Interpolation on the manifold of K component GMMs

[†]Hyunwoo J. Kim [†]Nagesh Adluru Monami Banerjee[§] Baba C. Vemuri[§] Vikas Singh[†]
[†]University of Wisconsin–Madison [§]University of Florida
http://pages.cs.wisc.edu/~hwkim/projects/k-gmm

Abstract

Probability density functions (PDFs) are fundamental objects in mathematics with numerous applications in computer vision, machine learning and medical imaging. The feasibility of basic operations such as computing the distance between two PDFs and estimating a mean of a set of PDFs is a direct function of the representation we choose to work with. In this paper, we study the Gaussian mixture model (GMM) representation of the PDFs motivated by its numerous attractive features. (1) GMMs are arguably more interpretable than, say, square root parameterizations (2) the model complexity can be explicitly controlled by the number of components and (3) they are already widely used in many applications. The main contributions of this paper are numerical algorithms to enable basic operations on such objects that strictly respect their underlying geometry. For instance, when operating with a set of K component GMMs, a first order expectation is that the result of simple operations like interpolation and averaging should provide an object that is also a K component GMM. The literature provides very little guidance on enforcing such requirements systematically. It turns out that these tasks are important internal modules for analysis and processing of a field of ensemble average propagators (EAPs), common in diffusion weighted magnetic resonance imaging. We provide proof of principle experiments showing how the proposed algorithms for interpolation can facilitate statistical analysis of such data, essential to many neuroimaging studies. Separately, we also derive interesting connections of our algorithm with functional spaces of Gaussians, that may be of independent interest.

1. Introduction

Gaussian mixture models (GMM) are a fundamental statistical tool deployed in a broad spectrum of applications in computer vision. These include modeling the foreground scribbles for segmentation [22], tracking [12], discriminant analysis [29], registration [15], action recognition [21], image indexing [26] and computing motion features [5]. Their properties are well studied and efficient implementations are available as part of popular software libraries in computer vision, machine learning and statistics.

A K component GMM (K-GMM for short) is a probability density function given as a weighted sum of K Gaussian densities,

$$p(x|\Theta) = \sum_{j=1}^{K} \pi^j \mathcal{N}(x|\mu^j, \Sigma^j) \tag{1}$$

where the mean and covariance of the mixing components are given by μ^j and Σ^j respectively, π^j gives the corresponding weight and $\Theta = \{\mu^j, \Sigma^j\}_{j=1}^K$. Let $\mathbf{G} = \{\mathcal{G}_1^K, \cdots, \mathcal{G}_N^K\}$ denote a set of N K-GMMs. This paper studies the problem of interpolating between $\mathcal{G}_1^K, \cdots, \mathcal{G}_N^K$ to derive an interpolant, $\hat{\mathcal{G}}$. Our main requirement on $\hat{\mathcal{G}}$ is that it should correspond to a K-GMM for a given K. In addition to this constraint, based upon the needs of the specific application, the interpolation task may correspond to an averaging operation over $\mathbf{G}$ or alternatively, when $|\mathbf{G}| = \mathbf{2}$, we may ask for a continuous interpolation $\Gamma(\mathcal{G}_i^K, t)$ such that $\Gamma(\mathcal{G}_i^K, 0) = \mathcal{G}_i^K$ and $\Gamma(\mathcal{G}_i^K, 1) = \mathcal{G}_j^K$ for any i, j and for any offset, $t \in [0, 1]$. The question of whether this problem permits efficient solution schemes is interesting enough in its own right to merit careful investigation. It turns out that such an algorithm, if available, will be immediately applicable to (or facilitate) a variety of tasks in computer vision, machine learning and medical imaging with minor changes. Below, as motivation, we provide a sampling of such applications.

Problem 1: Spatial transformations of diffusion PDFs [10, 7, 8]. An important scientific frontier today is to establish a connectome of the human brain [23]. Diffusion weighted magnetic resonance (MR) is one of the tools being used to help answer the underlying analysis questions. It exploits the physical phenomenon of diffusion of water to image the microstructure of the white matter pathways in the brain [7]. An object estimated from such MR measurements is the so-called ensemble average propagator (EAP), a PDF describing the diffusivity profiles of water molecules on spheres of varying radii at the micrometer scale. The

EAP can be conveniently represented as a K-GMM which can help resolve up to K crossing of white matter pathways at a voxel. Now, given two images (source and target) where each voxel has a K-GMM, the registration task involves applying a spatial transform to the source image to align it with the target image. Recall that the most basic routine needed in applying such a transformation is a way to estimate a 'value' for each voxel in the transformed image via interpolation (e.g., bi-linear). Since both the source and target images are a field of K-GMMs, an interpolation routine for K-GMMs is essential – in contrast, a naïve interpolation here will output a (NK)-GMM if $|\mathbf{G}| = N$, clearly blowing up the model complexity.

Problem 2: Matching point sets [15]. Consider the problem of matching one point set to another where we seek the best alignment between the transformed "model" set and the target "scene" set — common in shape matching and model-based segmentation. In contrast to identifying point-to-point correspondence, a class of fairly successful recent approaches [18] statistically model each of the two point sets by a PDF. Then, a suitable distance measure between the two distributions, $d(\cdot,\cdot)$ is minimized over the transformation parameters, τ. Kernel density based and GMM based representations are quite popular. Assume that the two point sets are defined as S and T. To align K-GMM$(\tau(S))$ and K-GMM(T), the optimization proceeds by taking incremental steps along $\nabla_\tau d$, until convergence. However, right after the first gradient update, we leave the feasibility region of K component GMMs. As a result, most methods are unable to provide intermediate evolution steps along the transformation that are members of the same set as the source and the target models, i.e., a K-GMM. In contrast, with a minor modification (i.e., plugging in our method), this ability can be obtained with a nominal additional cost.

Problem 3: Statistical compressed sensing [27]. Let $\mathbf{f} \in \mathbf{R}^p$ be a function (or signal) and $\Phi \in \mathbf{R}^{N\times p}$ denote the so-called sensing matrix. We are provided measurements $\mathbf{y} = \Phi\mathbf{f}$. The recovery of $\mathbf{f}$ from $\Phi\mathbf{f}$ is ill-posed in general when $N \ll p$. Compressed sensing significantly generalizes the regime under which such recovery is possible based on incoherence between the sensing and a certain 'representation' basis, see [9]. Statistical compressed sensing (SCS) takes this argument further by considering the situation where one is interested in reconstructing not just one but an entire sequence of signals, $\mathbf{f}_1, \mathbf{f}_2 \cdots$. Here, SCS assumes that $\mathbf{f}_i$ is drawn from a GMM — which enables additional improvements in recovery. When deployed in a 'streaming' setup, the current GMM prior in SCS (say, at time t) is incrementally updated based on the current measurement $(t + 1)$. Our proposed algorithm offer a potential improvement: by providing a *moving average* version of the to-be-updated GMM prior by constructing a weighted (or unweighted) mean of the previous t GMMs. This will likely be immune to local fluctuations or noise in the streaming measurements.

The main **contributions** of this paper are to develop a systematic framework for performing interpolation on the manifold of K component GMMs. It will take in as input a set of GMMs and a specific interpolation task and provide a K-GMM as an output that optimizes the interpolation objective. While the primary focus of this work is theoretical, we provide experiments demonstrating the expected behavior of the algorithm. Separately, we highlight some interesting connections of this formulation with *functional* spaces of Gaussians. Next, section 2 introduces some basic concepts relevant to K-GMMs. Sections 3 and 4 present our core algorithms followed by experimental results and conclusions in sections 5 and 6 respectively.

2. Preliminaries

To our knowledge, there are no existing algorithms for interpolating a set of K-GMMs; on the other hand, there *is* a mature body of research for tackling the setting where the objects to be interpolated are probability density functions (PDFs) [24, 4, 19, 17]. So one might ask, why not simply use PDFs? We will present several specific reasons in the section below.

Observe that the actual formulation for interpolation will depend on the specific parameterization we choose to represent the PDF as well as the distance metric. To make this point concrete, let us review a few example parameterizations and distance metrics. With these two pieces, the corresponding interpolation/averaging operation is simple to derive. Evaluation of their advantages or limitations in the K-GMM setting will then become apparent.

2.1. PDF parameterizations and distances

Parameterization. First, let us consider a simple expression for computing the mean of probability densities $\{f_i\}_{i=1}^N$,

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{N} w_i d(\Phi(f), \Phi(f_i))^2 \qquad (2)$$

where $\Phi(\cdot)$ is a mapping function for parameterizing the given probability densities, $d(\cdot,\cdot)$ is a distance metric and w_i is a weight for f_i. Some parameterizations will allow using tools from differential geometry for deriving efficient algorithms [24]. Clearly, there are multiple options for parameterization but some specific ones form a set (the so called unit Hilbert sphere in ℓ_2-space) and are mathematically convenient. We can parameterize a given set of PDFs so that they lie in this set. The mapping is bijective when restricted to non-negative functions i.e., *every* element in the unit Hilbert sphere can be mapped back to a PDF. For example, the square root parameterization simply takes the square-root of the PDF value. If, for example, the PDF was parameterized using a K-GMM then,

$$f(x|\Theta) = \sqrt{p(x|\Theta)} = \sqrt{\sum_{j=1}^{K} \pi^j \mathcal{N}(x|\mu^j, \Sigma^j)} \qquad (3)$$

By inspection, the ℓ_2-norm of f is always 1 since $\sqrt{\int f(x)f(x)dx} = \int p(x)dx = 1$. Notice that this is a *re*-parameterization of the original PDF (which was provided as a K-GMM).

Normalization. Alternatively, we can normalize the PDFs by dividing by the ℓ_2-norm, which only changes the scale and *not* the shape of the model.

$$p'(x) = p(x)/\|p(x)\|_2, \qquad (4)$$

where $\|\cdot\|_2$ is the standard ℓ_2-norm for functions. For the special case of GMMs, we have

$$\|\mathcal{G}_i\|_2^2 = \sum_{j}^{K}\sum_{j'}^{K} \pi^j \pi^{j'} \mathcal{N}(\mu^j|\mu^{j'}, \Sigma^j + \Sigma^{j'}), \qquad (5)$$

where $\mathcal{G}_i$ denotes a representative GMM.

Distances. Let us now consider the calculation of distances. Let $p'_i(x) = p_i(x)/\|p_i(x)\|_2$. Recall that for two different functions, the ℓ_2-distance is given as

$$\|f_1 - f_2\|_2 = \left(\int_X |f_1(x) - f_2(x)|^2 d\mu(x)\right)^{1/2}. \qquad (6)$$

Then, the normalized ℓ_2-distance ($d_{n\text{-}\ell_2}$) is simply the ℓ_2-distance between the normalized PDFs [13],

$$\begin{aligned} d_{n\text{-}\ell_2}(p_1, p_2) &= \int (p'_1(x) - p'_2(x))^2 dx \qquad (7) \\ &= 2(1 - \int_{\mathcal{X}} p'_1(x)p'_2(x)dx). \end{aligned}$$

Geodesics and Divergences. Instead of the ℓ_2-distance, we can also calculate the geodesic distance on the unit Hilbert sphere. Let $p'_i(x) = p_i(x)/\|p_i(x)\|_2$. Then, the geodesic distance between normalized PDFs is

$$d_{n\text{-geo}}(p_1, p_2) = \cos^{-1}\langle p'_1, p'_2\rangle_2 = \cos^{-1}(\int_{\mathcal{X}} p'_1(x)p'_2(x)dx)$$

This is interesting because the geodesic distance here admits a closed form solution.

