Parametric Manifold Learning of Gaussian Mixture Models

Ziquan Liu[1*], **Lei Yu**[1], **Janet H. Hsiao**[2] and **Antoni B. Chan**[1]
[1]Department of Computer Science, City University of Hong Kong
[2]Department of Psychology, The University of Hong Kong
{ziquanliu2-c, leiyu6-c}@my.cityu.edu.hk, jhsiao@hku.hk, abchan@cityu.edu.hk

Abstract

The Gaussian Mixture Model (GMM) is among the most widely used parametric probability distributions for representing data. However, it is complicated to analyze the relationship among GMMs since they lie on a high-dimensional manifold. Previous works either perform clustering of GMMs, which learns a limited discrete latent representation, or kernel-based embedding of GMMs, which is not interpretable due to difficulty in computing the inverse mapping. In this paper, we propose Parametric Manifold Learning of GMMs (PML-GMM), which learns a parametric mapping from a low-dimensional latent space to a high-dimensional GMM manifold. Similar to PCA, the proposed mapping is parameterized by the principal axes for the component weights, means, and covariances, which are optimized to minimize the reconstruction loss measured using Kullback-Leibler divergence (KLD). As the KLD between two GMMs is intractable, we approximate the objective function by a variational upper bound, which is optimized by an EM-style algorithm. Moreover, We derive an efficient solver by alternating optimization of subproblems and exploit Monte Carlo sampling to escape from local minima. We demonstrate the effectiveness of PML-GMM through experiments on synthetic, eye-fixation, flow cytometry, and social check-in data.

1 Introduction

Probabilistic models are effective tools for representing real-world data in the presence of noise. For example, Gaussian mixture models (GMMs) are regarded as a *universal visual vocabulary* in computer vision [Sánchez *et al.*, 2013; Winn *et al.*, 2005; Perronnin *et al.*, 2006] due to its capacity to model the mean and correlation in image patches; hidden Markov models (HMMs) are often used to model the speech sequence in natural language processing [Rabiner, 1989] because it can model the dynamics of both hidden (underlying) processes and the observed speech sequences; linear dynamical systems (dynamic textures, DTs) are used to describe a video since it can abstract complex patterns of motion and appearance [Doretto *et al.*, 2003]. Clustering probabilistic models can produce hierarchical representations of data, which can be used for retrieval, annotation, indexing and codebook generation. Previous works have developed clustering algorithms for Gaussian distributions [Yu *et al.*, 2018; Vasconcelos and Lippman, 1999], DTs [Chan *et al.*, 2010], and HMMs [Coviello *et al.*, 2012].

While clustering probabilistic models can give hierarchical representations, it cannot learn a *continuous* and *interpretable* manifold on which we can see the continuous change between probabilistic models. In various application domains, e.g., medical diagnosis [Carter *et al.*, 2009], behavior analysis [Chan *et al.*, 2018] and visual recognition [Perronnin *et al.*, 2006], such an interpretable manifold provides better insight into the subject differences and the underlying mechanisms. For example, suppose we collect data from several subjects and learn a subject-level GMM for each subject's data. The GMMs could be clustered to obtain common patterns among the subjects, but this provides a limited discrete representation (one of K clusters). Alternatively, if the GMMs are embedded into a manifold, then the subject's coordinates on the manifold are continuous and their relationship with other subject properties (e.g., subject age, performance) could be revealed using correlation analysis. Furthermore, directions on the manifold would correspond to changes in the structure of the GMMs, providing insight on the underlying mechanisms of the revealed correlations. Despite its importance, manifold learning for probabilistic models has not been well-explored.

In this paper, we propose to learn a smooth and interpretable manifold for GMMs, where the inverse mapping between the low-dimensional latent space and the high-dimensional statistical manifold can be obtained easily. Inspired by PCA, we propose a *parametric* approach for learning manifolds of distributions. The GMM parameters for the component priors, means, and covariances $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K_b}$ are each represented by their own principal axes. The latent space $(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y})$ is the coefficients on the principal axes, and projecting them onto the principal axes yields the GMM parameters. Figure 1a illustrates the mappings between the statistical manifold and latent space. By minimizing the KL divergence between the original GMMs and their reconstructions through the latent space, we obtain the parameters in f^{-1}, and also a continuous and inter-

*Contact Author

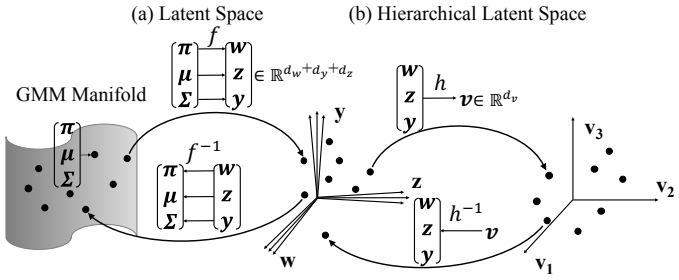

Figure 1: Learning a Parametric Manifold for GMMs. (a) The GMM parameters $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ for the component priors, means, and covariances are mapped to a latent space $(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y})$ corresponding to coefficients on the principal axes of the parameters. The forward mapping f is obtained via an optimization problem to minimize reconstruction error (KL divergence), while the inverse mapping f^{-1} is obtained directly by projecting the latent variables onto the principal axes of the parameters. (b) The latent space $(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y})$ is mapped to a hierarchical latent space $\boldsymbol{v}$ to further reduce the dimension, which models dependencies among the component priors, means, and covariances.

pretable latent space for the statistical manifold. In addition, the latent space $(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y})$ is mapped to a hierarchical latent space $\boldsymbol{v}$ to further reduce the dimension (see Figure 1b), which models dependencies among the latent space coefficients (and hence among the component priors, means, and covariances).

