Functional Finite Mixture Modelling and Estimation

by

Alexander Byron Sharp

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Julien Jacques
Professeur des Universités de classe exceptionnelle,
Mathématiques Appliquées, Directeur du laboratoire ERIC,
Université Lumière Lyon 2

Supervisor(s): Ryan P. Browne
Associate Professor,
Dept. of Statistics and Actuarial Science,
University of Waterloo

Internal Member: Reza Ramezan
Continuing Lecturer,
Dept. of Statistics and Actuarial Science,
University of Waterloo

Internal Member: Greg Rice
Associate Professor,
Dept. of Statistics and Actuarial Science,
University of Waterloo

Internal-External Member: Hans De Sterck
Professor,
Dept. of Applied Mathematics,
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis consists of four manuscripts, which have all either been published or submitted for publication, and were written solely by Alexander Sharp under the supervision of Dr. Ryan P. Browne. The sole exception is Chapter 4, which includes some contributions from another graduate student. Details ensue.

- Chapter 3: Alex Sharp and Ryan P. Browne. Functional Data Clustering by Projection in Latent Generalized Hyperbolic Subspaces. *Advances in Data Analysis and Classification*, 15(3):735-757, Sept 2021. ISSN 1862-5355. doi: 10.1007/s11634-020-00432-5.

  Work on this manuscript was initiated during Alex Sharp's MMath degree at the University of Waterloo. Contributions during the PhD program include the simulation studies on parameter recovery, model selection, and competing algorithms, as well as the analysis of the Symbols dataset.

- Chapter 4: Alex Sharp, Glen Chalatov, and Ryan P. Browne. A Dual Subspace Parsimonious Mixture of Matrix Normal Distributions. *Advances in Data Analysis and Classification*, Nov 2022b. ISSN 1862-5355. doi: 10.1007/s11634-022-00526-2.

  Work on this manuscript was done in collaboration with Glen Chalatov, an MMath student under the supervision of Dr. Ryan P. Browne. His contributions include some work on the algorithm, initial model code and exploratory simulations.

- Chapter 5: Alex Sharp and Ryan P. Browne. A Joint Factor Analyzer and Functional Subspace Model for Clustering Multivariate Functional Data. *Statistics and Computing*, 32(5), August 2022a. ISSN 1573-1375. doi: 10.1007/s11222-022-10128-9.

- Chapter 6: Alex Sharp and Ryan P. Browne. Maximum Contribution to the Likelihood: Increasing Estimation Precision in the Stochastic Expectation-Maximization Algorithm. *Submitted to the Electronic Journal of Statistics, March 2023.*

R code and packages related to these manuscripts will be made available at https://github.com/ab2sharp.

Abstract

Functional data analysis is a branch of statistics that studies models for information represented by functions. Meanwhile, finite mixture models serve as a conerstone in the field of cluster analysis, offering a flexible probabilisitic framework for the representation of heterogeneous data. These models posit that the observed data are drawn from a mixture of several different probability distributions from the same family, where each is conventionally thought to represent a distinct group within the overall population. However, their representation in terms of densities makes their application to function-valued random variables, the foundation of functional data analysis, difficult. Herein, we utilize density surrogates derived from the Karhunen-Loeve expansion to circumvent this discrepancy and develop functional finite mixture models for the clustering of functional data. Models developed for real-valued and vector-valued functions of a single variable. Estimation of all models is done using the expectation-maximization algorithm, and copious amounts of simulations and data examples are provided to demonstrate the properties and performance of the methodologies. Additionally, we present a new estimation approach to be used in tandem with the stochastic expectation-maximization algorithm. This estimation method offers increased precision in estimation with respect to the algorithm chain length when compared to averaging the chain. Asymptotic properties of the estimator are derived, and simulation studies are given to demonstrate its performance.

Acknowledgements

I would like to thank all the people who made this thesis possible.

Dedication

This is dedicated to the people.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The present thesis constitutes a contribution to the literature of the relatively unexplored domain of functional data clustering with finite mixture models. The work is structured as a compilation of four distinct studies, each contributing either directly or indirectly to this field.

The initial three works primarily focus on the development of models for clustering functional data, with finite mixture models as the clustering mechanism of choice. Model development is facilitated by an interplay between the Karhunen-Loève expansion, a principal component analysis for stochastic processes, and finite mixture models, probabilistic models that represent the presence of subpopulations within an overall population. A symbiotic connection between these two concepts is the heart of the proposed modelling approaches, leading to a coherent framework for unsupervised learning with functional data.

Mixture models, as a class of probabilistic models, have emerged as a particularly effective tool in the field of clustering, where they serve to discern and represent homogeneous subpopulations within an overall population. Their ability to unveil underlying structure in complex data sets, coupled with their inherent specification flexibility, has spurred a wealth of literature dedicated to their development and application. However, their tendency to suffer from the curse of dimensionality, a phenomenon that exponentially expands the parameter space in higher dimensions, poses a significant drawback. To mitigate this issue, parsimonious parameter specifications are often invoked. By efficiently scaling back the number of model parameters, these specifications aim to alleviate the curse of dimensionality, thereby bolstering the feasibility and performance of mixture models in high-dimensional data contexts.

One such approach posits the existence of component-specific latent subspaces, which are presumed to be of substantially lower dimension. Intuitively, this specification presumes the observations specific to each mixture component aggregate around this low-dimensional subspace in a noisy fashion. This assumption facilitates a parsimonious parameter specification for each component distribution, while the modularity of the mixture density permits flexibility in the modelling of this space for each component. The proposed functional finite mixture models extend this notion of parsimony to the case of three-way data, and later space curves, thereby enhancing the feasibility of the resulting models in application.

Mixture models are frequently fitted using the Expectation-Maximization (EM) algorithm, a deterministic optimization procedure often touted for its property of monotonic convergence. The EM algorithm operates through the iteration of two steps: the E-step, which computes the expected value of the complete-data log-likelihood, and the M-step, which maximizes this expected value with respect to the parameters. However, the E-step can sometimes prove intractable, particularly in complex or high-dimensional settings. In such instances, a stochastic expectation-maximization algorithm may be employed as an alternative. Unlike its deterministic counterpart, stochastic EM does not converge to a limit point, but rather, a stationary distribution. Although this necessitates the inclusion of an additional estimation step in the algorithm, stochasticity also provides the benefit allowing the algorithm to slip free of local maxima, enabling superior navigation of the parameter space.

