Learning Mixtures of Gaussian Processes through Random Projection

Emmanuel Akeweje [1] **Mimi Zhang** [1 2]

Abstract

We propose an ensemble clustering framework to uncover latent cluster labels in functional data generated from a Gaussian process mixture. Our method exploits the fact that the projection coefficients of the functional data onto any given projection function follow a univariate Gaussian mixture model (GMM). By conducting multiple one-dimensional projections and learning a univariate GMM for each, we create an ensemble of GMMs. Each GMM serves as a base clustering, and applying ensemble clustering yields a consensus clustering. Our approach significantly reduces computational complexity compared to state-of-the-art methods, and we provide theoretical guarantees on the identifiability and learnability of Gaussian process mixtures. Extensive experiments on synthetic and real datasets confirm the superiority of our method over existing techniques.

1. Introduction

Gaussian process (GP) models are fundamental in (Bayesian) machine learning. (Rasmussen & Williams, 2006) gave an overview of the mathematical foundations and practical applications of GP for regression and classification tasks. The GP mixture model naturally generalizes the concept of Gaussian mixture model (GMM), making it a powerful tool for the statistical or cluster analysis of heterogeneous functional data. However, documented works on GP mixture models all perform parameter estimation before any inference or cluster analysis, resulting in cubic computational complexity with the number of data points. To promote the application of GP mixture models for mixture modelling or cluster analysis, we here take an innovative approach that performs cluster analysis before parameter estimation, and once the hidden cluster labels are revealed, the problem of learning the mixture of GPs reduces to learning each GP component independently.

A GP mixture model can not be defined through the notion of probability density, which generally does not exist for random functions. We here define the GP mixture model in accordance with prior pooling (Seitz, 2021): a random function X is a K-mixture of GPs if the sub-populations are characterized by K Gaussian random functions $\{X_k\}_{k=1}^K$, and that with probability $\pi_k (> 0)$, a random sample is a realization of the kth random function $X_k \sim GP(\mu_k, \Sigma_k)$. Here, we let $GP(\mu_k, \Sigma_k)$ denote the GP with a mean function μ_k and a covariance function Σ_k. Note that, in the above definition, the unknown weights $\{\pi_k\}_{k=1}^K$ are fixed parameters. There are a few attempts at learning the GP mixture model with the above definition. Let $\{x_i\}_{i=1}^n$ denote the functional data from the GP mixture, and $z_i \in \{1, \ldots, K\}$ the hidden cluster label of the sample function x_i. (James & Sugar, 2003) assumed that each Gaussian random function X_k has an individual mean function μ_k yet a common covariance function $\Sigma_k = \Sigma$. They adopted the mixed-effects model that, given $z_i = k$ and the basis expansion $x_i(t) = \mu_k(t) + [x_i(t) - \mu_k(t)] = \sum_{v=1}^m \alpha_{kv} b_v(t) + \sum_{v=1}^m a_{iv} b_v(t)$, the n coefficient vectors $\{\boldsymbol{a}_i = (a_{i1}, \ldots, a_{im})^T\}_{i=1}^n$ are identically distributed and have a zero-mean Gaussian distribution. An EM-type algorithm was developed for parameter estimation. In both (Shi & Wang, 2008) and (Huang et al., 2014), each Gaussian random function X_k has a different mean function μ_k and a different covariance function Σ_k. (Huang et al., 2014) developed an EM-type algorithm for parameter estimation. In (Shi & Wang, 2008), each subject is characterized by both a sample function x_i and a covariate vector $\boldsymbol{c}_i$. The covariates $\{\boldsymbol{c}_i\}_{i=1}^n$ are fitted by a logistic regression model to predict the latent cluster labels. They developed an EM-type algorithm, where the E-step updates the expected log-likelihood function, and the M-step updates all parameter estimates. (Wu & Ma, 2018) divided the function domain $\mathcal{T} \subset \mathbb{R}$ into disjoint intervals: $\mathcal{T} = \cup_j \mathcal{T}_j$, and assumed that $X_k(t)$ over each interval $\mathcal{T}_j$ is a different GP; that is, the stochastic process $\{X_k(t) : t \in \mathcal{T}\}$ is piece-wise Gaussian. They developed an EM-type algorithm, where the E-step is built on an MCMC technique for generating latent cluster labels.

[1] School of Computer Science and Statistics, Trinity College Dublin, Ireland [2] I-Form Advanced Manufacturing Research Centre, Science Foundation Ireland, Ireland. Correspondence to: Emmanuel Akeweje <eakeweje@tcd.ie>, Mimi Zhang <mimi.zhang@tcd.ie>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Another branch of works organizes the generative model into a hierarchical structure. (Shi et al., 2005) assumed that the weights $(\pi_1, \ldots, \pi_K)$ have a Dirichlet distribution; they employed the Gibbs sampler for simulating the latent cluster labels and a hybrid Monte Carlo method for simulating the unknown model parameters. (Rasmussen & Ghahramani, 2001), (Jackson et al., 2007), (Ross & Dy, 2013), (Li & Ma, 2023), and a few others extended the Bayesian nonparametric model from multivariate data to functional data. In particular, the building blocks of the hierarchy are: (1) $G \sim DP(G_0, \alpha)$, a Dirichlet process having a base distribution G_0 and a concentration parameter α; (2) $\Theta = (\mu, \Sigma) \sim G(\Theta)$, a joint distribution over the pair (μ, Σ); (3) $x \sim GP(\mu, \Sigma)$. For the 2nd block, although one can utilize the GP and Wishart process together to directly sample the pair (μ, Σ), the two functions μ and Σ normally take a parametric form in applications, and we define a Dirichlet process over their parameters. The popular stick-breaking process gives an explicit construction of G: $G(d\Theta) = \sum_k \pi_k \delta_{\Theta_k}$; that is, Θ is generated from a random distribution G that concentrates a probability mass π_k on the atom Θ_k, and the π_k's are given by a stick-breaking process. For performing approximate inference, (Rasmussen & Ghahramani, 2001) and (Jackson et al., 2007) developed a Markov chain procedure relying on Gibbs sampling, while (Ross & Dy, 2013) and (Li & Ma, 2023) applied the variational Bayes technique.

(Zhang & Parnell, 2023) recently conducted a thorough review of clustering methods for functional data, from which we identified three pertinent studies (Jacques & Preda, 2013; Bouveyron et al., 2015; Rivera-García et al., 2019). They all approached the learning problem through defining a pseudo-density for random functions. In particular, given the spectral decomposition $\Sigma_k(s,t) = \sum_{v=1}^{\infty} \lambda_{kv} b_{kv}(s) b_{kv}(t)$, the latent label $z_i = k$, and hence the cluster-wise functional principal component (fPC) decomposition $x_i(t) = \mu_k(t) + \sum_{v=1}^{m_k} a_{ikv} b_{kv}(t)$, (Jacques & Preda, 2013) defined the pseudo-density of $[x_i | z_i = k]$ to be $\Pi_{v=1}^{m_k} \phi(a_{ikv}; 0, \lambda_{kv})$; that is, they assumed that the distribution of the coefficient vector $(a_{ik1}, \ldots, a_{ikm_k})^T$ is zero-mean Gaussian with a diagonal covariance matrix diag$(\lambda_{k1}, \ldots, \lambda_{km_k})$. (Rivera-García et al., 2019) assumed that the distribution of the coefficient vector $(a_{ik1}, \ldots, a_{ikm})^T$, with $m > m_k$, is $\mathcal{N}(\mathbf{0}, \text{diag}(\lambda_{k1}, \ldots, \lambda_{km_k}, \lambda_k, \ldots, \lambda_k))$; that is, the additional $(m - m_k)$ random coefficients are identically distributed. The work (Bouveyron et al., 2015) differs from (Rivera-García et al., 2019) mainly in that (Bouveyron et al., 2015) assumed that the transformed random function $g(X_k)$ is a Gaussian process, where the function g is the feature map of a kernel function. The three works all developed an EM-type algorithm, where the E step consists in computing the conditional expectation of the multivariate Gaussian mixture log-likelihood, and the M step involves updating the fPC decomposition for each cluster.

Following the definition of the GP mixture model by (Seitz, 2021), we address the learning problem within an ensemble clustering framework. Ensemble clustering methods typically produce a variety of base clusterings and then extract a consensus clustering from the base clusterings. The consensus clustering encompasses all the information contained in the ensemble, offering an improvement over the individual base clusterings (Zhang, 2022). Our learning algorithm for GP mixtures is extremely simple in that the base clusterings in the ensemble are univariate GMMs, obtained by projecting the functional data onto multiple projection functions. When the projection functions are randomly generated, our approach parallels the random projection method for (high-dimensional) multivariate data (Yellamraju & Boutin, 2018).

The paper is organized as follows. Section 2 provides the formal definition of the GP mixture model. Section 3 delves into the methodological details, and Section 4 offers a theoretical analysis. In Section 5, we present the results of extensive experimental studies. Section 6 concludes with a summary. All proofs are given in the appendix.

2. Gaussian Process Mixture

Let $\mathcal{T}$ denote a closed interval of $\mathbb{R}$, and X a random function that is defined on a probability space $(\Omega, \mathcal{F}, \Pr)$ and taking values in the Hilbert space $\mathcal{H}(\mathcal{T}, \mathbb{R}) = \{x : \mathcal{T} \mapsto \mathbb{R}\}$ of square-integrable functions. A sample function $x = X(\cdot, \omega) \in \mathcal{H}(\mathcal{T}, \mathbb{R})$ is the value of the random function X at the outcome $\omega \in \Omega$. Alternatively, we can view the function value $x(t)$ as a random realization of the real-valued random variable $X(t, \cdot)$, a mapping from $(\Omega, \mathcal{F})$ to $(\mathbb{R}, \mathcal{B}_{\mathbb{R}})$, where $\mathcal{B}_{\mathbb{R}}$ is the Borel σ-algebra of $\mathbb{R}$. The random function X is called Gaussian if and only if the random vector $(X(t_1), \ldots, X(t_r))^T$ is multivariate Gaussian for any finite collection of time indices $\{t_j \in \mathcal{T}\}_{j=1}^r$. A GP is characterized by the mean function $\mu(t) = \text{E}[X(t)]$ and the covariance function $\Sigma(t,s) = \text{E}[(X(t) - \mu(t))(X(s) - \mu(s))]$.

In the context of cluster analysis, we assume that the sample functions $\{x_1, \ldots, x_n\}$ are random realizations of K Gaussian random functions $\{X_1, \ldots, X_K\}$, with $K (\geq 2)$ being unknown. For each Gaussian random function X_k, we write $X_k \sim GP(\mu_k, \Sigma_k)$. The complete data are in the form of $\{(x_i, z_i)\}_{i=1}^n$ that are independent realizations of the couple (X, Z), where Z is the hidden cluster-indicator variable with $\Pr(Z = k) = \pi_k$, $\pi_k > 0$ and $\sum_{k=1}^K \pi_k = 1$. Given a value of Z, e.g., $Z = k$, the conditional distribution of the random function X is that of X_k. If the sample function x_i is a random realization of X_k, then for any finite collection of input points $\underline{t}_i = \{t_{i1}, \cdots, t_{ir_i}\}$, the vector of function values $x_i(\underline{t}_i)$ has the multivariate Gaussian distribution $\mathcal{N}(\mu_k(\underline{t}_i), \Sigma_k(\underline{t}_i, \underline{t}_i))$, where $\mathcal{N}(\cdot, \cdot)$ is the notation for multivariate Gaussian distribution. Here, we employ compact notation for functions applied to collections of input points, and $\Sigma_k(\underline{t}_i, \underline{t}_i)$ is the $r_i \times r_i$ covariance matrix.

In real applications, the observation of x_i at any point $t \in \mathcal{T}$, denoted by $y_i(t)$, may come with an additive error: $y_i(t) = x_i(t) + \epsilon_i(t)$, where ϵ_i is the noise process with $\mathrm{E}[\epsilon_i(t)] = 0$ and $\mathrm{E}[\epsilon_i^2(t)] = \sigma^2(t)$. If the sample function x_i is from $GP(\mu_k, \Sigma_k)$, the conditional distribution of the sample path $y_i(\underline{t}_i)$ is again multivariate Gaussian $\mathcal{N}(\mu_k(\underline{t}_i), \Sigma_k(\underline{t}_i, \underline{t}_i) + \sigma^2\mathbf{I})$, where $\mathbf{I}$ is the identity matrix of appropriate dimension. Therefore, given the prior beliefs $\{\pi_k\}_{k=1}^K$, the marginal distribution of $y_i(\underline{t}_i)$ is a multivariate Gaussian mixture:

$$y_i(\underline{t}_i) \sim \sum_{k=1}^K \pi_k \mathcal{N}(\mu_k(\underline{t}_i), \Sigma_k(\underline{t}_i, \underline{t}_i) + \sigma^2\mathbf{I}).$$

Let $\mathcal{D} = \{y_i(\underline{t}_i) : i = 1, \cdots, n\}$ denote the set of n sample paths. The log-likelihood function of the n sample paths is

$$L(\Theta; \mathcal{D}) = \sum_{i=1}^n \log(\sum_{k=1}^K \pi_k \phi(y_i(\underline{t}_i); \mu_k(\underline{t}_i), \Sigma_k(\underline{t}_i, \underline{t}_i) + \sigma^2\mathbf{I})),$$

where ϕ is the Gaussian density function, and Θ is the set of all model parameters: $\Theta = \{\pi_k, \mu_k, \Sigma_k, \sigma\}_{k=1}^K$.

Existing methods typically treat the mixture-learning problem as a parameter-estimation problem and develop an EM-type algorithm for parameter estimation. However, this line of approach suffers from two significant drawbacks: (1) The computational load is heavy, rendering EM-type algorithms impractical when r_i is large, due to the need to invert the covariance matrices. (2) EM-type algorithms are local-search heuristic, and the output is heavily influenced by the initial input, leading to local-optimal results with no quality guarantees. To address these computational barriers, we approach the mixture-learning problem from the cluster analysis perspective, where the direct objective is to learn the latent cluster labels $\{z_i\}_{i=1}^n$, not the parameter set Θ.

3. Methodology

3.1. From GP Mixture to Univariate Gaussian Mixture

According to the spectral theorem for compact self-adjoint operators, the covariance function $\Sigma_k(s,t)$ of X_k has a series representation:

$$\Sigma_k(s,t) = \sum_{v=1}^\infty \lambda_{kv} b_{kv}(s) b_{kv}(t),$$

where $\lambda_{k1} \geq \lambda_{k2} \geq \cdots \geq 0$ are the eigen-values, and $\{b_{kv}\}_{v\in\mathbb{N}}$ are the orthonormal eigen-functions. Then a sample function x_i, with $z_i = k$, has the following fPC decomposition:

$$x_i(t) = \mu_k(t) + \sum_{v=1}^\infty a_{ikv} b_{kv}(t), \qquad (1)$$

where $a_{ikv} = \langle x_i - \mu_k, b_{kv}\rangle$ is the fPC score associated with the eigen-function b_{kv}. Given that X_k is Gaussian, the fPC score a_{ikv} has the univariate Gaussian distribution $\mathcal{N}(0, \lambda_{kv})$ and is independent from a_{ikr} for any $r \neq v$ (Hall et al., 2006). Note that the fPC scores $\{a_{ik1}, a_{ik2}, \ldots\}$ are always uncorrelated, but the independence is only guaranteed when X_k is a Gaussian random function. If the sample functions $\{x_i\}_{i=1}^n$ are all from the kth GP, then the fPC scores $\{a_{ikv}\}_{i=1}^n$ are i.i.d. from $\mathcal{N}(0, \lambda_{kv})$, and are independent from the fPC scores $\{a_{ikr}\}_{i=1}^n$ of any other eigen-function b_{kr}, $r \neq v$.