The KL-divergence [16] is another possibility, albeit *not* a metric, that can be used as a information theoretic divergence between probability density functions $f(x)$ and $g(x)$. It is also known as relative entropy and given by

$$D(f||g) := \int f(x) \log \frac{f(x)}{g(x)} dx. \qquad (8)$$

The KL-divergence between two GMMs cannot be obtained analytically and so various approximations have been proposed [11]. Shortly, we will discuss the relationship between the KL-divergence/cross entropy and the log likelihood which will suggest natural EM style algorithms.

How many components? PDFs and K-GMMs. With these concepts in hand, it is easy to verify what happens when we seek to interpolate GMMs but the only tool we have available is an interpolation routine for PDFs. In general, given a set of GMMs, if we consider them simply as PDFs, the mean derived from the geodesic distance (with the square root parameterization) may not even be a GMM. However, it turns out (proof given later), that the simple arithmetic mean of PDFs, i.e., $\bar{f} = \sum_i^N f_i/N$ is optimal with respect to the ℓ_2-metric for PDFs. Unfortunately, the main difficulty is that when given N GMMs with K components each, the arithmetic mean solution will *not* be a K component GMM (instead, a GMM with $N \times K$ components),

$$\bar{\mathcal{G}} = \sum_{i}^{n} \mathcal{G}_i/N = \underbrace{\sum_{i=1}^{N}\sum_{j=1}^{K}}_{N\times K \text{ components}} \frac{\pi_i^j}{N} \mathcal{N}(\mu_i^j, \Sigma_i^j).$$

If one needs the interpolation of K-GMMs to be a K-GMM, to our knowledge, there are no existing solutions. We address this problem in the later sections with a focus on ℓ_2-distance and KL-divergence/cross entropy which, roughly speaking, corresponds to the least squares and log-likelihood functions of a finite number of samples in the classical GMM setting.

3. A gradient descent scheme for ℓ_2-distance

Let $\mathbf{G}^{(K)}$ denote the manifold of K-GMMs. We will first describe an optimization scheme to directly minimize the ℓ_2-distance in $\mathbf{G}^{(K)}$ which is used for the interpolation objective.

Computing the ℓ_2-mean in $\mathbf{G}^{(K)}$. First, we will derive an algorithm for calculating the mean for a set $\mathbf{F} = \{\mathcal{F}_1, \cdots, \mathcal{F}_n\}$ where $\forall j, \mathcal{F}_i \in \mathbf{G}^{(K)}$ w.r.t ℓ_2 metric. Second, for the case where $|\mathbf{F}| = 2$, we will derive a 'path' from $\mathcal{F}_i$ to $\mathcal{F}_j$, which never leaves the feasibility region i.e., $\mathbf{G}^{(K)}$. This construction will provide a meaningful distance measure which respects the geometry of $\mathbf{G}^{(K)}$.

The ℓ_2-mean (arithmetic mean) of $\{\mathcal{F}_n\}_{n=1}^N$ minimizes the sum of squared ℓ_2-distances to each $\mathcal{F}_i \in \mathbf{F}$,

$$\bar{\mathcal{F}} = \arg\min_{\mathcal{G}} \sum_{n=1}^{N} \|\mathcal{G} - \mathcal{F}_n\|_2^2 \qquad (9)$$

As discussed in Section 2, we have $\bar{\mathcal{F}} \in \mathbf{G}^{(NK)}$ (the blowup in the number of components). Instead, we require a GMM $\hat{\mathcal{G}} \in \mathbf{G}^{(K)}$. Our algorithm has two steps. First, we find $\bar{\mathcal{F}}$ and then find the closest K-component GMM to $\bar{\mathcal{F}}$, i.e., we will minimize (10)

$$\hat{\mathcal{G}} = \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} \|\mathcal{G} - \bar{\mathcal{F}}\|_2^2 \tag{10}$$

This may seem like a very loose relaxation. That is, is there a $\hat{\mathcal{G}}' \in \mathbf{G}^{(K)}$ that is farther from $\bar{\mathcal{F}}$ but achieves a lower objective function value for (9)? The following result shows that this cannot be the case.

Lemma 1. *The mean of a finite number of functions $\{\mathcal{F}_n\}_n^N$ with respect to ℓ_2 metric is the closest $\mathcal{G}^*$ to the ℓ_2-mean $\bar{\mathcal{F}} = \sum_n^N \frac{\mathcal{F}_n}{N}$.*

Proof.

$$\begin{aligned}
\mathcal{G}^* &= \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} \sum_n^N \|\mathcal{F}_n - \mathcal{G}\|_2^2 \\
&= \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} \sum_n^N \|\mathcal{F}_n\|_2^2 - 2\sum_n^N \langle \mathcal{F}_n, \mathcal{G}\rangle_2 + N\|\mathcal{G}\|_2^2 \\
&= \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} \frac{1}{N}\sum_n^N \|\mathcal{F}_n\|_2^2 - 2\left\langle \frac{\sum_n^N \mathcal{F}_n}{N}, \mathcal{G}\right\rangle_2 + \|\mathcal{G}\|_2^2 \\
&= \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} -2\left\langle \frac{\sum_n^N \mathcal{F}_n}{N}, \mathcal{G}\right\rangle_2 + \|\mathcal{G}\|_2^2 \\
&= \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} \|\frac{1}{N}\sum_n^N \mathcal{F}_n\|_2^2 - 2\langle \frac{1}{N}\sum_n^N \mathcal{F}_n, \mathcal{G}\rangle_2 + \|\mathcal{G}\|_2^2 \\
&= \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} \|\frac{1}{N}\sum_n^N \mathcal{F}_n - \mathcal{G}\|_2^2 \\
&= \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} \|\bar{\mathcal{F}} - \mathcal{G}\|_2^2 \qquad \square
\end{aligned}$$

This result suggests that (10) is indeed equivalent to (9) with the constraint $\mathcal{G} \in \mathbf{G}^{(K)}$.

Optimization scheme. To optimize (10), we first initialize the solution and then perform incremental gradient descent steps. The main terms in the gradient update step are described below and are computed using $\bar{\mathcal{F}}$ and $\mathcal{G}$, the former has $L(= NK)$ components and the latter has K components.

Let $\mathcal{L}$ denote the objective function in (10). The three main variables to optimize over are the component weights $\pi_{\mathcal{G}}^j$, means $\mu_{\mathcal{G}}^j$ and covariances $\Sigma_{\mathcal{G}}^j$, where i and j index components in $\bar{\mathcal{F}}$ and $\mathcal{G}$ respectively. Let $c_{\mathcal{G},\bar{\mathcal{F}}}^{j,i} := \mathcal{N}(\mu_{\mathcal{G}}^j | \mu_{\mathcal{F}}^i, \Sigma_{\mathcal{G}}^j + \Sigma_{\mathcal{F}}^i)$. The derivative w.r.t. $\pi_{\mathcal{G}}^j$ takes the form,

$$\frac{\partial \mathcal{L}}{\partial \pi_{\mathcal{G}}^j} = 2\left(\sum_{j'=1}^K \pi_{\mathcal{G}}^{j'} c_{\mathcal{G},\mathcal{G}}^{j,j'} - \sum_{i=1}^L \pi_{\bar{\mathcal{F}}}^i c_{\mathcal{G},\bar{\mathcal{F}}}^{j,i}\right)$$

The derivative w.r.t. $\mu_{\mathcal{G}}^j$ is given as

$$\frac{\partial \mathcal{L}}{\partial \mu_{\mathcal{G}}^j} = 2\pi_{\mathcal{G}}^j \left(\sum_{j' \neq j}^K \pi_{\mathcal{G}}^{j'} \frac{\partial}{\partial \mu_{\mathcal{G}}^j} c_{\mathcal{G},\mathcal{G}}^{j,j'} - \sum_{i=1}^L \pi_{\bar{\mathcal{F}}}^i \frac{\partial}{\partial \mu_{\mathcal{G}}^j} c_{\mathcal{G},\bar{\mathcal{F}}}^{j,i}\right),$$

whereas the derivative $\frac{\partial \mathcal{L}}{\partial \Sigma_{\mathcal{G}}^j}$ is

$$\left(\pi_{\mathcal{G}}^j\right)^2 \frac{\partial}{\partial \Sigma_{\mathcal{G}}^j} c_{\mathcal{G},\mathcal{G}}^{j,j} + 2\pi_{\mathcal{G}}^j \left(\sum_{j' \neq j}^K \pi_{\mathcal{G}}^{j'} \frac{\partial}{\partial \Sigma_{\mathcal{G}}^j} c_{\mathcal{G},\mathcal{G}}^{j,j'} - \sum_{i=1}^L \pi_{\mathcal{F}}^i \frac{\partial}{\partial \Sigma_{\mathcal{G}}^j} c_{\mathcal{G},\bar{\mathcal{F}}}^{j,i}\right).$$

The extended version of the paper includes the detailed derivations. The gradient is calculated by putting together the three terms above and the step size is determined using a standard line search procedure [20]. We repeat until convergence.

Special case: Identifying a path in $\mathbf{G}^{(K)}$ between $\mathcal{F}_{start}$ and $\mathcal{F}_{end}$. A special case for the interpolation scheme above is when we want to interpolate between just two K component GMMs, $\mathcal{F}_{\text{start}}$ and $\mathcal{F}_{\text{end}}$, and recover a shortest path $\{\mathcal{G}_t\}_{t=1}^T$ that does not leave the feasibility region, $\mathbf{G}^{(K)}$ and $\mathcal{G}_0 = \mathcal{F}_{\text{start}}$ and $\mathcal{G}_{T+1} = \mathcal{F}_{\text{end}}$. As can be expected, one can identify such a path with a minor change of the algorithm described above. Then, our objective function is,

$$\min_{\{\mathcal{G}_t\}_{t=1}^T} \sum_{t=0}^T \|\mathcal{G}_t - \mathcal{G}_{t+1}\|_2^2, \text{ s.t. } \mathcal{G}_t \in \mathbf{G}^{(K)} \ \forall t. \tag{11}$$

Letting $d_T := \sum_{t=0}^T \|\mathcal{G}_t^* - \mathcal{G}_{t+1}^*\|_2$, we have $\lim_{T\to\infty} d_T = d(\mathcal{F}_{\text{start}}, \mathcal{F}_{\text{end}})$, the geodesic distance between $\mathcal{F}_{\text{start}}$ and $\mathcal{F}_{\text{end}}$ in $\mathbf{G}^{(K)}$. Further details on the minimization are given in the extended version.

4. An EM algorithm for KL-divergence

Our initial experiments reveal that minimizing ℓ_2-distance via gradient descent with the constraint of staying on the $\mathbf{G}^{(K)}$ manifold is technically correct but prone to instability due to many local optima. For example, the gradient descent method works well when the covariance matrices are diagonally dominant (isotropic) but tends to yield unsatisfactory results when the estimated covariances matrices need to be projected back to satisfy the "$\succeq 0$" constraint. To address this issue, we describe an alternate algorithm that avoids such a projection step. To motivate this setup, observe that in the preceding section, the overall interpolation task comprised of modules/steps for finding the closest K-GMM to a given L component GMM, see (10). So, any potential solution to the foregoing numerical issue must be addressed at the level of this module.

Consider a very special case of the module above where L is arbitrary but $K = 1$. Interestingly, it turns out that if we use cross-entropy instead of the ℓ_2-distance between GMMs, Lemma 2 suggests that that there *is* a closed form solution which involves no numerical difficulties. Notice that no such result exists for ℓ_2-distance. So, if we can extend this result to the case where $K > 1$, we can efficiently solve the problem while ensuring that the procedure is numerically stable. In fact, this idea will form the core of our proposal described next where we first decouple the components in the “E” step and use a closed form solution for each component in the “M” step. In fact, our scheme optimizes the KL-divergence which is equivalent to cross-entropy in this case.

