean vector projected on $X^{(i,j,l)}$ is a sub-vector of the mean vector $\mu_k^{(i,j)}$ keeping the corresponding dimensions in $X^{(i,j,l)}$, and the covariance matrix projected on $X^{(i,j,l)}$ is a sub-matrix of the covariance $\Sigma_k^{(i,j)}$ keeping the entries for the covariances between the dimensions in $X^{(i,j,l)}$.

By definition of maximum partition, $\mathcal{P}_1 \not> \mathcal{P}_2$ and vice versa $\mathcal{P}_1 \not< \mathcal{P}_2$. Therefore there exists a $X^{(1,j)}$ such that the variables in this block do not belong to a single block in $\mathcal{P}_2$. Without loss of generality and for brevity of notation, assume $X^{(1,1)}$ is such a block and its variables fall into K blocks in $\mathcal{P}_2$. We divide block $X^{(1,1)}$ into K sub-blocks such that each sub-block belongs to a distinct block in $\mathcal{P}_2$. Denote the K sub-blocks by $X^{(1,1,k)}$. $X^{(1,1)} = \{X^{(1,1,1)}, X^{(1,1,2)}, ..., X^{(1,1,K)}\}$. Again without loss of generality assume sub-block $X^{(1,1,k)} \subset X^{(2,k)}$, $k = 1, ..., K$.

Suppose the number of components for each block in $\mathcal{P}_1$ is M_1, ..., M_{T_1}, and the number of components for each block in $\mathcal{P}_2$ is M'_1, ..., M'_{T_2}. Without loss of generality, we can assume the lattice Θ and Θ' corresponding to $\mathcal{P}_1$ and $\mathcal{P}_2$ are non-redundant. The fact all the component parameters in Θ (or Θ') exist is guaranteed already by $\mathcal{P}_1$ (or $\mathcal{P}_2$) being a tight partition. If any $\Theta^{(t)}$ is not distinct (containing identical components), we can always combine those components without affecting the partition $\mathcal{P}_1$. Therefore we can assume Θ and Θ' are non-redundant without the loss of generality.

Let the underlying component identity be s_j, $j = 1, ..., T_1$ for $\mathcal{P}_1$ and s'_j, $j = 1, ..., T_2$ for $\mathcal{P}_2$. Because the GMM is a lattice-GMM on both $\mathcal{P}_1$ and $\mathcal{P}_2$, the marginal density of variable block $X^{(1,1)}$ according to partition $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively is given by

$$f_{X^{(1,1)}}(x^{(1,1)}) = \sum_{i_1=1}^{M_1} P(s_1 = i_1) \phi(x^{(1,1)} \mid \theta_{i_1}^{(1,1)}), \tag{10}$$

$$f_{X^{(1,1)}}(x^{(1,1)}) = \sum_{i'_1=1}^{M'_1} \cdots \sum_{i'_K=1}^{M'_K} P(s'_1 = i'_1, ..., s'_K = i'_K) \prod_{j=1}^{K} \phi(x^{(1,1,j)} \mid \theta^{(2,j)}_{i'_j}(X^{(1,1,j)})). \quad (11)$$

Because both $\mathcal{P}_1$ and $\mathcal{P}_2$ are tight partitions, the priors in Eqs. (10) and (11) are all positive. The components in mixture (10) are distinct by the non-redundancy of Θ, while the components in (11) may not because $X^{(1,1,j)}$'s are sub-vectors of the blocks in $\mathcal{P}_2$. However, if they are not distinct, we can combine components without losing the positive priors on the components. To be strict, suppose we combined components and mixture model (11) becomes the following model containing distinct components:

$$f_{X^{(1,1)}}(x^{(1,1)}) = \sum_{i''_1=1}^{M''_1} \cdots \sum_{i''_K=1}^{M''_K} P(s''_1 = i''_1, ..., s''_K = i''_K) \prod_{j=1}^{K} \phi(x^{(1,1,j)} \mid \theta^{(2,j)}_{i''_j}(X^{(1,1,j)})). \quad (12)$$

By the identifiability of GMM (Lemma 1), the number of terms in Eq. (12), $M''_1 \times M''_2 \cdots \times M''_K = M_1$ and for each i_1, there is a unique K-tuple $(i''_1, i''_2, ..., i''_K)$ such that

$$\phi(x^{(1,1)} \mid \theta^{(1,1)}_{i_1}) = \prod_{j=1}^{K} \phi(x^{(1,1,j)} \mid \theta^{(2,j)}_{i''_j}(X^{(1,1,j)})). \quad (13)$$

The density of the full-dimensional X according to $\mathcal{P}_1$ is

$$f_{X^{(1,1)}}(x^{(1,1)}) = \sum_{i_1=1}^{M_1} \cdots \sum_{i_T=1}^{M_T} \pi(i_1, i_2, ..., i_T) \prod_{j=1}^{T} \phi(x^{(1,j)} \mid \theta^{(1,j)}_{i_j}).$$

Because the map from i_1 to $(i''_1, i''_2, ..., i''_K)$ is bijective, we can define a set of prior

$$\tilde{\pi}(i''_1, i''_2, ..., i''_K, i_2, i_3, ..., i_T) = \pi(i_1, i_2, ..., i_T).$$

By Eq. (13), we can write

$$\begin{aligned} f_X(x) \;=\; & \sum_{i''_1=1}^{M''_1} \cdots \sum_{i''_K=1}^{M''_K} \sum_{i_2=1}^{M_2} \cdots \sum_{i_T=1}^{M_T} \tilde{\pi}(i''_1, ..., i''_K, i_2, ..., i_T) \cdot \\ & \prod_{l=1}^{K} \phi(x^{(1,1,l)} \mid \theta^{(2,l)}_{i''_l}(X^{(1,1,l)})) \prod_{j=2}^{T} \phi(x^{(1,j)} \mid \theta^{(1,j)}_{i_j}). \end{aligned} \quad (14)$$