In application, we do not know the true cluster label of x_i, and therefore cannot perform the cluster-wise fPC decomposition as in (1). Let μ denote the population mean function: $\mu(t) = \sum_{k=1}^K \pi_k \mu_k(t)$. Let $\{\beta_v\}_{v\in\mathbb{N}}$ denote a (arbitrary) basis of the Hilbert space $\mathcal{H}(\mathcal{T}, \mathbb{R})$. Then any sample function x_i admits the following expansion

$$x_i(t) = \mu(t) + \sum_{v=1}^\infty \alpha_{iv} \beta_v(t), \qquad (2)$$

where $\alpha_{iv} = \langle x_i - \mu, \beta_v\rangle$ is the projection coefficient onto the basis function β_v. Given that $z_i = k$, we replace x_i with its fPC decomposition in Equation (1) and obtain

$$\alpha_{iv} = \langle x_i - \mu, \beta_v\rangle = \langle \mu_k - \mu, \beta_v\rangle + \sum_{l=1}^\infty a_{ikl}\langle b_{kl}, \beta_v\rangle,$$

where we have utilized the linearity of the inner product. Recall that the fPC scores $\{a_{ik1}, a_{ik2}, \ldots\}$ are independent and Gaussian, and that a linear combination of independent Gaussian variables is still Gaussian. Therefore, conditioning on $z_i = k$, the distribution of α_{iv} is univariate Gaussian $\mathcal{N}(\langle \mu_k - \mu, \beta_v\rangle, \sum_{l=1}^\infty \lambda_{kl}\langle b_{kl}, \beta_v\rangle^2)$. The marginal distribution of α_{iv} is hence a univariate Gaussian mixture:

$$\alpha_{iv} \sim \sum_{k=1}^K \pi_k \mathcal{N}(\langle \mu_k - \mu, \beta_v\rangle, \sum_{l=1}^\infty \lambda_{kl}\langle b_{kl}, \beta_v\rangle^2). \qquad (3)$$

That is, for any basis $\{\beta_v\}_{v\in\mathbb{N}}$ of the Hilbert space $\mathcal{H}(\mathcal{T}, \mathbb{R})$, the projection coefficients of the functions $\{x_i - \mu\}_{i=1}^n$ onto, e.g., the basis function β_v are distributed according to the univariate GMM $\sum_{k=1}^K \pi_k \mathcal{N}(\langle \mu_k - \mu, \beta_v\rangle, \sum_{l=1}^\infty \lambda_{kl}\langle b_{kl}, \beta_v\rangle^2)$.

Define the integral operator $\mathbb{K}$: $\mathcal{H}(\mathcal{T}, \mathbb{R}) \to \mathcal{H}(\mathcal{T}, \mathbb{R})$ by

$$[\mathbb{K}x](s) = \int_{\mathcal{T}} \Sigma(s,t) x(t) dt,$$

where $\Sigma(s,t) = \sum_{k=1}^K \pi_k \mathrm{E}[(X_k(t) - \mu(t))(X_k(s) - \mu(s))]$ is the population kernel. We have the following theorem.

Theorem 3.1. *The operator $\mathbb{K}$ is compact, positive and self-adjoint.*

Built on Theorem 3.1, we can invoke the spectral theorem for compact self-adjoint operators to conclude that $\mathbb{K}$ has a complete set of eigen-functions in $\mathcal{H}(\mathcal{T}, \mathbb{R})$. Therefore, in Equation (3), we can let $\{\beta_v\}_{v\in\mathbb{N}}$ be the set of eigen-functions of the operator $\mathbb{K}$, and fit a univariate GMM to the projection coefficients (namely, the fPC scores) $\{\alpha_{iv}\}_{i=1}^n$ for each $v \geq 1$. However, although the eigenfunctions of the operator $\mathbb{K}$ are more efficient in explaining the variation in the functional data, our inclination leans toward basis functions onto which the projected Gaussians are well separated. Finally, we note that the projection functions $\{\beta_v\}_{v=1}^V$ need not be from a basis of $\mathcal{H}(\mathcal{T}, \mathbb{R})$. It can be readily proved that, for any function $\beta \in \mathcal{H}(\mathcal{T}, \mathbb{R})$, the distribution of the projection coefficients $\{\langle x_i - \mu, \beta\rangle : i = 1, \ldots, n\}$ is always a univariate GMM: $\sum_{k=1}^K \pi_k \mathcal{N}(\langle \mu_k - \mu, \beta\rangle, \sum_{l=1}^\infty \lambda_{kl}\langle b_{kl}, \beta\rangle^2)$. Therefore, the projection functions $\{\beta_v\}_{v=1}^V$ can be randomly generated. In Section 4.2, we delve deeper into the performance analysis of our clustering method in the context of random projection functions.

3.2. The Ensemble Clustering Method

The motivation for taking the ensemble clustering approach is explained in Appendix B, and the pseudo code of the clustering method is given in Algorithm 1. Before we expand on step 7, a few points are noted: (a) Step 1 calls for a smoothing technique, and our Python package[1] *GPmix* offers a range of options. (b) In step 4, we apply both the method of moments and the EM algorithm for parameter estimation, where the parameter estimates from the method of moments are for initializing the EM algorithm. We prove in Section 4.3 that the method of moments learns the parameters in polynomial time and a polynomial number of samples. (c) It is important to note that cluster labels are symbolic. If we want to directly aggregate the V cluster membership matrices $\{\mathbf{M}_v\}_{v=1}^V$, we need to solve the label correspondence problem. The method outlined in step 9 utilizes the concept of objects co-occurrence, where the relation $[\mathbf{B}_v\mathbf{B}_v^T]_{ij} = 1$ $(1 \leq i, j \leq n)$ indicates that the sample functions x_i and x_j are in the same cluster w.r.t. the vth clustering.

When $\{\beta_v\}_{v\in\mathbb{N}}$ are the eigen-functions w.r.t. the population kernel $\Sigma(s,t) = \sum_{k=1}^K \pi_k \mathrm{E}[(X_k(t) - \mu(t))(X_k(s) - \mu(s))]$, the density plot of the fPC scores $\{\alpha_{iv}\}_{i=1}^n$ has fewer distinct peaks for a larger eigen-dimension v. In other words, the component Gaussian distributions in the mixture model $\sum_{k=1}^K \pi_{vk}\phi(\alpha; u_{vk}, \sigma_{vk}^2)$ overlap significantly when v is large. This motivates us to define the weight w_v for the vth base clustering in relation to the overlapping degree between the component Gaussian distributions.

Algorithm 1 The GP mixture learning algorithm.

```
Input: The raw data $\mathcal{D} = \{y_i(\underline{t}_i)\}_{i=1}^n$, the projection functions $\{\beta_v\}_{v=1}^V$, and the number of clusters $K$.
Output: The learned cluster labels $\{z_i\}_{i=1}^n$.
1: Estimate the population mean function $\mu$ and the $n$ sample functions $\{x_i\}_{i=1}^n$.
2: for $v = 1, \ldots, V$ do
3:     Calculate the $n$ projection coefficients:
           $\alpha_{iv} = \langle x_i - \mu, \beta_v\rangle, \quad 1 \leq i \leq n.$
4:     Train a univariate GMM from the data $\{\alpha_{iv}\}_{i=1}^n$, denoted by $\sum_{k=1}^K \pi_{vk}\phi(\alpha; u_{vk}, \sigma_{vk}^2)$.
5:     Obtain the cluster membership matrix $\mathbf{M}_v$:
           $m_{ik}^v = \frac{\pi_{vk}\phi(\alpha_{iv}; u_{vk}, \sigma_{vk}^2)}{\sum_{j=1}^K \pi_{vj}\phi(\alpha_{iv}; u_{vj}, \sigma_{vj}^2)},$
       for $1 \leq i \leq n, \ 1 \leq k \leq K$.
6:     Construct a binary membership indicator matrix $\mathbf{B}_v$:
           $b_{ik}^v = \begin{cases} 1, & \text{if } k = \arg\max_{1\leq j\leq K}\{m_{ij}^v\}; \\ 0, & \text{otherwise.} \end{cases}$
7:     Calculate the weight $w_v (> 0)$: $\sum_{v=1}^V w_v = 1$.
8: end for
9: Apply a multivariate clustering method on the affinity matrix $\mathbf{A} = \sum_{v=1}^V w_v \mathbf{B}_v \mathbf{B}_v^T$ and return the identified cluster labels $\{z_i\}_{i=1}^n$.
```

The development of an effective statistic for assessing the separation between two Gaussian distributions traces back to the work of (Dasgupta, 1999), where the concept of c-separation was introduced. We here adopt the definition given by (Maitra & Melnykov, 2010), which is in keeping with our clustering objective. For two multivariate Gaussians $\mathcal{N}(\mathbf{u}_k, \Sigma_k)$ and $\mathcal{N}(\mathbf{u}_j, \Sigma_j)$ with mixing proportions π_k and π_j, define $\varepsilon_{j|k}$ as the probability that an instance from the kth mixture component $\mathcal{N}(\mathbf{u}_k, \Sigma_k)$ is misclassified into the jth mixture component:

$$\varepsilon_{j|k} = \Pr\big(\pi_k\phi(\mathbb{X}; \mathbf{u}_k, \Sigma_k) < \pi_j\phi(\mathbb{X}; \mathbf{u}_j, \Sigma_j) \\ | \mathbb{X} \sim \mathcal{N}(\mathbf{u}_k, \Sigma_k)\big).$$

Analytic calculation of $\varepsilon_{j|k}$ is impractical, but numerical computation can be readily done for univariate GMMs. The total probability of misclassification, denoted by ε_v, can be evaluated by either $\varepsilon_v = \sum_{k=1}^K \sum_{j\neq k} \varepsilon_{j|k}$ or $\varepsilon_v = \sum_{k=1}^K \pi_{vk}(\sum_{j\neq k} \varepsilon_{j|k})$, where the later is weighted by the estimated mixing proportions. Then the weight w_v for the vth base clustering is $w_v = \frac{\varepsilon_v^{-1}}{\sum_{j=1}^V \varepsilon_j^{-1}}$.

[1] https://github.com/EAkeweje/GPmix

Both the number of projection functions V and the number of clusters K can be determined according to proper internal clustering validation criteria. Given that we employ the parametric model GMM on the projection coefficients, we can alternatively identify the optimal K value through the Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC). In our Python package, we fit multiple GMMs to the projection coefficients $\{\langle x_i - \mu, \beta\rangle\}_{i=1}^n$, where β is the first eigen-function of the population kernel $\Sigma(s,t)$, and the optimal K is with the GMM having the minimum BIC or AIC value.

4. Theoretical Analysis

4.1. Identifiability of GP Mixtures

A finite mixture model is identifiable if a given dataset leads to a uniquely determined set of estimated parameters up to a permutation of the clusters. (Teicher, 1961) and (Teicher, 1963) pioneered the study of identifiability for finite mixture distributions; (Yakowitz & Spragins, 1968) showed that finite mixtures of multivariate Gaussian distributions with variable mean vectors and covariance matrices are identifiable. We here give sufficient conditions for the identifiability of GP mixtures.

In the Karhunen-Loève expansion, the Gaussian random function X_k can be written down as

$$X_k(t) = \mu_k(t) + \sum_{v=1}^{\infty} a_{kv} b_{kv}(t),$$

where the random coefficient a_{kv} is given by $a_{kv} = \langle X_k - \mu_k, b_{kv}\rangle$, and satisfy the following: $\mathrm{E}[a_{kv}] = 0, \mathrm{var}(a_{kv}) = \lambda_{kv}$ and $\mathrm{E}[a_{kv}a_{kr}] = 0$ for any $r \neq v$. Even though the expansion is infinite dimensional, in application, a finite number V_k exists for a given functional dataset, such that the first leading V_k eigen-functions can efficiently represent the sample functions. Therefore, we approximate X_k by the truncated Karhunen-Loève expansion:

$$\hat{X}_k(t) = \mu_k(t) + \sum_{v=1}^{V_k} a_{kv} b_{kv}(t), \quad k = 1, \ldots, K.$$

We let $H_k = \{\mu_k(t) + \sum_{v=1}^{V_k} a_v b_{kv}(t) : a_v \in \mathbb{R}, v = 1, \ldots, V_k\}$ denote the finite-dimensional linear space spanned by the mean function and eigen-functions from the truncated Karhunen-Loève expansion of X_k.

If we project X_k onto the finite-dimensional space H_l attributed to X_l, we obtain the projection:

$$\mathcal{P}_{H_l}(X_k) = \mu_l(t) + \sum_{v=1}^{V_l} a_{lv}^k b_{lv}(t).$$

If the two distributions $GP(\mu_k, \Sigma_k)$ and $GP(\mu_l, \Sigma_l)$ are non-identifiable, then the expected value (w.r.t. the distribution of X_k) of the L^2-norm $\|\hat{X}_k - \mathcal{P}_{H_l}(X_k)\|$ will be approximately zero. Hence the magnitude of the discrepancy $\|\hat{X}_k - \mathcal{P}_{H_l}(X_k)\|$ unravels the identifiability between the GP mixture components X_k and X_l. If the expected value of the discrepancy $\|\hat{X}_k - \mathcal{P}_{H_l}(X_k)\|$ is large for any mixture component X_l $(l \neq k)$, then the true cluster membership of any $x \sim GP(\mu_k, \Sigma_k)$ can be easily identified.

Theorem 4.1. *Let $R_k = X_k - \hat{X}_k = \sum_{r=V_k+1}^{\infty} a_{kr} b_{kr}$ denote the residual random function for X_k. The squared L^2-distance between $\hat{X}_k$ and $\mathcal{P}_{H_l}(X_k)$ has the following form:*

$$\begin{aligned}\|\hat{X}_k - \mathcal{P}_{H_l}(X_k)\|^2 &= \|\mu_k - \mu_l\|^2 + 2\sum_{v=1}^{V_k} a_{kv}\langle \mu_k - \mu_l, b_{kv}\rangle \\ &+ \Big[\sum_{v=1}^{V_k} a_{kv}^2 - \sum_{v=1}^{V_l} \langle \hat{X}_k - \mu_l, b_{lv}\rangle^2\Big] + \sum_{v=1}^{V_l} \langle R_k, b_{lv}\rangle^2.\end{aligned} \tag{4}$$

If the eigen-value λ_{kv} decays rapidly for $v > V_k$ such that $V_l \sum_{v=V_k+1}^{\infty} \lambda_{kv}$ converges to 0 as $V_k \to \infty$ and $V_l \to \infty$, then the last term $\sum_{v=1}^{V_l} \langle R_k, b_{lv}\rangle^2$ converges to 0 in probability.