The interpolation of multiple GMMs is obtained by minimizing,

$$\mathcal{G}^* = \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} \sum_{n=1}^{N} D(\mathcal{F}_n || \mathcal{G}) \tag{12}$$

We observe that the expression in (12) is equivalent to,

$$\begin{aligned} &\arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} D(\bar{\mathcal{F}} || \mathcal{G}) \\ &= \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} \int \bar{\mathcal{F}}(x) \log \frac{\bar{\mathcal{F}}(x)}{\mathcal{G}(x)} dx \\ &= \arg \min_{\mathcal{G} \in \mathbf{G}^{(K)}} - \int \bar{\mathcal{F}}(x) \log \mathcal{G}(x) dx. \end{aligned} \tag{13}$$

Letting $\mathcal{G}(x) = \sum_{j=1}^{K} w_j g_j(x)$, the objective function is given by

$$\begin{aligned} \mathcal{G}^* &= \arg \max_{\mathcal{G} \in \mathbf{G}^{(K)}} \int \bar{\mathcal{F}}(x) \log \sum_{j=1}^{K} w_j g_j(x) dx, \\ &= \arg \max_{g \in \mathbf{G}^{(K)}} \mathbb{E}_{\bar{\mathcal{F}}(x)} [\log \sum_{j=1}^{K} w_j g_j(x)]. \end{aligned} \tag{14}$$

We note that this formulation can also be interpreted as finding the best code book in $\mathbf{G}^{(K)}$, namely, $\mathcal{G}^*(x)$ to represent $\bar{\mathcal{F}}(x)$.

The E and M steps are presented in Fig. 1. Detailed derivations are provided in the extended version.

Lemma 2. *Given GMM $f(x) := \sum_i^L \pi_i f_i(x)$, where $f_i(x)$ is a Gaussian distribution, the minimum cross entropy / KL-divergence between $f(x)$ and an unknown single Gaussian $g := \mathcal{N}(x; \mu, \Sigma)$ is obtained by (μ^*, Σ^*),*

$$(\mu^*, \Sigma^*) = \arg \min_{\mu, \Sigma} H(f(x), \mathcal{N}(x; \mu, \Sigma)), \tag{16}$$

where $\mu^* = \mathbb{E}_{f(x)}[x]$ and $\Sigma^* = \mathbb{E}_{f(x)}[(x - \mu^*)(x - \mu^*)^T]$.

The closed form of (μ^*, Σ^*) is in (15). Proof is provided in the extended paper.

In the case of EAPs (introduced in section 5), the GMMs have a special property that all μ_js are zero. It is easy to verify that if the input EAPs are comprised of zero mean Gaussians, the algorithm in Fig. 1 does yield a valid EAP (k-GMM with zero means). However, our goal is not to merely obtain ‘valid’ EAPs but to minimize the potential change in anisotropy (of the EAPs) using our interpolation. EAPs (with zero mean Gaussians) imply that their components overlap significantly at their modes. We found that their differences are much less accurately captured by cross-entropy. In practice, this may lead the algorithm towards inaccurately big ellipsoids since it averages different Gaussians (in the EAPs) with relatively similar responsibility γ_{ij}.

E-step: Let $\Theta = \{w_j, \mu_j, \Sigma_j\}_{j=1}^{K}$, $\bar{\mathcal{F}}(x) = \sum_{i=1}^{NK} \pi_i f_i(x)$ and X_i be a set of points with density function $f_i(x)$. Then we have,

$$\gamma_{ij} := p(z_i = j | X_i, \Theta) = \frac{w_j \exp\left[-H(f_i, g_j)\right]}{\sum_{j'}^{K} w_{j'} \exp[-H(f_i, g_{j'})]}$$

Note that γ_{ij} is the likelihood that the i^{th} component of $\bar{\mathcal{F}}$ corresponds to j^{th} in $\mathcal{G}$[†]. $H(f_i, g_j)$ is analytically obtained as,

$$\frac{1}{2}\{k \log 2\pi + \log |\Sigma_j| + \mathrm{tr}[\Sigma_j^{-1} \Sigma_i] + (\mu_i - \mu_j)^T \Sigma_j^{-1} (\mu_i - \mu_j)\}$$

M-step:

$$\begin{aligned} w_j &= \frac{\sum_{i=1}^{NK} \pi_i \gamma_{ij}}{\sum_{j'=1}^{K} \sum_{i'=1}^{NK} \pi_{i'} \gamma_{i'j'}} \\ \mu_j &= \mathbb{E}_{\bar{\mathcal{F}}'(x)}[x] = \sum_{i=1}^{NK} \pi_i' \mu_i \\ \Sigma_j &= \mathbb{E}_{\bar{\mathcal{F}}'(x)}[(x - \mu_j)(x - \mu_j)^T] \\ &= \sum_{i=1}^{NK} \pi_i' \Sigma_i + \sum_{i=1}^{NK} \pi_i' (\mu_i - \mu_j)(\mu_i - \mu_j)^T \end{aligned} \tag{15}$$

where $\bar{\mathcal{F}}' = \sum_{i=1}^{NK} \pi_i' f_i(x)$, and $\pi_i' = \frac{\pi_i \gamma_{ij}}{\sum_i \pi_i \gamma_{ij}}$, for fixed j.

[†] It is probably more natural to write $p(z_i = j | f_i, \theta)$ rather than $p(z_i = j | X_i, \theta)$. We choose the latter notation because it is closer to how an EM procedure for classical GMMs is typically explained and better shows the relationship between log-likelihood and cross-entropy.

Figure 1: **EM algorithm minimizing cross entropy.**

This problem is directly addressed by our modified EM algorithm in Fig 2. First, we use the ℓ_2 distance for the E-step to capture the differences. In addition, by introducing the simplest covariance function C_j for each component, we allow each component to have different densities in the functional space. In other words, even though some Gaussians within an EAP may overlap substantially, if C_j is small enough, our algorithm is still able to distinguish them nicely and assign significantly different responsibility. This makes our approach very robust.

This modified algorithm, which estimates *four* parameters for each component $(w_j, \mu_j, \Sigma_j, C_j)$, drives the EAP experiments presented in this paper. Note that as a by-product of EM algorithms, all our EM-algorithms described

E-step: Estimate the responsibilities of data PDFs to components of our model,

$$\gamma_{ij} = \frac{w_j C_j^{-1} \exp\left(-\frac{1}{2C_j^2}\|f_i - g_j\|_2^2\right)}{\sum_{k=1}^{K} w_k C_k^{-1} \exp\left(-\frac{1}{2C_k^2}\|f_i - g_k\|_2^2\right)} \tag{17}$$

M-step: Maximize cross entropy given assignments over model parameters (a weight w_j, mean function $\mathcal{N}(\mu_j, \Sigma_j)$ and a covariance function C_j).

$$C_j^2 = \sum_{i=1}^{NK} \gamma_{ij}\pi_i \|f_i - g_j\|_2^2 / \sum_{i'=1}^{NK} \gamma_{i'j}\pi_{i'} \tag{18}$$

w_j and μ_j, Σ_j are updated using Eqs. (15).

Figure 2: **Modified EM for operations on EAPs.**

in this section clusters Gaussian distributions f_i of $\bar{\mathcal{F}}$ in the functional space.

5. Experiments

In this section, we introduce the diffusion PDF of interest (EAP) and demonstrate the results of various operations such as upsampling resolution, denoising, spatial transformations on the EAP field where the basic underlying module is interpolation. We also show experiments showing that interpolation on the K-GMM manifold provides benefits in terms of controlling the number of components when one needs to perform repeated interpolations. Controlling the number of components has a direct impact on our ability to resolve the peaks in the EAP profiles which is crucial in generating tractography, a key component in deriving brain connectivity information from such imaging data [2].

Ensemble average propagator (EAP). White matter architecture can be probed by analyzing thermal diffusivity profiles of water molecules in the brain. Thermal diffusion of water causes signal decay in the measured MR signal. The decay, under certain assumptions of the MR pulse sequencing used to acquire the signal satisfies the following relationship

$$E(q\mathbf{u}) = \int_{\mathbb{R}^3} P(R\mathbf{r})\exp(2\pi i q r \mathbf{u}^T \mathbf{r})\mathrm{d}R\mathbf{r}, \tag{19}$$

where $\mathbf{u}, \mathbf{r}$ are unit vectors in $\mathbb{R}^3$, q is proportional to the amplitude of the magnetic field gradient along $\mathbf{u}$ and $P(R\mathbf{r})$ is called the ensemble average propagator (EAP) describing the diffusion displacements of water molecules [25, 3]. Assuming antipodal (radial) symmetries for the signal decay (i.e., $E(q\mathbf{u}) = E(-q\mathbf{u})$) and EAP ($P(R\mathbf{r}) = P(-R\mathbf{r})$), the following relationship holds [7]

$$P(R\mathbf{r}) = \int_{\mathbb{R}^3} E(q\mathbf{u})\cos(2\pi q r \mathbf{u}^T \mathbf{r})\mathrm{d}q\mathbf{u}. \tag{20}$$

The EAP is a PDF whose domain is $\mathbb{R}^3$. In our experiments, we use a K-GMM representation of the EAP [14]. We would like to note that our approach is also applicable to such operations on the so called orientation distribution functions (ODFs) [10, 6].

Upsampling and denoising. Signal to noise ratio (SNR) of the MR signal is proportional to the volume size of a voxel. Diffusion weighted MRI faces challenges in terms of achieving high SNR due to rapid acquisitions and hence the voxel resolution acquired on typical scanners is usually 8 mm^3. For applications like tractography, recent investigations recommend a resolution of 1.95313 mm^3[23]. But acquiring such a scan requires drastic improvements to the scanner gradient capabilities and adds significant scanning time (∼55 mins. vs. ∼10 mins.) [23]. Hence providing an upsampling algorithm and a denoising modules that can reconstruct the EAPs respecting its native geometry can be practically very useful. We simulate EAP profiles at $R = 15\mu$m in voxels at the four corners of a 6 × 6 grid as shown in Fig. 3(a) and fill in such severely undersampled data in the remainder of the grid with our algorithm. We perform a simple bi-linear interpolation to fill in the grid as shown in Fig. 3(b) using the operations introduced in Section 3. We can observe that the diffusion PDFs are smoothly interpolated respecting the geometry of the crossing fibers. To demonstrate the denoising capabilities of our algorithm we add Wishart noise to the EAPs (Fig. 3(c)). The denoised EAPs using Gaussian filtering and anisotropic filtering are shown in Figs. 3(d) and (e).

Since EAP profiles are affected by the architecture of the white matter pathways we additionally simulate EAP data to reflect crossing and curving pathways [7] and demonstrate Gaussian and anisotropic filtering as shown in Fig. 4(a). We can observe that the anisotropic filtering does near perfect recovery of the underlying signal. The red boxes in (c) highlight the differences between Gaussian and anisotropic filtering.

Spatial transformations. One of the key steps in statistical analysis of neuroimaging data is to spatially normalize the images from different subjects i.e., transform/warp each of the individual subject's image data onto a group-level standard grid. Although spatial transformations of diffusion tensor images (single component GMMs) is widely studied and used in clinical studies [28], currently there are no widely available tools for advanced diffusion PDFs such as EAPs. Note that there *are* Riemmannian interpolation schemes available [10, 6, 8] in the literature but not specifically for K-GMMs. Using our algorithm, we rotate two EAP fields by 30° and also apply affine transformations.