The final contribution of this thesis proposes a new way to implement the estimation step of a stochastic EM algorithm. The benefit of this approach is that it increases estimator precision in terms of the algorithm chain length. As a result, fewer iterations of the algorithm are needed to obtain precision thresholds, leading to a more computationally efficient algorithm. Additionally, enhancing the precision of the estimation process increases the robustness and reliability of the fitted model, thereby contributing to the ongoing development and application of these versatile models.

Chapter 2

Background

2.1 Functional Data Analysis

One of the fundamental scenarios of statistics is the analysis of data observations drawn independently from a given probability distribution. Frequently, the support of this distribution is embedded within Euclidean space, so that observations arise in the form of finite dimensional vectors. There are however, areas of statistics that study random variables taking values in spaces that are not necessarily Euclidean. For example, some data naturally arise in the form of a continuous observation over time, and are properly considered as functions. Airplane trajectories and speech pitch are two examples. The study of such data and their underlying constructs is known as functional data analysis.

In probability theory, random variables are defined to be measurable maps from some underlying probability space into the real numbers. That definition clearly does not work when the context is shifted to functional data, although the same idea can be applied. Indeed, function-valued random variables are often characterized as random elements of an infinite-dimensional, separable Hilbert space. Typically, this space is chosen to be $L^2(\mathcal{T}, \mu)$, where $\mathcal{T}$ is a closed interval and μ is the Lebsegue measure. This is the Hilbert space of functions defined over $\mathcal{T}$ which are square integrable with respect to μ. An alternative definition posits functional data to be observed paths of some underlying stochastic process.

When this process is mean-square continuous with continuous paths, the Hilbert space perspective and the stochastic process perspective coincide.

The infinite dimensional nature of functional data poses challenges for the formulation of useful models. A useful tool in this regard is the Karhunen-Loeve expansion. Let $(\Omega, \mathcal{A}, P)$ be a probability space and let $X : \mathcal{T} \times \Omega \to \mathbb{R}$ be a second-order, mean-square continuous stochastic process taking values in $L^2(\mathcal{T})$. Then, there are zero-mean random variables C_i and a countable orthonormal basis ψ_i for $L^2(\mathcal{T})$ such that $X(t, \omega)$ can be represented as

$$X(t, \omega) = \sum_{i=1}^{\infty} C_i(\omega)\psi_i(t).$$

Modelling may then proceed by, for example, assuming that the function-valued random variable of interest lives in a space spanned by finitely many of the ψ_j, or by assuming that much of the variation in the process is governed by the first few C_i.

Another important point regarding functional data analysis is that technological limitations often inhibit our ability to fully capture observations. In such cases, functional data are only truly functional in a theoretical sense. In particular, it is quite common that only finitely many entries of the infinite dimensional data point are actually observed. In many applications then, functional data are recorded as finite dimensional vectors, an object which we have already explained that the statistics community is quite comfortable handling. However, (Ramsay and Silverman, 2005) cautions against the temptation to analyze these data using established multivariate techniques, suggesting that methods specifically taking the functional origin of the data into account will provide a better analysis. Stemming from this notion, statistical researchers have busied themselves extending many of the familiar methods used in multivariate analysis to accommodate data of a functional nature. For example, linear models (Cardot et al. (1999), Cardot et al. (2003), Chen et al. (2011)) , graphical models (Zhu et al. (2016), Qiao et al. (2019)), principal component analysis (Dauxois et al. (1982), Rice and Silverman (1991), Silverman (1996), Jacques and Preda (2014b)), and hypothesis testing (Hall and Keilegom (2007), Zhang et al. (2011), Fremdt et al. (2013)) have all been modified to accommodate function-valued random variables. A good introduction to functional data analysis is provided in Ramsay and Silverman (2005), while Wang et al. (2016) provides a high level overview of more contemporary methods in the functional data literature.

2.2 Model-Based Clustering and Parsimony

A data observation $\mathbf{X}$ is said to arise from a finite mixture model with G components if the density of the distribution can be expressed as a convex combination of G component densities,

$$\mathbf{X} \sim \sum_{g=1}^{G} \pi_g f(\mathbf{x} \mid \boldsymbol{\theta}_g) \quad \text{such that,} \quad \sum_{g=1}^{G} \pi_g = 1 \quad \text{and,} \quad \pi_g > 0, \forall\, g.$$

Model-based clustering is the employment of finite mixture models to identify latent homogeneous subgroups within data. The standard interpretation is that each of the G components in the fitted mixture model correspond to a latent group in the data (McNicholas, 2016). The first known use of finite mixture models for this purpose is Wolfe (1965), while the idea was popularized by works such as Duda and Hart (1973), Dempster et al. (1977), McLachlan and Peel (2000), and Fraley and Raftery (2002).

One unfortunate issue is the tendency for mixture model inference to suffer from the curse of dimensionality (Bellman, 1954)—the number of parameters required to fit the mixture increases rapidly with data dimension. This issue is the driving force behind research into parsimonious mixture models—models in which parameters are provided parsimonious specifications, reducing the rate at which the total number of parameters increases with dimensionality of the data.

One clever way to foster parsimony is to assume that common parameters exist across the components. This is the approach taken by Banfield and Raftery (1993), which considered estimating the covariance matrix through the components of its spectral decomposition. In this way, volume, orientation, and shape of each component covariance matrix can be controlled; specifying that some or any of these are also equal across groups introduces parsimony. The work of Celeux and Govaert (1995) outlines estimation procedures for fitting these models. Fraley and Raftery (2003) reports the release of publicly available software for fitting models of this class.

Another method for skirting dimensionality issues is subspace clustering, which leverages the *empty space phenomenon* (Scott and Thompson, 1983) and seeks to identify component specific, low-dimensional subspaces in which the data are well represented. The mixture of factor analyzers (Ghahramani and Hinton, 1996), and their specific application to high dimensional data (McLachlan et al., 2003; McNicholas and Murphy, 2008), are an example of such a model, where complexity is increased only to characterize subspaces of interest.

Another methodology for subspace clustering is proposed in Bouveyron et al. (2007). This approach assumes that trailing covariance eigenvalues are all equal, hence any information within the subspace spanned by the associated eigenvectors is contained in the projection of the data onto the orthogonal complement of the free eigenvectors. As such, the subspace spanned by the free eigenvectors becomes the subspace of interest.