Theorem 4.1 was adapted from Theorem 1 of (Chiou & Li, 2007). In Equation (4), the two terms in the upper line are related to the two mean functions, while the terms within the square brackets are related to the two sets of eigen-functions. In general, if the two mean functions μ_k and μ_l are not identical (with respect to the Lebesgue measure on $\mathcal{T}$), or if $\mu_k \notin H_l$, then the expected value of the L^2-norm $\|\hat{X}_k - \mathcal{P}_{H_l}(X_k)\|$ will be bounded away from 0. In particular, if the following two pathological phenomena do not occur, then the discrepancy $\|\hat{X}_k - \mathcal{P}_{H_l}(X_k)\|$ is expected to be distinguishable:

- If the two mean functions μ_k and μ_l are identical, and moreover $b_{kv} \in \{\sum_{r=1}^{V_l} a_r b_{lr} : a_r \in \mathbb{R}, r = 1, \ldots, V_l\}$, for $v = 1, \ldots, V_k$, then the discrepancy $\|\hat{X}_k - \mathcal{P}_{H_l}(X_k)\|$ reduces to the residual term, and hence its expected value converges to 0.
- If the two mean functions μ_k and μ_l are not identical, but $\mu_k \in H_l$ and $b_{kv} \in \{\sum_{r=1}^{V_l} a_r b_{lr} : a_r \in \mathbb{R}, r = 1, \ldots, V_l\}$ for $v = 1, \ldots, V_k$, then again the discrepancy reduces to the residual term, and hence $GP(\mu_k, \Sigma_k)$ and $GP(\mu_l, \Sigma_l)$ are non-identifiable.

4.2. Random Projection

In the context of learning high-dimensional GMMs, projection to lower-dimensional subspaces has proven to be indispensable (Dasgupta, 1999; Bingham & Mannila, 2001). One important property is that data from a mixture of K

Gaussians can be projected into subspaces of $O(\log(K))$ dimensions, while still retaining the approximate level of separation between the Gaussian components. In this section, we study the probability that a 1-dimensional random projection achieves a separation of ϵ or higher.

Let $\mathcal{H}(\mathcal{T},\mathbb{R})$ be a separable Hilbert space. (When $\mathcal{T}$ is a closed and bounded interval, then for any continuous function $x : \mathcal{T} \mapsto \mathbb{R}$, we have $x \in \mathcal{H}(\mathcal{T},\mathbb{R})$.) For any function $\beta \in \mathcal{H}(\mathcal{T},\mathbb{R})$, the marginal distribution of the projection coefficient $\langle x_i - \mu, \beta\rangle$ is the univariate GMM $\sum_{k=1}^K \pi_k \mathcal{N}(\langle \mu_k - \mu, \beta\rangle, \sum_{l=1}^\infty \lambda_{kl}\langle b_{kl}, \beta\rangle^2)$. A Borel probability measure g on $\mathcal{H}(\mathcal{T},\mathbb{R})$ is called Gaussian if each of its one-dimensional projections is Gaussian. It is non-degenerate if, in addition, each of its one-dimensional projections is non-degenerate. Theorem 4.2 is reproduced from Theorem 4.1 of (Cuesta-Albertos et al., 2007).

Theorem 4.2. *Let X and Y be two random elements taking values in $\mathcal{H}(\mathcal{T},\mathbb{R})$. Let $X \sim GP(\mu,\Sigma)$ be a Gaussian random function. Let g be a non-degenerate Gaussian measure on $\mathcal{H}(\mathcal{T},\mathbb{R})$, independent of the probability laws of X and Y. If $g\left(\{\beta \in \mathcal{H}(\mathcal{T},\mathbb{R}) : \langle \beta, X\rangle \overset{d}{=} \langle \beta, Y\rangle\}\right) > 0$, where the notation $\overset{d}{=}$ stands for equality in distribution, then the probability law of Y is $GP(\mu,\Sigma)$.*

Theorem 4.1 in (Cuesta-Albertos et al., 2007) does not assume that the random function X is Gaussian, but only that the distribution of X is determined by its moments. A random function X is moment-determined if, e.g., the absolute moments $m_r = \mathrm{E}[\|X\|^r]$ are finite and satisfy the Carleman condition: $\sum_{r\ge 1} m_r(X)^{-1/r} = \infty$. It can be readily proved that Gaussian random functions satisfy the Carleman condition and hence are moment-determined. Theorem 4.2 establishes that, given a reference Gaussian probability measure g, if two Gaussian random functions have different probability laws, then the g-probability of finding two one-dimensional projections identically distributed is zero. Therefore, Theorem 4.2 justifies our implementation of random projection in step 3 of Algorithm 1.

The projection function β is generated at random from the Gaussian distribution g. One appropriate Gaussian measure is the strictly stationary Ornstein-Uhlenbeck process with mean 0 and covariance function $\mathrm{cov}(U_t, U_s) = e^{-|s-t|}$; that is, the mean-reversion coefficient is 1, and $\beta(t) \sim \mathcal{N}(0,1)$ for any $t \in \mathcal{T}$. We below explain another approach, and provide an upper bound on the expected number of projections required to achieve ϵ-separation.

Let $\{b_v\}_{v\in\mathbb{N}}$ denote a functional basis of $\mathcal{H}(\mathcal{T},\mathbb{R})$, and we approximate each Gaussian random function X_k by $\hat{X}_k$: $\hat{X}_k(t) = \sum_{v=1}^p \alpha_{kv} b_v(t)$, where p is large enough to offer a good approximation for each random function. To generate a projection function $\beta(t)$, we randomly generate a coefficient vector $\mathbf{a} = (a_1,\ldots,a_p)^T$ from the standard normal distribution: $a_v \sim \mathcal{N}(0,1)$ for $v = 1,\ldots,p$, and write $\beta(t) = \sum_{v=1}^p a_v b_v(t)$. It can be readily proved that the projection function β is from a Gaussian distribution. Note that the Gaussian measure associated with β is degenerate; a non-degenerate Gaussian process can be $\beta + \beta_g$, with β_g a Gaussian process tightly concentrated around zero, albeit employing β or $\beta + \beta_g$ has negligible effects in practice.

We then replace the 1-dimensional projection $\langle X_k, \beta\rangle$ by $\langle \hat{X}_k, \beta\rangle$, and we have

$$\langle \hat{X}_k, \beta\rangle = \langle \sum_{v=1}^p \alpha_{kv} b_v, \sum_{r=1}^p a_r b_r\rangle = (\mathbf{B}\boldsymbol{\alpha}_k)^T \mathbf{a}, \quad (5)$$

where $\mathbf{B} = [\langle b_v, b_r\rangle]_{p\times p}$, and $\boldsymbol{\alpha}_k = (\alpha_{k1},\ldots,\alpha_{kp})^T$ is a Gaussian random vector. Equation (5) reduces the 1-dimensional projection of the random function $\hat{X}_k$ to the 1-dimensional projection of the random vector $\mathbf{B}\boldsymbol{\alpha}_k$. For $k = 1,\ldots,K$, let $\mathcal{N}(\mathbf{u}_k, \boldsymbol{\Sigma}_k)$ denote the Gaussian distribution of $\mathbf{B}\boldsymbol{\alpha}_k$. By projecting the functional data $\{x_i\}_{i=1}^n$ onto the linear space expanded by $\{b_v\}_{v=1}^p$, the projected multivariate data confirm to the multivariate GMM $\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{u}_k, \boldsymbol{\Sigma}_k)$. For two p-variate Gaussian distributions $\mathcal{N}(\mathbf{u}_k, \boldsymbol{\Sigma}_k)$ and $\mathcal{N}(\mathbf{u}_j, \boldsymbol{\Sigma}_j)$, where the covariance matrices are semi-positive definite, we have the following theorem (reproduced from (Kushnir et al., 2019)):

Theorem 4.3. *Let $\mathbf{a} = (a_1,\ldots,a_p)^T$ denote the random projection vector, where $a_v \sim \mathcal{N}(0,1)$ for $v = 1,\ldots,p$. Then the probability that the two projected Gaussians achieve a separation of ϵ or higher:*

$$\Pr\Big(\frac{|\langle \mathbf{u}_k, \mathbf{a}\rangle - \langle \mathbf{u}_j, \mathbf{a}\rangle|}{\sqrt{\mathbf{a}^T\boldsymbol{\Sigma}_k\mathbf{a}} + \sqrt{\mathbf{a}^T\boldsymbol{\Sigma}_j\mathbf{a}}} \ge \epsilon\Big),$$

is lower bounded by

$$Q\left(\sqrt{(1+\tau)\frac{p-1}{p-\zeta}\zeta}\right)\left[1-\exp\left(-\frac{p-1}{2}[\tau-\log(1+\tau)]\right)\right],$$

where $\tau > 0$ is a free parameter, $Q(\mathbb{x}) = \Pr(\mathbb{X} \ge \mathbb{x} | \mathbb{X} \sim \mathcal{N}(0,1))$ is the complementary cumulative distribution function, and

$$\zeta = \frac{2p\epsilon^2 \lambda_{max}(\boldsymbol{\Sigma}_k + \boldsymbol{\Sigma}_j)}{\|\mathbf{u}_k - \mathbf{u}_j\|^2}.$$

Let $\mathbb{n}(\epsilon)$ denote the expected number of random projections required to attain ϵ-separation in 1-dimension, we have $\lim_{p\to\infty} \mathbb{n}(\epsilon) \le \frac{1}{2Q(\sqrt{\zeta})}$. If ϵ is such that $\zeta = (\log(\log(p)))^{1-\eta}$, where $\eta > 0$ is a free parameter, then $\mathbb{n}(\epsilon)$ is sub-logarithmic in p: $\mathbb{n}(\epsilon) = o(\log(p))$.

Theorem 4.3 establishes that for sufficiently separated Gaussians, after $o(\log(p))$ random projections, a good direction will be found that yields ϵ-separation in 1-dimension.

4.3. Polynomial Learnability

Given the 1-dimensional GMM of ϵ-separation from step 3, we now turn to the algorithmic problem of provably recovering good estimates of the unknown parameters in the univariate GMM, in polynomial time and a polynomial number of samples.

We adapt the ideas from (Moitra & Valiant, 2010) to provide the theoretical guarantees on step 4 of Algorithm 1. Theorem 4.4 below is on the polynomial learnability in the sample size, and Theorem 4.5 below is on the polynomial learnability in the runtime. Our proofs for the two theorems differ from the approach taken by (Moitra & Valiant, 2010).

Theorem 4.4. *Let $\mathbb{X}$ follow the univariate GMM $\sum_{k=1}^{K} \pi_k \mathcal{N}(u_k, \sigma_k^2)$, and let $\mathbb{x}_1, \mathbb{x}_2, \dots, \mathbb{x}_m$ be independent draws from the univariate GMM. The univariate GMM is in isotropic position, and $\pi_k \geq \epsilon$ for any $1 \leq k \leq K$. Then with probability at least $1 - \delta$,*

$$|\frac{1}{m}\sum_{i=1}^{m} \mathbb{x}_i^r - E\left[\mathbb{X}^r\right]|^2 \leq \frac{1}{m\delta} O\left(\epsilon^{-r}\right),$$

where $E\left[\mathbb{X}^r\right]$ is the rth order raw moment, and the hidden constant on the big-Oh depends on the order r.

Theorem 4.5. *Let $\mathbb{X}$ follow the univariate GMM $\sum_{k=1}^{K} \pi_k \mathcal{N}(u_k, \sigma_k^2)$. Assume that $\pi_k \geq \epsilon$ and $|u_k - u_j| + |\sigma_k^2 - \sigma_j^2| \geq \epsilon$, for any $1 \leq j \neq k \leq K$, and that $\mathbb{X}$ has zero mean and a bounded variance: $\mathrm{E}[\mathbb{X}] = 0$ and var$(\mathbb{X}) \leq 1$. Given a target accuracy $\xi \leq \epsilon$ and an integer R, we can find (in polynomial time) the parameter estimate $(\hat{\pi}_1, \hat{u}_1, \hat{\sigma}_1^2, \dots, \hat{\pi}_K, \hat{u}_K, \hat{\sigma}_K^2)$ that gives $|E[\mathbb{X}^r] - E[\hat{\mathbb{X}}^r]| \leq \xi$, for any $1 \leq r \leq R$. Here, $\hat{\mathbb{X}}$ is the random variable defined by the univariate GMM $\sum_{k=1}^{K} \hat{\pi}_k \mathcal{N}(\hat{u}_k, \hat{\sigma}_k^2)$. Moreover, we have $\hat{\pi}_k \geq \epsilon/2$ and $|\hat{u}_k - \hat{u}_j| + |\hat{\sigma}_k^2 - \hat{\sigma}_j^2| \geq \epsilon/2$, for any $1 \leq j \neq k \leq K$.*

In Theorem 4.4, for any target accuracy $\xi \leq \epsilon$, if we let the sample size m be polynomial in ξ^{-1}, δ^{-1} and ϵ^{-1}: $m = (\xi\delta)^{-1} O\left(\epsilon^{-R}\right)$, then with probability at least $1 - \delta$, the rth-order sample moment will be within ξ of the corresponding true moment, for any $r = 1, \dots, R$. The proof for Theorem 4.5 indicates that, through a brute-force search over a uniform grid, we will find a grid point $(\hat{\pi}_1, \hat{u}_1, \hat{\sigma}_1^2, \dots, \hat{\pi}_K, \hat{u}_K, \hat{\sigma}_K^2)$ that, for any $r = 1, \dots, R$, the moment $\mathrm{E}[\hat{\mathbb{X}}^r]$ is within ξ from the true moment $\mathrm{E}[\mathbb{X}^r]$. Moreover, the grid width γ is polynomial in the target accuracy ξ: $\gamma = O(\xi^{\frac{R}{2}+1})$, with the big-Oh depending on K, and therefore the runtime is polynomial in the target accuracy. In practice, we can only calculate the difference between the estimated and sample moments, i.e., $|\mathrm{E}[\hat{\mathbb{X}}^r] - \frac{1}{m}\sum_{i=1}^{m} \mathbb{x}_i^r|$, rather than calculating the difference between the estimated and true moments $|\mathrm{E}[\hat{\mathbb{X}}^r] - \mathrm{E}[\mathbb{X}^r]|$; Theorem 4.4 and 4.5 together indicate that, with probability at least $1 - \delta$, the method of moments will find a grid point (in polynomial time and and a polynomial number of samples) at which the raw moment $\mathrm{E}[\hat{\mathbb{X}}^r]$ is within 2ξ of the true moment $\mathrm{E}[\mathbb{X}^r]$, for any $r = 1, \dots, R$.