Eq. (14) shows that the GMM is a lattice-GMM on the partition

$$\mathcal{P}_3 = \{X^{(1,1,1)}, X^{(1,1,2)}, \cdots, X^{(1,1,K)}, X^{(1,2)}, ..., X^{(1,T)}\},$$

and $\mathcal{P}_3$ is a tight partition. Clearly, $\mathcal{P}_3 \succ \mathcal{P}_1$ and $\mathcal{P}_3 \neq \mathcal{P}_1$. This contradicts the assumption that $\mathcal{P}_1$ is a maximum partition. We have thus proved the maximum partition is unique.

■

Appendix C

In this section, we prove Theorem 6.

Proof We follow the idea of the proof of Theorem 2 in Maugis et al. (2009). According to the definition, $\hat{\mathbf{C}} = \text{argmin}_{\mathbf{C}\in\mathcal{C}} BIC(\mathbf{C})$, where $\mathcal{C}$ is the set of all possible partitions under $T = T^0$,

$$BIC(\mathbf{C}) = -2\sum_{i=1}^{n} \log(f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C})})) + \lambda(\mathbf{C})\log(n),$$

where $\lambda(\mathbf{C})$ is the number of model parameters for variable blocks $\mathbf{C}$. Thus

$$\begin{aligned} P(\hat{\mathbf{C}} = \mathbf{C}^0) &= P(BIC(\mathbf{C}^0) \leq BIC(\mathbf{C}), \forall \mathbf{C} \in \mathcal{C}) \\ &= P(BIC(\mathbf{C}) - BIC(\mathbf{C}^0) \geq 0, \forall \mathbf{C} \in \mathcal{C}). \end{aligned}$$

Denote by $\nu(\mathbf{C}) = \lambda(\mathbf{C}) - \lambda(\mathbf{C}^0)$ and $\Delta BIC(\mathbf{C}) = BIC(\mathbf{C}) - BIC(\mathbf{C}^0)$, then

$$\Delta BIC(\mathbf{C}) = 2n\{\frac{1}{n}\sum_{i=1}^{n} \log[\frac{f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C}^0)})}{g(\mathbf{x}_i)}] - \frac{1}{n}\sum_{i=1}^{n}[\frac{f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C})})}{g(\mathbf{x}_i)}]\} + \nu(\mathbf{C})\log(n). \tag{15}$$

All the possible partition $\mathcal{C}$ can be decomposed as follows

$$\mathcal{C} = \mathbf{C}^0 \cup \{\mathbf{C} \in \mathcal{C}; D_{KL}(g||f(.|\gamma^*_{(\mathbf{C})})) \neq 0\}.$$

Let $\mathcal{C}_1 = \{\mathbf{C} \in \mathcal{C}; D_{KL}(g||f(.|\gamma^*_{\mathbf{C}})) \neq 0\}$. We only need to prove that

$$\forall \mathbf{C} \in \mathcal{C}_1,\ P(\Delta BIC(\mathbf{C}) < 0) \xrightarrow[n\to\infty]{} 0.$$

Let $\mathcal{Q}_n(\mathbf{C}) = \frac{1}{n}\sum_{i=1}^{n} \log[\frac{f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C})})}{g(\mathbf{x}_i)}]$, and $\mathcal{Q}(\mathbf{C}) = -D_{KL}(g||f(.|\gamma^*_{(\mathbf{C})}))$,
Following Eq. (15), we then have

$$\begin{aligned} &P(\Delta BIC(\mathbf{C}) < 0) = P(2n[\mathcal{Q}_n(\mathbf{C}^0) - \mathcal{Q}_n(\mathbf{C})] + \nu(\mathbf{C})\log(n) < 0) \\ =&P(\mathcal{Q}_n(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}^0) + \mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) + \mathcal{Q}(\mathbf{C}) - \mathcal{Q}_n(\mathbf{C}) + \frac{\nu(\mathbf{C})\log(n)}{2n} < 0). \end{aligned}$$

Thus, for all $\epsilon > 0$, and according to the Lemma 8 below,

$$\begin{aligned} P(\Delta BIC(\mathbf{C}) < 0) \leq& P(\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}_n(\mathbf{C}^0) > \epsilon) + P(\mathcal{Q}_n(\mathbf{C}) - \mathcal{Q}(\mathbf{C}) > \epsilon) + \\ & P(\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) + \frac{\nu(\mathbf{C})\log(n)}{2n} < 2\epsilon). \end{aligned}$$

From Proposition 1 below, $\forall \mathbf{C}, \mathcal{Q}_n(\mathcal{C}) \xrightarrow[n\to\infty]{P} \mathcal{Q}(\mathcal{C})$. Thus,

$$\forall \epsilon > 0, P(\mathcal{Q}_n(\mathbf{C}) - \mathcal{Q}(\mathbf{C}) > \epsilon) \leq P(|\mathcal{Q}_n(\mathbf{C}) - \mathcal{Q}(\mathbf{C})| > \epsilon) \xrightarrow[n\to\infty]{} 0.$$

Then,

$$P(\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) + \frac{\nu(\mathbf{C})\log(n)}{2n} < 2\epsilon) \leq P(\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) - 2\epsilon < |\frac{\nu(\mathbf{C})\log(n)}{2n}|).$$