There are, of course, multiple options for introducing parsimony into models, although the two mentioned here are among the most popular in the current model-based clustering meta. Good reviews and references of parsimonious model-based clustering can be found in Bouveyron and Brunet-Saumard (2014) and Bouveyron et al. (2019).

2.3 Stochastic EM Algorithm

The EM algorithm is a numerical optimization procedure, in which an objective function, often a likelihood, is maximized indirectly by iteratively defining a lower bound and then proceeding to maximize it. This algorithm was used heavily in classical latent variable modelling literature due to its ease of implementation with data models derived from exponential family distributions. It is also the de facto algorithm for fitting finite mixture models for frequentist inference, owing to its effective accomodation of the latent cluster memberships.

Each iteration of the algorithm is comprised of two steps, the E-step and the M-step. In the E-step, the complete-data loglikelihood, which is the likelihood formed from the joint distribution of the observed and latent data, is integrated with respect to the latent data conditional on the observed data and the current best estimate of the unknown parameter. The resulting construct is the aforementioned lower bound, which is more rightfully referred to as a minorizer. Obtaining the argmax of this minorizer is the duty of the M-step. Once obtained, this value will serve as the next best guess of the unknown parameter, and thanks to the properties of minorizers, this estimate is guaranteed not to be worse than that of the previous iteration.

Both steps of the EM algorithm can present challenges when it comes to implementation. The E-step, for example, requires the conditional distribution of the latent data given the observed, which may be difficult to obtain. It also involves integration, which may prove intractable. In this latter scenario, a common workaround is to use a Monte Carlo approximation to the E-step. This results in the *stochastic* EM algorithm, so-called because the aforementioned Monte Carlo approximation injects an element of randomness into the algorithm's trajectory. When stochastic EM is employed in lieu of the standard algorithm, an additional estimation step is required, as the resulting sequence of parameter values no longer correspond to monotonic increases in the observed-data likelihood. Common estimation approaches include averaging the elements of the chain, and choosing the element of the chain corresponding to the largest likelihood value. For more details see Chapter 6.

Chapter 3

Functional Data Clustering by Projection into Latent Generalized Hyperbolic Subspaces

3.1 Introduction

Model-based clustering is an unsupervised learning algorithm which enables the clustering of unlabelled data into homogeneous groups. In contrast to non-parametric approaches, which amalgamate data into clusters through some measure of closeness, model-based clustering assumes each cluster arose from a known probability distribution having unknown parameter values. These unknown parameters are estimated using an appropriate numerical method, and cluster assignment is completed using the posterior probability of belonging to the g^{th} group. Model-based clustering has a rich literature, and many researchers continue to work on extending the applicability of this clustering approach into previously problematic scenarios.

Model-based clustering often falters when extended to the high dimensional setting, due to the large number of parameter estimates required. This is related to what Bellman (1954), called “The curse of dimensionality." Many strides have been taken to amend this deficiency, with a multitude of more recent research focusing on developing adaptations to model-based clustering that extend efficiently to the high dimensional setting without sacrificing the properties that make the model-based approach desirable. These include regularization of the covariance matrix (Hastie et al. (1995), Bickel and Levina (2008)), parsimonious model specification (Banfield and Raftery (1993), Celeux and Govaert (1995)), mixtures of factor analyzers (Mclachlan et al. (2007), Baek et al. (2010)) among others. For a relatively recent review of the model-based clustering meta, refer to Bouveyron and Brunet-Saumard (2014).

One notable adaptation, aptly named High Dimensional Data Clustering (HDDC), is outlined in Bouveyron et al. (2007). The results of this research sprouted from the clever notion of projecting separate group clusters into distinct, lower dimensional subspaces. It goes on to demonstrate that employing this approach in the context of a GMM reduces the total number of required parameter estimates to be a linear function of the data dimension, p. The idea of subspace clustering was not originally developed in Bouveyron et al. (2007). As Bouveyron and Brunet-Saumard (2014) notes, some of the earliest methods correspond to the work of Ghahramani and Hinton (1997) and McLachlan et al. (2003). Extensions of HDDC to alternative distributions are given in Pesevski et al. (2017), which develops a subspace clustering method for a multivariate-t distribution, and Kim and Browne (2018) which adapts subspace clustering to the aforementioned mixture of generalized hyperbolic distributions. For a complete review of high dimensional data clustering methods, and subspace clustering methods in particular, see Bouveyron and Brunet-Saumard (2014) and Parsons et al. (2004), where the latter details mainly nonparamteric approaches. Methods such as these, together with the aforementioned methods, contribute to the successful extension of model-based clustering to the high dimensional setting.

A relatively new area in which model-based clustering is being experimented with is functional data analysis. Often, functional data arise in clearly defined groups, such as longitudinal biological measurements of a control and treatment groups, or the observed trajectories of different types of baseball pitches. The problem, however, is that the notion of a probability density or likelihood function is extremely difficult to define on general, infinite dimensional function spaces, see for example Lin et al. (2018). This suggests that the model-based approach to clustering is not feasible as a direct approach in this context.

As a consequence, most early methods for functional clustering relied on either simple dimension reduction — through regularization approaches applied directly to the discretely observed functional data—or basis projection, in which the basis expansion coefficients are treated as multivariate data. The former approach is referred to as “raw-data clustering" while the later are called “two-stage" approaches, according to Jacques and Preda (2014a). As noted in James and Sugar (2003) these methods “break down" in the face of fairly realistic conditions, such as non-uniformity across the sets of time points on which the functional data are observed, or if the functional data are sampled sparsely. Accordingly, the work in James and Sugar (2003) alleviates these issues by proposing a latent random effects model on the basis expansion coefficients, allowing them to "borrow information across curves." This could be considered the first model-based approach to clustering functional data. Some of the more recent works in this area are Bouveyron and Jacques (2011), which extends HDDC to the domain of functional data clustering, Schmutz et al. (2020) which extends clustering methods to multivariate functions, and Bouveyron et al. (2015) which chooses a discriminating subspace. For a full review of functional data clustering algorithms, see Jacques and Preda (2014a).

In this chapter, we propose a method for clustering functional data. The method extends the ideas of the latent subspace approach presented in Bouveyron and Jacques (2011), by generalizing to a mixture of jointly generalized hyperbolic distributions. Parameter estimation is done via the EM algorithm, although here we employ a multicycle extension of the algorithm. The number of mixture components will be chosen using the Bayesian Information Criterion (BIC). The remainder of the chapter will have the following structure: In section 3.2 we provide an overview of the distribution that will be imposed on the data, and some of its desirable properties. Section 3.3 will then provide the theoretical framework of our method, while section 3.4 outlines the specifics of parameter estimation. Section 3.5 gives the results of running the proposed method against a predecessor on simulated and real world data, and finally, section 3.6 gives a full summary of our results and a discussion on directions for further work.