We might let $\Theta = (\pi_1, u_1, \sigma_1^2, \dots, \pi_K, u_K, \sigma_K^2)$ and its estimate $\hat{\Theta} = (\hat{\pi}_1, \hat{u}_1, \hat{\sigma}_1^2, \dots, \hat{\pi}_K, \hat{u}_K, \hat{\sigma}_K^2)$. We note that the moment difference function $\psi_r(\Theta, \hat{\Theta}) := \mathrm{E}[\mathbb{X}^r] - \mathrm{E}[\hat{\mathbb{X}}^r]$ is a polynomial of $6K$ variables (i.e., the $6K$ model parameters). Let I_r be the ideal in the (Noetherian) ring of polynomials, which are generated by the polynomials $\{\psi_1, \psi_2, \dots, \psi_r\}$. Then we have an increasing sequence of ideals $I_1 \subset I_2 \subset I_3 \subset \cdots$. Define $I = \cup_{r=1}^{\infty} I_r$, which is an ideal according to the ascending chain condition. By the Hilbert basis theorem, the ideal I is finitely generated. In particular, there exists an integer R such that $I_R = I_{R+1} = \cdots$; that is, I_R contains all the generators of I: for any $r > R$, we can write

$$\psi_r(\Theta, \hat{\Theta}) = \sum_{j=1}^{R} c_j(\Theta, \hat{\Theta}) \psi_j(\Theta, \hat{\Theta}),$$

where each coefficient $c_j(\Theta, \hat{\Theta})$ is a polynomial of the $6K$ parameters. We can conclude that, if the two moments $\mathrm{E}[\mathbb{X}^r]$ and $\mathrm{E}[\hat{\mathbb{X}}^r]$ coincide for all orders from 1 to R (namely, $\psi_r(\Theta, \hat{\Theta}) = 0$ for $r = 1, \dots, R$), then the two moments $\mathrm{E}[\mathbb{X}^r]$ and $\mathrm{E}[\hat{\mathbb{X}}^r]$ coincide for any order $r \geq 1$. Therefore, we claim that the distribution function of $\mathbb{X}$ (not the parameter Θ) can be uniquely identified via the moments $\{\mathrm{E}[\mathbb{X}^r] : r = 1, \dots, R\}$. Note that (1) if the univariate GMM $f(x; \Theta) = \sum_{k=1}^{K} \pi_k \phi(x; u_k, \sigma_k^2)$ is identifiable, in that $f(x; \Theta_1) \neq f(x; \Theta_2)$ for any $\Theta_1 \neq \Theta_2$, then the set of parameters Θ can be uniquely identified via the moments $\{\mathrm{E}[\mathbb{X}^r] : r = 1, \dots, R\}$; (2) we here only prove that finitely many moments are able to uniquely identify the distribution function of $\mathbb{X}$; (Moitra & Valiant, 2010) proved that the exact number of moments required is $R = 4K - 2$.

For many distribution families, identifying the values of model parameters uniquely is impossible, due to the fact that multiple parameter values can yield the same probability distribution. We now prove that, for the univariate GMM $f(x; \Theta)$, if the mixture components have non-zero pairwise parameter distance and non-zero weights, then the model parameter Θ is identifiable.

Theorem 4.6. *Let $\mathbb{X}$ follow the univariate GMM $\sum_{k=1}^{K} \pi_k \mathcal{N}(u_k, \sigma_k^2)$. Assume that $\pi_k \geq \epsilon$ and $|u_k - u_j| + |\sigma_k^2 - \sigma_j^2| \geq \epsilon$, for any $1 \leq j \neq k \leq K$. Then the model parameter Θ is identifiable.*

5. Experiments

We validate the efficacy of GPmix (Algorithm 1) on 12 synthetic datasets and 10 real datasets, benchmarking it against existing functional data clustering algorithms available in

R or Python. We adopt the widely used Adjusted Rand Index (ARI) and Adjusted Mutual Information (AMI) for performance evaluation. In Appendix G.1, we explain the arguments in our GPmix Python package and elaborate on the specific configurations tailored to each dataset for all the seven algorithms involved in the benchmarking study.

5.1. Simulated Datasets

To ensure a comprehensive assessment, we investigate 12 simulation scenarios, covering a range of cluster sizes, sample sizes, and noise levels. Some scenarios are adopted from prior studies, while others draw inspiration from the same sources. Details of these simulation scenarios are provided in Appendix G.2. For each scenario, we apply the clustering algorithms on 100 randomly generated datasets. The resulting mean and standard deviation of the 100 AMI scores are given in Table 1, and for the ARI scores, in Table 2.

The GPmix algorithm consistently ranks within the top three, emerging as the top performer in 6 out of the 12 simulation scenarios. For scenario J, all the seven algorithms struggled to identify the true underlying structure. This difficulty arises because the clusters in this scenario have an identical mean function, making the GP mixture unidentifiable.

We note that the first six simulation scenarios (A to F) are not a GP mixture model, yet our algorithm adeptly generated clusters that accurately mirror the underlying data structure in all of these scenarios. The latter six datasets (G to L) are from the GP mixture model, wherein Algorithm 1 is theoretically expected to excel. As observed, the algorithm effectively fulfills these expectations by consistently producing either the optimal clustering or one that closely rivals the best in all six scenarios.

5.2. Real Datasets

We evaluated the seven algorithms on 10 real datasets from the UEA & UCR Time Series Classification Repository: ArrowHead (AH), BirdChicken (BC), CBF, DiatomSizeReduction (DSR), ECG200 (ECG), FaceFour (FF), GunPoint (GuP), Meat, Strawberry (SB), and Symbols (SYM). These datasets, characterized by their substantial sizes and diverse underlying structures, present a notable challenge for many clustering algorithms.

Each algorithm underwent the clustering procedure ten times to mitigate the impact of initialization, and we reported the highest scores for each dataset in Table 3. Our GPmix algorithm outperforms the others on 8 out of the 10 datasets. Generally, the AMI and ARI scores are lower than those observed in the simulation study, indicating the complexity of these real datasets. However, in comparison to the other six algorithms, our GPmix algorithm consistently demonstrates superior performance. In Appendix G.3, we provide a detailed illustration of the clustering procedure of the GPmix algorithm using two datasets: one simulated data (Scenario F) and one real data (CBF).

Table 1. Mean (upper line) and standard deviation (lower line) of the AMI score for the 12 simulation scenarios.

Sim.	GPmix	FEM	HDD	clu	FC	km	ADP
A	**0.79**	0.66	0.63	0.16	0.71	0.40	0.62
	0.02	0.01	0.06	0.18	0.01	0.08	0.02
B	0.32	0.00	0.15	0.01	**0.52**	0.00	0.11
	0.23	0.01	0.25	0.02	**0.21**	0.01	0.05
C	0.49	0.41	0.47	0.00	**0.64**	0.07	0.52
	0.06	0.09	0.05	0.01	**0.16**	0.05	0.07
D	0.54	0.08	0.47	0.04	**0.62**	0.01	0.45
	0.10	0.04	0.10	0.07	**0.11**	0.01	0.11
E	0.97	0.00	0.48	0.36	0.97	0.00	**1.00**
	0.02	0.01	0.05	0.22	0.14	0.01	**0.03**
F	**0.97**	0.41	0.74	0.30	0.83	0.40	0.88
	0.03	0.08	0.07	0.17	0.03	0.11	0.10
G	0.46	0.06	**0.49**	0.01	0.49	0.07	0.41
	0.05	0.05	**0.05**	0.02	0.08	0.04	0.07
H	**0.97**	0.23	0.72	0.18	0.95	0.26	0.95
	0.02	0.26	0.12	0.28	0.02	0.11	0.02
I	**0.74**	0.10	0.13	0.12	0.67	0.27	0.73
	0.06	0.02	0.03	0.24	0.02	0.11	0.05
J	**0.08**	0.00	0.00	0.00	0.00	0.00	0.00
	0.10	0.01	0.01	0.00	0.00	0.00	0.01
K	0.96	0.99	**1.00**	0.00	0.60	0.51	0.70
	0.04	0.07	**0.01**	0.00	0.32	0.41	0.18
L	**0.70**	0.03	0.04	0.00	0.67	0.26	0.67
	0.08	0.02	0.02	0.01	0.15	0.11	0.09

Table 2. Mean (upper line) and standard deviation (lower line) of the ARI score for the 12 simulation scenarios.

Sim.	GPmix	FEM	HDD	clu	FC	km	ADP
A	**0.74**	0.57	0.53	0.14	0.63	0.33	0.52
	0.03	0.03	0.08	0.16	0.02	0.09	0.04
B	0.34	0.00	0.18	0.01	**0.61**	0.00	0.09
	0.30	0.02	0.31	0.03	**0.27**	0.01	0.06
C	0.42	0.29	0.36	0.00	**0.57**	0.04	0.43
	0.08	0.09	0.07	0.01	**0.21**	0.04	0.09
D	0.55	0.08	0.44	0.04	**0.60**	0.00	0.43
	0.12	0.04	0.11	0.06	**0.13**	0.01	0.11
E	0.98	0.00	0.41	0.35	0.96	0.00	**1.00**
	0.02	0.01	0.05	0.23	0.16	0.01	**0.03**
F	**0.96**	0.27	0.63	0.23	0.76	0.29	0.82
	0.05	0.08	0.09	0.14	0.04	0.12	0.15
G	0.37	0.04	0.42	0.00	**0.43**	0.05	0.35
	0.05	0.04	0.06	0.01	**0.09**	0.04	0.07
H	**0.99**	0.23	0.59	0.20	0.98	0.26	0.98
	0.01	0.30	0.21	0.31	0.01	0.13	0.01
I	0.75	0.09	0.11	0.13	0.71	0.27	**0.78**
	0.14	0.02	0.03	0.28	0.08	0.14	**0.10**
J	**0.11**	0.00	0.00	0.00	0.00	0.00	0.00
	0.13	0.01	0.01	0.01	0.01	0.01	0.01
K	0.98	0.99	**1.00**	0.00	0.65	0.54	0.78
	0.02	0.05	**0.01**	0.01	0.36	0.42	0.19
L	**0.75**	0.03	0.03	0.00	0.69	0.26	0.72
	0.08	0.02	0.02	0.01	0.18	0.12	0.10

Table 3. AMI scores (upper line) and ARI scores (lower line) for the 10 real datasets.

DATA	GPMIX	FEM	HDD	CLU	FC	KM	ADP
AH	**0.37**	0.25	0.22	0.05	0.24	0.28	0.19
	0.36	0.29	0.21	0.01	0.25	0.26	0.18
BC	**0.24**	0.03	0.06	0.22	0.08	0.10	0.08
	0.29	0.04	0.07	0.15	0.10	0.10	0.10
CBF	**0.84**	0.37	0.47	0.01	0.53	0.34	0.40
	0.87	0.35	0.44	0.00	0.44	0.31	0.31
DSR	**0.94**	0.79	0.82	0.00	0.83	0.72	0.78
	0.95	0.83	0.86	0.01	0.86	0.73	0.82
ECG	**0.37**	0.15	0.17	0.03	0.37	0.17	0.07
	0.38	0.26	0.28	0.03	0.37	0.28	0.14
FF	**0.77**	0.47	0.40	0.06	0.56	0.50	0.44
	0.76	0.41	0.36	0.08	0.54	0.45	0.32
GUP	**0.34**	0.00	0.00	0.02	0.00	0.15	0.01
	0.25	0.00	0.00	0.02	0.00	0.07	0.01
MEAT	0.70	**0.93**	0.54	0.36	0.54	0.66	0.72
	0.69	**0.95**	0.44	0.37	0.49	0.69	0.69
SB	**0.32**	0.08	0.00	0.03	0.12	0.07	0.03
	0.30	0.00	0.00	0.03	0.04	0.07	0.05
SYM	0.75	0.63	0.77	0.00	**0.85**	0.69	0.37
	0.66	0.53	0.67	0.00	**0.80**	0.62	0.30

Table 4 outlines the computation runtimes for the seven algorithms, each executed with the optimal configuration. All experiments were conducted on a PC with a 3.20GHz processor, 16 CPU cores, and 32GB of RAM. We excluded the time spent on smoothing from the benchmark timings, since this preprocessing step is essential for all seven clustering algorithms. Clearly, clu, FC and ADP struggle when confronted with complex datasets. In contrast, GPmix proves to be both effective and efficient, clustering the real datasets in less than a second. The SYM dataset comprises 1020 sample curves, each evaluated at 398 points. Comparing the computation runtime of GPmix on SYM to that of other algorithms demonstrates the scalability of the GPmix algorithm.

Table 4. The run time for the real datasets.

DATASET	RUNTIME (MILLISECONDS)						
	GPMIX	FEM	HDD	CLU	FC	KM	ADP
AH	107	120	**59**	2112	290K	215	396
BC	**18**	27	20	242	2M	66	45
CBF	**242**	2174	3959	102K	57K	893	5909
DSR	625	**481**	3256	6338	1.2M	507	1034
ECG	53	49	**37**	20K	14K	121	378
FF	**157**	413	147	390	1.4M	192	158
GUP	**50**	283	106	18K	47K	134	400
MEAT	**58**	744	108	1548	2M	160	167
SB	**144**	1608	1546	1.6M	570K	683	14K
SYM	**551**	9803	3992	1.2M	3.4M	2917	40K

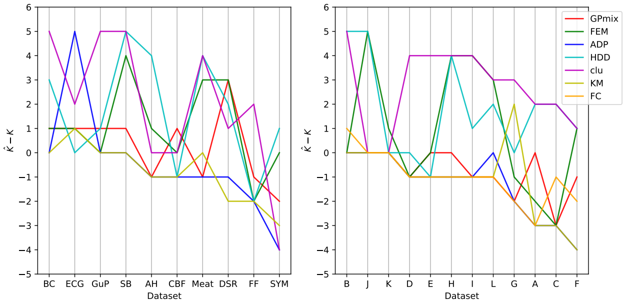

Figure 1. Difference between actual and predicted cluster quantities, with datasets ordered by increasing cluster counts.

5.3. Number of Clusters

In the above study, for each algorithm, the number of clusters is fixed at the true count. We now contrast their performance by comparing their estimated cluster numbers. Step 4 in Algorithm 1 allows us to utilize model selection techniques such as BIC or AIC for determining the optimal number of mixture components. Specifically, in our Python package, we conduct eigen-decomposition of the population kernel $\Sigma(s,t)$, followed by evaluating the AIC/BIC score of the GMM fitted to the fPC scores associated with the first eigen-function, as it explains the most variation in the data. The mixture model with the lowest AIC/BIC score indicates the optimal model, and its number of components represents the number of clusters. Figure 1 plots the estimation errors on the number of clusters for each dataset. The datasets are arranged in ascending order according to their cluster numbers. FEM, HDD, and clu base their cluster numbers on the BIC score, while the remaining three algorithms rely on the Silhouette score. GPmix provides estimates within one unit of the true count in 10 simulation scenarios and 8 real datasets. Following GPmix, the FC algorithm demonstrated commendable performance.

6. Conclusion

We developed a simple yet efficient technique for learning GP mixture models. Our method involves projecting functional data onto multiple one-dimensional functions, and learning a univariate GMM for each projection. We established a lower bound on the expected number of projections required to achieve effective separation within the 1-dimensional mixture components. For univariate GMMs, our algorithm ensures accurate estimation of unknown parameters in polynomial time and with a polynomial number of samples. This development significantly extends the applicability of GP mixture models in cluster analysis. Notably, our numerical study demonstrated the robust performance of our method even in cases where the functional data are not Gaussian.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101002240) and funding from the Science Foundation Ireland under Grant number 21/RC/10295_P2. We are grateful to the four anonymous reviewers for their valuable comments.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Atienza, N., Garcia-Heras, J., and Muñoz-Pichardo, J. A new condition for identifiability of finite mixture distributions. *Metrika*, 63(2):215 – 221, 2006. doi: 10.1007/s00184-005-0013-z.

Bingham, E. and Mannila, H. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 245–250. Association for Computing Machinery, 2001. doi: 10.1145/502512.502546.

Bouveyron, C., Fauvel, M., and Girard, S. Kernel discriminant analysis and clustering with parsimonious Gaussian process models. *Statistics and Computing*, 25(6):1143–1162, 2015. doi: 10.1007/s11222-014-9505-x.