We know that $\frac{\nu(\mathbf{C})\log(n)}{2n} \xrightarrow[n\to\infty]{} 0$ and $\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) > 0$ because $\mathbf{C} \in \mathcal{C}_1$. Taking $\epsilon = \frac{\mathcal{Q}(\mathbf{C}^0)-\mathcal{Q}(\mathbf{C})}{4} > 0$, we get

$$P(\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) + \frac{\nu(\mathbf{C})\log(n)}{2n} < 2\epsilon) \leq P(\frac{\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C})}{2} < |\frac{\nu(\mathbf{C})\log(n)}{2n}|) \xrightarrow[n\to\infty]{} 0.$$

Finally, $P(\Delta BIC(\mathbf{C}) < 0) \xrightarrow[n\to\infty]{} 0$. ■

Lemma 8 *Let A and B be two real random variables,*

$$\forall \epsilon \in \mathcal{R}, P(A + B \leq 0) \leq P(A \leq \epsilon) + P(-B > \epsilon).$$

Proposition 1 *Under Assumptions* **A1** *and* **A2**, $\forall \mathbf{C} \in \mathcal{C}$,

$$\frac{1}{n}\sum_{i=1}^{n} \log\{\frac{g(\mathbf{x}_i)}{f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C})})}\} \xrightarrow[n\to\infty]{P} D_{KL}(g||f(.|\gamma^*_{(\mathbf{C})})).$$

Proof By the law of large numbers, if $E[|\log(g(X))|] < \infty$,

$$\frac{1}{n}\sum_{i=1}^{n} \log[g(\mathbf{x}_i)] \xrightarrow[n\to\infty]{P} E_X[\log(g(X))]. \tag{16}$$

If we can show that for the family

$$\mathcal{G}_{(\mathbf{C})} := \{\log[f(.|\gamma)]; \gamma \in \Gamma'_{(\mathbf{C})}\},$$

the following holds

$$\frac{1}{n}\sum_{i=1}^{n} \log[f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C})})] \xrightarrow[n\to\infty]{P} E_X[\log f(X|\gamma^*{}_{(\mathbf{C})})], \tag{17}$$

then Proposition 1 is proved by combining Eqs. (16) and (17).

Now we only need to prove that $E_X[\log f(X)] < \infty$. First, by the Assumption **A**1, $\Gamma'_{(\mathbf{C})}$ is a compact metric space. And for all $\mathbf{x} \in \mathcal{R}^p$, $\gamma_{(\mathbf{C})} \in \Gamma'_{(\mathbf{C})} \to \log[f(\mathbf{x}|\gamma_{(\mathbf{C})})]$ is continuous. We just need to verify that there is an envelope function G of $\mathcal{G}_{(\mathbf{C})}$ being g-integrable.

Denote by $\mathcal{M} = [M_2^0] \times [M_3^0] \times \cdots \times [M_T^0] = \{1, 2, ..., M_2^0\} \times \{1, 2, ..., M_3^0\} \times \cdots \times \{1, 2, ..., M_T^0\}$. We first write out $\log[f(\mathbf{x}|\gamma_{(\mathbf{C})})]$ explicitly:

$$\begin{aligned}
\log[f(\mathbf{x}|\gamma_{(\mathbf{C})})] &= \log\Big[\sum_{k_1=1}^{M_1^0} \pi_{k_1} \sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}\Pi_{j=1}^{T} N(x^{(C_j)}|\mu_{k_j}^{(j)}, \Sigma_{k_j}^{(j)})\Big] \\
&= \log\Big[\sum_{k_1=1}^{M_1^0} \pi_{k_1} \sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T} \\
&\quad \Pi_{j=1}^{T}(2\pi)^{-|C_j|/2}|\Sigma_{k_j}^{(j)}|^{-1/2}\exp\{-\frac{(x^{(C_j)}-\mu_{k_j}^{(j)})'{\Sigma_{k_j}^{(j)}}^{-1}(x^{(C_j)}-\mu_{k_j}^{(j)})}{2}\} \\
&\leq \log\Big[\sum_{k_1=1}^{M_1^0} \pi_{k_1} \sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}(2\pi a)^{-\frac{d}{2}}\Big] \\
&\leq -\frac{d}{2}\log[2\pi a],
\end{aligned}$$

where $x^{(C_j)}$ being the C_j-th variable block. The inequality is obtained according to Lemma 9 and $(x^{(C_j)}-\mu_{k_j}^{(j)})'{\Sigma_{k_j}^{(j)}}^{-1}(x^{(C_j)}-\mu_{k_j}^{(j)}) \leq 0$, for $j = 1, 2, ..., T$, hence is exponential function bounded by 1.

Now we need to obtain a lower bound, and we use the concavity of the logarithm function:

$$\begin{aligned}
\log[f(\mathbf{x}|\gamma_{(\mathbf{C})})] &\geq \sum_{k_1=1}^{M_1^0} \pi_{k_1} \sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T} \\
&\quad \Pi_{j=1}^{T}(2\pi)^{-|C_j|/2}|\Sigma_{k_j}^{(j)}|^{-1/2}\exp\{-\frac{(x^{(C_j)}-\mu_{k_j}^{(j)})'{\Sigma_{k_j}^{(j)}}^{-1}(x^{(C_j)}-\mu_{k_j}^{(j)})}{2}\} \\
&= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\sum_{k_1=1}^{M_1^0} \pi_{k_1} \sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}\Big\{\sum_{j=1}^{T}\log[|\Sigma_{k_j}^{(j)}|] \\
&\quad + \sum_{j=1}^{T}(x^{(C_j)}-\mu_{k_j}^{(j)})'{\Sigma_{k_j}^{(j)}}^{-1}(x^{(C_j)}-\mu_{k_j}^{(j)})\Big\}.
\end{aligned}$$