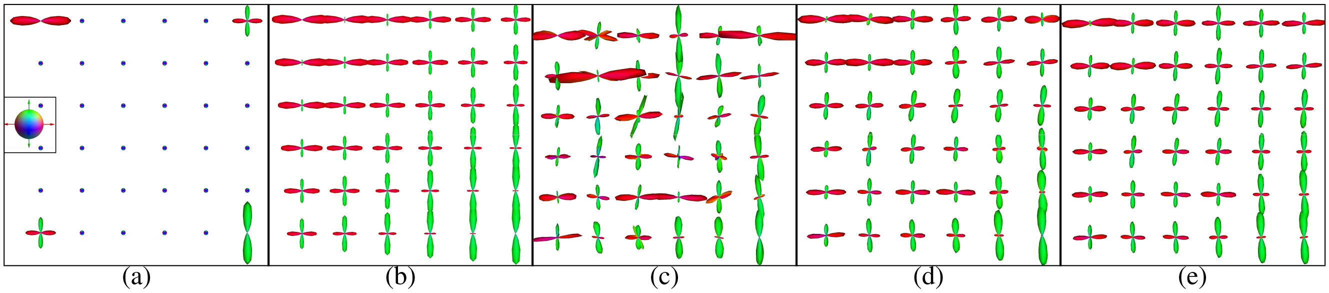

Figure 3: (a) Input data with just four voxels in the foreground, (a)-(e) are all at the same scale. The color mapping scheme used to visualize the profiles is shown in the box overlaid on the background voxels which are set to have isotropic diffusivity. (b) Result of upsampling with bi-linear interpolation. (c) Noisy EAPs. (d) Gaussian filtering. (e) Anisotropic filtering.

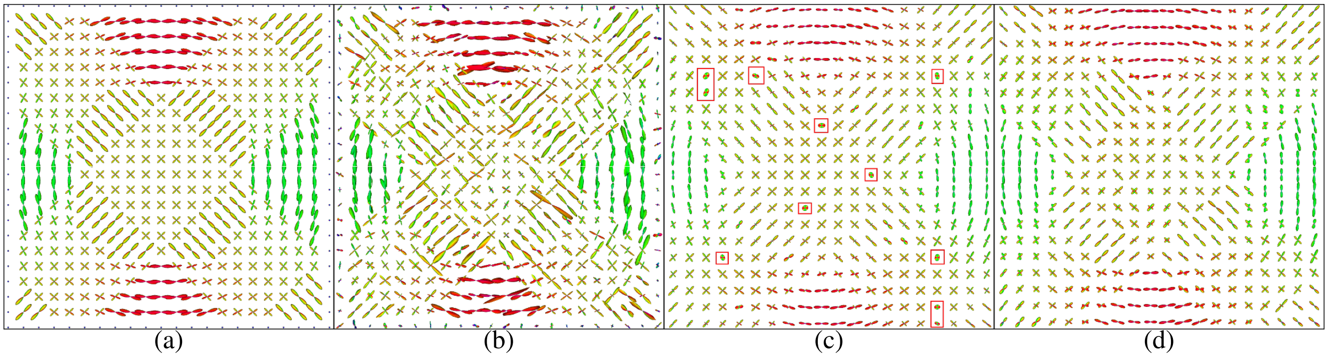

Figure 4: (a) Simulated EAP profiles. (b) EAP profiles with added Wishart noise. (c) Gaussian filtering. (d) Anisotropic filtering.

The results are shown in Fig. 5. When performing non-orthonormal transformations on the EAP fields, one needs to extract the rotation transformation to reorient the profiles. To do so, we use the finite strain method [1]. We observe that even in cases of really complex architecture our interpolation and reorientation preserve the organizational features (crossing and circular nature of the profiles) of the profiles. The shearing effects where the crossing fiber region stretches increasing the number of crossing fibers and the circular organization becomes elliptical.

Peak preserving complexity reduction. In this experiment, we demonstrate that model complexity can interfere with simple peak finding algorithms and hence it is advantageous to operate on a fixed K-GMM manifold. The error in peak detection is computed as follows,

Let K^* be the true number of peaks in the simulated EAP field. Then, the error at each voxel in an estimated/interpolated EAP field is measured by

$$\epsilon = \min_{\Pi} \sum_{i=1}^{K^*} \cos^{-1} |V_i^T U_{\Pi(i)}|. \tag{21}$$

where V_i and U_i are eigen vectors of the K^* largest weight components of ground truth and estimated EAP, respectively. $\Pi(i)$ is the best permutation which has the minimum error, i.e., when $K^* = 2$, ϵ is the minimum of angular errors between $\{V_1, V_2\}$ and $\{U_1, U_2\}$ with all pos-

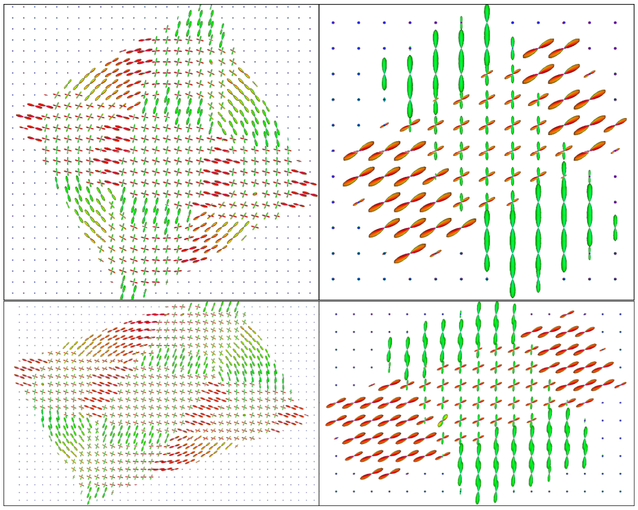

Figure 5: **Top row:** Rotated EAP profiles. **Bottom row:** Results of affine transformation of the EAP fields.

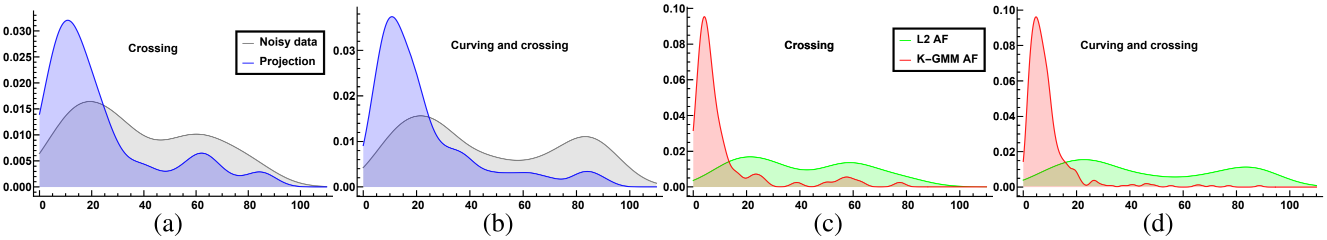

Figure 6: The distributions of angular deviations of the peaks. Comparing projected and noisy data in (a) crossing fiber phantom, and (b) curving and crossing phantom. Comparing anisotropic filtering with K-GMM (ours) and ℓ_2 interpolation in (c) crossing fiber phantom, and (d) curving and crossing phantom.

sible permutations. Hence the range of ϵ in each voxel is $[0, K^* \times 90°]$. We first add Wishart noise to the numerically simulated crossing and curving EAP profiles (see Fig. 5). Figs. 6(a) and (b) show the deviations of the peaks detected by projecting (*without any filtering*) from $\mathbf{G}^{(10)}$ to $\mathbf{G}^{(2)}$ the noisy data for crossing and curving phantoms respectively. The distribution corresponds to errors in all voxels in an image. As we can see, the errors are reduced just by reducing the number of components. Figs. 6(c) and (d) show the distributions of the angular deviations for crossing and curving after anisotropic filtering with K-GMM and ℓ_2 method. We can observe that the K-GMM method significantly outperforms the ℓ_2 method. The K-GMM method deviates on average about $10°$ while the errors with ℓ_2 are spread further especially in crossing fiber regions (i.e. $\epsilon > 90°$).

6. Conclusions

This paper describes a numerically robust scheme for performing interpolation on the manifold of K component GMMs, where few solutions are available in the literature today. Such operations are needed to perform theoretically sound processing of a field of EAPs, fundamental objects in diffusion weighted Magnetic resonance imaging. We first derive a gradient descent scheme and then use those ideas towards an efficient and numerically stable EM style method. The algorithm is general and applicable to other situations where interpolation is needed for objects such as functions, probability distributions and so on (though for some special cases, more specialized algorithms are known). Separately, notice that operating directly on the functional space of Gaussians (and their mixtures) suggested insights that were useful in obtaining our numerical procedures. Some of these issues are briefly mentioned in passing in the paper (see last paragraphs of Section 2 and Section 4) and described in more detail in the extended paper for the interested reader. We believe that with the growing interest in using advanced image analysis and statistical techniques for analyzing and making sense of rich datasets being collected worldwide (e.g., the Human Connectome project), algorithms such as the one proposed here will be valuable in ensuring that the underlying processing remains faithful to the geometry/structure of the data. Doing so will not only improve the statistical analysis but put us in the best position to extract scientifically interesting hypotheses from such images.

Acknowledgment

This work was supported in part by NIH grants AG040396 (VS), NS066340 (BCV), IIS-1525431 (BCV), NSF CAREER award 1252725 (VS), 1UL1RR025011 (NA), and P30 HD003352-45 (NA). Partial support was also provided by the Center for Predictive Computational Phenotyping (CPCP) at UWMadison (AI117924). We thank Jia Xu and Greg Plumb for contributing their time to various discussions related to the ideas described in this paper.

References

[1] D. C. Alexander, C. Pierpaoli, P. J. Basser, and J. C. Gee. Spatial transformations of diffusion tensor magnetic resonance images. *IEEE Transactions on Medical Imaging*, 20(11):1131–1139, 2001. 7

[2] M. Bastiani, N. J. Shah, R. Goebel, and A. Roebroeck. Human cortical connectome reconstruction from diffusion weighted MRI: the effect of tractography algorithm. *Neuroimage*, 62(3):1732–1749, 2012. 6

[3] P. T. Callaghan. *Principles of nuclear magnetic resonance microscopy*, volume 3. Clarendon Press Oxford, 1991. 6

[4] H. E. Cetingul, B. Afsari, M. J. Wright, P. M. Thompson, and R. Vidal. Group action induced averaging for HARDI processing. In *IEEE International Symposium on Biomedical Imaging*, pages 1389–1392, 2012. 2

[5] D. Chen and J. Yang. Exploiting high dimensional video features using layered Gaussian mixture models. In *International Conference on Pattern Recognition*, volume 2, pages 1078–1081, 2006. 1

[6] J. Cheng, A. Ghosh, T. Jiang, and R. Deriche. A Riemannian framework for orientation distribution function computing. In *Medical Image Computing and Computer-Assisted Intervention*, pages 911–918. 2009. 6

[7] J. Cheng, A. Ghosh, T. Jiang, and R. Deriche. Model-free and analytical EAP reconstruction via spherical polar fourier

diffusion MRI. In *Medical Image Computing and Computer-Assisted Intervention*, pages 590–597. 2010. 1, 6

[8] J. Cheng, A. Ghosh, T. Jiang, and R. Deriche. Diffeomorphism invariant Riemannian framework for ensemble average propagator computing. In *Medical Image Computing and Computer-Assisted Intervention*, pages 98–106. 2011. 1, 6

[9] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 2

[10] A. Goh, C. Lenglet, P. M. Thompson, and R. Vidal. A nonparametric Riemannian framework for processing high angular resolution diffusion images (HARDI). In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2496–2503, 2009. 1, 6