Chen, H., Reiss, P. T., and Tarpey, T. Optimally weighted L^2 distance for functional data. *Biometrics*, 70(3):516–525, 2014.

Chiou, J.-M. and Li, P.-L. Functional clustering and identifying substructures of longitudinal data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(4):679–699, 2007.

Cuesta-Albertos, J. A., Fraiman, R., and Ransford, T. A sharp form of the cramér-wold theorem. *Journal of Theoretical Probability*, 20(2):201 – 209, 2007. doi: 10.1007/s10959-007-0060-7.

Dasgupta, S. Learning mixtures of Gaussian. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pp. 634–644, 1999. doi: 10.1109/SFFCS.1999.814639.

Golovkine, S., Klutchnikoff, N., and Patilea, V. Clustering multivariate functional data using unsupervised binary trees. *Computational Statistics & Data Analysis*, 168: 107376, 2022.

Hall, P., Müller, H.-G., and Wang, J.-L. Properties of principal component methods for functional and longitudinal data analysis. *Annals of Statistics*, 34(3):1493 – 1517, 2006. doi: 10.1214/009053606000000272.

Huang, M., Li, R., Wang, H., and Yao, W. Estimating mixture of Gaussian processes by kernel smoothing. *Journal of Business and Economic Statistics*, 32(2):259–270, 2014. doi: 10.1080/07350015.2013.868084.

Jackson, E., Davy, M., Doucet, A., and Fitzgerald, W. J. Bayesian unsupervised signal classification by Dirichlet process mixtures of Gaussian processes. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 3, 2007. doi: 10.1109/ICASSP.2007.366870.

Jacques, J. and Preda, C. Funclust: A curves clustering method using functional random variables density approximation. *Neurocomputing*, 112:164–171, 2013. doi: 10.1016/j.neucom.2012.11.042.

Jacques, J. and Preda, C. Model-based clustering for multivariate functional data. *Computational Statistics & Data Analysis*, 71:92–106, 2014. doi: 10.1016/j.csda.2012.12.004.

James, G. M. and Sugar, C. A. Clustering for sparsely sampled functional data. *Journal of the American Statistical Association*, 98(462):397–408, 2003. doi: 10.1198/016214503000189.

Jiang, J., Lin, H., Peng, H., Fan, G.-Z., and Li, Y. Cluster analysis with regression of non-Gaussian functional data on covariates. *Canadian Journal of Statistics*, 50(1):221–240, 2022. doi: 10.1002/cjs.11680.

Kushnir, D., Jalali, S., and Saniee, I. Towards clustering high-dimensional gaussian mixture clouds in linear running time. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1379–1387. PMLR, 16–18 Apr 2019.

Li, T. and Ma, J. Dirichlet process mixture of Gaussian process functional regressions and its variational EM algorithm. *Pattern Recognition*, 134:109129, 2023. doi: 10.1016/j.patcog.2022.109129.

Maitra, R. and Melnykov, V. Simulating data to study performance of finite mixture modeling and clustering algorithms. *Journal of Computational and Graphical Statistics*, 19(2):354–376, January 2010. doi: 10.1198/jcgs.2009.08054.

Meng, Y., Liang, J., Cao, F., and He, Y. A new distance with derivative information for functional k-means clustering algorithm. *Information Sciences*, 463-464:166–185, 2018. doi: 10.1016/j.ins.2018.06.035.

Moitra, A. and Valiant, G. Settling the polynomial learnability of mixtures of Gaussians. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 93–102, 2010. doi: 10.1109/FOCS.2010.15.

Rasmussen, C. and Ghahramani, Z. Infinite mixtures of Gaussian process experts. In Dietterich, T., Becker, S., and Ghahramani, Z. (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

Rivera-García, D., García-Escudero, L. A., Mayo-Iscar, A., and Ortega, J. Robust clustering for functional data based on trimming and constraints. *Advances in Data Analysis and Classification*, 13(1):201–225, 2019. doi: 10.1007/s11634-018-0312-7.

Ross, J. and Dy, J. Nonparametric mixture of Gaussian processes with constraints. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1346–1354, 2013.

Seitz, S. Mixtures of Gaussian Processes for regression under multiple prior distributions. *arXiv*, 2021. doi: 10.48550/ARXIV.2104.09185.

Shi, J. and Wang, B. Curve prediction and clustering with mixtures of Gaussian process functional regression models. *Statistics and Computing*, 18(3):267–283, 2008. doi: 10.1007/s11222-008-9055-1.

Shi, J. Q., Murray-Smith, R., and Titterington, D. M. Hierarchical Gaussian process mixtures for regression. *Statistics and Computing*, 15(1):31–41, 2005. doi: 10.1007/s11222-005-4787-7.

Teicher, H. Identifiability of mixtures. *The annals of Mathematical statistics*, 32(1):244–248, 1961.

Teicher, H. Identifiability of finite mixtures. *The annals of Mathematical statistics*, 34(4):1265–1269, 1963.

Wu, D. and Ma, J. A two-layer mixture model of Gaussian process functional regressions and its MCMC EM algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10):4894–4904, 2018. doi: 10.1109/TNNLS.2017.2782711.

Yakowitz, S. J. and Spragins, J. D. On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 39(1):209 – 214, 1968. doi: 10.1214/aoms/1177698520.

Yellamraju, T. and Boutin, M. Clusterability and clustering of images and other "real" high-dimensional data. *IEEE Transactions on Image Processing*, 27(4):1927–1938, 2018. doi: 10.1109/TIP.2017.2789327.

Zhang, M. Weighted clustering ensemble: A review. *Pattern Recognition*, 124:108428, 2022. doi: 10.1016/j.patcog.2021.108428.

Zhang, M. and Parnell, A. Review of clustering methods for functional data. *ACM Trans. Knowl. Discov. Data*, 2023. doi: 10.1145/3581789.

A. Proof of Theorem 3.1

The function domain $\mathcal{T}$ is a compact interval, and hence if $\Sigma(s,t)$ is a Hilbert-Schmidt kernel, then the integral operator $\mathbb{K}$ will be compact. We have

$$\begin{aligned}
&\int_{\mathcal{T}}\int_{\mathcal{T}}|\Sigma(s,t)|^2 dsdt\\
&=\int_{\mathcal{T}}\int_{\mathcal{T}}\Big|\sum_{k=1}^{K}\pi_k\mathrm{E}[(X_k(t)-\mu(t))(X_k(s)-\mu(s))]\Big|^2 dsdt\\
&\le\int_{\mathcal{T}}\int_{\mathcal{T}}\sum_{k=1}^{K}\pi_k^2\Big|\mathrm{E}[(X_k(t)-\mu(t))(X_k(s)-\mu(s))]\Big|^2 dsdt \qquad (6)\\
&\le\int_{\mathcal{T}}\int_{\mathcal{T}}\sum_{k=1}^{K}\pi_k^2\mathrm{E}[(X_k(t)-\mu(t))^2]\mathrm{E}[(X_k(s)-\mu(s))^2]dsdt \qquad (7)\\
&=\sum_{k=1}^{K}\pi_k^2\Big(\int_{\mathcal{T}}\mathrm{E}[(X_k(t)-\mu(t))^2]dt\Big)^2\\
&=\sum_{k=1}^{K}\pi_k^2\Big(\int_{\mathcal{T}}\mathrm{E}[(X_k(t)-\mu_k(t)+\mu_k(t)-\mu(t))^2]dt\Big)^2\\
&<\infty. \qquad (8)
\end{aligned}$$

Here, we have applied Jensen's inequality for (6) and Cauchy-Schwarz inequality for (7). The random functions $\{X_k\}$ are Gaussian, and hence the inequality (8) is valid, from which we conclude that $\Sigma(s,t)$ is a Hilbert-Schmidt kernel and that $\mathbb{K}$ is compact.

Next, we show that $\langle\mathbb{K}x,x\rangle\ge 0$ for every $x\in\mathcal{H}(\mathcal{T},\mathbb{R})$:

$$\begin{aligned}
&\langle\mathbb{K}x,x\rangle\\
&=\int_{\mathcal{T}}(\mathbb{K}x)(t)x(t)dt\\
&=\int_{\mathcal{T}}\Big(\int_{\mathcal{T}}\Sigma(s,t)x(s)ds\Big)x(t)dt\\
&=\int_{\mathcal{T}}\int_{\mathcal{T}}\sum_{k=1}^{K}\pi_k\mathrm{E}[(X_k(t)-\mu(t))(X_k(s)-\mu(s))]x(s)x(t)dsdt\\
&=\sum_{k=1}^{K}\pi_k\mathrm{E}\Big[\int_{\mathcal{T}}\int_{\mathcal{T}}[X_k(t)-\mu(t)][X_k(s)-\mu(s)]x(s)x(t)dsdt\Big] \qquad (9)\\
&=\sum_{k=1}^{K}\pi_k\mathrm{E}\Big[(\int_{\mathcal{T}}[X_k(t)-\mu(t)]x(t)dt)^2\Big]\\
&\ge 0,
\end{aligned}$$

where we have applied Fubini's theorem for (9) to swap the order of expectation and integration. Therefore, the operator $\mathbb{K}$ is positive.

The proof for $\mathbb{K}$ being self-adjoint is straightforward: for every $y\in\mathcal{H}(\mathcal{T},\mathbb{R})$,

$$\begin{aligned}
\langle\mathbb{K}x,y\rangle&=\int_{\mathcal{T}}(\mathbb{K}x)(t)y(t)dt\\
&=\int_{\mathcal{T}}\Big(\int_{\mathcal{T}}\Sigma(s,t)x(s)ds\Big)y(t)dt\\
&=\int_{\mathcal{T}}x(s)\Big(\int_{\mathcal{T}}\Sigma(s,t)y(t)dt\Big)ds\\
&=\langle x,\mathbb{K}y\rangle.
\end{aligned}$$

B. Motivation for the Ensemble Approach

Based on the relation presented in Equation (3), it is evident that fitting a univariate GMM to the projection coefficients $\{\alpha_{iv}\}_{i=1}^{n}$ would enable us to infer each cluster label z_i from its posterior distribution. However, the quality of the inferred cluster labels $\{z_i\}_{i=1}^{n}$ is highly influenced by the degree of overlap among the component univariate Gaussian distributions. For example, a mixture of two univariate Gaussian distributions with equal standard deviations is bimodal only if their means differ by at least twice the common standard deviation. Therefore, if we want to correctly identify the hidden cluster labels from the mixture modeling of only one coefficient set $\{\alpha_{iv}\}_{i=1}^{n}$, the K component univariate Gaussian distributions have to be adequately far apart from each other and have low overlapping degree among them. Apparently, it is a formidable computational and statistical challenge to determine the function β_v onto which the projections of the GP components are well separated. Therefore, rather than finding one perfect projection function, we utilize multiple imperfect projection functions and employ an ensemble clustering method to aggregate the different univariate GMMs.

Note that, for any sample function x_i, while two fPC scores a_{ikv} and a_{ikr} are always independent, the projection coefficients α_{ikv} and α_{ikr} are not independent:

$$\begin{aligned}
\mathrm{cov}(\alpha_{iv},\alpha_{ir}|z_i=k)&=\mathrm{E}[\sum_{l=1}^{\infty}\sum_{q=1}^{\infty}a_{ikl}\langle b_{kl},\beta_v\rangle a_{ikq}\langle b_{kq},\beta_r\rangle]\\
&=\sum_{l=1}^{\infty}\lambda_{kl}\langle b_{kl},\beta_v\rangle\langle b_{kl},\beta_r\rangle.
\end{aligned}$$

In other words, the two base clusterings obtained from the two sets of projection coefficients $\{\alpha_{iv}\}_{i=1}^{n}$ and $\{\alpha_{ir}\}_{i=1}^{n}$ are dependent. This is a blessing rather than a curse. Otherwise, if the datasets $\{\alpha_{iv}\}_{i=1}^{n}$ for $v=1,2,\ldots$ are completely independent, then there is no "strength to be borrowed" across the different projections.

C. Proof of Theorem 4.1

We re-arrange the terms in Equation (4):

$$\begin{aligned}
&\|\hat{X}_k - \mathcal{P}_{H_l}(X_k)\|^2 \\
&= \|\mu_k - \mu_l\|^2 + \sum_{v=1}^{V_k} a_{kv}^2 + 2\sum_{v=1}^{V_k} a_{kv}\langle \mu_k - \mu_l, b_{kv}\rangle \\
&- \sum_{v=1}^{V_l}\langle \hat{X}_k - \mu_l, b_{lv}\rangle^2 + \sum_{v=1}^{V_l}\langle R_k, b_{lv}\rangle^2. \qquad (10)
\end{aligned}$$

Building on the orthonormality of the eigen-functions $\{b_{lv}\}_{v=1}^{\infty}$, we have

$$\begin{aligned}
&\|\hat{X}_k - \mathcal{P}_{H_l}(X_k)\|^2 \\
&= \|\hat{X}_k - \mu_l - \sum_{v=1}^{V_l} a_{lv}^k b_{lv}\|^2 \\
&= \|\hat{X}_k - \mu_l\|^2 + \|\sum_{v=1}^{V_l} a_{lv}^k b_{lv}\|^2 - 2\langle \hat{X}_k - \mu_l, \sum_{v=1}^{V_l} a_{lv}^k b_{lv}\rangle \\
&= \|\hat{X}_k - \mu_l\|^2 + \sum_{v=l}^{V_l}[(a_{lv}^k)^2 - 2a_{lv}^k\langle \hat{X}_k - \mu_l, b_{lv}\rangle].
\end{aligned}$$

For the first term, we have

$$\begin{aligned}
&\|\hat{X}_k - \mu_l\|^2 \\
&= \|\mu_k - \mu_l + \sum_{v=1}^{V_k} a_{kv} b_{kv}\|^2 \\
&= \|\mu_k - \mu_l\|^2 + \|\sum_{v=1}^{V_k} a_{kv} b_{kv}\|^2 + 2\sum_{v=1}^{V_k} a_{kv}\langle \mu_k - \mu_l, b_{kv}\rangle \\
&= \|\mu_k - \mu_l\|^2 + \sum_{v=1}^{V_k} a_{kv}^2 + 2\sum_{v=1}^{V_k} a_{kv}\langle \mu_k - \mu_l, b_{kv}\rangle.
\end{aligned}$$

For the second term, with the decomposition $a_{lv}^k = \langle X_k - \mu_l, b_{lv}\rangle = \langle \hat{X}_k - \mu_l, b_{lv}\rangle + \langle R_k, b_{lv}\rangle$, we have

$$\begin{aligned}
&\sum_{v=l}^{V_l}[(a_{lv}^k)^2 - 2a_{lv}^k\langle \hat{X}_k - \mu_l, b_{lv}\rangle] \\
&= \sum_{v=l}^{V_l}[\Big(\langle \hat{X}_k - \mu_l, b_{lv}\rangle + \langle R_k, b_{lv}\rangle\Big)^2 \\
&\quad - 2\left(\langle \hat{X}_k - \mu_l, b_{lv}\rangle + \langle R_k, b_{lv}\rangle\right)\langle \hat{X}_k - \mu_l, b_{lv}\rangle] \\
&= -\sum_{v=l}^{V_l}\langle \hat{X}_k - \mu_l, b_{lv}\rangle^2 + \sum_{v=l}^{V_l}\langle R_k, b_{lv}\rangle^2.
\end{aligned}$$

Piecing together the above equations completes the proof of Equation (10).