Our contributions are three-fold: 1) we propose a parametric manifold of GMMs that can be explicitly generated from latent variables; 2) we propose an optimization method based on variational approximation to learn the parametric mapping, and derive a fast solver by alternating optimization and Metroplis-Hastings sampling; 3) we empirically show that our method can reconstruct GMMs well and learn a smooth and interpretable latent space in several application domains.

2 Related Work

Given a set of probabilistic models (PMs), with each PM representing one example in the dataset, the relationship among the PMs can be uncovered through clustering, to obtain discrete groups of common models, or through dimensionality reduction, to obtain a latent space where directions in the latent space correspond to changes in the PM.

The hierarchical EM (HEM) algorithm is a seminal work in clustering PMs [Vasconcelos and Lippman, 1999], and was first proposed to cluster Gaussian distributions. The Gaussians are collected into a "base" GMM, from which a "reduced" GMM is estimated with fewer number of components. The components in the reduced GMM serve as the representative Gaussians for the clusters, and the cluster memberships map between the Gaussians of the base GMM and the reduced GMM. To avoid high computation cost, virtual samples are generated from the base GMM and a closed-form solution is derived that expresses the reduced GMM in terms of the parameters of the base GMM. Later, HEM was extended to cluster time-series PMs: DTs [Chan *et al.*, 2010] and HMMs [Coviello *et al.*, 2012]. As the original HEM algorithm was derived for clustering, [Yu *et al.*, 2018] recently propose a density-preserving HEM algorithm for simplifying a GMM into an equivalent GMM with fewer components, while minimizing the distortion as measured by KLD.

These clustering methods only obtain a discrete representation of the set of PMs, i.e., a finite set of representative models and cluster assignments. In contrast, we propose to learn a manifold of PMs, which is a *continuum* of representative models specified by the latent space coefficients. Hence, our model can better visualize changes in the structure of the PMs.

In our learning algorithm, we adopt a variational approximation to the expected log-likelihood, which is similar to that of DPHEM [Yu *et al.*, 2018]. The main difference is that DPHEM takes a *single* input GMM and reduces the number of components to obtain a single output GMM. In contrast, our formulation takes a *set* of GMMs and embeds them into a parametric manifold by minimizing the reconstruction loss of the GMMs. DPHEM is a special case of our framework when there is one input GMM and $K_m < K_b$, where K_m and K_b are the number of components in the reconstruction GMM and the input GMM, respectively. Also, the EM algorithm in our paper has no closed-form solution in the M-step, whereas the simpler M-step of DPHEM has a closed-form solution.

There are two general approaches for dimensionality reduction for PMs: kernel embedding and latent variable models. Kernel embedding explicitly models the mapping from input space to latent variables using a kernel function (or distance function). Hence, PMs can be embedded into a low-dimensional space by using a suitably-defined kernel function over probability distributions. For example, kernel PCA [Schölkopf *et al.*, 1998] can be used with the KL kernel [Moreno *et al.*, 2004] or probability product kernel [Jebara *et al.*, 2004] to perform dimensionality reduction on a set of GMMs. Based on information geometry [Amari and Nagaoka, 2007], [Carter *et al.*, 2009] propose Fisher Information Nonparametric Embedding (FINE), which computes geodesics on the Riemannian manifold of distributions, and then uses multi-dimensional scaling (MDS) to obtain embeddings. The advantage of these kernel methods is that the forward mapping from distributions to embedding coordinates can be obtained explicitly. However, the disadvantage is that the inverse mapping from embedding coordinates to distribution is difficult and requires solving the pre-image problem, which hinders interpretation of the embedding space and its relationship with the input space. In contrast to kernel methods, our method explicitly constructs the inverse mapping from latent space to probability space.

Latent variable models solve the problem in the opposite way: they model the generative process, i.e. inverse mapping from low-dimensional latent variables to high-dimensional variables. For example, Gaussian Process Latent Variable Models (GPLVM) [Lawrence, 2004] obtains non-linear inverse mappings using a kernel matrix on the latent variables. However, the high-dimensional variables are still treated as vectors, and thus GPLVM cannot naturally represent structured non-vector data, such as probability distributions. While it is possible to also kernelize the high-dimensional variable, this leads to the same pre-image problem as KPCA above. Similar to GPLVM, our method is also a generative model, but in contrast to GPLVM, we construct an explicit parametric mapping from the latent space to the probability distribution.

AutoEncoders (AEs) are another type of nonlinear embedding method with latent variables. AEs first map input vectors to a latent space (encoder) and then reconstruct input vectors from the latent variables (decoder). Using neural network encoder/decoders, AEs can model complex mapping functions for high-dim input vectors. However, previous AE works have not considered how to handle multi-modal distribution input such as GMMs – using vectorized GMMs as inputs to the AE does not address the identifiability problem caused by component permutations. Our method is invariant to component permutation due to the KLD loss and the variational parameters as assignment indicators.