[11] J. R. Hershey and P. A. Olsen. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages IV–317, 2007. 3

[12] H. Idrees, N. Warner, and M. Shah. Tracking in dense crowds using prominence and neighborhood motion concurrence. *Image and Vision Computing*, 32(1):14–26, 2014. 1

[13] J. H. Jensen, D. P. Ellis, M. G. Christensen, and S. H. Jensen. Evaluation distance measures between Gaussian mixture models of MFCCs. In *International Conference on Music Information Retrieval*, pages 107–108, 2007. 3

[14] B. Jian and B. C. Vemuri. A unified computational framework for deconvolution to reconstruct multiple fibers from diffusion weighted MRI. *IEEE Transactions on Medical Imaging*, 26(11):1464–1471, 2007. 6

[15] B. Jian and B. C. Vemuri. Robust point set registration using Gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, 2011. 1, 2

[16] S. Kullback. *Information theory and statistics*. Courier Corporation, 1997. 3

[17] J. Li, Y. Shi, and A. W. Toga. Diffusion of fiber orientation distribution functions with a rotation-induced Riemannian metric. In *Medical Image Computing and Computer-Assisted Intervention*, pages 249–256. 2014. 2

[18] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010. 2

[19] S. Ncube, Q. Xie, and A. Srivastava. A geometric analysis of ODFs as oriented surfaces for interpolation, averaging and denoising in HARDI data. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 1–6, 2012. 2

[20] J. Nocedal and S. J. Wright. *Least-Squares Problems*. Springer, 2006. 4

[21] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013. 1

[22] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004. 1

[23] K. Setsompop, R. Kimmlingen, E. Eberlein, T. Witzel, J. Cohen-Adad, J. A. McNab, B. Keil, M. D. Tisdall, P. Hoecht, P. Dietz, et al. Pushing the limits of in vivo diffusion MRI for the Human Connectome Project. *Neuroimage*, 80:220–233, 2013. 1, 6

[24] A. Srivastava, I. Jermyn, and S. Joshi. Riemannian analysis of probability density functions with applications in vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 2

[25] E. Stejskal and J. Tanner. Spin diffusion measurements: spin echoes in the presence of a time-dependent field gradient. *The journal of chemical physics*, 42(1):288–292, 1965. 6

[26] N. Vasconcelos. Image indexing with mixture hierarchies. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1–8, 2001. 1

[27] G. Yu and G. Sapiro. Statistical compressed sensing of gaussian mixture models. *IEEE Transactions on Signal Processing*, 59(12):5842–5858, 2011. 2

[28] H. Zhang, P. A. Yushkevich, D. C. Alexander, and J. C. Gee. Deformable registration of diffusion tensor MR images with explicit orientation optimization. *Medical image analysis*, 10(5):764–785, 2006. 6

[29] W. Zheng, H. Tang, Z. Lin, and T. S. Huang. Emotion recognition from arbitrary view facial images. In *European Conference on Computer Vision*, pages 490–503. 2010. 1

Interpolation on the manifold of k component GMMs (supplement)

0. Summary

In this supplement, we provide proofs, detailed derivations, additional discussion and experiments. This includes 1) Full derivation of the gradient of the objective function (9) in the main 2) More details on identification of a path in $\mathbf{G}^{(K)}$ 3) Proof of Lemma 2 4) Derivation of the EM algorithm 5) Discussion of the relationship between our EM algorithms and functional clustering (or Gaussian process mixture models) 6) Visualization, AF weights and internal representation of EAP.

1. Derivation of the gradient of (9) and implementation

We begin with the Gaussian distribution and its derivatives and using this result, we derive gradient of (9) in the main text.

1.1. Gaussian distribution and its derivatives

Suppose $X \in \mathbf{R}^{d\times d}$. We have the following.

(i) $\det(cX) = c^d \det(X)$, where A is a $n \times n$ matrix.

(ii) $\frac{\partial \det(X)}{\partial X} = \det(X)X^{-T}$, see [2].

(iii) $\frac{\mathbf{a}^T X^{-1}\mathbf{b}}{\partial X} = -X^T\mathbf{a}\mathbf{b}^T X^{-T}, \forall \mathbf{a}, \mathbf{b} \in \mathbf{R}^d$, see [2].

(iv) $\frac{\partial}{\partial X} \log|\det(X)| = X^{-T}$

where for a matrix A, A^{-T} denotes an inverse followed by a transpose. We will use the facts above in derivations. Places where they are used, appear over the equality e.g. " $\overset{\text{(i)}}{=}$ ".

The density function of Gaussian is given by

$$f(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{1}$$

The derivative w.r.t. μ is given by

$$\frac{\partial f(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)}{\partial \boldsymbol{\mu}} = f(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \tag{2}$$

The derivative w.r.t. Σ is given by

$$\begin{aligned}
\frac{\partial f(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)}{\partial \Sigma} &= \frac{\partial}{\partial \Sigma}\left[\det(2\pi\Sigma)^{-1/2} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]\right] \\
&\overset{\text{(i)}}{=} (2\pi)^{-d/2}\frac{\partial}{\partial \Sigma}\left[\det(\Sigma)^{-1/2} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]\right] \\
&= (2\pi)^{-d/2}\left(\frac{\partial}{\partial \Sigma}\det(\Sigma)^{-1/2}\right) \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \\
&\quad + (2\pi)^{-d/2}\det(\Sigma)^{-1/2}\left(\frac{\partial}{\partial \Sigma}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(x-\boldsymbol{\mu})\right]\right) \\
&\overset{\text{(ii)}}{=} (2\pi)^{-d/2}\left(-\frac{1}{2}\det(\Sigma)^{-3/2}\det(\Sigma)\Sigma^{-T}\right) \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \\
&\quad + (2\pi)^{-d/2}\det(\Sigma)^{-1/2}\left(\frac{\partial}{\partial \Sigma}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]\right) \\
&= -\frac{1}{2}f(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)\Sigma^{-T} \\
&\quad + (2\pi)^{-d/2}\det(\Sigma)^{-1/2}\left(\frac{\partial}{\partial \Sigma}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]\right) \\
&\overset{\text{(iii)}}{=} -\frac{1}{2}f(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)\Sigma^{-T} + \frac{1}{2}f(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)\Sigma^{-T}(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-T}
\end{aligned}$$

1.2. Complete derivation of (9) in the main text

In the main text, the loss function (9) consists of the ℓ_2 norms of GMMs. We complete the derivation of (9) in the main with the partial derivatives of $c^{j,i}_{\mathcal{G},\mathcal{F}} := \mathcal{N}(\boldsymbol{\mu}^j_{\mathcal{G}}|\boldsymbol{\mu}^i_{\bar{\mathcal{F}}}, \Sigma^i_{\mathcal{G}} + \Sigma^i_{\mathcal{F}})$ w.r.t $\boldsymbol{\mu}_{\mathcal{G}}$ and $\Sigma_{\mathcal{G}}$. The derivatives are obtained by derivatives of Gaussian in (2) and (3).

The derivatives w.r.t $\boldsymbol{\mu}^j_{\mathcal{G}}$ are given by

$$\begin{aligned}
\frac{\partial c^{j,i}_{\mathcal{G},\bar{\mathcal{F}}}}{\partial \boldsymbol{\mu}^j_{\mathcal{G}}} &= -c^{j,i}_{\mathcal{G},\bar{\mathcal{F}}}(\Sigma^j_{\mathcal{G}} + \Sigma^i_{\bar{\mathcal{F}}})^{-1}(\boldsymbol{\mu}^j_{\mathcal{G}} - \boldsymbol{\mu}^i_{\bar{\mathcal{F}}}), \text{ by (2)} \\
\frac{\partial c^{j,j'}_{\mathcal{G},\mathcal{G}}}{\partial \boldsymbol{\mu}^j_{\mathcal{G}}} &= -c^{j,j'}_{\mathcal{G},\mathcal{G}}(\Sigma^j_{\mathcal{G}} + \Sigma^{j'}_{\mathcal{G}})^{-1}(\boldsymbol{\mu}^j_{\mathcal{G}} - \boldsymbol{\mu}^{j'}_{\mathcal{G}}), \text{ by (2)}
\end{aligned} \tag{3}$$

The derivatives w.r.t $\Sigma^j_{\mathcal{G}}$ are given by

$$\begin{aligned}
\frac{\partial c^{j,i}_{\mathcal{G},\bar{\mathcal{F}}}}{\partial \Sigma^j_{\mathcal{G}}} &= -\frac{1}{2}c^{j,i}_{\mathcal{G},\bar{\mathcal{F}}}(\Sigma^j_{\mathcal{G}} + \Sigma^i_{\bar{\mathcal{F}}})^{-T} \\
&\qquad [I - (\boldsymbol{\mu}^j_{\mathcal{G}} - \boldsymbol{\mu}^i_{\bar{\mathcal{F}}})(\boldsymbol{\mu}^j_{\mathcal{G}} - \boldsymbol{\mu}^i_{\bar{\mathcal{F}}})^T(\Sigma^j_{\mathcal{G}} + \Sigma^i_{\bar{\mathcal{F}}})^{-T}] \\
\frac{\partial c^{j,j'}_{\mathcal{G},\mathcal{G}}}{\partial \Sigma^j_{\mathcal{G}}} &= -\frac{1}{2}c^{j,j'}_{\mathcal{G},\mathcal{G}}(\Sigma^j_{\mathcal{G}} + \Sigma^{j'}_{\mathcal{G}})^{-T} \\
&\qquad [I - (\boldsymbol{\mu}^j_{\mathcal{G}} - \boldsymbol{\mu}^{j'}_{\mathcal{G}})(\boldsymbol{\mu}^j_{\mathcal{G}} - \boldsymbol{\mu}^{j'}_{\mathcal{G}})^T(\Sigma^j_{\mathcal{G}} + \Sigma^{j'}_{\mathcal{G}})^{-T}]
\end{aligned} \tag{4}$$

When $j = j'$, it is simplified as

$$\begin{aligned}\frac{\partial c^{j,j}_{\mathcal{G},\mathcal{G}}}{\partial \Sigma^j_{\mathcal{G}}} &= -2^{-(d/2+1)} c^{j,j}_{\mathcal{G},\mathcal{G}} (\Sigma^j_{\mathcal{G}})^{-T} \\ &= -\frac{1}{2}\mathcal{N}(\boldsymbol{\mu}^j_{\mathcal{G}} | \boldsymbol{\mu}^j_{\mathcal{G}}, 2\Sigma^j_{\mathcal{G}})(\Sigma^j_{\mathcal{G}})^{-1}\end{aligned} \tag{5}$$

2. Identifying a path in $\mathrm{G}^{(K)}$ between $\mathcal{F}_{start}$ and $\mathcal{F}_{end}$ w.r.t l-2 distance

Given two GMMs, $\mathcal{F}_{\text{start}}$ and $\mathcal{F}_{\text{end}}$, we seek to find the shortest path which does not leave the feasibility region, $\mathbf{G}^{(K)}$. The result from such a procedure will directly provide a potentially more meaningful distance measure between two samples in $\mathbf{G}^{(K)}$.

To do so, we will approximate it by a set of smaller paths along other GMMs with K components. By minimizing the sum of the squared distances between adjacent GMMs, we will approximate the shortest path. It is similar to chordal distance approximation on the sphere, see Figure 1. In the limit, this will be the true shortest length.

On a Riemannian manifold $\mathcal{M}$ with metric tensor g, the length of a continuously differentiable curve $\gamma : [a, b] \to \mathcal{M}$ is defined by

$$L(\gamma) = \int_a^b \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}\, dt \tag{6}$$

where L is its length and $g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))$ is the inner product of $\dot{\gamma}(t)$ at $\gamma(t)$ w.r.t g. When γ is the shortest geodesic curve, it is called *geodesic distance*.