According to Lemma 9 and $\forall k_j$, $\mu_{k_j}^{(j)} \in \mathcal{B}(\eta, |C_j|)$

$$\begin{aligned}
(x^{(C_j)}-\mu_{k_j}^{(j)})'{\Sigma_{k_j}^{(j)}}^{-1}(x^{(C_j)}-\mu_{k_j}^{(j)}) &\leq \frac{||x^{(C_j)}-\mu_{k_j}^{(j)}||^2}{a} \\
&\leq \frac{2(||x^{(C_j)}||^2+||\mu_{k_j}^{(j)}||^2)}{a} \\
&\leq \frac{2(||x^{(C_j)}||^2+\eta^2)}{a}.
\end{aligned}$$

Hence,

$$\log[f(\mathbf{x}|\gamma_{(\mathbf{C})})] \geq -\frac{d}{2}\log(2\pi) - \frac{1}{2}\sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}$$
$$\Big\{\log(b^d) + \sum_{j=1}^{T}\frac{2}{a}(||x^{(C_j)}||^2+\eta^2)\Big\}$$
$$= -\frac{d}{2}\log(2\pi b) - \frac{||\mathbf{x}||^2+2\eta}{a}.$$

Hence, each function of the family $\mathcal{G}_{(\mathbf{C})}$ is bounded by

$$-\frac{d}{2}\log(2\pi b) - \frac{||\mathbf{x}||^2+2\eta}{a} \leq \log[f(\mathbf{x}|\gamma_{(\mathbf{C})})] \leq \frac{d}{2}\log[2\pi a].$$

Therefore, for all $\gamma_{(\mathbf{C})} \in \Gamma'_{(\mathbf{C})}$ and all $\mathbf{x} \in \mathcal{R}^p$,

$$|\log[f(\mathbf{x}|\gamma_{(\mathbf{C})})]| \leq Z_1(a,b,\eta) + Z_2(\eta,a)||\mathbf{x}||^2$$

defining the envelope function G, where $Z_1(a,b,\eta)$ and $Z_2(\eta,a)$ are two positive constants.

In order to verify that G is g-integrable, we need to show that $\int||\mathbf{x}||^2 g(\mathbf{x})d\mathbf{x} < \infty$.

$$\int||\mathbf{x}||^2 g(\mathbf{x})d\mathbf{x} = \int||\mathbf{x}||^2 f(\mathbf{x}|\gamma^*_{(\mathbf{C}^0)})d\mathbf{x}$$
$$= \sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}\int||\mathbf{x}||^2\phi(\mathbf{x}|\mu_{(k_1,k_2,...,k_T)},\Sigma_{(k_1,k_2,...,k_T)})d\mathbf{x}$$
$$\leq \sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}(2\sum_{j=1}^{T}||\mu_{k_j}^{(j)}||^2+2\sum_{j=1}^{T}tr(\Sigma_{k_j}^{(j)})),$$

where $\mu_{(k_1,k_2,...,k_T)} = (\mu_{k_1}^{(1)}, \mu_{k_2}^{(2)}, ..., \mu_{k_T}^{(T)})$ and $\Sigma_{(k_1,k_2,...,k_T)}$ is block diagonal with the t-th diagonal block being $\Sigma_{k_t}^{(t)}$, for $t = 1, 2, ..., T$. Because $\int||\mathbf{x}||^2\phi(\mathbf{x}|0,\Sigma)d\mathbf{x} = tr(\Sigma)$. Then using the triangle inequality, we get $\int||\mathbf{x}||^2\phi(\mathbf{x}|\mu,\Sigma)d\mathbf{x} \leq 2(||\mu||^2+tr(\Sigma))$. Further, from Lemma 9,

$$\int||\mathbf{x}||^2 g(\mathbf{x})d\mathbf{x} \leq 4\eta^2 + 2bp.$$

Hence, G is g-integrable. Since $\log(g) \in \mathcal{G}_{(\mathbf{C}^0)}$, it implies that $E[|\log g(X)|] \leq E[G(X)] < \infty$. We then prove the theorem. ■

Lemma 9 *Let $\Sigma \in \mathcal{D}_d$, where $\mathcal{D}_d$ is defined in* **A**1. *Then*

1. $a^d \leq |\Sigma| \leq b^d$ *and* $tr(\Sigma) \leq bd$
2. $\forall \mathbf{x} \in \mathcal{R}^d$, $\mathbf{x}'\mathbf{x}/b \leq \mathbf{x}'\Sigma^{-1}\mathbf{x} \leq \mathbf{x}'\mathbf{x}/a$

Proof See for instance Maugis et al. (2007). ■

Appendix D

We prove Theorem 7 in Section 4.3. A configuration of a set of states is a particular tuple of values for these states. Use notation $\Psi(\cdot)$ for the set of all possible configurations of any group of states.