3 Parametric Manifold Learning of GMMs

Let $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K_b}$ be the parameters of a GMM with K components, where π_k is the prior probability of the kth component, and $\boldsymbol{\mu}_k \in \mathbb{R}^D$ and $\boldsymbol{\Sigma}_k \in \mathbb{S}_+^{D\times D}$ are the mean and covariance matrix of the kth component. The probability distribution for a GMM is $p(\boldsymbol{x}) = \sum_{k=1}^{K_b} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Our goal is to learn a mapping between the space of GMMs and a lower-dimensional latent space, i.e., to learn the forward mapping $f : p(\boldsymbol{x}) \rightarrow (\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y})$ where $\boldsymbol{w} \in \mathbb{R}^{d_w}$, $\boldsymbol{z} \in \mathbb{R}^{d_z}$, $\boldsymbol{y} \in \mathbb{R}^{d_y}$ are the latent variables for component prior, mean and covariance, respectively, as well as an inverse mapping $f^{-1} : (\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y}) \rightarrow p(\boldsymbol{x})$.

3.1 Parametric Manifold for GMMs

In contrast to kernel methods, here we focus on explicitly constructing the inverse mapping from latent space to probability space. Following PCA reconstruction, we define a set of principal axes and corresponding coefficients $(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y})$ for each GMM parameter (prior, mean, covariance), from which the parameters can be reconstructed. The latent space variables are used to reconstruct a GMM with K_m components (possibly different from K_b), with parameters $\{\hat{\pi}_r, \hat{\boldsymbol{\mu}}_r, \hat{\boldsymbol{\Sigma}}_r\}_{r=1}^{K_m}$. The rth component of the reconstructed GMM is defined by a set of principal axes $\{\boldsymbol{a}_r, \{\boldsymbol{m}_{rl}\}_{l=1}^{d_z}, \{\boldsymbol{C}_{rl}\}_{l=1}^{d_y}\}$ and offsets $\{\boldsymbol{b}_r, \beta_r\}$, where $\boldsymbol{a}_r \in \mathbb{R}^{d_w}$, $\boldsymbol{m}_{rl}, \boldsymbol{b}_r \in \mathbb{R}^D$, $\boldsymbol{C}_{rl} \in \mathbb{R}^{D\times D}$ and $\beta_r \in \mathbb{R}$, according to

$$\begin{aligned} \hat{\pi}_r &= \frac{\sigma(\boldsymbol{w}^T\boldsymbol{a}_r)}{\sum_{n=1}^{K_m}\sigma(\boldsymbol{w}^T\boldsymbol{a}_n)}, \;\; \hat{\boldsymbol{\mu}}_r = \sum_{l=1}^{d_z} z_l \boldsymbol{m}_{rl} + \boldsymbol{b}_r, \\ \hat{\boldsymbol{\Sigma}}_r^{-1} &= \sum_{l=1}^{d_y} \log(1+\exp(y_l))\boldsymbol{C}_{rl}\boldsymbol{C}_{rl}^T + \beta_r^2\boldsymbol{I}, \end{aligned} \tag{1}$$

where $\sigma(x) = 1/(1+\exp(x))$ is the sigmoid function, and y_l, z_l are the l-th coefficients of $\boldsymbol{y}$ and $\boldsymbol{z}$.

Similar to PCA, the mean $\hat{\boldsymbol{\mu}}_r$ is a linear combination of principal axes $\boldsymbol{m}_{rl}$, weighted by z_l, and an offset vector $\boldsymbol{b}_r$. The reconstructed precision matrix $\hat{\boldsymbol{\Sigma}}_r^{-1}$ is a linear combination of $\boldsymbol{C}_{rl}\boldsymbol{C}_{rl}^T$ and an offset $\beta_r^2\boldsymbol{I}$. The reason for reconstructing the precision matrix in this way is three-fold. First, the positive definite constraint of $\hat{\boldsymbol{\Sigma}}_r$ is naturally fulfilled since the weights $\log(1+\exp(y_l))$ are always non-negative. Second, when the latent variable $\boldsymbol{y} \ll \boldsymbol{0}$, then the precision matrix will be a "default" value (i.e., a fixed level of uncertainty). For increasing values of the latent variable $\boldsymbol{y}$, the precision will increase, i.e., the covariance (uncertainty) decreases. Thus the latent variable naturally interpolates between different shapes of covariance matrices and a default covariance. Third, the gradients are easier to compute when defining the reconstruction through the precision matrix. For priors $\hat{\pi}_r$, the sigmoid function has more stable gradients, c.f., the gradients of the softmax function that change rapidly due to the exponential function. Also note that the probability constraints (non-negative and sum to 1) on the prior are naturally fulfilled by the formulation.

Note that the latent variables $(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y})$ are shared among all the components of the reconstructed GMM, although each reconstructed component has its own set of principal axes. Furthermore, there is no need to define an explicit correspondence between the rth component of the reconstructed GMM and kth component of the input GMM, since the learning algorithm uses the reconstruction loss between the whole input GMM and the whole reconstruction GMM – the ordering of the components in the input GMMs will not affect the embedding. Finally, PCA is a special case of our formulation in (1) when there is only one component $K_b = K_m = 1$, and the latent variable $\boldsymbol{y} \ll \boldsymbol{0}$ and β_r is a constant (see supplemental [Liu *et al.*, 2019]). From this perspective, our method is more universal than vanilla PCA.