3.2 Distributional Overview

The model component of our functional clustering algorithm will rely on the generalized hyperbolic distribution (GHD). The benefits of employing the GHD in modelling has been demonstrated in the fields of quantitative risk management and model-based clustering by McNeil et al. (2015), and Browne and McNicholas (2015) respectively. The present section will provide a brief overview of the generalized hyperbolic distribution.

3.2.1 Generalized Hyperbolic Distribution

A random variable X is said to have a p dimensional generalized hyperbolic distribution, denoted by $\mathcal{G}_p(\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \lambda, \omega, \eta)$, if it satisfies,

$$X \overset{d}{=} \boldsymbol{\mu} + W\eta\boldsymbol{\beta} + \sqrt{W\eta}\boldsymbol{U} \tag{3.1}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\beta}$ are p dimensional vectors, $U \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$, and $W\eta$ is a generalized inverse-Gaussian (GIG) distribution with pdf

$$h(w|\omega, \eta, \lambda) = \frac{(w/\eta)^{\lambda-1}}{2\eta K_\lambda(\omega)} \exp\left\{ -\frac{\omega}{2}\left(\frac{w}{\eta} + \frac{\eta}{w}\right)\right\}. \tag{3.2}$$

In this parameterization, η is a scale parameter, ω is a concentration parameter, and λ is an index parameter. To indicate that a random variable has a GIG distribution with this parameterization we write $\mathcal{I}(\omega, \eta, \lambda)$. As discussed in Browne and McNicholas (2015), the given formulation of the generalized hyperbolic distribution has an identifiability issue in the sense that,

$$\mathcal{G}_p(\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \lambda, \omega, \eta) \overset{d}{=} \mathcal{G}_p(\boldsymbol{\mu}, c\boldsymbol{\beta}, c\boldsymbol{\Sigma}, \lambda, \omega, c^{-1}\eta),$$

for any $c \in \mathbb{R}^+$. McNeil et al. (2015) dicuss multiple ways to remedy this identifiability issue. For consistency, we follow the convention adopted by Browne and McNicholas (2015), who choose to set $\eta = 1$ to obtain indentifiable parameters. With this choice, the pdf of X is given by,

$$f(\boldsymbol{x}|\boldsymbol{\theta}) = \left[\frac{\omega + \delta(\boldsymbol{x}, \boldsymbol{\mu}|\boldsymbol{\Sigma})}{\omega + \boldsymbol{\beta}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}}\right]^{(\lambda-p/2)/2} \frac{K_{\lambda-p/2}\left(\sqrt{[\omega + \boldsymbol{\beta}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}][\omega + \delta(\boldsymbol{x}, \boldsymbol{\mu}|\boldsymbol{\Sigma})]}\right)}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}K_\lambda(\omega)\exp(-(\boldsymbol{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta})} \tag{3.3}$$

where $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \lambda, \omega)$ is the set of parameters required to specify a GHD, and K_λ is the modified Bessel function of the third kind with index λ.

What makes the GHD alluring for clustering applications is its wide assortment of special-case and limiting distributions. Distributions that fall under this category are the multivariate t, Laplace, Gaussian, skew-normal. In a sense, it allows for a mixture model consisting of varying mixing distributions, under a framework in which the distributions are decided by parameter estimation of a single parent distribution. Borrowing the notation used in Browne and McNicholas (2015), we indicate a random variable X has a p dimensional GHD by writing, $X \sim \mathcal{G}_p^*(\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \lambda, \omega)$.

3.3 Functional Latent Mixture Model

Assume that there is a set of observed curves $\{\boldsymbol{x}_i\}^n$ that we wish to cluster into G homogeneous groups. We will accomplish this task by implementing a model-based, subspace clustering approach that imposes a mixture of jointly generalized hyperbolic distributions (JGHD). This choice further increases flexibility of the mixing distributions, as detailed in the section that follows.

3.3.1 Basis Expansion

Observed functional data often take the form of a discrete time series. In functional data analysis, we want our models to account for the fact that these are functional observations, so quite often the first step is to return the data to a functional form. This is done through a basis expansion approach, where the data are assumed to have arisen from a function space spanned by a discrete set of basis functions. That is, if X is the L_2 continuous process that generated the observed curves in $L_2[0, T]$, we assume there exists a set of basis functions $\{\psi_j\}_{j=1}^p$ such that X can be expressed as,

$$X(t) = \sum_{j=1}^{p} \upsilon_j(X)\psi_j(t) \tag{3.4}$$

where the vector of coefficients $\mathbf{\Upsilon} = (\upsilon_j(X))_{j=1}^{p}$ is assumed to be a random vector in $\mathbb{R}^p$. In particular, we specifically assume there exists a basis such that the resulting coefficients for each of the G groups are distributed according to a JGHD. Once an appropriate basis is chosen, we project each of the observed curves $\{\boldsymbol{x}_i\}$ into this space. We implement basis expansion using least-squares smoothing. If the observations were assumed to be exactly correct, basis expansion could be approached using an interpolation procedure. Upon basis expansion of the data, our observations of the functional random variable X can be equivalently thought of as individual observations of the finite dimensional, vector-valued random variable $\mathbf{\Upsilon}$. It is this interpretation that will allow us to proceed with clustering these functional data.

3.3.2 A Functional Latent Model

Restricting attention to one particular group g in G, we consider the random variable $\mathbf{\Upsilon}_g$ that generates corresponding coefficients. That is, we consider the function,

$$X(t) = \langle \mathbf{\Upsilon}_g, \mathbf{\Psi}(t) \rangle \tag{3.5}$$

generating observations in the g^{th} group, where $\mathbf{\Psi}(t) = (\psi_j(t))_{j=1}^{p}$. We assume that the observed curves in this group, $\{\boldsymbol{\upsilon}_{gi}\}_{i=1}^{n_g}$ are generated by making n_g independent draws from the random variable $\mathbf{\Upsilon}_g$ and substituting them into the equation above.