The arc length L of γ can be approximated by

$$L(C) = \sup_{a=t_0<t_1<\cdots<t_n=b} \sum_{i=0}^{n-1} d(\gamma(t_i), \gamma(t_{i+1})) \tag{7}$$

where the supremum is taken over all possible partitions of $[a, b]$ and n is unbounded. Then, the objective is to minimize the sum of squared distances of each chordal segment.

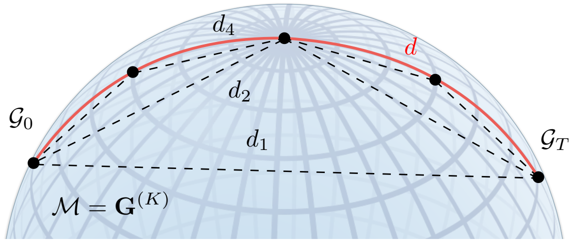

Figure 1: ℓ_2 distance between $\mathcal{G}_0$ and $\mathcal{G}_T$ is d_1. Similar to chordal distance, the geodesic distance between two GMMs can be approximated by the sum of ℓ_2 distances between many intermediate GMMs. It converges to the real path length as the number of chords t increases. Here $d_1 \leq d_2 \leq \ldots \leq d$, where d is the true distance.

So, we square each term in the objective function and minimize their sum.

Let $\mathcal{G}_0$ ($\mathcal{G}_{T+1}$ resp.) be $\mathcal{G}_{\text{start}}$ ($\mathcal{G}_{\text{end}}$ resp.). Then, our objective function is given as

$$\min \mathcal{L} := \min_{\{\boldsymbol{\mu}_t, \boldsymbol{\pi}_t, \boldsymbol{\Sigma}_t\}_{t=1}^T} \sum_{t=0}^{T} \|\mathcal{G}_t - \mathcal{G}_{t+1}\|_2^2, \tag{8}$$

using the shorthand notations $\boldsymbol{\pi}_t := \{\boldsymbol{\pi}_t^j\}_{j=1}^K$, $\boldsymbol{\mu}_t := \{\mu_t^j\}_{j=1}^K$ and $\boldsymbol{\Sigma}_t := \{\boldsymbol{\Sigma}_t^j\}_{j=1}^K$.

Again, to compute the gradient, we take the derivative with respect to the relevant variables which include the component weights, means and their covariances. Similarly, define $c^{j,j'}_{t,t'} := \mathcal{N}(\boldsymbol{\mu}^j_t | \boldsymbol{\mu}^{j'}_{t'}, \Sigma^j_t + \Sigma^{j'}_{t'})$. The derivatives of (7) w.r.t π^j_t, μ^j_t, Σ^j_t are related to only the following terms

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_t} &:= \frac{\partial}{\partial \boldsymbol{\mu}_t}\left[\|\mathcal{G}_{t-1} - \mathcal{G}_t\|_2^2 + \|\mathcal{G}_t - \mathcal{G}_{t+1}\|_2^2\right] \\ &= \frac{\partial}{\partial \boldsymbol{\mu}_t}\left[\|\mathcal{G}_{t-1}\|_2^2 - 2\langle \mathcal{G}_{t-1}, \mathcal{G}_t\rangle_2 + 2\|\mathcal{G}_t\|_2^2 - 2\langle \mathcal{G}_t, \mathcal{G}_{t+1}\rangle_2 + \|\mathcal{G}_{t+1}\|_2^2\right] \\ &= \frac{\partial}{\partial \boldsymbol{\mu}_t}\left[2\|\mathcal{G}_t\|_2^2 - 2\langle \mathcal{G}_t, \mathcal{G}_{t+1}\rangle_2 - 2\langle \mathcal{G}_t, \mathcal{G}_{t-1}\rangle_2\right].\end{aligned} \tag{9}$$

Recall that the inner product of two GMMs $\mathcal{G}_t$ and $\mathcal{G}_{t'}$ in $\mathbf{G}^{(K)}$ is given by

$$\langle \mathcal{G}_t, \mathcal{G}_{t'}\rangle_2 = \sum_{j=1}^K \sum_{j'=1}^K \pi^j_t \pi^{j'}_{t'} \mathcal{N}(\mu^j_t | \mu^{j'}_{t'}, \Sigma^j_t + \Sigma^{j'}_{t'}) = \sum_{j=1}^K \sum_{j'=1}^K \pi^j_t \pi^{j'}_{t'} c^{j,j'}_{t,t'} \tag{10}$$

Then the derivatives w.r.t π^j_t, μ^j_t, Σ^j_t can be written with $c^{j,j'}_{t,t'}$. The derivative w.r.t π^j_t is given as

$$\frac{\partial \mathcal{L}}{\partial \pi^j_t} = 4\sum_{j'}^K \pi^{j'}_t c^{j,j'}_{t,t} - 2\sum_{j'}^K \pi^{j'}_{i+1} c^{j,j'}_{t,t+1} - 2\sum_{j'}^K \pi^{j'}_{i-1} c^{j,j'}_{t,t-1} \tag{11}$$

The derivative w.r.t μ^j_t is

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mu^j_t} &= 4\sum_{j'\neq j}^K \pi^j_t \pi^{j'}_t \frac{\partial}{\partial \mu^j_t} c^{j,j'}_{t,t} - 2\sum_{j'=1}^K \pi^j_t \pi^{j'}_{t+1} \frac{\partial}{\partial \mu^j_t} c^{j,j'}_{t,t+1} \\ &\quad - 2\sum_{j'=1}^K \pi^j_t \pi^{j'}_{i-1} \frac{\partial}{\partial \mu^j_t} c^{j,j'}_{t,t-1}\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \Sigma^j_t} &= 2\pi^j_t \pi^j_t \frac{\partial}{\partial \Sigma^j_t} c^{j,j}_{t,t} + 4\sum_{j'\neq j}^K \pi^j_t \pi^{j'}_t \frac{\partial}{\partial \Sigma^j_t} c^{j,j'}_{t,t} \\ &\quad - 2\sum_{j'=1}^K \pi^j_t \pi^{j'}_{t+1} \frac{\partial}{\partial \Sigma^j_t} c^{j,j'}_{t,t+1} - 2\sum_{j'=1}^K \pi^j_t \pi^{j'}_{t-1} \frac{\partial}{\partial \Sigma^j_t} c^{j,j'}_{t,t-1}\end{aligned} \tag{12}$$

In $\mathbf{G}^{(2)}$, our experimental result is shown in Fig 2.

3. Minimum cross entropy between a given GMM and an unknown single Gaussian

Lemma 2. *Given a GMM $f(x) := \sum_i^L \pi_i f_i(x)$, where $f_i(x)$ is a Gaussian distribution, the minimum cross entropy / KL-divergence between $f(x)$ and an unknown single Gaussian $g := \mathcal{N}(x; \mu, \Sigma)$ is obtained by (μ^*, Σ^*),*

$$(\mu^*, \Sigma^*) = \arg\min_{\mu,\Sigma} H(f(x), \mathcal{N}(x; \mu, \Sigma)), \tag{13}$$

where $\mu^* = \mathbb{E}_{f(x)}[x]$ and $\Sigma^* = \mathbb{E}_{f(x)}[(x - \mu^*)(x - \mu^*)^T]$.

Proof. One easily observes that

$$\arg\min_g D(f||g) = \arg\min_g H(f, g) \tag{14}$$

since f is fixed. Recall that cross entropy is given by

$$H(f, g) := E_f[-\log g(x)] = -\int f(x) \log g(x) dx \tag{15}$$

Take the derivative of objective function $H(f, g)$ w.r.t μ and set it to zero. Then we get,

$$\begin{aligned}
-\frac{\partial}{\partial \boldsymbol{\mu}} \int f(\boldsymbol{x}) \log g(\boldsymbol{x}) dx &= \kappa \frac{\partial}{\partial \boldsymbol{\mu}} \int f(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) dx \\
&= \kappa' \int f(\boldsymbol{x}) \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) dx = c' \Sigma^{-1} (\int f(\boldsymbol{x}) \boldsymbol{x} dx - \boldsymbol{\mu}) = \mathbf{0} \\
&\Leftrightarrow \boldsymbol{\mu} = \int f(\boldsymbol{x}) \boldsymbol{x} dx
\end{aligned}$$

κ and κ' are some constants. Therefore $\mu^* = \int f(x) x dx$, since Σ is invertible. Now take the derivative of objective function $H(f, g)$ w.r.t. Σ we get,

$$\begin{aligned}
&-\frac{\partial}{\partial \Sigma} \int f(x) \left(\log \left(\frac{2\pi^{-d/2}}{\det(\Sigma)^{1/2}} \right) - \frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right) dx \\
&= c \int f(x) \frac{\partial}{\partial \Sigma} \left(\log\det(\Sigma) + (x - \mu)^T \Sigma^{-1} (x - \mu) \right) dx \\
&\overset{\text{(iii \& iv)}}{=} c \int f(x) \left(\Sigma^{-T} - \Sigma^{-T} (x - \mu)(x - \mu)^T \Sigma^{-T} \right) dx, \quad \because X \succ 0
\end{aligned}$$

Set the derivative to zero we get,

$$\begin{aligned}
\Sigma^{-T} &= \int \Sigma^{-T} (x - \mu)^T (x - \mu) \Sigma^{-T} f(x) dx \\
&= \Sigma^{-T} \int (x - \mu)^T (x - \mu) f(x) dx \Sigma^{-T}
\end{aligned} \tag{16}$$

Then,

$$\Sigma = \int (x - \mu)^T (x - \mu) f(x) dx, \quad \because \Sigma = \Sigma^T \tag{17}$$

□

4. Derivation of EM algorithms

4.1. Mean and covariance of samples from a GMM

Let $f'(x) = \sum_{i=1}^L \pi_i' f_i(x)$ be a GMM, namely, each $f_i(x; \mu_i, \Sigma_i)$ is a Gaussian distribution and $\sum_{i=1}^L \pi_i' = 1$. Then the mean and covariance of GMM $f'(x)$ is given by

$$\mathbb{E}_{f'}(x)[x] = \sum_{i=1}^L \pi_i' \mu_i =: \bar{\mu} \tag{18}$$

$$\mathbb{E}_{f'}(x)[(x - \bar{\mu})(x - \bar{\mu})^T] = \sum_{i=1}^L \pi_i' \Sigma_i + \sum_{i=1}^L \pi_i' (\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^T \tag{19}$$

The mean is obtained by

$$\begin{aligned}
\mathbb{E}_{f'(x)}[x] &= \int x f'(x) dx = \int_x x \sum_{i=1}^L \pi_i' f_i(x) dx \\
&= \sum_{i=1}^L \pi_i' \int_x x f_i(x) dx = \sum_{i=1}^L \pi_i' \mu_i =: \bar{\mu}
\end{aligned} \tag{20}$$

Now, the covariance is obtained by

$$\begin{aligned}
&\mathbb{E}_{f'(x)}[(x - \bar{\mu})(x - \bar{\mu})^T] = \int (x - \bar{\mu})(x - \bar{\mu})^T f'(x) dx \\
&= \int_x (x - \bar{\mu})(x - \bar{\mu})^T \sum_{i=1}^L \pi_i' f_i(x) dx \\
&= \sum_{i=1}^L \pi_i' \int_x (x - \bar{\mu})(x - \bar{\mu})^T f_i(x) dx \\
&= \left(\sum_{i=1}^L \pi_i' \int_x x x^T f_i(x) dx - 2\pi_i' \int x \bar{\mu}^T f_i(x) dx \right) + \bar{\mu}\bar{\mu}^T \\
&= \left(\sum_{i=1}^L \pi_i' \int_x x x^T f_i(x) dx \right) - \bar{\mu}\bar{\mu}^T \\
&= \left(\sum_{i=1}^L \pi_i' [\Sigma_i + \mu_i \mu_i^T] \right) - \bar{\mu}\bar{\mu}^T \\
&= \sum_{i=1}^L \pi_i' \Sigma_i + \sum_{i=1}^L \pi_i' (\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^T
\end{aligned} \tag{21}$$

Remarks: our proposed method can be interpreted as a functional clustering algorithm for a set of Gaussian distributions $\{f_i\}_{=1}^L$ whereas the classical GMM clusters a set of points $\{x_i\}_{=1}^L$. In our case, γ_{ij} represents the soft assignment of f_i in $\bar{\mathcal{F}}$ to g_j in $\mathcal{G}$. M-step can be interpreted as searching parameters for one representative Gaussian $g_j \in \mathcal{G}$ (roughly speaking, a mean function restricted to a Gaussian) for a set of assigned Gaussians $\{f_i\}_{i=1}^L \in \bar{\mathcal{F}}$ with $\{\gamma_{ij}\}_{i=1}^L$.