3.2 Learning with EM Optimization

We next propose a learning algorithm for estimating the reconstruction parameters and latent variables from training data. Given a training set of N GMMs, let $p_i(\boldsymbol{x})$ be the distribution for the ith GMM with parameters $\{\pi_{ik}, \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}\}_{k=1}^{K_b}$. Denote the corresponding latent variables as $(\boldsymbol{w}_i, \boldsymbol{z}_i, \boldsymbol{y}_i)$, the reconstructed GMM as $\{\hat{\pi}_{ir}, \hat{\boldsymbol{\mu}}_{ir}, \hat{\boldsymbol{\Sigma}}_{ir}\}_{r=1}^{K_m}$, and its distribution as $\hat{p}_i(\boldsymbol{x})$. The reconstruction parameters $\boldsymbol{\Theta} = \{\boldsymbol{a}_r, \{\boldsymbol{m}_{rl}\}_{l=1}^{d_z}, \{\boldsymbol{C}_{rl}\}_{l=1}^{d_y}, \boldsymbol{b}_r, \beta_r\}_{r=1}^{K_m}$ and latent variables for the training data $\boldsymbol{\Omega} = \{\boldsymbol{w}_i, \boldsymbol{z}_i, \boldsymbol{y}_i\}_{i=1}^{N}$ are obtained by minimizing the reconstruction loss between $p_i(\boldsymbol{x})$ and $\hat{p}_i(\boldsymbol{x})$, measured by KLD [Kullback, 1997],

$$\{\boldsymbol{\Theta}^*, \boldsymbol{\Omega}^*\} = \underset{\boldsymbol{\Theta},\boldsymbol{\Omega}}{\operatorname{argmin}} \sum_{i=1}^{N} D_{KL}(p_i \| \hat{p}_i), \tag{2}$$

where $D_{KL}(p\|q) = \int p(x)\log\frac{p(x)}{q(x)}dx$ is the KLD between p and q. Decomposing the KLD and removing the first term, which is a constant w.r.t. $\{\boldsymbol{\Theta}, \boldsymbol{\Omega}\}$ yields an equivalent optimization problem to minimize the cross-entropy loss,

$$\begin{aligned} J_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}) &= -\sum_{i=1}^{N}\int p_i(\boldsymbol{x})\log\hat{p}_i(\boldsymbol{x}|\boldsymbol{\Theta}, \boldsymbol{\Omega}_i)d\boldsymbol{x} \\ &= -\sum_{i=1}^{N}\int\sum_{k=1}^{K_b}\pi_{ik}\mathcal{N}_{ik}(\boldsymbol{x})\log\sum_{r=1}^{K_m}\hat{\pi}_{ir}\hat{\mathcal{N}}_{ir}(\boldsymbol{x})d\boldsymbol{x}, \end{aligned} \tag{3}$$

where $\mathcal{N}_{ik}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})$ is the original Gaussian and $\hat{\mathcal{N}}_{ir}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\hat{\boldsymbol{\mu}}_{ir}, \hat{\boldsymbol{\Sigma}}_{ir})$ is its reconstruction.

Variational Approximation

As the cross-entropy between two GMMs in (3) is intractable, we derive an approximation based on a variational upper-bound inspired by [Yu *et al.*, 2018]. Introducing the variational parameters $\boldsymbol{q} = \{q_{kr}^{(i)}\}$, (3) is approximated (see supplemental

[Liu *et al.*, 2019]),

$$J_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}) \leq \tilde{J}_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \boldsymbol{q}) \quad (4)$$
$$= -\sum_{i=1}^{N}\sum_{k=1}^{K_b}\pi_{ik}\sum_{r=1}^{K_m} q_{kr}^{(i)}\left[\log \tfrac{\hat{\pi}_{ir}}{q_{kr}^{(i)}} + \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}_{ik}}[\log \hat{\mathcal{N}}_{ir}(\boldsymbol{x})]\right]$$

The variational parameter $q_{kr}^{(i)}$ can be interpreted as a soft assignment value for assigning the kth component of the ith GMM to the rth component of its reconstruction.

Variational Optimization Algorithm

Using $\tilde{J}_{CE}$, we minimize an upper bound to J_{CE},

$$\{\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Omega}}, \hat{\boldsymbol{q}}\} = \underset{\boldsymbol{\Theta},\boldsymbol{\Omega},\boldsymbol{q}}{\operatorname{argmin}} \tilde{J}_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \boldsymbol{q}). \quad (5)$$

We adopt an alternating (variational EM) algorithm to solve the optimization problem:

(i) *Variational M-step*: Given $\hat{\boldsymbol{q}}$, optimize the manifold and latent variables: $\{\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Omega}}\} = \operatorname{argmin}_{\boldsymbol{\Theta},\boldsymbol{\Omega}} \tilde{J}_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \hat{\boldsymbol{q}})$.

(ii) *Variational E-step*: Given $\{\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Omega}}\}$, calculate the optimal variational parameters: $\hat{\boldsymbol{q}} = \operatorname{argmin}_{\boldsymbol{q}} \tilde{J}_{CE}(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Omega}}, \boldsymbol{q})$.