For any full rank transformation of the space $L_2[0,T]$ given by an orthogonal matrix $\mathbf{\Gamma}_g$, we have,

$$\begin{aligned} X(t) &= \langle \mathbf{\Gamma}_g' \mathbf{\Upsilon}_g, \mathbf{\Gamma}_g' \mathbf{\Psi}(t) \rangle \\ &= \langle \mathfrak{X}, \mathbf{\Phi}_g(t) \rangle + \langle \boldsymbol{\xi}, \mathbf{\Phi}_g^{\perp}(t) \rangle \end{aligned}$$

where $\mathbf{\Phi}_g(t) = (\phi_j(t))_{j=1}^{d_g}$ is a vector of basis functions which span the d_g dimensional subspace $H_g[0,T]$ of $L_2[0,T]$. Now suppose there exists a particular $\mathbf{\Gamma}_g$ such that the true function $X(t)$ is adequately represented in the resulting functional latent space $H_g[0,T]$. That is, we assume

$$X(t) \approx \langle \mathfrak{X}, \mathbf{\Phi}(t) \rangle \tag{3.6}$$

where the approximation $\langle \mathfrak{X}, \mathbf{\Phi}(t)\rangle$ accounts for much of the variance among the data, while the dimension of the space $H_g[0,T]$ is such that $d_g \ll p$. Now, if the matrix $\mathbf{\Gamma}_g$ has the form $\mathbf{\Gamma}_g = [\,\boldsymbol{Q}_g\, \boldsymbol{E}_g\,]$, where $\boldsymbol{Q}_g$ is the $p \times d_g$ matrix containing the first d_g columns of $\mathbf{\Gamma}_g$, then it is possible to write the coefficient random variable $\boldsymbol{\Upsilon}_g$ as,

$$\boldsymbol{\Upsilon}_g = \boldsymbol{Q}_g \mathfrak{X} + \boldsymbol{\varepsilon} \tag{3.7}$$

where $\mathfrak{X}$ is a random variable which generates the coefficients for the function $X(t)$ expressed in the latent space $H_g[0,T]$, and $\boldsymbol{\varepsilon}$ is an independent random noise term in $\mathbb{R}^p$. By extension of the assumption that the observed coefficients $\{\boldsymbol{\upsilon}_{gi}\}_{i=1}^{n_g}$ were drawn independently, the latent expansion coefficients of the observed curves $\{\mathfrak{x}_{ig}\}_{i=1}^{n_g}$ are also assumed to be independent observations of the random variable $\mathfrak{X}$.

We now proceed with some distributional assumptions on the latent random vectors $\mathfrak{X}$ and $\boldsymbol{\varepsilon}$. First, we assume that $\mathfrak{X}$ is distributed according to a generalized hyperbolic distribution with dimension d_g,

$$\mathfrak{X} = \boldsymbol{\mu}_{1g} + W_{1g}\boldsymbol{\beta}_{1g} + \sqrt{W_{1g}}\boldsymbol{U}_{1g} \tag{3.8}$$

where $U_1 \sim N(\mathbf{0}, \boldsymbol{S}_g)$, $W_{1g} \sim \mathcal{I}(\omega_1, 1, \lambda_1)$ and $\boldsymbol{S}_g = \text{diag}(s_{1g}, ..., s_{d_g g})$. We also assume that the error term $\boldsymbol{\varepsilon}$ is such that the $(p - d_g)$ dimensional vector $\boldsymbol{\xi}$ has distribution,

$$\boldsymbol{\xi} = \boldsymbol{E}_g'\boldsymbol{\varepsilon} = \boldsymbol{\mu}_{2g} + W_2\boldsymbol{\beta}_{2g} + \sqrt{W_{2g}}\boldsymbol{U}_{2g} \tag{3.9}$$

where $\boldsymbol{U}_2 \sim N(\mathbf{0}, b_g I_{(p-d_g)})$ and $W_{2g} \sim \mathcal{I}(\omega_2, 1, \lambda_2)$. Therefore, the distribution of the coefficients for the g^{th} cluster, $\boldsymbol{\Upsilon}_g$, is a multiple-scaled generalized hyperbolic distribution with stochastic relationship,

$$\boldsymbol{\Upsilon}_g = \mathbf{\Gamma}_g\boldsymbol{\mu}_g + \mathbf{\Gamma}_g\Delta_W\boldsymbol{\beta}_g + \mathbf{\Gamma}_g\boldsymbol{V}_g \tag{3.10}$$

where, $\boldsymbol{\mu}_g = (\boldsymbol{\mu}_{1g}, \boldsymbol{\mu}_{2g})^\top$, $\boldsymbol{\beta}_g = (\boldsymbol{\beta}_{1g}, \boldsymbol{\beta}_{2g})^\top$, and $\boldsymbol{V}_g \sim N(\mathbf{0}, \Delta_W\boldsymbol{D}_g)$ with $\boldsymbol{D}_g = \text{diag}(\boldsymbol{S}_g, b_g I_{(p-d_g)})$. We also have that Δ_W is given by,

$$\Delta_W = \begin{bmatrix} W_{1g}\,\mathrm{I}_{d_g} & 0 \\ 0 & W_{2g}\mathrm{I}_{p-d_g} \end{bmatrix}. \tag{3.11}$$

The distribution of $\boldsymbol{\Upsilon}_g$ can then be written as,

$$f_{\boldsymbol{\Upsilon}}(\boldsymbol{\upsilon}) = \left[\frac{\omega_1 + \delta(\boldsymbol{Q}_g'\boldsymbol{\upsilon}, \boldsymbol{\mu}_{1g}|\boldsymbol{S}_g)}{\omega_1 + \boldsymbol{\beta}_{1g}'\boldsymbol{S}_g^{-1}\boldsymbol{\beta}_{1g}}\right]^{\frac{\lambda_1 - d_g/2}{2}} \frac{K_{\lambda_1 - d_g/2}\left(\sqrt{[\omega_1 + \boldsymbol{\beta}_{1g}'\boldsymbol{S}_g^{-1}\boldsymbol{\beta}_{1g}][\omega_1 + \delta(\boldsymbol{Q}_g'\boldsymbol{\upsilon}, \boldsymbol{\mu}_{1g}|\boldsymbol{S}_g)]}\right)}{(2\pi)^{\frac{d_g}{2}}|\boldsymbol{S}_g|^{\frac{1}{2}}K_{\lambda_1}(\omega_1)\exp\{-(\boldsymbol{Q}_g'\boldsymbol{\upsilon} - \boldsymbol{\mu}_{1g})\boldsymbol{S}_g^{-1}\boldsymbol{\beta}_{1g}\}} \times$$