Now we investigate the residual term in Equation (10):

$$\begin{aligned}
\sum_{v=1}^{V_l}\langle R_k, b_{lv}\rangle^2 &= \sum_{v=1}^{V_l}\left(\sum_{r=V_k+1}^{\infty} a_{kr}\langle b_{kr}, b_{lv}\rangle\right)^2 \\
&= \sum_{v=1}^{V_l}\sum_{r=V_k+1}^{\infty}\sum_{s=V_k+1}^{\infty} a_{kr}a_{ks}\langle b_{kr}, b_{lv}\rangle\langle b_{ks}, b_{lv}\rangle.
\end{aligned}$$

Taking expectation w.r.t. the distribution of X_k and utilizing the uncorrelatedness of a_{kr} and a_{ks}, it follows that

$$\begin{aligned}
&\sum_{v=1}^{V_l}\mathrm{E}[\langle R_k, b_{lv}\rangle^2] \\
&= \sum_{v=1}^{V_l}\sum_{r=V_k+1}^{\infty}\sum_{s=V_k+1}^{\infty}\mathrm{E}[a_{kr}a_{ks}]\langle b_{kr}, b_{lv}\rangle\langle b_{ks}, b_{lv}\rangle \\
&= \sum_{v=1}^{V_l}\sum_{r=V_k+1}^{\infty}\lambda_{kr}\langle b_{kr}, b_{lv}\rangle^2 \\
&\leq \sum_{v=1}^{V_l}\sum_{r=V_k+1}^{\infty}\lambda_{kr}\langle b_{kr}, b_{kr}\rangle\langle b_{lv}, b_{lv}\rangle \\
&= V_l\sum_{r=V_k+1}^{\infty}\lambda_{kr},
\end{aligned}$$

which converges to 0 by assumption, implying that $\sum_{v=1}^{V_l}\langle R_k, b_{lv}\rangle^2$ converges to 0 in probability.

D. Proof of Theorem 4.4

By Chebyshev's inequality, we have that for any $\kappa > 0$,

$$\Pr\left(|\frac{1}{m}\sum_{i=1}^{m}\mathbb{x}_i^r - \mathrm{E}\left[\mathbb{X}^r\right]|^2 \geq \kappa^2\mathrm{var}(\frac{1}{m}\sum_{i=1}^{m}\mathbb{x}_i^r)\right) \leq \kappa^{-2}.$$

Let $\delta = \kappa^{-2}$. Then with probability at least $1-\delta$,

$$|\frac{1}{m}\sum_{i=1}^{m}\mathbb{x}_i^r - \mathrm{E}\left[\mathbb{X}^r\right]|^2 \leq \frac{1}{\delta}\mathrm{var}(\frac{1}{m}\sum_{i=1}^{m}\mathbb{x}_i^r).$$

We further give a crude upper bound on the right term:

$$\frac{1}{\delta}\mathrm{var}(\frac{1}{m}\sum_{i=1}^{m}\mathbb{x}_i^r) = \frac{1}{m\delta}\mathrm{var}(\mathbb{X}^r) \leq \frac{1}{m\delta}\mathrm{E}[\mathbb{X}^{2r}].$$

Recall that the univariate GMM $\sum_{k=1}^{K}\pi_k\mathcal{N}(u_k, \sigma_k^2)$ is in isotropic position: $\mathrm{E}[\mathbb{X}^2] = 1$, and that the mixing weights are all bounded by ϵ. Therefore, we have

$$\begin{aligned}
\mathrm{E}[\mathbb{X}^2] &= \sum_{k=1}^{K}\pi_k\mathrm{E}\left[\mathbb{Y}^2|\mathbb{Y}\sim\mathcal{N}(u_k, \sigma_k^2)\right] \\
&\geq \epsilon\sum_{k=1}^{K}\mathrm{E}\left[\mathbb{Y}^2|\mathbb{Y}\sim\mathcal{N}(u_k, \sigma_k^2)\right],
\end{aligned}$$

from which we obtain $\mathrm{E}\left[\mathbb{Y}^2|\mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)\right] \le \epsilon^{-1}$. Given $\mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)$, it follows that

$$u_k^2 = (\mathrm{E}[\mathbb{Y}])^2 \le \mathrm{E}[\mathbb{Y}^2] \le \epsilon^{-1},$$

and

$$\sigma_k^2 = \mathrm{E}[\mathbb{Y}^2] - (\mathrm{E}[\mathbb{Y}])^2 \le \mathrm{E}[\mathbb{Y}^2] \le \epsilon^{-1}.$$

Finally, we have that

$$\mathrm{E}\left[\mathbb{X}^{2r}\right] = \sum_{k=1}^{K} \pi_k \mathrm{E}\left[\mathbb{Y}^{2r}|\mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)\right]$$

and that

$$\begin{aligned}
&\mathrm{E}\left[\mathbb{Y}^{2r}|\mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)\right] \\
&= \mathrm{E}\left[(u_k + \sigma_k \mathbb{Z})^{2r}|\mathbb{Z} \sim \mathcal{N}(0,1)\right] \\
&= \sum_{j=0}^{r} \binom{2r}{2j} u_k^{2j} \sigma_k^{2r-2j} \mathrm{E}[\mathbb{Z}^{2r-2j}] \\
&\le \Big[\sum_{j=0}^{r} \binom{2r}{2j} (2r-2j-1)!!\Big] \epsilon^{-r},
\end{aligned}$$

from which the theorem follows.

E. Proof of Theorem 4.5

A brute force approach for parameter estimation is to (1) evaluate the moment $\mathrm{E}[\mathbb{X}^r]$ over a uniform grid in the parameter space; (2) compare the sample moment $\frac{1}{m}\sum_{i=1}^{m} \mathbb{x}_i^r$ with the moments $\mathrm{E}[\mathbb{X}^r]$ evaluated at all the grid points. The final parameter estimate is the grid point at which the moments of order from 1 to R all well match the sample moments. We let the identical width of the grid cell be denoted by γ, and hence every parameter estimate is a multiple of γ. Let $(\hat\pi_1, \hat u_1, \hat\sigma_1^2, \dots, \hat\pi_K, \hat u_K, \hat\sigma_K^2)$ denote the optimal grid point w.r.t. the agreement with the sample moments and the constraint that $\sum_{k=1}^{K} \hat\pi_k = 1$. Then we have $|\pi_k - \hat\pi_k| \le \gamma$, $|u_k - \hat u_k| \le \gamma$ and $|\sigma_k^2 - \hat\sigma_k^2| \le \gamma$, for any $1 \le k \le K$.

From the problem definition, we have $\min\{\pi_k : k = 1, \dots, K\} \ge \epsilon$. If $\epsilon \le 2\gamma$, then $\min\{\hat\pi_k : k = 1, \dots, K\} \ge \gamma \ge \epsilon/2$. If $\epsilon > 2\gamma$, then $|\pi_k - \hat\pi_k| \le \gamma < \epsilon/2$ and therefore $\hat\pi_k \ge \pi_k - \gamma > \epsilon - \epsilon/2 = \epsilon/2$. To conclude, we always have $\min\{\hat\pi_k : k = 1, \dots, K\} \ge \epsilon/2$.

Given $|u_k - u_j| + |\sigma_k^2 - \sigma_j^2| \ge \epsilon$, we might let $|u_k - u_j| = 0$ and hence $|\sigma_k^2 - \sigma_j^2| \ge \epsilon$. With the constraint that $\sum_{k=1}^{K} \hat\pi_k = 1$, the difference $|\hat\sigma_k^2 - \hat\sigma_j^2|$ is a non-zero multiple of γ. Moreover, we can prove that $|\sigma_k^2 - \hat\sigma_k^2| + |\sigma_j^2 - \hat\sigma_j^2| \le \gamma$. Therefore, if $\epsilon \le 2\gamma$, then $|\hat\sigma_k^2 - \hat\sigma_j^2| \ge \gamma \ge \epsilon/2$. If $\epsilon > 2\gamma$, then $|\hat\sigma_k^2 - \hat\sigma_j^2| \ge |\sigma_k^2 - \sigma_j^2| - (|\sigma_k^2 - \hat\sigma_k^2| + |\sigma_j^2 - \hat\sigma_j^2|) \ge \epsilon - \gamma > \epsilon/2$. Therefore, we always have $|\hat\sigma_k^2 - \hat\sigma_j^2| \ge \epsilon/2$. By analogy, if $|u_k - u_j| + |\sigma_k^2 - \sigma_j^2| \ge \epsilon$ and $|\sigma_k^2 - \sigma_j^2| = 0$, we can prove that $|\hat u_k - \hat u_j| \ge \epsilon/2$. For the general case, we can write $|u_k - u_j| = \epsilon_1$, $|\sigma_k^2 - \sigma_j^2| = \epsilon_2$ and $\epsilon_1 + \epsilon_2 \ge \epsilon$. We then independently prove that $|\hat u_k - \hat u_j| \ge \epsilon_1/2$ and $|\hat\sigma_k^2 - \hat\sigma_j^2| \ge \epsilon_2/2$; and therefore we have $|\hat u_k - \hat u_j| + |\hat\sigma_k^2 - \hat\sigma_j^2| \ge \epsilon/2$, for any $1 \le j \ne k \le K$.

We now evaluate the moment difference between the true and estimated GMMs. We have $\mathrm{var}(\mathbb{X}) = \mathrm{E}[\mathbb{X}^2] = \sum_{k=1}^{K} \pi_k \mathrm{E}\left[\mathbb{Y}^2|\mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)\right]$, and therefore

$$\begin{aligned}
1 &\ge \sum_{k=1}^{K} \pi_k \mathrm{E}\left[\mathbb{Y}^2|\mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)\right] \\
&\ge \epsilon \sum_{k=1}^{K} \mathrm{E}\left[\mathbb{Y}^2|\mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)\right].
\end{aligned}$$

Again, given $\mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)$, we have

$$u_k^2 = (\mathrm{E}[\mathbb{Y}])^2 \le \mathrm{E}[\mathbb{Y}^2] \le \epsilon^{-1},$$

and

$$\sigma_k^2 = \mathrm{E}[\mathbb{Y}^2] - (\mathrm{E}[\mathbb{Y}])^2 \le \mathrm{E}[\mathbb{Y}^2] \le \epsilon^{-1}.$$

If r is even, we write $r = 2v$ and

$$\begin{aligned}
\mathrm{E}\left[\mathbb{Y}^r|\mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)\right] &= \mathrm{E}\left[(u_k + \sigma_k \mathbb{Z})^r|\mathbb{Z} \sim \mathcal{N}(0,1)\right] \\
&= \sum_{j=0}^{v} \binom{2v}{2j} u_k^{2j} \sigma_k^{2v-2j} \mathrm{E}[\mathbb{Z}^{2v-2j}] \\
&\le O(\epsilon^{-v}).
\end{aligned} \tag{11}$$

Then the difference in the moments is

$$\begin{aligned}
&\mathrm{E}\left[\hat{\mathbb{Y}}^r - \mathbb{Y}^r|\hat{\mathbb{Y}} \sim \mathcal{N}(\hat u_k, \hat\sigma_k^2), \mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)\right] \\
&= \sum_{j=0}^{v} \binom{2v}{2j} (2v-2j-1)!! [\hat u_k^{2j} \hat\sigma_k^{2v-2j} - u_k^{2j} \sigma_k^{2v-2j}].
\end{aligned}$$

We might let $u_k > 0$. With the difference between the true and estimated parameters being bounded by γ: $|u_k - \hat u_k| \le \gamma$ and $|\sigma_k^2 - \hat\sigma_k^2| \le \gamma$, we have the following inequality:

$$\begin{aligned}
&|\hat u_k^{2j} \hat\sigma_k^{2v-2j} - u_k^{2j} \sigma_k^{2v-2j}| \\
&\le (u_k + \gamma)^{2j} (\sigma_k^2 + \gamma)^{v-j} - u_k^{2j} \sigma_k^{2v-2j} \\
&= u_k^{2j} \sigma_k^{2v-2j} \big[(1 + \frac{\gamma^2 + 2\gamma u_k}{u_k^2})^j (1 + \frac{\gamma}{\sigma_k^2})^{v-j} - 1 \big] \\
&\le \epsilon^{-v} (2^v - 1) \max\{\frac{\gamma^2 + 2\gamma u_k}{u_k^2}, \frac{\gamma}{\sigma_k^2}\},
\end{aligned}$$

where we have utilized the inequality that, when $c < 1$, $(1+c)^v - 1 \le (2^v - 1)c$. Therefore, we have $|\mathrm{E}[\hat{\mathbb{Y}}^r] - \mathrm{E}[\mathbb{Y}^r]| \le O(\epsilon^{-v}\gamma)$.

If we were given the true mixing weights, the difference in the moments $|\mathrm{E}[\hat{\mathbb{X}}^r] - \mathrm{E}[\mathbb{X}^r]|$ would be bounded by K times of $O(\epsilon^{-v}\gamma)$. In the brute force approach, we have

$|\pi_k - \hat{\pi}_k| \leq \gamma$, and Equation (11) gives the upper bound on every component moment; therefore, the rounding of the weight will contribute an extra of at most $O(\epsilon^{-v}\gamma)$. Adding the two bounds together, we get that each moment $\mathrm{E}[\hat{\mathbb{X}}^r]$ can be off from the true one $\mathrm{E}[\mathbb{X}^r]$ by at most $K \times O(\epsilon^{-v}\gamma) + O(\epsilon^{-v}\gamma)$. Therefore, letting $\gamma = c_K \xi^{v+1} = c_K \xi^{\frac{r}{2}+1}$, where the constant c_K depends on K, then the moment $\mathrm{E}[\hat{\mathbb{X}}^{2v}]$ will be within ξ of $\mathrm{E}[\mathbb{X}^{2v}]$.

If r is odd, then we write $r = 2v + 1$ and

$$\begin{aligned}
&\mathrm{E}\left[\mathbb{Y}^r | \mathbb{Y} \sim \mathcal{N}(u_k, \sigma_k^2)\right] \\
&= \sum_{j=0}^{v} \binom{2v+1}{2j+1} (2v-2j-1)!! u_k^{2j+1} \sigma_k^{2v-2j} \\
&\leq O(\epsilon^{-(v+\frac{1}{2})}).
\end{aligned}$$

By analogy, we have

$$\begin{aligned}
&|\hat{u}_k^{2j+1} \hat{\sigma}_k^{2v-2j} - u_k^{2j+1} \sigma_k^{2v-2j}| \\
&\leq (u_k + \gamma)^{2j+1} (\sigma_k^2 + \gamma)^{v-j} - u_k^{2j+1} \sigma_k^{2v-2j} \\
&= u_k^{2j+1} \sigma_k^{2v-2j} \left[(1 + \frac{\gamma}{u_k})^{2j+1} (1 + \frac{\gamma}{\sigma_k^2})^{v-j} - 1\right] \\
&\leq \epsilon^{-(v+\frac{1}{2})} (2^{v+j+1} - 1) \max\{\frac{\gamma}{u_k}, \frac{\gamma}{\sigma_k^2}\},
\end{aligned}$$

Therefore, the error bound for $|\mathrm{E}[\hat{\mathbb{X}}^{2v+1}] - \mathrm{E}[\mathbb{X}^{2v+1}]|$ is

$$K \times O\epsilon^{-(v+\frac{1}{2})}\gamma) + O(\epsilon^{-(v+\frac{1}{2})}\gamma).$$

Letting $\gamma = c_K \xi^{v+\frac{3}{2}} = c_K \xi^{\frac{r}{2}+1}$, then the moment $\mathrm{E}[\hat{\mathbb{X}}^{2v+1}]$ will be within ξ of $\mathrm{E}[\mathbb{X}^{2v+1}]$.