For (ii), the optimal values of the variational parameters $\hat{q}_{kr}^{(i)}$ is derived analytically,

$$\hat{q}_{kr}^{(i)} = \frac{\hat{\pi}_{ir}\hat{\mathcal{N}}_{ir}(\boldsymbol{\mu}_{ik})\exp\{-\frac{1}{2}\operatorname{tr}(\hat{\boldsymbol{\Sigma}}_{ir}^{-1}\boldsymbol{\Sigma}_{ik})\}}{\sum_{n=1}^{K_m}\hat{\pi}_{in}\hat{\mathcal{N}}_{in}(\boldsymbol{\mu}_{ik})\exp\{-\frac{1}{2}\operatorname{tr}(\hat{\boldsymbol{\Sigma}}_{in}^{-1}\boldsymbol{\Sigma}_{ik})\}}. \quad (6)$$

The M-step is solved efficiently by alternating optimization of the parameters, i.e., using fast solvers for sub-problems of optimizing one set of parameters while keeping others fixed. For example, $\boldsymbol{m}_{rl}$ and $\boldsymbol{b}_r$ can be obtain in closed-form, $\boldsymbol{C}_{rl}\boldsymbol{C}_{rl}^T$ can be solved by semidefinite programming, $\boldsymbol{a}_r$ and β_r can be solved by the Newton-Raphson method. Finally, the optimizer may converge to a local minimum due to $\hat{\boldsymbol{q}}$. To help escape from local minima, after each iteration, we use a Metropolis-Hasting sampler for $\hat{\boldsymbol{q}}$ [Metropolis *et al.*, 1953; Hastings, 1970], which randomly swaps assignments for a random Gaussian component (see supplemental [Liu *et al.*, 2019]).

Regularization

In (1), the latent variables and principal axes are unconstrained, and thus multiple equivalent solutions exist by scaling the latent variables and principal axes in opposite directions. To remove this ambiguity, we apply regularization on the latent variables, which effectively constrains the principal axes, using a regularized objective function,

$$\tilde{J}_{CE}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \boldsymbol{q}) + \sum_{i=1}^{N}\left(c_w\|\boldsymbol{w}_i\|^2 + c_z\|\boldsymbol{z}_i\|^2 + c_y\|\boldsymbol{y}_i\|^2\right).$$

To better condition the assignment variables $q_{kr}^{(j)}$ and prevent degeneration to uniform assignments, following previous work [Vasconcelos and Lippman, 1999; Yu *et al.*, 2016], we introduce virtual samples where the variables $\boldsymbol{x}$ are replicated with i.i.d. distributions. Using N_v virtual samples, the optimal variational parameters are,

$$\hat{q}_{kr}^{(i)} = \frac{\hat{\pi}_{ir}\hat{\mathcal{N}}_{ik}(\boldsymbol{\mu}_{ik})^{N_v}\exp\{-\frac{1}{2}N_v\operatorname{tr}(\hat{\boldsymbol{\Sigma}}_{ir}^{-1}\boldsymbol{\Sigma}_{ik})\}}{\sum_{n=1}^{K_m}\hat{\pi}_{in}\hat{\mathcal{N}}_{in}(\boldsymbol{\mu}_{ik})^{N_v}\exp\{-\frac{1}{2}N_v\operatorname{tr}(\hat{\boldsymbol{\Sigma}}_{in}^{-1}\boldsymbol{\Sigma}_{ik})\}}.$$

This expression is similar to the deterministic annealing [Rose, 1998], derived from the maximum entropy principle to avoid poor local optima (see supplemental [Liu *et al.*, 2019]).

Dataset	Metric	KPCA	GPLVM	FINE	PML-GMM
Synthetic	KL Loss	505.6	4.678	-	**5.419e-2**
	LDA Acc	-	-	-	-
Eye Fixations	KL Loss	1.749	0.8257	-	**0.7100**
	LDA Acc	51.5%	51.5%	45.5%	**81.8%**
Flow Cyto (K_b=2)	KL Loss	5.196	4.284	-	**2.447**
	LDA Acc	90.0%	**95.0%**	50%	**95.0%**
Flow Cyto (K_b=1)	KL Loss	5.258	1.689	-	**1.680**
	LDA Acc	35%	**100%**	90%	**100%**
Social Checkin	KL Loss	7.668	16.784	-	**2.7398**
	LDA Acc	43.8%	43.8%	62.5%	**81.3%**

Table 1: KL reconstruction loss for held-out test GMMs and LDA classification accuracy in the latent space.

3.3 Inference

After learning the manifold $\hat{\boldsymbol{\Theta}}$, a novel GMM is embedded in the manifold by minimizing the cross-entropy between the novel GMM $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ and its reconstruction,

$$(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y}) = \underset{\boldsymbol{w},\boldsymbol{z},\boldsymbol{y}}{\operatorname{argmin}} -\int\sum_{k=1}^{K_b}\pi_k\mathcal{N}_k(\boldsymbol{x})\log\sum_{r=1}^{K_m}\hat{\pi}_r\hat{\mathcal{N}}_r(\boldsymbol{x})d\boldsymbol{x} + c_w\|\boldsymbol{w}\|^2 + c_z\|\boldsymbol{z}\|^2 + c_y\|\boldsymbol{y}\|^2. \quad (7)$$

The optimization problem is solved using the same algorithm in Section 3.2, but keeping the manifold parameters $\hat{\boldsymbol{\Theta}}$ fixed.

3.4 Hierarchical Latent Space

In (1), we use different latent variables to embed the prior, means and covariances to allow flexibility in representation. However, this treats the generation of each set of parameters independently. The dependencies among the prior, mean, and covariances is further modeled using a hierarchical latent space (HLS), which reduces the dimension of the latent space (LS). The HLS can also be used to visualize the GMM manifold in a 2D or 3D space. We assume a linear relationship between HLS and LS, $[\boldsymbol{w}_i^T\boldsymbol{y}_i^T\boldsymbol{z}_i^T]^T = \boldsymbol{H}\boldsymbol{v}_i$, where $\boldsymbol{v}_i \in \mathbb{R}^{d_v}$ are the HLS variables, and the matrix $\boldsymbol{H} \in \mathbb{R}^{(d_w+d_y+d_z)\times d_v}$ consists of orthonormal basis vectors. Now the Variational M-step optimizes w.r.t. $\{\boldsymbol{\Theta}, \boldsymbol{H}, \boldsymbol{v}\}$ (see supplemental [Liu *et al.*, 2019]).