Interestingly, our methods are applicable to ordinary vector data in $\mathbf{R}^d$ and offer some advantages in the following cases.

Case 1) Weighted samples

Case 2) Noisy observations with known Gaussian errors

Case 3) Ill-conditioned covariance matrices, e.g., small number of high dimensional samples $d \gg L$, or many clusters $K \sim L$

Here, we show a simple construction. Suppose that a set of ordinary vector samples $\{x_i\}_{=1}^{L}$ is given. Now, one wants to utilize additional knowledge on the noise of observation (case 2). First, we perform kernel density estimation (KDE) over samples with Gaussian kernels centered at each data point with a known covariance Σ. Then, we have L Gaussians distributions. Our method allows to naturally incorporate this knowledge in the model. Similarly, for ill-conditioned covariance matrices (case 3), by treating samples as a Gaussian with sufficiently small covariance, this problem can be resolved in our framework. Weights of samples (case 1) can be trivially exploited by π_i.

How can we impose a bias on the covariance matrices to resolve the ill-conditioned problem of covariance matrices? Observe that given the responsibilities γ_{ij}, the M step in the proposed method differs by $\sum_{=1}^{L} \pi_i' \Sigma_i$ in (21) from one in classical GMM. Considering the KDE case, each instance corresponds to a Gaussian distribution with mean $\mu_i = x_i$ and covariance Σ. The covariance in classical GMM is $\Sigma = \sum_{i=1}^{L} \pi_i'(x_i - \bar{\mu})(x_i - \bar{\mu})^T$, where π_i' is a weight of x_i and $\sum_{i=1}^{L} \pi_i' = 1$. This corresponds to $\sum_{i=1}^{L} \pi_i'(\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^T$ in (19) in our method. Hence KDE increases the covariance in (19) by $\sum_{=1}^{L} \pi_i' \Sigma = \Sigma$ which is exactly same as the covariance of Gaussian kernel in KDE. This observation implies that by adding the covariances on each sample, we may have better conditioned covariance matrices.

4.2. Cross entropy between two Gaussians

Cross entropy is the optimal code length given data p and codebook q, which is as

$$H(p,q) := \mathbb{E}_p[-\log q(x)] = -\int p(x) \log q(x) dx. \quad (22)$$

The cross entropy between two Gaussian distributions has an analytic form. Let $\mathcal{N}_p$ and $\mathcal{N}_q$ be multivariate Gaussian distributions with (μ_p, Σ_p) and (μ_q, Σ_q) respectively. Cross entropy $H(p,q)$ is given as

$$\begin{aligned}\mathbb{E}_p[-\log q(x)] = \frac{1}{2}\Big\{&k \log 2\pi + \log|\Sigma_q| + \mathrm{tr}[\Sigma_q^{-1}\Sigma_p] \\ &+(\mu_p - \mu_q)^T \Sigma_q^{-1} (\mu_p - \mu_q)\Big\}\end{aligned} \quad (23)$$

Proof. First, Gaussian density function is given by

$$q(x) = \frac{1}{\sqrt{(2\pi)^k |\Sigma_q|}} \exp\left(-\frac{1}{2}(x-\mu_q)^T \Sigma_q^{-1} (x - \mu_q)\right) \quad (24)$$

Let us take the log of $q(x)$.

$$\log q(x) = -\frac{k}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_q| - \frac{1}{2}(x-\mu_q)^T \Sigma_q^{-1}(x-\mu_q) \quad (25)$$

Then, the cross entropy $H(p,q)$ is

$$\begin{aligned}\int -p(x)\log q(x) dx &= \frac{k}{2}\log 2\pi + \frac{1}{2}\log|\Sigma_q| \\ &+ \frac{1}{2}[\mu_q^T \Sigma_q^{-1}\mu_q - 2\mu_p^T \Sigma_q^{-1} \mu_q + \int x^T \Sigma_q^{-1} x p(x) dx]\end{aligned} \quad (26)$$

We know that $\mathrm{tr}(\cdot)$ and $\mathbb{E}[\cdot]$ are linear operators, so $\mathrm{tr} \circ E = E \circ \mathrm{tr}$. Using this fact, we have

$$\begin{aligned}\int x^T \Sigma_q^{-1} x p(x) dx &= \mathbb{E}_p[x^T \Sigma_q^{-1} x] = \mathbb{E}_p[\mathrm{tr}[x^T \Sigma_q^{-1} x]] \\ &= \mathbb{E}_p[\mathrm{tr}[\Sigma_q^{-1} x x^T]] = \mathrm{tr}[\mathbb{E}_p[\Sigma_q^{-1} x x^T]] \\ &= \mathrm{tr}[\Sigma_q^{-1}\mathbb{E}_p[x x^T]] = \mathrm{tr}[\Sigma_q^{-1}(\Sigma_p + \mu_p \mu_p^T)] \\ &= \mathrm{tr}[\Sigma_q^{-1}\Sigma_p] + \mu_p \Sigma_q^{-1} \mu_p^T\end{aligned} \quad (27)$$

Replacing $\int x^T \Sigma_q^{-1} x p(x) dx$ in (26) with $\mathrm{tr}[\Sigma_q^{-1}\Sigma_p] + \mu_p \Sigma_q^{-1} \mu_p^T$ completes the proof. □

4.3. EM derivation

EM-algorithm w.r.t cross entropy. As EM-algorithm for classical GMM, this proposed method comprises of two steps: E-step and M-step. Our result shows that M-step maximizes *negative* cross entropy between reweighted data GMM $\sum_{i=1}^{L} \pi' f_i$ and a Gaussian component in a model GMM g_j as the classical GMM increases the likelihood of reweighted samples. Let us compare each step of classical GMM and our proposed method.

E-step in classical GMM is given by

$$\gamma_{ij} := p(z_i = j | x_i, \theta) = \frac{p(z_i = j|\theta) g(x_i | z_i = j, \theta)}{\sum_{j'}^{K} p(z_i = j'|\theta) g(x_i|z_i = j', \theta)} \quad (28)$$

where i is the index for instance and j is the index for component in g (K-GMM).

E-step with cross entropy, we estimate the responsibilities between f_i and g_j rather than a point x_i and g_j in the classical GMM. Let X_i be a set of points which belong to set i with density function $f_i(x)$. j is defined as above.

Then, the responsibilities γ_{ij} of f_i to g_j is given by

$$\begin{aligned}
&\gamma_{ij} := p(z_i = j|X_i, \theta) \\
&= \frac{p(z_i = j|\theta)p(X_i|z_i = j, \theta)}{\sum_{j'=1}^{K} p(z_i = j'|\theta)p(X_i|z_i = j', \theta)} \\
&= \frac{p(z_i = j|\theta)\prod_{x\in X_i} g(x|z_i = j, \theta)^{f_i(x)}}{\sum_{j'}^{K} p(z_i = j'|\theta)\prod_{x\in X_i} p(x|z_i = j', \theta)^{f_i(x)}} \\
&= \frac{p(z_i = j|\theta)\exp\left[\sum_{x\in X_i} f_i(x)\log g(x|z_i = j, \theta)\right]}{\sum_{j'}^{K} p(z_i = j'|\theta)\exp\left[\sum_{x\in X_i} f_i(x)\log g(x|z_i = j', \theta)\right]} \\
&= \frac{p(z_i = j|\theta)\exp\left[\int_x f_i(x)\log g(x|z_i = j, \theta)dx\right]}{\sum_{j'}^{K} p(z_i = j'|\theta)\exp\left[\int_x f_i(x)\log g(x|z_i = j', \theta)dx\right]} \\
&= \frac{w_j\exp\left[\int_x f_i(x)\log g_j(x|\theta)dx\right]}{\sum_{j'}^{K} w_{j'}\exp\left[\int_x f_i(x)\log g_{j'}(x|\theta)dx\right]} \\
&= \frac{w_j\exp\left[-H(f_i, g_j)\right]}{\sum_{j'}^{K} w_{j'}\exp[-H(f_i, g_{j'})]}
\end{aligned} \tag{29}$$

Note that γ_{ij} means the membership of the i-th Gaussian distribution in l-GMM to the j-th component in k-GMM. $H(f_i, g_j)$ is analytically obtained as (23).

M-step in classical GMM is,

$$\begin{aligned}
w_j &= \frac{1}{L}\sum_i \gamma_{ij} \\
\mu_j &= \frac{\sum_{i=1}^{L}\gamma_{ij}x_i}{\sum_{i'=1}^{L}\gamma_{i'j}} \\
\Sigma_j &= \frac{\sum_{i=1}^{L}\gamma_{ij}(x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i'=1}^{L}\gamma_{i'j}}
\end{aligned} \tag{30}$$

M-step with cross entropy is,

$$\begin{aligned}
w_j &= \sum_{i=1}^{L}\pi_i\gamma_{ij}, \text{ since } \sum_{i=1}^{L}\pi_i = 1 \\
\mu_j &= \mathbb{E}_{f'}(x)[x] = \sum_{i=1}^{L}\pi_i'\mu_i \\
\Sigma_j &= \mathbb{E}_{f'}(x)[(x - \mu_j)(x - \mu_j)^T] \\
&= \sum_i \pi_i'\Sigma_i + \sum_i \pi_i'(\mu_i - \mu_j)(\mu_i - \mu_j)^T
\end{aligned} \tag{31}$$

where $f' = \sum_{i=1}^{L}\pi_i' f_i(x)$, and $\pi_i' = \frac{\pi_i\gamma_{ij}}{\sum_i \pi_i\gamma_{ij}}$, for fixed j.

M step in classical GMM can be interpreted as expectations: x and $(x - \mu)(x - \mu)^T$ over a discrete probability distribution $\{\gamma_{ij}/\gamma_j\}_{i=1}^{L}$, where $\gamma_j := \sum_{i=1}^{L}\gamma_{ij}$. Similarly, M-step in our proposed method also can be interpreted as expectations over a reweighted GMM f', which is a continuous probability distribution.