$$\left[\frac{\omega_2 + b_g^{-1}||\boldsymbol{E}_g'\boldsymbol{\upsilon}||^2}{\omega_2}\right]^{\frac{\lambda_2 - (p-d_g)/2}{2}} \frac{K_{\lambda_2 - (p-d_g)/2}\left(\sqrt{\omega_2\left[\omega_2 + b_g^{-1}||\boldsymbol{E}_g'\boldsymbol{\upsilon}||^2\right]}\right)}{(2\pi)^{\frac{p-d_g}{2}} b_g^{\frac{p-d_g}{2}} K_{\lambda_2}(\omega_2)},$$

where $\delta(\boldsymbol{x}, \boldsymbol{\mu}|\boldsymbol{A}) = (\boldsymbol{x} - \boldsymbol{\mu})'\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{\mu})$ is the squared Mahalanobis distance between $\boldsymbol{x}$ and $\boldsymbol{\mu}$, and K_λ is the modified Bessel function of the third kind, with index λ.

From (3.10) it follows that,

$$\operatorname{cov}(\boldsymbol{\Gamma}_g'\boldsymbol{\Upsilon}_g \mid \Delta_W) = \begin{bmatrix} w_{1g}^2\boldsymbol{S}_g & \boldsymbol{0} \\ \boldsymbol{0} & w_{2g}^2 b_g I_{(p-d_g)} \end{bmatrix} \tag{3.12}$$

with $s_{ig} > b_g$ for $i = 1, ..., d_g$. By projecting the data into the space $H_g[0, T]$, the coefficients of the observed curves become independent with variances $w_{1g}^2 s_{ig}$, while the error due to restricting the data exclusively to the space $H_g[0, T]$, is assumed to have expected value $\boldsymbol{0}$ and spherical variance given by $w_{2g}^2 b_g I_{(p-d_g)}$. Thus, by rotating the data, we find that, for each group $g \in G$, only the first d_g eigenvalues are important, while the remaining eigenvalues can be sufficiently represented by a single value b_g.

3.3.3 Functional Latent Mixture Model

We now turn attention to the entire set of observed curves, $\{\boldsymbol{x}_i\}_{i=1}^n$. Our desire is to cluster these observations into G homogeneous groups, and therefore assign to each observation a group designation. Let $\boldsymbol{Z} = (Z_g)_{g=1}^G$ be an unobserved random variable dictating the group membership of an observation $\boldsymbol{x}$. If $\boldsymbol{x}$ belongs to group g, then $Z_g = 1$, otherwise $Z_g = 0$. Group membership is assumed to be mutually exclusive, so that each observation is generated by exactly one of the $\boldsymbol{\Upsilon}_g$'s. However, each time we draw an observation from X, we assume that each $\boldsymbol{\Upsilon}_g$ has a non-zero probability of being chosen. Thus, our model for the coefficients becomes,

$$p(\boldsymbol{\upsilon}) = \sum_{g=1}^{G} \pi_g f_{\boldsymbol{\Upsilon}_g}(\boldsymbol{\upsilon}|\boldsymbol{\theta}_g) \tag{3.13}$$

where $f(\cdot|\boldsymbol{\theta}_g)$ is the density of a multiple-scaled generalized hyperbolic distribution, $\boldsymbol{\theta}_g$ is the corresponding vector of parameters, i.e.

$$\boldsymbol{\theta}_g = (\boldsymbol{\mu}_g, \boldsymbol{\beta}_g, \omega_1, \omega_2, \lambda_1, \lambda_2, \boldsymbol{S}_g, \sigma_g, \boldsymbol{Q}_g),$$

and π_g is the prior probability of the g^{th} group.

3.4 Parameter Estimation

The specified model is fit using a multicycle ECM algorithm, which proceeds with multiple E-steps performed before partial, and mutually exclusive M-steps. We define the missing data to be the group membership indicators z_{ig}, as well as the observations of the latent generalized inverse Gaussian distribution, w_{i1g} and w_{i2g}. The complete data is then given by the set of observed coefficients $\{\boldsymbol{\upsilon}_i\}_{i=1}^{n}$, along with the corresponding z_{ig}, w_{i1g} and w_{i2g}. Therefore, the complete data likelihood is given by,

$$\mathscr{L}_c(\{\boldsymbol{\upsilon}_i\}^n|\boldsymbol{\theta}) = \prod_{i=1}^{n}\prod_{g=1}^{G} \left[\pi_g f_{\boldsymbol{\Upsilon}_g}(\boldsymbol{\Gamma'}_g\boldsymbol{\upsilon}_i|\Delta_{w_{ig}})h_{\Delta_{w_g}}(\Delta_{w_{ig}})\right]^{z_{ig}}$$

It follows that the complete data log-likelihood is,

$$
\begin{aligned}
\ell_c(\boldsymbol{\theta}; \upsilon) = & \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig} \log \pi_g - \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig} \log |\boldsymbol{S}_g| \\
& - \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig} \left[w_{i1g}^{-1} (\boldsymbol{Q}_g' \boldsymbol{\upsilon}_i - \boldsymbol{m}_{1g})' \boldsymbol{S}_g^{-1} (\boldsymbol{Q}_g' \boldsymbol{\upsilon}_i - \boldsymbol{m}_{1g}) \right] \\
& + \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig} \log h_{w_1}(w_{i1g} | \omega_{1g}, \lambda_{1g}) \\
& - \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig} \left[(p - q_g) \log \sigma_g + \frac{1}{w_{i2g}} \frac{||\boldsymbol{E}_g' \boldsymbol{\upsilon}_i||^2}{\sigma_g} \right] \\
& + \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig} \log h_{w_2}(w_{i2g} | \omega_{2g}, \lambda_{2g}) + C,
\end{aligned}
$$

where $\boldsymbol{m}_{1g} := \boldsymbol{\mu}_{1g} + w_{1g}\boldsymbol{\beta}_{1g}$ and C is a collection of terms that do not depend on any of the model parameters. Our multicycle ECM performs two CM steps, with an E-step performed before each. The first CM step provides updates for $(\pi_g, \boldsymbol{\mu}_g, \boldsymbol{\beta}_g, \omega_{1g}, \omega_{2g}, \lambda_{1g}, \lambda_{2g}, \boldsymbol{S}_g, \sigma_g)$ for each group g, while the second CM step provides the updates for $\boldsymbol{Q}_g$'s. Note that efficiency is gained in that all terms involving $\boldsymbol{E}_g$ can be written in terms of $\boldsymbol{Q}_g$, so that estimation of $\boldsymbol{E}_g$ is not required.