F. Proof of Theorem 4.6

Our proof is built on the sufficient condition given by (Atienza et al., 2006) for a finite mixture of distributions to be identifiable. In Lemma F.1, A^c denotes the accumulation set of $A \subset \mathbb{R}^d$, consisting of all points for which every neighborhood contains infinitely many distinct points of A.

Lemma F.1. *Let $\mathcal{F}$ be a family of distributions. Let M be a linear (one-to-one) mapping which transforms any $F \in \mathcal{F}$ into a real function M_F with domain $D_F \subset \mathbb{R}^d$. Let $D_F^* = \{\mathbb{x} \in D_F : M_F(\mathbb{x}) \neq 0\}$. Suppose that there exists a point $\mathbb{x}_0$ verifying*

$$\mathbb{x}_0 \in \left[\cap_{1 \leq k \leq K} D_{F_k}^*\right]^c,$$

for any finite collection of distributions $F_1, F_2, \ldots, F_K \in \mathcal{F}$. If the order

$$F_1 \prec F_2 \textit{ if and only if } \lim_{\mathbb{x} \to \mathbb{x}_0} \frac{M_{F_2}(\mathbb{x})}{M_{F_1}(\mathbb{x})} = 0$$

is a total ordering on $\mathcal{F}$, then any finite mixture of distributions of $\mathcal{F}$ is identifiable.

If we have $\sum_{k=1}^{K_1} \pi_k F_k = \sum_{k=1}^{K_2} \hat{\pi}_k \hat{F}_k$, where $F_1 \preceq \hat{F}_1$, $F_k \prec F_{k+1}$, $\hat{F}_k \prec \hat{F}_{k+1}$ and $K_1 \leq K_2$, then the proof for Lemma F.1 affirms that $\hat{\pi}_k = \pi_k$ and $\hat{F}_k = F_k$, for $k = 1, \ldots, K_1$, and that $\sum_{k=K_1+1}^{K_2} \hat{\pi}_k \hat{F}_k = 0$.

Let $\mathcal{F}$ be the family of univaritate Gaussian cumulative distribution functions: $\mathcal{F} = \{F_k = \mathcal{N}(u_k, \sigma_k^2) : 1 \leq k \leq K\}$. Let M be the map which transforms $F_k \in \mathcal{F}$ into its moment generating function, a real function. In fact, M is an integral transform and we can readily prove that it is a linear and one-to-one mapping.

For any $F_k \in \mathcal{F}$, we have

$$M_{F_k}(\mathbb{x}) = \exp\left(u_k \mathbb{x} + \frac{1}{2}\sigma_k^2 \mathbb{x}^2\right).$$

The domain of the moment generating function M_{F_k} is the real line: $D_{F_k}^* = D_{F_k} = (-\infty, +\infty)$, and therefore the accumulation set is $(D_{F_k}^*)^c = [-\infty, +\infty]$. We pick $\mathbb{x}_0 = +\infty$, and it follows that

$$\mathbb{x}_0 \in \left[\cap_{1 \leq k \leq K} D_{F_k}^*\right]^c.$$

For any $F_k, F_j \in \mathcal{F}$, we have

$$\frac{M_{F_j}(\mathbb{x})}{M_{F_k}(\mathbb{x})} = \exp\left([u_j - u_k]\mathbb{x} + \frac{1}{2}[\sigma_j^2 - \sigma_k^2]\mathbb{x}^2\right).$$

To have the property that $\lim_{\mathbb{x} \to \mathbb{x}_0} \frac{M_{F_j}(\mathbb{x})}{M_{F_k}(\mathbb{x})} = 0$, the exponent need approach to $-\infty$, or equivalently,

$$[\sigma_j^2 - \sigma_k^2 < 0] \text{ or } [\sigma_j^2 - \sigma_k^2 = 0 \text{ and } u_j - u_k < 0]. \quad (12)$$

The condition (12) naturally leads to our definition of the total order $\prec$; that is, we have $F_k \prec F_j$ if either $[\sigma_j^2 - \sigma_k^2 < 0]$ or $[\sigma_j^2 - \sigma_k^2 = 0$ and $u_j - u_k < 0]$.

Finally, we note that the condition (12) is equivalent to the condition that $|\sigma_j^2 - \sigma_k^2| + |u_j - u_k| \geq \epsilon$. The requirement on $\pi_k \geq \epsilon$ is intuitive, and the proof is complete.

G. Supplementary Materials for the Experiments

G.1. Details on Algorithm Configuration

For each algorithm, the argument for the number of clusters is set to the true cluster number in the dataset.

1. FEM (from R package funFEM): The other arguments are set to `model = "all"`, `crit = "bic"`, `init = "kmeans"`, `maxit = 50`, `eps = 1e-06`. This configuration enables the application of all 12 supported models ("DkBk", "DkB", "DBk", "DB", "AkjBk", "AkjB", "AkBk", "AkBk", "AjBk", "AjB", "ABk", "AB"). For each dataset, the clustering result is given by the model with the lowest BIC value.

2. HDD (from R package funHDDC): The other arguments are set to `model = c("AkjBkQkDk", "AkjBQkDk", "AkBkQkDk", "ABkQkDk", "AkBQkDk", "ABQkDk"), init="kmeans", threshold=0.1, criterion = "bic", itermax = 200`. For each dataset, the clustering result is given by the model with the lowest BIC value.

3. clu (from R package Funclustering): The other arguments are set to `nbInit = 20, thd = 0.05, increaseDimension = FALSE, hard = FALSE, fixedDimension = integer(0)`.

4. FC (from R package fdapace): The package supports two clustering methods "EMCluster" and "kCFC". For the "kCFC" method, the other arguments are set to `cmethod = "kCFC", optnsFPCA = NULL, optnsCS = NULL`, and for the "EMCluster" method, the other arguments are set to`cmethod = "EMCluster", optnsFPCA = NULL, optnsCS = NULL`. For each dataset, the clustering result is given by the method with the highest AMI or ARI score.

5. km (the `kmeans_align` function from R package fdasrvf): The other arguments are set to `seeds = NULL, centroid_type = "mean", alignment = FALSE`.

6. ADP (from R package FADPclust): The package supports two clustering methods "FADP1" and "FADP2". For the "FADP1" method, the other arguments are set to `method = "FADP1", proportion = NULL, f.cut = 0.15`, and for the "FADP2" method, the other arguments are set to `method = "FADP2", proportion = NULL, f.cut = 0.15`. For each dataset, the clustering result is given by the method with the highest AMI or ARI score.

For the GPmix algorithm, we need to specify the family of projection functions, the number of projection functions, and pertinent hyper-parameters associated with the chosen projection family. Our package offers six types of projection functions: eigen-functions from the fPC decomposition (fPC), random linear combinations of eigen-functions (rl-fPC), B-splines, Fourier basis, discrete wavelets, and Ornstein-Uhlenbeck (OU) random functions. A detailed explanation of each projection family is available in the package's documentation file. Table 5 gives the configuration of the GPmix algorithm for the ten real datasets and 12 simulation scenarios. For the wavelet families, namely {db10, haar, rbio1.3, rbio6.8, sym17, bior2.4}, we need to specify a lower resolution, which together with the number of projection functions determine the location and scale shifts of the mother wavelet. For B-splines, we need to specify the order. In Table 5, the optimal configuration was selected in a grid search strategy according to the ARI and AMI scores. In real applications, we rely on internal clustering validation indices such as the Silhouette validation index (SIL) or Davies-Bouldin score.

Table 5. Specification of the projection family, the number of projection functions, and if applicable, the hyper-parameter (HP) value for the selected projection family.

Data	Family	No.	HP	Sim.	Family	No.	HP
AH	db10	8	1	A	haar	64	1
BC	b-spline	2	1	B	rbio1.3	2	2
CBF	haar	14	1	C	bior6.8	32	1
DSR	OU	32	-	D	haar	32	1
ECG	haar	6	4	E	fPC	2	-
FF	bior2.4	64	8	F	b-spline	16	3
GuP	bior2.4	6	1	G	rl-fPC	16	-
Meat	bior2.4	10	8	H	fourier	2	-
SB	rbio6.8	6	2	I	haar	64	1
SYM	Fourier	16	-	J	b-spline	16	3
				K	rbio6.8	8	4
				L	rl-fPC	16	-

G.2. Details on the Real and Simulated Data

In Figures 2 and 3, we plot the functions for each dataset, colored by their cluster labels. For the 10 real datasets from the UEA & UCR Time Series Classification Repository, we included the sample curves from both the training and testing datasets. The configurations of the 12 simulation scenarios are explained below.

- **Scenario A** (Golovkine et al., 2022): We randomly simulate a set of 1000 curves. Each curve is evaluated at 101 equidistant points in the interval $[0, 1]$. These curves are generated from a mixture of 5 components, each having an equal mixing proportion of 0.2. The random functions are formulated as follows:

$$
\begin{aligned}
X_1(t) &= \mu_1(t) + a\phi_1(t) + b\phi_2(t) + c\phi_3(t),\\
X_2(t) &= \mu_1(t) + d\phi_1(t) + e\phi_2(t) + f\phi_3(t),\\
X_3(t) &= \mu_2(t) + a\phi_1(t) + b\phi_2(t) + c\phi_3(t),\\
X_4(t) &= \mu_2(t) + d\phi_1(t) + e\phi_2(t) + f\phi_3(t),\\
X_5(t) &= \mu_2(t) + d\phi_1(t) + e\phi_2(t) + f\phi_3(t) - 15t,
\end{aligned}
$$

where

$$\phi_k(t) = \sqrt{2}\sin\left((k-0.5)\pi t\right), \quad k = 1, 2, 3,$$

and

$$\mu_1(t) = \frac{20}{1+\exp(-t)}, \; \mu_2(t) = \frac{-25}{1+\exp(-t)}.$$

The coefficients are Gaussian variables: $a \sim \mathcal{N}(0, 16)$, $b \sim \mathcal{N}(0, 64/9)$, $c \sim \mathcal{N}(0, 16/9)$, $d \sim \mathcal{N}(0, 1)$, $e \sim \mathcal{N}(0, 4/9)$, and $f \sim \mathcal{N}(0, 1/9)$.

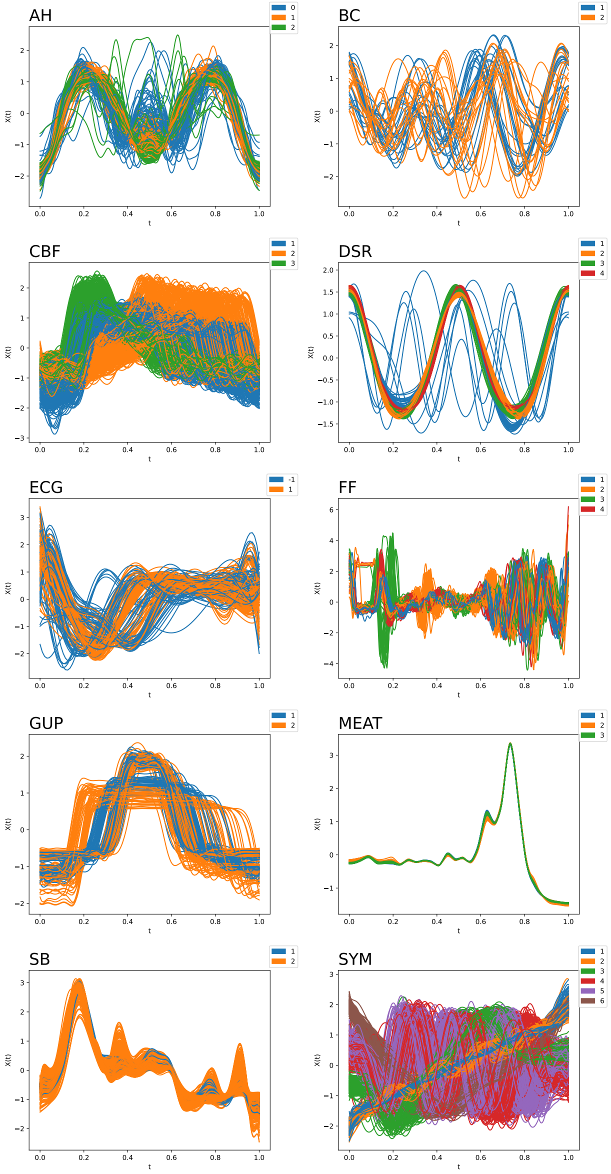

Figure 2. Plots of smoothed version of the 10 real datasets.

- **Scenario B** (Jacques & Preda, 2013): We randomly generate 200 curves from a population of two clusters, where the mixing proportions are 0.7 and 0.3. The curves are evaluated at 51 equidistant points in the interval $[1, 21]$. The random functions are formulated as:

$$X_1(t) = ah_1(t) + bh_2(t) + \epsilon(t),$$
$$X_2(t) = ah_1(t) + \epsilon(t),$$

where $h_1(t) = [6 - |t - 7|]_+$, $h_2(t) = [6 - |t - 15|]_+$, $a \sim \mathcal{N}(0, 1/12)$, $b \sim \mathcal{N}(0, 1/12)$, and $\epsilon(t)$ is a white noise such that $\text{var}(\epsilon(t)) = 1/12$.

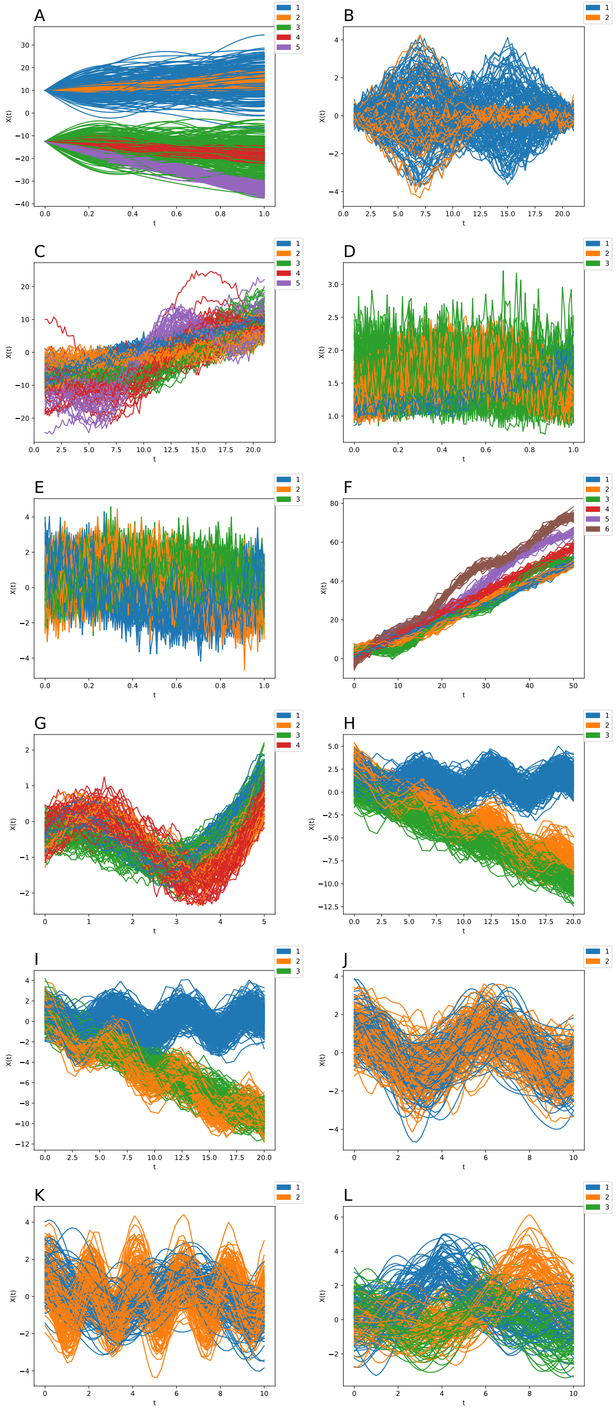

Figure 3. Plots of the 12 simulated datasets.