Proof For a HMM-VB, the configuration of the whole state sequence $\mathbf{s} = \{s_1, s_2, ..., s_T\}$ is treated as the index for one mixture component in the mapped GMM. The set of all possible configurations is $\Psi(\{s_1, ..., s_T\})$. Let $\hat{\mathcal{S}} = \Psi(\{s_1, ..., s_T\})$ and $\mathcal{S}_t = \Psi(s_t)$. Each component is a Gaussian distribution with mean $\mu_{\mathbf{s}} = (\mu_{s_1}^{(1)}, \mu_{s_2}^{(2)}, ..., \mu_{s_T}^{(T)})$ (column-wise stack of vectors) and a covariance matrix, denoted by $\Sigma_{\mathbf{s}}$, that is block diagonal. The tth diagonal block in $\Sigma_{\mathbf{s}}$ is $\Sigma_{s_t}^{(t)}$ with dimension $d_t \times d_t$

$$\Sigma_{\mathbf{s}} = \begin{pmatrix} \Sigma_{s_1}^{(1)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \Sigma_{s_2}^{(2)} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \Sigma_{s_T}^{(T)} \end{pmatrix}.$$

If we apply MEM directly to HMM-VB and keep in mind that $\mathbf{s}$ is the index for the mixture component, we need to compute the posterior $P(\mathbf{s} \mid \mathbf{x})$ in the E-step and

$$\left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \Sigma_{\mathbf{s}}^{-1} \right)^{-1} \left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \Sigma_{\mathbf{s}}^{-1} \mu_{\mathbf{s}} \right)$$

in the M-step. The computational hurdle is that the number of possible sequences $\mathbf{s}$, that is, $|\hat{\mathcal{S}}|$, grows exponentially with T (assuming similar $|\mathcal{S}_t|$).

Because $\Sigma_{\mathbf{s}}$ is block diagonal, we have

$$\begin{aligned} &\left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \Sigma_{\mathbf{s}}^{-1} \right)^{-1} \left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \Sigma_{\mathbf{s}}^{-1} \mu_{\mathbf{s}} \right) \\ = &\begin{pmatrix} \left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_1}^{(1)} \right)^{-1} \right)^{-1} \left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_1}^{(1)} \right)^{-1} \mu_{s_1}^{(1)} \right) \\ \left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_2}^{(2)} \right)^{-1} \right)^{-1} \left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_2}^{(2)} \right)^{-1} \mu_{s_2}^{(2)} \right) \\ \vdots \\ \vdots \\ \left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_T}^{(T)} \right)^{-1} \right)^{-1} \left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_T}^{(T)} \right)^{-1} \mu_{s_T}^{(T)} \right) \end{pmatrix}. \end{aligned}$$

Hence the tth variable block of $\mathbf{x}$ is given by

$$x^{(t)} = \left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_t}^{(t)} \right)^{-1} \right)^{-1} \left(\sum_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_t}^{(t)} \right)^{-1} \mu_{s_t}^{(t)} \right), \quad t = 1, 2, ..., T.$$

Let $I(\cdot)$ be the indicator function that equals 1 when the argument is true. Note that

$$\begin{aligned}\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s}\mid\mathbf{x})\left(\Sigma_{s_t}^{(t)}\right)^{-1} &= \sum_{k\in\mathcal{S}_t}\left[\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s}\mid\mathbf{x})I(s_t=k)\right]\cdot\left(\Sigma_k^{(t)}\right)^{-1}\\ &= \sum_{k\in\mathcal{S}_t} L_k(\mathbf{x},t)\cdot\left(\Sigma_k^{(t)}\right)^{-1}\end{aligned}$$

according to the definition of $L_k(\mathbf{x},t)$ in Eq. (4). By the same technique, we can show that

$$\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s}\mid\mathbf{x})\left(\Sigma_{s_t}^{(t)}\right)^{-1}\mu_{s_t}^{(t)} = \sum_{k\in\mathcal{S}_t} L_k(\mathbf{x},t)\cdot\left(\Sigma_k^{(t)}\right)^{-1}\mu_k^{(t)}.$$

■

It is interesting to note that Theorem 7 extends easily to a model more general than HMM-VB. For HMM-VB, s_1, ..., s_T follow a finite state Markov chain. *Bayesian network (BN)* is a generalization of Markov chain on a *directed acyclic graph (DAG)*, which is defined by all the conditional distributions of every state given any configuration of its parent states. If there is no parent, the marginal distribution of a state is specified. See Jensen (1996) for an introduction to BN. If we assume that the latent states s_1, ..., s_T are governed by a BN, and the observed $X^{(t)}$ conditioned on s_t follows a parametric distribution and is conditionally independent of any other $X^{(t')}$ and $s_{t'}$, then we obtain a special lattice MM more broadly defined than HMM-VB. We call this model *Mixture with Latent Bayesian Network* (*MLBN*). Define the following probability for MLBN:

$$L_k(\mathbf{x},t) = P(s_t=k\mid\mathbf{x}),\ \ k\in\mathcal{S}_t,\ t=1,...,T. \tag{18}$$

Clearly, the above definition is the same as Eq. (4) in the special case of HMM-VB. The MBW algorithm in Section 3.2 for HMM-VB applies to MLBN in general. We only need to use the definition in Eq. (18) for $L_k(\mathbf{x}^{[r]},t)$. Theorem 7 can be proved for MLBN in the same way.

Appendix E

In this section, we provide the details for the data generation schemes of the simulation studies. We follow the notations in Eq. (3).

Simulation study in Section 5.1.1

It is a two-block design. The first variable block has 5 variables and is generated from a normal mixture of 7 components. We set $\pi = (0.51, 0.09, 0.20, 0.07, 0.10, 0.01, 0.02)$,

$$\mu^{(1)} = \begin{pmatrix} \mu_1^{(1)} \\ \mu_2^{(1)} \\ \vdots \\ \mu_7^{(1)} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4.5 & 0 & 0 \\ 4.5 & -2 & 0 & 0 & 0 \\ 0 & 0 & 4 & -1 & 4 \\ 0 & 3 & 0 & 5 & 0 \\ 0 & 7 & 7 & 0 & 0 \\ 0 & 7.7 & 8 & 0 & 0 \end{pmatrix},$$

$\Sigma_{1:5}^{(1)} = 1.5I_5$, $\Sigma_6^{(1)} = I_5$, $\Sigma_7^{(1)} = 0.5I_5$.