4 Experiments

To demonstrate the effectiveness of our method, we conduct experiments on GMMs on four different domains: 1) synthetic data, 2) flow cytometry data [Aghaeepour *et al.*, 2013], 3) eye-fixation data [Chan *et al.*, 2018] and 4) social check-in data [Cho *et al.*, 2011]. For comparisons, we also learn the latent space using GPLVM and KPCA, both using the Gaussian kernel for vector inputs. To convert a GMM into a vector, we first transform its parameters so that valid GMMs can be obtained in the reconstruction stage – we map the prior to an unconstrained space using the inverse softmax function, and use a Cholesky decomposition for the covariance matrix. The transformed parameters of the GMM are then concatenated into a long vector. As the GMM vector is determined by the order of components during the concatenation, the component orders will affect the result of embedding. We normalize the order as follows. For all GMMs in the dataset, the parameters for each Gaussian component are converted into vectors. and then

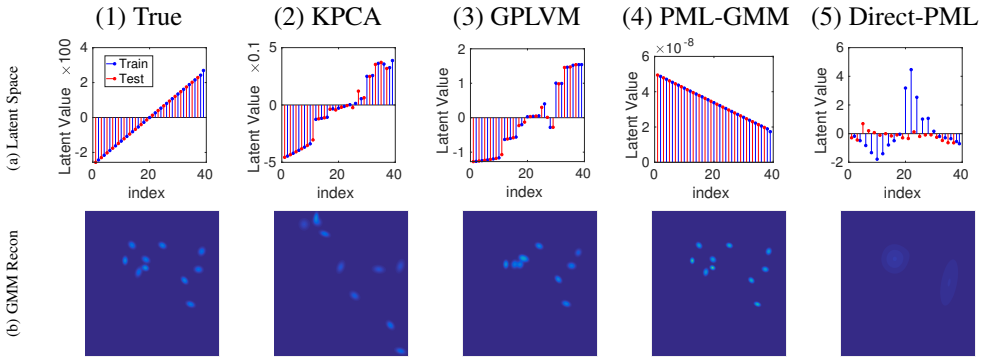

Figure 2: Experiment on Synthetic Data. The positions of latent variables are kept the same as that of ground truth. PML-GMM learns a smooth latent structure and achieve the best reconstruction.

mapped to a 1-D line using PCA. The PCA coefficient for each Gaussian component then determines the order during concatenation. Finally, we compare with FINE [Carter *et al.*, 2009], which originally embeds Gaussians into a low-dimensional space, by extending it to embed GMMs. The KLD between GMMs is computed by variational approximation [Hershey and Olsen, 2007].

We evaluate the methods in 4 ways: 1) the KLD reconstruction loss on a held-out test-set of GMMs; 2) the correlation of the latent space with respect to other dependent variables (metadata); 3) classification accuracy using latent discriminant analysis (LDA) in the latent space; 4) visualization of the GMM manifold. Specifically, for (3), we use LDA to learn a 1D discriminant space from the trained latent variables, then map test latent variables to the same space and use k Nearest Neighbors to do classification. The accuracy measures the correlation between latent space and data labels, as well as the effectiveness of latent variables in downstream tasks. The class labels for the eye fixation, flow cytometry, and social check-in datasets are older/young, healthy/unhealthy, and living city, respectively. Note that we put the experiment result of social checkin data in Supplemental [Liu *et al.*, 2019].

4.1 Summary of Quantitative Results

We first present a quantitative evaluation of the methods based on the reconstruction loss of the held-out test data, and test accuracy of using LDA in the latent space. Table 1 shows the result. In terms of reconstruction loss, PML-GMM outperforms other methods, which demonstrates that it generalizes well to novel GMMs. For the accuracy metric, PML-GMM outperforms KPCA and FINE. PML-GMM obtains the same performance as GPLVM when the component number is small, K_b=$\{1, 2\}$ on Flow Cytometry. However, when K_b increases (Synthetic and Social Checkin), GPLVM performs worse due to the component ordering problem when vectorizing the GMM parameters. These results show that the latent space obtained by PML-GMM is more effective at revealing the underlying structure in the data.

4.2 Synthetic Data

The synthetic dataset consists of 39 synthetic GMMs with 10 components, whose mean and covariances are generated from a 1D latent space (Fig. 2(a1)), according to some latent functions (see supplemental [Liu *et al.*, 2019]), and the priors are uniform. 20 GMMs are used for training (blue points) and the other 19 (red points) are reserved for testing. For a fair comparison, the latent space dimension is set as 1 for all methods. PML-GMM learns both a smooth latent space and achieve the best reconstruction error (Table 1), compared with other methods. KPCA neither reconstructs GMMs nor learns a smooth latent space. GPLVM reconstructs some of the GMMs but fails on others, and the latent space is not consistent with ground truth. See more visualizations in [Liu *et al.*, 2019].