4.4. Detailed derivation of EM algorithm

In this section, we provide full derivation of EM algorithm for our method. Let $f = \sum_i \pi_i f_i$ and $g = \sum_j \pi_j f_j$ be GMMs. Our EM algorithm maximizes the *negative* cross entropy $-H(f, g) := \int f(x)\log g(x)$. First, we derive Q function from the objective function.

$$\begin{aligned}
&\arg\max_{g\in\mathbf{G}^{(K)}} \int \sum_{i=1}^{L}\pi_i f_i(x)\log\sum_{j=1}^{K} w_j g_j(x)dx \\
&= \arg\max_{g\in\mathbf{G}^{(K)}} \int \sum_{i=1}^{L}\pi_i f_i(x)\log\sum_{j=1}^{K} P(z_i = j|X_i, \theta)\frac{w_j g_j(x)}{P(z_i = j|X_i, \theta)}dx \\
&\geq \arg\max_{g\in\mathbf{G}^{(K)}} \int \sum_{i=1}^{L}\sum_{j=1}^{K}\pi_i f_i(x)P(z_i = j|X_i, \theta)\log\frac{w_j g_j(x)}{P(z_i = j|X_i, \theta)}dx \\
&= \arg\max_{g\in\mathbf{G}^{(K)}} \int \sum_{i=1}^{L}\sum_{j=1}^{K}\pi_i f_i(x)P(z_i = j|X_i, \theta)\log w_j g_j(x)dx \\
&\quad - \int \sum_{i=1}^{L}\sum_{j=1}^{K}\pi_i f_i(x)P(z_i = j|X_i, \theta)\log P(z_i = j|X_i, \theta)dx
\end{aligned} \tag{32}$$

The inequality above is obtained by Jensen's inequality. Now, we define $Q(\theta|\theta_n)$ with the first term of the last equation as

$$Q(\theta|\theta_n) := \int \sum_{i=1}^{L}\sum_{j=1}^{K}\pi_i f_i(x)P(z_i = j|X_i, \theta_n)\log w_j g_j(x)dx. \tag{33}$$

Once we define $Q(\theta|\theta_n)$, we are ready to derive EM algorithm. First, E step is merely to estimate $P(z_i = j|X_i, \theta) =: \gamma_{ij}$ by (29). Second, we derive M step. To do so, we will maximize $Q(\theta|\theta_n)$ w.r.t. $\{w_j, \boldsymbol{\mu}_j, \Sigma_j\}_{j=1}^{K}$. The Q function can be rewritten as

$$Q(\theta|\theta_n) = \int \sum_{i=1}^{L}\sum_{j=1}^{K}\pi_i f_i(x)\gamma_{ij}\log g_j(x)dx + \sum_{i=1}^{L}\sum_{j=1}^{K}\pi_i\gamma_{ij}\log w_j. \tag{34}$$

To maximize over μ_j and Σ_j, one needs to maximize the following.

$$\begin{aligned}
&\underset{\{\boldsymbol{\mu}_j\}_{j=1}^{K}, \{\Sigma_j\}_{j=1}^{K}}{\operatorname{argmax}} \quad Q(\theta|\theta_n) \\
&= \underset{\{\boldsymbol{\mu}_j\}_{j=1}^{K}, \{\Sigma_j\}_{j=1}^{K}}{\operatorname{argmax}} \int \sum_{i=1}^{L}\sum_{j=1}^{K}\pi_i\gamma_{ij}f_i(x)\log g_j(x)dx
\end{aligned} \tag{35}$$

Since, given γ_{ij}, one can decompose the maximization into for each component j, one has

$$\begin{aligned}
&\underset{\boldsymbol{\mu}_j, \Sigma_j}{\operatorname{argmax}} \int \sum_{i=1}^{L}\pi_i\gamma_{ij}f_i(x)\log g_j(x)dx \\
&= \underset{\boldsymbol{\mu}_j, \Sigma_j}{\operatorname{argmax}} \int \frac{\sum_{i=1}^{L}\pi_i\gamma_{ij}}{\sum_{i'=1}^{L}\pi_{i'}\gamma_{i'j}}f_i(x)\log g_j(x)dx
\end{aligned} \tag{36}$$

since dividing the objective function by a constant doesn't change the problem. Now, let $\pi_i^{'} := \frac{\pi_i\gamma_{ij}}{\sum_{i'=1}^{L}\pi_{i'}\gamma_{i'j}}$. Then the maximization over μ_j and Σ_j is reduced to Lemma 2 that maximizes the *negative* entropy between a reweighted GMM $f'(x) := \sum_{i=1}^{L} \pi_i^{'} f_i(x)$ and a Gaussian with μ_j and Σ_j. The optimal solution μ_j^*, Σ_j^* is given in (31).

To maximize over w_j, one needs to maximize $Q(\theta|\theta_n)$ with a constraint $\sum_{j=1}^{K} w_j = 1$. So the objective function with a Lagrange multiplier is defined by $\mathcal{L} := Q(\theta|\theta_n) + \lambda(\sum_{j=1}^{K} w_j - 1)$. Now, take the derivative of $\mathcal{L}$ w.r.t w_j.

$$\frac{\partial \mathcal{L}}{\partial w_j} = \frac{\partial}{\partial w_j} \sum_{i=1}^{L} \pi_i \gamma_{ij} \log w_j + \lambda \tag{37}$$

Let's set the derivative above to zero. Then, one gets

$$w_j = -\frac{\sum_{i=1}^{L} \pi_i \gamma_{ij}}{\lambda}. \tag{38}$$

The primal feasibility $\sum_{j=1}^{K} w_j = 1$ and the result above yield $\lambda = -\sum_{j=1}^{K} \sum_{i=1}^{L} \pi_i \gamma_{ij}$. Hence the optimal weight is given by

$$w_j^* = \frac{\sum_{i=1}^{L} \pi_i \gamma_{ij}}{\sum_{j'=1}^{K} \sum_{i'=1}^{L} \pi_{i'} \gamma_{i'j'}}. \tag{39}$$

This completes the derivation of our EM algorithm. For more details on the theory of EM algorithms, we refer [1].

5. Functional clustering and construction of modified EM for EAPs

In our main text and at the end of section 4.1 in this supplementary material, we discussed the relationship between functional clustering and our proposed EM method. In this section, we compare our modified EM for EAPs (in Fig 2 in the main) with EM algorithm originally proposed method (in Fig 1 in the main) from the functional clustering perspective. For the rest of this supplementary, we denote algorithm 1 and and algorithm 2 for EM algorithm and modified EM algorithm respectively.

First, algorithm 1 seeks a representative Gaussian g_j to cover assigned Gaussians in $\bar{\mathcal{F}}$. This is similar to functional k-means algorithm with a restriction on the structure of the representative function (roughly speaking, mean function). Algorithm 2 corresponds to the GMM in the functional space. It learns the representative function for each cluster and the deviation of members from the representative function as well.

More precisely, modified EM algorithm can be interpreted as a restricted Gaussian Process Mixture Model. Each component in Algorithm 2 has four parameters $(w_j, \mu_j, \Sigma_j, C_j)$. As Algorithm 1, w_j is a weight for a component, μ_j, Σ_j are for a Gaussian mean function. C_j is a constant for covariance function. In our algorithm 2, we use the simplest covariance function $K(x, x') = C_j^2$ if $x = x'$ and otherwise $k(x, x') = 0$. Let $m_j(x) := \mathcal{N}(x|\mu_j, \Sigma_j)$ be a mean function of GP. Then the GP is given by

$$f \sim \mathcal{N}(m_j(x), K_j(x, x')) \tag{40}$$

Using this observation, we modify E step for EAPs. **E-step** in algorithm 2 estimates the responsibilities of $f_i \in \bar{\mathcal{F}}$ to component j in GPMM as

$$\gamma_{ij} = \frac{w_j C_j^{-1} \exp\left(-\frac{1}{2C_j^2} \|f_i - g_j\|_2^2\right)}{\sum_{k=1}^{K} w_k C_k^{-1} \exp\left(-\frac{1}{2C_k^2} \|f_i - g_k\|_2^2\right)}. \tag{41}$$

M-step is same as Algorithm 1 except C_j update as

$$C_j^2 = \sum_i \gamma_{ij} \pi_i \|f_i - g_j\|^2 / \sum_{i'} \gamma_{i'j} \pi_{i'} \tag{42}$$

w_j and μ_j, Σ_j are updated using Eqs.(15) in the main text.

6. Simulation experiments

Both **Gaussian filtering** and **aniostropic filtering** are performed using a 3×3 neighborhood size kernel. For Gaussian filtering, we use isotropic Gaussian weights. For anisotropic filtering kernel, the weights were determined using the inner product of the EAP profiles evaluated at 321 tessellated points spread uniformly on a sphere. We would like to note that we may also use the inner product of the K-GMM models themselves. Since the visualization tool only uses PDF values on a sphere, for our experiments, we used the former approach.

Interpolation path is shown in Fig. 2. We show an interpolation path between two 2-GMMs, which is obtained by minimizing (8).

References

[1] M. Mak, S. Kung, and S. Lin. Expectation Maximization Theory. *Biometric Authentication: A Machine Learning Approach*, 61(1):503–512, 2004. 6

[2] K. B. Petersen and M. S. Pedersen. The matrix cookbook (version november 15, 2012). 2012. 1

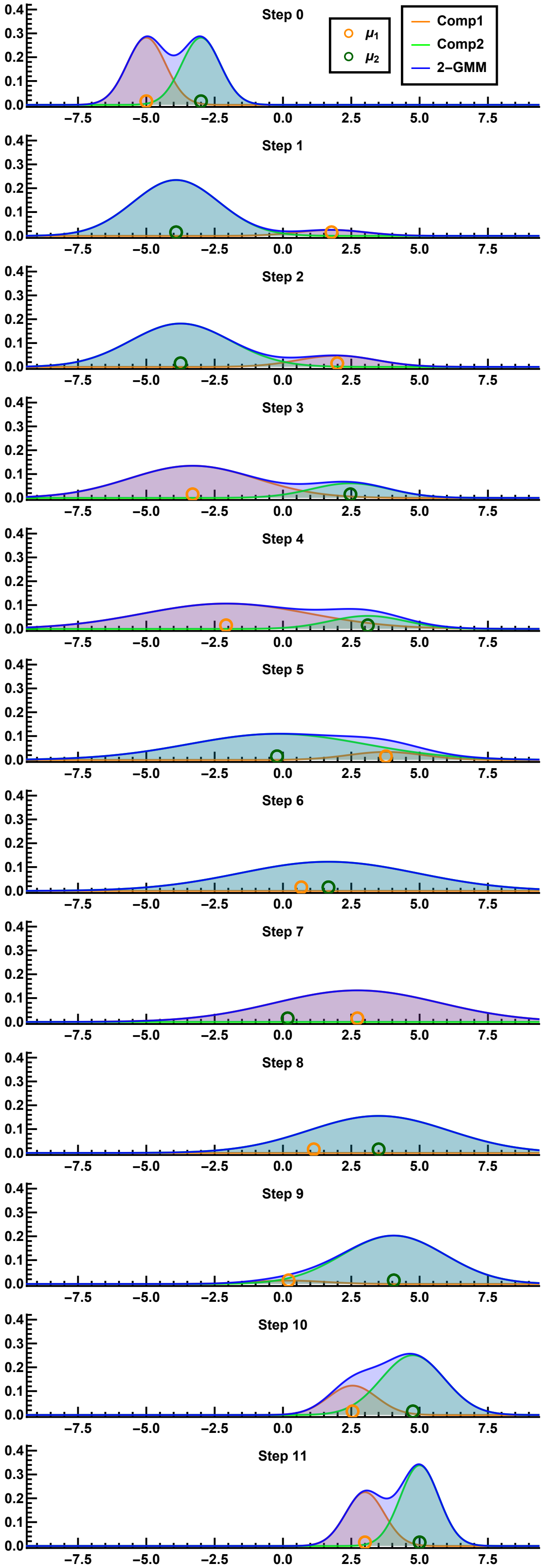

Figure 2: Interpolation path along 2-GMM manifold showing 10 steps from top (GMM0) to bottom (GMM11).