- **Scenario C** (Jacques & Preda, 2014): We generate 200 curves from a mixture of five random functions, each having an equal mixing proportion of 0.2. For $1 \leq k \leq 5$, the random function X_k takes the form:

$$X_k(t) = \frac{-21}{2} + t + kU_1 \cos\left(\frac{kt}{10}\right) + kU_2 \sin\left(k + \frac{t}{10}\right) + \epsilon(t),$$

where $U_1, U_2 \sim \mathcal{N}(1,1)$ and $\epsilon(t)$ is the unit-variance white noise. The curves are evaluated on 101 equidistant points in the interval $[1,21]$.

- **Scenario D** (Jiang et al., 2022): We generate 200 curves from a mixture of three random functions, each having an equal mixing proportion of $1/3$. For $1 \leq k \leq 3$, the random function X_k takes the form:

$$X_k(t) = \exp\left([g_k(t) + U_1 k + \epsilon_k(t)]/4\right),$$

where $g_1(t) = \exp(t) - 1$, $g_2(t) = \sin(\pi t)$, and $g_3(t) = -0.5t^2 + 0.5$. $\epsilon_k(t)$ is a GP with zero mean and covariance function $\text{cov}(\epsilon_k(t), \epsilon_k(s)) = \sigma_k^2 \rho_k^{|t-s|}$, where $(\sigma_1^2, \rho_1) = (0.1, 0.3)$, $(\sigma_2^2, \rho_2) = (0.15, 0.35)$, and $(\sigma_3^2, \rho_3) = (0.2, 0.4)$. The random value U_1 is generated from the $(0,1)$ uniform distribution.

- **Scenario E** (Meng et al., 2018): We generate 200 curves from a mixture of three random functions, each having an equal mixing proportion of $1/3$. The random functions are formulated as:

$$\begin{aligned}
X_1(t) &= \cos(1.5\pi t) + \epsilon(t), \\
X_2(t) &= \sin(1.5\pi t) + \epsilon(t), \\
X_3(t) &= \sin(\pi t) + \epsilon(t),
\end{aligned}$$

where $\epsilon(t)$ is a white noise of zero mean and unit variance. The curves are evaluated on 101 equidistant points in the interval $[0,1]$.

- **Scenario F**: With a Gaussian kernel $\Sigma(t,s) = \exp(-\frac{1}{2}(t-s)^2)$, we define the following models:

$$\begin{aligned}
C_1(p,t) \sim t &+ GP\left(p\cos(\frac{tp}{10}),\ \Sigma(t,t)\right) \\
&+ GP\left(p\sin(p + \frac{t}{10}),\ \Sigma(t,t)\right) + \epsilon(t), \\
C_2(p,t) \sim t &+ GP\left(p\sin(\frac{tp}{10}), \Sigma(t,t)\right) \\
&+ GP\left(p\cos(p + \frac{t}{10}),\ \Sigma(5t,5t)\right) \\
&+ GP\left(p\left(\sqrt{\frac{t}{10}} + \frac{t}{10}\right),\ \Sigma(8t,8t)\right) + \epsilon(t),
\end{aligned}$$

where $\epsilon(t)$ is a zero-mean white noise of variance $1/64$. We generate 200 curves from a mixture of 6 random functions, each having an equal mixing proportion of $1/6$: $X_1(t) = C_1(1,t)$, $X_2(t) = C_1(2,t)$, $X_3(t) = C_1(3,t)$, $X_4(t) = C_2(1,t)$, $X_5(t) = C_2(2,t)$, $X_6(t) = C_2(3,t)$. The curves are evaluated on 30 equidistant points in the interval $[0,50]$.

- **Scenario G** (Chen et al., 2014): We generate 200 curves from a GP mixture of four components, each having an equal mixing proportion of 1/4. The random functions are formulated as

$$\begin{aligned}
X_1(t) &\sim GP(\mu_1(t),\ \Sigma(t,t;1,0.1)), \\
X_2(t) &\sim GP(\mu_2(t),\ \Sigma(t,t;1,0.15)), \\
X_3(t) &\sim GP(\mu_3(t),\ \Sigma(t,t;1,0.20)), \\
X_4(t) &\sim GP(\mu_4(t),\ \Sigma(t,t;1,0.25)),
\end{aligned}$$

where the covariance function take the form

$$\Sigma(t,s;\beta,\sigma^2) = \sigma^2(2\beta)^{-1}\exp\left(-\beta|t-s|\right).$$

The mean functions are

$$\begin{aligned}
\mu_1(t) &= -\sin(t-1)\ln(t+0.5), \\
\mu_2(t) &= \cos(t)\ln(t+0.5), \\
\mu_3(t) &= -0.25 - 0.1\cos(0.5(t-1))t^{1.5}\sqrt{5t^{.5} + 0.5}, \\
\mu_4(t) &= 0.6\cos(t)\ln(t+0.5)\sqrt{(t+0.5)}.
\end{aligned}$$

Each curve is evaluated at 101 equidistant points in the interval $[0,5]$.

- **Scenario H**: We simulate 500 curves from a GP mixture of 3 components, where the mixing proportions are 0.5, 0.25 and 0.25. The random functions are formulated as:

$$\begin{aligned}
X_1(t) &\sim 1 + GP(\mu_1(t), \Sigma(t,t)) + \epsilon(t), \\
X_2(t) &\sim 2 + GP(\mu_2(t), \Sigma(t,t)) + \epsilon(t), \\
X_3(t) &\sim GP(\mu_3(t), \Sigma(t,t)) + \epsilon(t),
\end{aligned}$$

where $\Sigma(t,s) = \exp(-\frac{1}{2}(t-s)^2)$, $\mu_1(t) = \cos(t)$, $\mu_2(t) = \cos(t) - 0.5t$, $\mu_3(t) = -0.5t + 0.5$, and $\epsilon(t) \sim \mathcal{N}(0, 0.04)$ is a white noise. The curves are evaluated at 30 equidistant points in the interval $[0,20]$.

- **Scenario I**: The simulation configuration is the same as that of scenario H, except that there is no white noise in the component GP processes.

- **Scenario J**: To investigate the case where the component GPs have the same mean function but different covariance functions, we simulate 200 curves from a mixture of two components, where the mixing proportions are 0.4 and 0.6. The random functions are formulated as:

$$\begin{aligned}
X_1(t) &\sim GP(\mu(t), \Sigma_1(t,t)) \\
X_2(t) &\sim GP(\mu(t), \Sigma_2(t,t)),
\end{aligned}$$

where $\mu(t) = \cos(t)$, $\Sigma_1(t,s) = \exp(-\frac{1}{2}(t-s)^2)$, and $\Sigma_2(t,s) = (1 + \sqrt{3}|t-s|)\exp(-\sqrt{3}|t-s|)$. Each curve is evaluated at 50 equidistant points in the interval $[0,10]$.

- **Scenario K**: The simulation configuration is the same as that of scenario J, except that the two GPs now have different mean functions: $\mu_1(t) = \cos(t)$ and $\mu_2(t) = \cos(3t)$.
- **Scenario L**: We simulate 200 curves from a mixture of three components, where the mixing proportions are 0.4, 0.3 and 0.3. The components are GPs with different mean functions:

$$
\begin{aligned}
X_1(t) &\sim GP(\mu_1(t), \Sigma(t,t)),\\
X_2(t) &\sim GP(\mu_2(t), \Sigma(t,t)),\\
X_3(t) &\sim GP(\mu_3(t), \Sigma(t,t)),
\end{aligned}
$$

  where $\Sigma(t,s) = \exp(-\frac{1}{2}(t-s)^2)$, $\mu_1(t) = [3 - |t - 4|]_+$, $\mu_2(t) = [3 - |t - 8|]_+$, and $\mu_3(t) = \cos(t)$. Each curve is evaluated at 100 equidistant points in the interval $[0, 10]$.

G.3. Details on the Cluster Analysis of Two Datasets

Here, we provide an illustration of the clustering process using two datasets: one simulated data (Scenario F) and one real data (CBF). In Figure 4, we depict the projection functions for the simulated data (left) and the CBF data (right). The 16 projection functions in the left panel are orthonormalized B-splines, while the 14 projection functions in the right panel are wavelets from the Haar wavelet family.

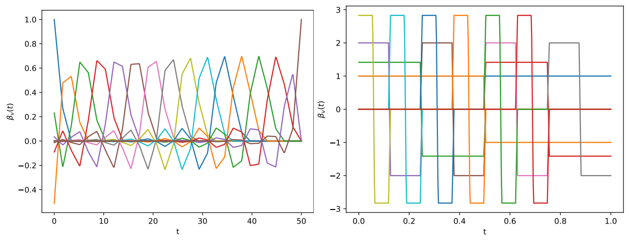

Figure 4. Projection functions for the simulated data (left) and the CBF data (right).

The projection process generates different sets of projection coefficients by projecting the curves onto the various projection functions. In Figure 5, we provide histogram plots of the different sets of projection coefficients for the simulated data, with the red curves representing Gaussian density functions from the estimated univariate GMM. Moving to Figure 6, we showcase the distributions of projection coefficients for the CBF data. Unlike the fPC decomposition, where the distribution of fPC scores tends to become more uni-modal, the histogram plots in Figure 5 and Figure 6 consistently exhibit different modes. In comparison to Figure 5, the density curves in Figure 6 exhibit a greater degree of overlapping, amplifying the complexity of the clustering problem. Nevertheless, our theoretical study asserts that, after $o(\log(p))$ random projections, a good direction will be found that yields good separation in 1-dimension. In Figure 6, the mixture components associated with the second projection function demonstrate clear separation, highlighting the effectiveness of the clustering approach.

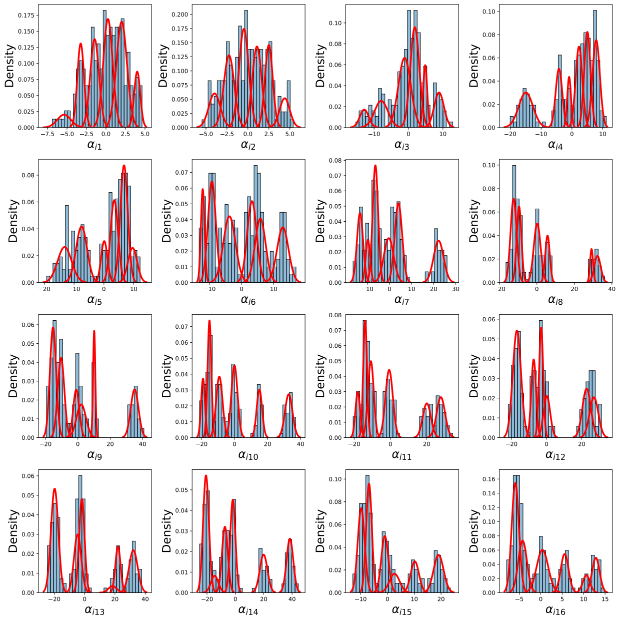

Figure 5. Histogram plots of the projection coefficients (for the simulated data), overlaid with the estimated density curves.

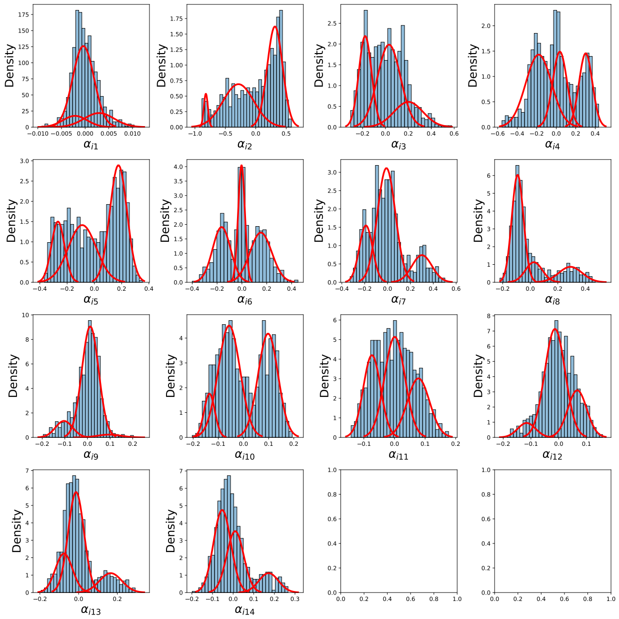

Figure 6. Histogram plots of the projection coefficients (for the CBF data), overlaid with the estimated density curves.

Following the learning of individual GMMs, we calculate the weight for each base clustering based on the degree of overlapping among its mixture components. Figure 7 plots the two sets of weights. For the CBF dataset, mixture components over the first projection function exhibits relatively higher overlapping, therefore receiving lowest weight. Similarly, for the simulated dataset, the mixture components over the tenth projection function demonstrate significantly lower overlap, resulting in the highest weight for the tenth base clustering. Figure 8 presents the two clustering results, with each curve color-coded according to its cluster label.

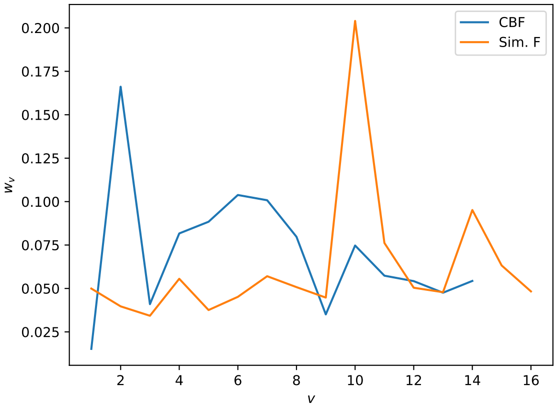

Figure 7. Base clustering weights, calculated according to the overlapping degree of mixture components.

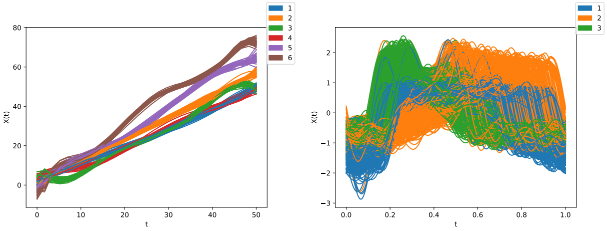

Figure 8. Final clustering results, left for the simulated data and right for the CBF data.