The second variable block has 3 variables, and is generated from a mixture of 10 Gaussian components. We set the transition probability matrix as

$$a^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.22 & 0.5 & 0.28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.19 & 0 & 0 & 0 & 0 & 0.46 & 0.35 \\ 0 & 0 & 0 & 0.50 & 0.2 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0.11 & 0.31 & 0.38 & 0 & 0 \\ 0 & 0 & 0.14 & 0.66 & 0 & 0.20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.15 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0.65 \end{pmatrix},$$

$$\mu^{(2)} = \begin{pmatrix} \mu_1^{(2)} \\ \mu_2^{(2)} \\ \vdots \\ \mu_{10}^{(2)} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ -4 & -4 & -4 \\ 6.5 & 6.5 & 6.5 \\ -1 & 5 & 0 \\ -1.5 & 0 & 5 \\ 6 & 7 & 6.5 \\ -4.0 & 2 & 4.5 \\ 5 & -5 & -5 \\ -4 & 0 & 0 \\ 5 & -4.5 & -6 \end{pmatrix},$$

$\Sigma_{1,4,5}^{(2)} = 2I_3$, $\Sigma_2^{(2)} = \begin{pmatrix} 2.0 & 0.2 & 0.2 \\ 0.2 & 2.0 & 0.2 \\ 0.2 & 0.2 & 2.0 \end{pmatrix}$, $\Sigma_3^{(2)} = \begin{pmatrix} 2.0 & 0.9 & 0.9 \\ 0.9 & 2.0 & 0.9 \\ 0.9 & 0.9 & 1.5 \end{pmatrix}$, $\Sigma_6^{(2)} = \begin{pmatrix} 2.0 & 0.9 & 0.9 \\ 0.9 & 1.5 & 0.9 \\ 0.9 & 0.9 & 2.0 \end{pmatrix}$,

$\Sigma_7^{(2)} = \begin{pmatrix} 2.0 & -0.6 & -0.6 \\ -0.6 & 2.0 & 0.6 \\ -0.6 & 0.6 & 2.0 \end{pmatrix}$, $\Sigma_8^{(2)} = \begin{pmatrix} 2.0 & -0.6 & -0.6 \\ -0.6 & 5/3 & 0.6 \\ -0.6 & 0.6 & 5/3 \end{pmatrix}$, $\Sigma_{9,10}^{(2)} = \begin{pmatrix} 5/3 & -0.6 & -0.6 \\ -0.6 & 5/3 & 0.6 \\ -0.6 & 0.6 & 5/3 \end{pmatrix}$.

Simulation study in Section 5.1.2

The data is generated from a GMM with 10 components. Hence all the variables form one single variable block. We set

$$\pi = (0.01, 0.02, 0.23, 0.1, 0.15, 0.098, 0.098, 0.098, 0.098, 0.098),$$

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_{10} \end{pmatrix} = \begin{pmatrix} 5 & 5 & 5 & 5 & 5 \\ -5 & -5 & -5 & -5 & -5 \\ 5 & 5 & 0 & 5 & 0 \\ 5 & 0 & 0 & 5 & 5 \\ 0 & 0 & 5 & 5 & 5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -5 & -5 & -5 \\ -5 & -5 & 0 & 0 & -5 \\ -5 & -5 & -5 & 0 & 0 \\ -5 & 0 & 0 & 0 & -5 \end{pmatrix}.$$

All the variance matrices are independently generated from an inverse Wishart distribution with 10 degrees of freedom and diagonal scale matrix $7I_5$.

Simulation study in Section 5.1.3

The first 10 dimensions of the data are generated from a 3-component GMM. We set $\pi = (0.05, 0.25, 0.70)$,

$$\mu^{(1)} = \begin{pmatrix} \mu_1^{(1)} \\ \mu_2^{(1)} \\ \mu_3^{(1)} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 \end{pmatrix},$$

and $\Sigma_{1:3}^{(1)}$ are independently generated from an inverse Wishart distribution with 15 degrees of freedom and diagonal scale matrix $7I_{10}$.

The rest 30 dimensions are conditionally generated from a 5-component GMM. We set the transition probability matrix as

$$a^{(1)} = \begin{pmatrix} 0.1 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0.28 & 0.72 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mu^{(2)} = \begin{pmatrix} \mu_1^{(2)} \\ \mu_2^{(2)} \\ \cdots \\ \mu_5^{(2)} \end{pmatrix} = \begin{pmatrix} \vec{0}_{30} \\ \vec{5}_{30} \\ \vec{-5}_{30} \\ \vec{-5}_{15}, \vec{5}_{15} \\ \vec{5}_{15}, \vec{-5}_{15} \end{pmatrix},$$

where $\vec{v}_l$ denotes a vector of the form $(v, v, ..., v)$ with l elements of the same value v. $\Sigma_{1:5}^{(2)}$ are block diagonals, which have the form of

$$\Sigma_k^{(2)} = \begin{pmatrix} \mathbf{A}_k^{(2)} & 0 \\ 0 & \mathbf{B}_k^{(2)} \end{pmatrix},$$

where $k = 1, ..., 5$, $\mathbf{A}_k^{(2)}$ is of size 10×10, which is independently generated from an inverse Wishart distribution with 15 degrees of freedom and diagonal scale matrix $7\mathbf{I}_{10}$. $\mathbf{B}_k^{(2)}$ is of size 20×20, and is independently generated from an inverse Wishart distribution with 25 degrees of freedom and diagonal scale matrix $7\mathbf{I}_{10}$.