E-Step: We compute the expected value of the complete data log-likelihood function with respect to the random variables representing the missing data. To do this, we need to compute the following expectations:

$$
E[Z_{ig} | \boldsymbol{\upsilon}_i] = \frac{\pi_g f_{\Upsilon}(\boldsymbol{\upsilon}_i | \theta_g)}{\sum_{k=1}^{G} \pi_k f_{\Upsilon}(\boldsymbol{\upsilon}_i | \theta_k)} =: \hat{z_{ig}},
$$

$$
E[W_{i1g} | \boldsymbol{\upsilon}_i, z_{ig} = 1] = \left[\frac{\mathfrak{u}_{i1g}}{\mathfrak{v}_{1g}} \right]^{\frac{1}{2}} \frac{K_{\lambda_{1g} - q_g/2 + 1}\left(\sqrt{\mathfrak{u}_{i1g}\mathfrak{v}_{1g}} \right)}{K_{\lambda_{1g} - q_g/2}\left(\sqrt{\mathfrak{u}_{i1g}\mathfrak{v}_{1g}} \right)} =: \hat{a}_{i1g},
$$

$$
E[W_{i1g}^{-1} | \boldsymbol{\upsilon}_i, z_{ig} = 1] = \left[\frac{\mathfrak{u}_{i1g}}{\mathfrak{v}_{1g}} \right]^{-\frac{1}{2}} \frac{K_{\lambda_{1g} - q_g/2 + 1}\left(\sqrt{\mathfrak{u}_{i1g}\mathfrak{v}_{1g}} \right)}{K_{\lambda_{1g} - q_g/2}\left(\sqrt{\mathfrak{u}_{i1g}\mathfrak{v}_{1g}} \right)} - \frac{2\lambda_{1g} - q_g}{\mathfrak{u}_{i1g}} =: \hat{b}_{i1g} \text{ and}
$$

$$E[\log W_{i1g}|\boldsymbol{v}_i, z_{ig}=1] = \frac{1}{2}\log\left[\frac{\mathfrak{u}_{i1g}}{\mathfrak{v}_{1g}}\right] + \frac{\partial}{\partial t}\log K_t\left(\sqrt{\mathfrak{u}_{i1g}\mathfrak{v}_{1g}}\right)\Bigg|_{t=\lambda_{1g}-q_g/2} =: \hat{c}_{i1g},$$

where $\mathfrak{u}_{i1g} = \omega_{1g} + \delta(\boldsymbol{Q}_g'\boldsymbol{v}_i, \boldsymbol{\mu}_{1g}|\boldsymbol{S}_g)$ and $\mathfrak{v}_{1g} = \omega_{1g} + \boldsymbol{\beta}_{1g}'\boldsymbol{S}_g^{-1}\boldsymbol{\beta}_{1g}$. We also have,

$$E[W_{i2g}|\boldsymbol{v}_i, z_{ig}=1] = \left[\frac{\mathfrak{u}_{i2g}}{\omega_{2g}}\right]^{\frac{1}{2}} \frac{K_{\lambda_{2g}-(p-q_g)/2+1}\left(\sqrt{\omega_{2g}\mathfrak{u}_{i2g}}\right)}{K_{\lambda_{2g}-(p-q_g)/2}\left(\sqrt{\omega_{2g}\mathfrak{u}_{i2g}}\right)} =: \hat{a}_{i2g},$$

$$E[W_{i2g}^{-1}|\boldsymbol{v}_i, z_{ig}=1] = \left[\frac{\mathfrak{u}_{i2g}}{\omega_{2g}}\right]^{-\frac{1}{2}} \frac{K_{\lambda_{2g}-(p-q_g)/2+1}\left(\sqrt{\omega_{2g}\mathfrak{u}_{i2g}}\right)}{K_{\lambda_{2g}-(p-q_g)/2}\left(\sqrt{\omega_{2g}\mathfrak{u}_{i2g}}\right)} - \frac{2\lambda_{2g}-(p-q_g)}{\mathfrak{u}_{i2g}} =: \hat{b}_{i2g} \text{ and}$$

$$E[\log W_{i2g}|\boldsymbol{v}_i, z_{ig}=1] = \frac{1}{2}\log\left[\frac{\mathfrak{u}_{i2g}}{\omega_{2g}}\right] + \frac{\partial}{\partial t}\log K_t\left(\sqrt{\omega_{2g}\mathfrak{u}_{i2g}}\right)\Bigg|_{t=\lambda_{2g}-(p-q_g)/2} =: \hat{c}_{i2g},$$

where $\mathfrak{u}_{i2g} = \omega_{2g} + \sigma_g^{-1}(||\boldsymbol{v}_i||^2 - ||\boldsymbol{Q}_g'\boldsymbol{v}_i||^2)$. Plugging these updates into the log-likelihood function we obtain the following expected complete data log-likelihood function,

$$\begin{aligned}
Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) = &\sum_{i=1}^{n}\sum_{g=1}^{G}\hat{z}_{ig}\log\pi_g - \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G}\hat{z}_{ig}\log|\boldsymbol{S}_g| \\
&- \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G}\hat{z}_{ig}\left[\hat{b}_{i1g}(\boldsymbol{Q}_g'\boldsymbol{v}_i - \hat{m}_{i1g})'\boldsymbol{S}_g^{-1}(\boldsymbol{Q}_g'\boldsymbol{v}_i - \hat{m}_{i1g})\right] \\
&+ \sum_{i=1}^{n}\sum_{g=1}^{G}\hat{z}_{ig}\hat{c}_{i1g} \\
&- \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G}\hat{z}_{ig}\left[(p-q_g)\log\sigma_g + \hat{b}_{i2g}\frac{||\boldsymbol{E}_g'\boldsymbol{v}_i||^2}{\sigma_g}\right] \\
&+ \sum_{i=1}^{n}\sum_{g=1}^{G}\hat{z}_{ig}\hat{c}_{i2g} + C.
\end{aligned}$$