Our framework uses a two-stage estimation procedure: 1) subsets of data (e.g., corresponding to subjects) are summarized using GMMs; 2) the GMM manifold is estimated from the individual GMMs. One reasonable alternative to our framework is to directly learn the GMM manifold from the data samples, denoted as Direct-PML (see supplemental [Liu *et al.*, 2019]). Fig. 2(5) shows an example using Direct-PML to learn a manifold of synthetic GMMs. Direct-PML can neither learn a good GMM manifold (Average KLD loss of 2.749) compared to PML-GMM, nor learn a right latent space.

4.3 Eye-Fixation Data

The eye-fixation data [Chan *et al.*, 2018] consists of eye-fixation coordinates of 34 young adults and 67 older adults (34 for training, 33 for testing) when recognizing faces. We model each person's eye-fixation pattern with a GMM, where each Gaussian component corresponds to a region-of-interest (ROI) on the face. As suggested by [Chan *et al.*, 2018], we use K=3 components corresponding to 3 ROIs. For PML-GMM, the latent space is set to 6 dimensions ($d_w = d_z = d_y = 2$) and the HLS is set to $d_v = 3$.

The latent space is shown in Fig. 3. Only in the latent space of PML-GMM (Fig. 3a) do we observe that there are different regions for older (AD quadrant) and young (CD quadrant) adults, and the test data consisting of older adults (black plus points) are all embedded into the older region (AD quadrant). In contrast, the other three methods embed testing GMMs into their latent spaces in an undesirable way (see LDA accuracy in Table 1). We examine the correlation between the HLS and the subject's age using multivariate linear regression analysis (see 2). The HLS of PML is correlated with ages at a statistically

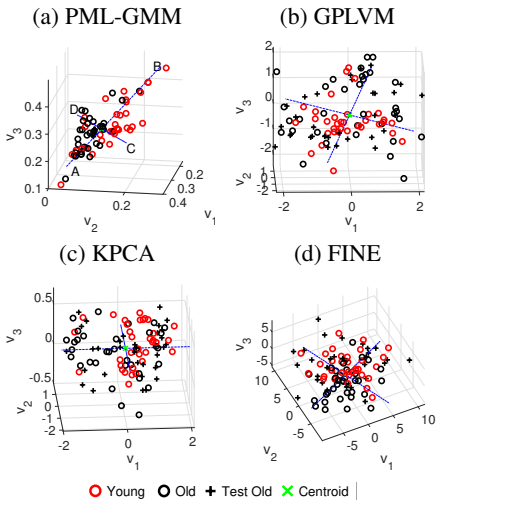

Figure 3: Latent spaces of Eye-Fixation Data. PML-GMM can learn meaningful regions corresponding to older (AD quadrant) and young (BC quadrant) adults, and embeds test data to the correct region.

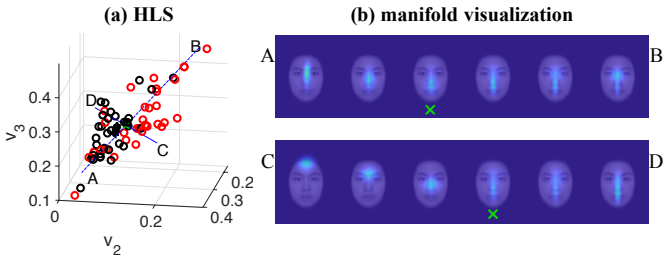

Figure 4: Manifold Visualization of Eye-Fixation Data. (b) shows the changes in eye-fixation regions of older adults (AD quadrant) to young ones (CB quadrant). Green crosses show GMMs nearest to the centroid.

significant level and has the largest R^2 statistic.

	PML-GMM	GPLVM	KPCA	FINE
p	**<0.0001**	0.1586	**0.0157**	**0.0001**
R^2	**0.3180**	0.0773	0.1485	0.2703

Table 2: Eye fixation data: correlation between the HLS and age using multivariate linear regression analysis.

For PML we visualize the manifold along the two principal axes in Fig. 4b. Along $\overline{AB}$, from the centroid towards A shows a vertically shaped ROI going up towards the upper center of the face, and towards B shows a vertically shaped ROI at the nose and a more horizontally shaped ROI around the eyes. Along $\overline{CD}$, from the centroid to C shows a horizontally shaped ROI going upwards, whereas towards D shows a vertically shaped ROI going downwards. As older adults' ROIs focus on the face midline, their ROIs are vertically shaped, and thus they are embedded into the AD quadrant. In contrast, young adults look around the eye regions and have horizontal ROIs around the eyes, and thus they are embedded into the BC quadrant. This finding is consistent with the previous paper

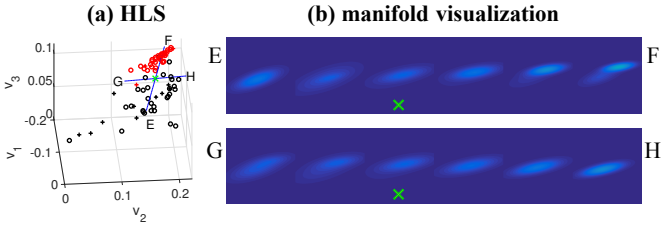

Figure 5: Manifold Visualization of Flow Cytometry Data. (a) HLS of PML-GMM. (b) $\overline{EF}$ visualizes the change from unhealthy cells to healthy cells and $\overline{GH}$ visualizes the middle region between healthy and unhealthy.

[Chan *et al.*, 2018] but we visualize the continuous change of eye gaze strategy, instead of only discrete clusters as in [Chan *et al.*, 2018].