References

Nima Aghaeepour, Radina Nikolic, Holger H. Hoos, and Ryan R. Brinkman. Rapid cell population identification in flow cytometry data. *Cytometry Part A*, 79A(1):6–13, 2011.

Nima Aghaeepour, Greg Finak, Holger Hoos, Tim R. Mosmann, Ryan Brinkman, Raphael Gottardo, and Richard H. Scheuermann. Critical assessment of automated flow cytometry data analysis techniques. *Nature Methods*, 10(3):228–238, Mar 2013.

Greg M. Allenby, Neeraj Arora, and James L. Ginter. On the heterogeneity of demand. *Journal of Marketing Research*, 35(3):384–389, 1998.

Adelchi Azzalini and Giovanna Menardi. Clustering via nonparametric density estimation: The r package pdfcluster. *Journal of Statistical Software, Articles*, 57(11):1–26, 2014.

Dmitry R. Bandura, Vladimir I. Baranov, Olga I. Ornatsky, Alexei Antonov, Robert Kinach, Xudong Lou, Serguei Pavlov, Sergey Vorobiev, John E. Dick, and Scott D. Tanner. Mass cytometry: Technique for real time single cell multitarget immunoassay based on inductively coupled plasma time-of-flight mass spectrometry. *Analytical Chemistry*, 81 (16):6813–6822, 2009.

Jeffrey D Banfield and Adrian E Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.

Burkhard Becher, Andreas Schlitzer, Jinmiao Chen, Florian Mair, Hermi R. Sumatoh, Karen Wei Weng Teng, Donovan Low, Christiane Ruedl, Paola Riccardi-Castagnoli, Michael Poidinger, Melanie Greter, Florent Ginhoux, and Evan W. Newell. High-dimensional analysis of the murine myeloid cell system. *Nature Immunology*, 15(12):1181–1189, Dec 2014. Resource.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

Michael J. Boedigheimer and John Ferbas. Mixture modeling approach to flow cytometry data. *Cytometry Part A*, 73A(5):421–429, 2008.

Kenneth P Burnham and David R Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer Science & Business Media, 2003.

Gilles Celeux and Grard Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995.

Gilles Celeux, Merrilee Hurn, and Christian P. Robert. Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, 95(451):957–970, 2000.

Cliburn Chan, Feng Feng, Janet Ottinger, David Foster, Mike West, and Thomas B. Kepler. Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry Part A*, 73A(8):693–701, 2008.

Cliburn Chan, Lin Lin, Jacob Frelinger, Valerie Hebert, Dominic Gagnon, Claire Landry, Rafick-Pierre Sékaly, Jennifer Enzor, Janet Staats, Kent J. Weinhold, Maria Jaimes, and Mike West. Optimization of a highly standardized carboxyfluorescein succinimidyl ester flow cytometry panel and gating strategy design using discriminative information measure evaluation. *Cytometry Part A*, 77A(12):1126–1136, 2010.

Pratip K. Chattopadhyay, Todd M. Gierahn, Mario Roederer, and J. Christopher Love. Single-cell technologies for monitoring immune systems. *Nature Immunology*, 15(2): 128–135, Feb 2014. Review.

Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

Lawrence Corey, Peter B. Gilbert, Georgia D. Tomaras, Barton F. Haynes, Giuseppe Pantaleo, and Anthony S. Fauci. Immune correlates of vaccine protection against hiv-1 acquisition. *Science Translational Medicine*, 7(310):310rv7, 2015.

Patricia A. Darrah, Dipti T. Patel, Paula M. De Luca, Ross W. B. Lindsay, Dylan F. Davey, Barbara J. Flynn, Soren T. Hoff, Peter Andersen, Steven G. Reed, Sheldon L. Morris, Mario Roederer, and Robert A. Seder. Multifunctional th1 cells define a correlate of vaccine-mediated protection against leishmania major. *Nature Methods*, 13(7):843–850, Jul 2007.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B(Statistical Methodology)*, 39(1):1–38, 1977.

Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

Greg Finak, Ali Bashashati, Ryan Brinkman, and Raphael Gottardo. Merging mixture components for cell population identification in flow cytometry. *Advances in Bioinformatics*, Article ID 247646, 2009.

Chris Fraley and Adrian E Raftery. Mclust version 3: an r package for normal mixture modeling and model-based clustering. Technical report, DTIC Document, 2006.

Keinosuke Fukunaga and Larry D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

Minzhe Guo, Hui Wang, S Steven Potter, Jeffrey A Whitsett, and Yan Xu. Sincera: a pipeline for single-cell rna-seq profiling analysis. *PLoS Computational Biology*, 11(11): e1004575, 2015.

Christian Hennig. Methods for merging gaussian mixture components. *Advances in Data Analysis and Classification*, 4(1):3–34, 2010.

Finn V. Jensen. *Introduction to Bayesian Networks*. Springer-Verlag New York, Inc., New York, NY, USA, 1996.

Hiroyuki Kasahara and Katsumi Shimotsu. Nonparametric identification of finite mixture models of dynamic discrete choices. *Econometrica*, 77(1):135–175, 2009.

Vladimir Yu Kiselev, Kristina Kirschner, Michael T. Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N. Natarajan, Wolf Reik, Mauricio Barahona, Anthony R Green, and Martin Hemberg. Sc3: consensus clustering of single-cell rna-seq data. *Nature Methods*, 2017.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

Hyangmin Lee and Jia Li. Variable selection for clustering by separability based on ridgelines. *Journal of Computational and Graphical Statistics*, 21(2):315–337, February 2012.