4.4 Flow Cytometry Data

Flow cytometry data is used in medical diagnosis to test for unhealthy disorders by measuring cell properties in patients. The dimensionality of flow cytometry data samples ranges from 5-8, and the number of points for one patient is often thousands, which makes direct analyses cumbersome. Here we use an open AML dataset [Aghaeepour *et al.*, 2013] with 7-dim features. 30 healthy and 30 unhealthy patients are used for training, and another 10 healthy and 10 unhealthy are used for testing. We use the same LS/HLS dimensions as Sec. 4.3. We run EM on each patient's data using $K \in \{2, 3, 4, 5\}$ and find that when $K > 2$ there are many subjects ($> 80\%$) with low-weight components. Hence, we use either K_b=2 or K_b=1 GMMs to model each patient. Although PML-GMM and GPLVM performs the same in classification accuracy, PML-GMM achieves 43% lower reconstruction loss than GPLVM (see Table 1). Furthermore, PML-GMM can easily visualize GMMs from the latent space (see Fig. 5), while GPLVM cannot give a good visualization due to the poor reconstruction loss. In Fig. 5b, $\overline{EF}$ shows that unhealthy cells change to healthy cells by gradually separating two components. $\overline{GH}$ shows the region between the unhealthy and healthy, and includes both 1 component (unhealthy) and 2 components GMMs (healthy). See [Liu *et al.*, 2019] for latent spaces learnt by KPCA, GPLVM and FINE.

5 Conclusion

In this paper, we propose a parametric method to learn the manifold of GMMs, which both learns a parametric mapping from latent space to GMM parameters, and obtains a continuous and interpretable latent space. Future work will increase the representation power of the HLS by nonlinear mapping and extend PML to other important probabilistic models like HMMs.

Acknowledgements

This work was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 17609117 and CityU 11200314).

References

[Aghaeepour *et al.*, 2013] Nima Aghaeepour, Greg Finak, Holger Hoos, Tim R Mosmann, Ryan Brinkman, Raphael Gottardo, Richard H Scheuermann, FlowCAP Consortium, DREAM Consortium, et al. Critical assessment of automated flow cytometry data analysis techniques. *Nature methods*, 10(3):228, 2013.

[Amari and Nagaoka, 2007] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2007.

[Carter *et al.*, 2009] Kevin M Carter, Raviv Raich, William G Finn, and Alfred O Hero III. Fine: Fisher information nonparametric embedding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2093–2098, 2009.

[Chan *et al.*, 2010] Antoni B Chan, Emanuele Coviello, and Gert RG Lanckriet. Clustering dynamic textures with the hierarchical EM algorithm. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2022–2029. IEEE, 2010.

[Chan *et al.*, 2018] Cynthia YH Chan, Antoni B Chan, Tatia MC Lee, and Janet H Hsiao. Eye-movement patterns in face recognition are associated with cognitive decline in older adults. *Psychonomic bulletin and review*, pages 1–8, 2018.

[Cho *et al.*, 2011] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility:user movement in location-based social networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, Ca, Usa, August*, pages 1082–1090, 2011.

[Coviello *et al.*, 2012] Emanuele Coviello, Gert R Lanckriet, and Antoni B Chan. The variational hierarchical EM algorithm for clustering hidden Markov models. In *Advances in neural information processing systems*, pages 404–412, 2012.

[Doretto *et al.*, 2003] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.

[Hastings, 1970] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[Hershey and Olsen, 2007] John R Hershey and Peder A Olsen. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *ICASSP*, 2007.

[Jebara *et al.*, 2004] Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5(Jul):819–844, 2004.

[Kullback, 1997] Solomon Kullback. *Information theory and statistics*. Courier Corporation, 1997.

[Lawrence, 2004] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems*, pages 329–336, 2004.

[Liu *et al.*, 2019] Ziquan Liu, Lei Yu, Janet H. Hsiao, and Antoni B. Chan. Parametric manifold learning of Gaussian mixture models-Supplementary. http://visal.cs.cityu.edu.hk/static/pubs/conf/ijcai19-pml-gmm-supp.pdf, 2019.

[Metropolis *et al.*, 1953] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[Moreno *et al.*, 2004] Pedro J Moreno, Purdy P Ho, and Nuno Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In *Advances in neural information processing systems*, pages 1385–1392, 2004.

[Perronnin *et al.*, 2006] Florent Perronnin, Christopher Dance, Gabriela Csurka, and Marco Bressan. Adapted vocabularies for generic visual categorization. In *European Conference on Computer Vision*, pages 464–475, 2006.

[Rabiner, 1989] Lawrence R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Rose, 1998] Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239, 1998.

[Sánchez *et al.*, 2013] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the Fisher vector: theory and practice. *International Journal of Computer Vision*, 105(3):222–245, Dec 2013.

[Schölkopf *et al.*, 1998] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[Vasconcelos and Lippman, 1999] Nuno Vasconcelos and Andrew Lippman. Learning mixture hierarchies. In *Advances in Neural Information Processing Systems*, pages 606–612, 1999.

[Winn *et al.*, 2005] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Tenth IEEE International Conference on Computer Vision*, pages 1800–1807 Vol. 2, 2005.

[Yu *et al.*, 2016] Lei Yu, Tianyu Yang, and Antoni B Chan. Approximate inference for generic likelihoods via density-preserving GMM simplification. In *Workshop on Advances in Neural Information Processing Systems*, 2016.

[Yu *et al.*, 2018] Lei Yu, Tianyu Yang, and Antoni B Chan. Density-preserving hierarchical EM algorithm: Simplifying Gaussian mixture models for approximate inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):1–1, 2018.