Jia Li. Clustering based on a multi-layer mixture model. *Journal of Computational and Graphical Statistics*, 14(3):547 – 568, 2005.

Jia Li, Surajit Ray, and Bruce G. Lindsay. A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, 8(8):1687–1723, August 2007.

Lin Lin, Cliburn Chan, Sine R. Hadrup, Thomas M. Froesig, Quanli Wang, and Mike West. Hierarchical bayesian mixture modelling for antigen-specific t-cell subtyping in combinatorially encoded flow cytometry studies. *Statistical Applications in Genetics and Molecular Biology*, 12(3):309–331, June 2013.

Lin Lin, Greg Finak, Kevin Ushey, Chetan Seshadri, Thomas R Hawn, Nicole Frahm, Thomas J. Scriba, Hassan Mahomed, Willem Hanekom, Pierre-Alexandre Bart, Giuseppe Pantaleo, Georgia D. Tomaras, Supachai Rerks-Ngarm, Jaranit Kaewkungwal, Sorachai Nitayaphan, Punnee Pitisuttithum, Nelson L. Michael, Jerome H. Kim, Merlin L. Robb, Robert J. O'Connell, Nicos Karasavvas, Peter Gilbert, Stephen C. De Rosa, M. Juliana McElrath, and Raphael Gottardo. COMPASS identifies t-cell subsets correlated with clinical outcomes. *Nature Biotechnology*, 33(6):610–616, June 2015a.

Lin Lin, Jacob Frelinger, Wenxin Jiang, Greg Finak, Chetan Seshadri, Pierre-Alexandre Bart, Giuseppe Pantaleo, Julie McElrath, Steve DeRosa, and Raphael Gottardo. Identification and visualization of multidimensional antigen-specific t-cell populations in polychromatic cytometry data. *Cytometry Part A*, 87(7):675–682, 2015b.

Lin Lin, Cliburn Chan, and Mike West. Discriminative variable subsets in bayesian classification with mixture models, with application in flow cytometry studies. *Biostatistics*, 17 (1):40–53, 2016.

Kenneth Lo, Ryan Remy Brinkman, and Raphael Gottardo. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A*, 73A(4):321–332, 2008.

Holden T. Maecker, J. Philip McCoy, and Robert Nussenblatt. Standardizing immunophenotyping for the human immunology project. *Nature Reviews Immunology*, 12(3):191–200, Mar 2012.

Cathy Maugis, Gilles Celeux, and Marie-Laure Martin-Magniette. *Variable selection for clustering with Gaussian mixture models*. PhD thesis, INRIA, 2007.

Cathy Maugis, Gilles Celeux, and Marie-Laure Martin-Magniette. Variable selection for clustering with gaussian mixture models. *Biometrics*, 65(3):701–709, 2009.

Geoff J. McLachlan, Richard W. Bean, and David Peel. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18(3):413–422, 2002.

Volodymyr Melnykov. Merging mixture components for clustering through pairwise overlap. *Journal of Computational and Graphical Statistics*, 25(1):66–90, 2016.

Volodymyr Melnykov and Ranjan Maitra. Finite mixture models and model-based clustering. *Statistics Surveys*, 4:80–116, 2010.

Stephen P. Perfetto, Pratip K. Chattopadhyay, and Mario Roederer. Seventeen-colour flow cytometry: unravelling the immune system. *Nature Reviews Immunology*, 4(8):648–655, Aug 2004.

Saumyadipta Pyne, Xinli Hu, Kui Wang, Elizabeth Rossin, Tsung-I Lin, Lisa M. Maier, Clare Baecher-Allan, Geoffrey J. McLachlan, Pablo Tamayo, David A. Hafler, Philip L. De Jager, and Jill P. Mesirov. Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Sciences*, 106(21):8519–8524, 2009.

William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

Surajit Ray and Saumyadipta Pyne. A computational framework to emulate the human perspective in flow cytometric data analysis. *PloS ONE*, 7(5):e35693, 2012.

Sylvia Richardson and Peter J. Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792, 1997.

Mario Roederer, Jason M. Brenchley, Michael R. Betts, and Stephen C. De Rosa. Flow cytometric analysis of vaccine responses: how many colors are enough? *Clinical Immunology*, 110(3):199 – 205, 2004.

Rahul Satija, Jeffrey A. Farrell, David Gennert, Alexander F. Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5): 495–502, 2015.

Chetan Seshadri, Lin Lin, Thomas J. Scriba, Glenna Peterson, David Freidrich, Nicole Frahm, Stephen C. DeRosa, D. Branch Moody, Jacques Prandi, Martine Gilleron, Hassan Mahomed, Wenxin Jiang, Greg Finak, Willem A. Hanekom, Raphael Gottardo, M. Juliana McElrath, and Thomas R. Hawn. T cell responses against mycobacterial lipids and proteins are poorly correlated in south african adolescents. *The Journal of Immunology*, 195(10):4595–4603, 2015.

Matthew H. Spitzer and Garry P. Nolan. Mass cytometry: Single cells, many features. *Cell*, 165(4):780–791, May 2016.

Matthew Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809, 2000.

Michael Titterington, Adrian F.M. Smith, and Ehud Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1985.

Bo Wang, Junjie Zhu, Emma Pierson, Daniele Ramazzotti, and Serafim Batzoglou. Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. *Nature Methods*, 14(4):414–416, 2017.

Sidney J. Yakowitz and John D. Spragins. On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 39(1):209–214, 1968.

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book*. Cambridge University Press, 1997.

Justina Žurauskienė and Christopher Yau. pcareduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, 17(1):140, Mar 2016.