First CM Step: We maximize all other parameters while keeping $\boldsymbol{Q}_g^{(t)}$ fixed. These updates are given by,

$$\begin{aligned}
\hat{\pi}_g &= \frac{n_g^{(t)}}{n}, \\
\boldsymbol{\mu}_{1g}^{(t+1)} &= \frac{\sum_{i=1}^{n} \hat{z}_{ig}^{(t)} \boldsymbol{Q}_g'^{(t)} \boldsymbol{\upsilon}_i \Big(\overline{a}_{1g} \hat{b}_{i1g} - 1 \Big)}{\sum_{i=1}^{n} \hat{z}_{ig}^{(t)} \Big(\overline{a}_{1g} \hat{b}_{i1g} - 1 \Big)} \text{ and} \\
\boldsymbol{\beta}_{1g}^{(t+1)} &= \frac{\sum_{i=1}^{n} \hat{z}_{ig}^{(t)} \boldsymbol{Q}_g'^{(t)} \boldsymbol{\upsilon}_i \Big(\overline{b}_{1g} - \hat{b}_{i1g} \Big)}{\sum_{i=1}^{n} \hat{z}_{ig}^{(t)} \Big(\overline{a}_{1g} \hat{b}_{i1g} - 1 \Big)}
\end{aligned}$$

respectively, where $n_g = \sum_{j=1}^{n} \hat{z}_{jg}^{(t)}$, $\overline{a}_{1g} = n_g^{-1} \sum_{j=1}^{n} \hat{z}_{jg}^{(t)} \hat{a}_{j1g}^{(t)}$, and $\overline{b}_{1g} = n_g^{-1} \sum_{j=1}^{n} \hat{z}_{jg}^{(t)} \hat{b}_{j1g}^{(t)}$. The update for the j^{th} diagonal element of $\boldsymbol{S}_g$, $j = 1, ..., d_g$ is then

$$\hat{s}_{jg} = \frac{\sum_{i=1}^{n} \hat{z}_{ig} \Big[\hat{b}_{i1g}^{(t)} (\boldsymbol{Q}_g'^{(t)} \boldsymbol{\upsilon}_i - \boldsymbol{\mu}_{1g}^{(t+1)})_j^2 - 2(\boldsymbol{Q}_g'^{(t)} \boldsymbol{\upsilon}_i - \boldsymbol{\mu}_{1g}^{(t+1)})_j \boldsymbol{\beta}_{1g(j)}^{(t+1)} + \hat{a}_{i1g}^{(t)} \boldsymbol{\beta}_{1g(j)}^{2\,(t+1)} \Big]}{n_g^{(t)}}.$$

The update for the remaining eigenvalues in each group, σ_g, is given by,

$$\hat{b}_g^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ig} \hat{b}_{i2g} (||\boldsymbol{\upsilon}_i||^2 - ||\boldsymbol{Q}_g' \boldsymbol{\upsilon}_i||^2)}{(p - q_g) n_g^{(t)}}.$$

We update ω_{kg} and λ_{kg}, $k = 1, 2$ by maximizing the function,

$$m_g(\omega_{kg}, \lambda_{kg}) = (\lambda_{kg} - 1)\overline{c}_{kg} - \log K_{\lambda_{kg}}(\omega_{kg}) - \frac{\omega_{kg}}{2}(\overline{a}_{kg} + \overline{b}_{kg}),$$

where $\overline{c}_{kg} = n_g^{-1} \sum_{j=1}^{n} \hat{z}_{jg}^{(t)} \hat{c}_{jkg}^{(t)}$. The updates are thus given by,

$$\begin{aligned}
\lambda_{kg}^{(t+1)} &= \overline{c}_{kg} \lambda_{kg}^{(t)} \left[\frac{\partial}{\partial t} \log K_t \Big(\omega_{kg}^{(t)} \Big) \bigg|_{t = \lambda_{kg}^{(t)}} \right]^{-1} \text{ and,} \\
\omega_{kg}^{(t+1)} &= \omega_{kg}^{(t)} - \left[\frac{\partial}{\partial t} m_g \Big(t, \lambda_{kg}^{(t+1)} \Big) \bigg|_{t = \omega_{kg}^{(t)}} \right] \left[\frac{\partial^2}{\partial t^2} m_g \Big(t, \lambda_{kg}^{(t+1)} \Big) \bigg|_{t = \omega_{kg}^{(t)}} \right]^{-1}.
\end{aligned}$$

Second CM-Step: To estimate $\boldsymbol{Q}_g = \boldsymbol{B}^{\frac{1}{2}}\boldsymbol{P}_g$ we need to maximize,

$$
\begin{aligned}
f(\boldsymbol{P}_g) &= -\frac{1}{2}\sum_{i=1}^{n}\hat{z}_{ig}\left[\hat{b}_{i1g}(\boldsymbol{Q}_g'\boldsymbol{\upsilon}_i - \hat{\boldsymbol{m}}_{i1g})'\boldsymbol{S}_g^{-1}(\boldsymbol{Q}_g'\boldsymbol{\upsilon}_i - \hat{\boldsymbol{m}}_{i1g})\right] \\
&= -\frac{1}{2}\mathrm{tr}\left(\sum_{i=1}^{n}\hat{z}_{ig}\hat{b}_{i1g}\boldsymbol{\upsilon}_i\boldsymbol{\upsilon}_i'\boldsymbol{Q}_g\boldsymbol{S}_g^{-1}\boldsymbol{Q}_g'\right) + \mathrm{tr}\left(\sum_{i=1}^{n}\boldsymbol{S}_g^{-1}(\hat{b}_{i1g}\boldsymbol{\mu}_{1g} + \boldsymbol{\beta}_{1g})\boldsymbol{\upsilon}_i'\boldsymbol{Q}_g\right) + C \\
&= -\frac{1}{2}\mathrm{tr}\left(\sum_{i=1}^{n}\hat{z}_{ig}\hat{b}_{i1g}\boldsymbol{B}'^{\frac{1}{2}}\boldsymbol{\upsilon}_i\boldsymbol{\upsilon}_i'\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{P}_g\boldsymbol{S}_g^{-1}\boldsymbol{P}_g'\right) + \mathrm{tr}\left(\sum_{i=1}^{n}\boldsymbol{S}_g^{-1}(\hat{b}_{i1g}\boldsymbol{\mu}_{1g} + \boldsymbol{\beta}_{1g})\boldsymbol{\upsilon}_i'\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{P}_g\right) + C.
\end{aligned}
$$

Maximizing this function is equivalent to minimizing,

$$
f(\boldsymbol{P}_g) = \frac{1}{2}\mathrm{tr}\left(\sum_{i=1}^{n}\hat{z}_{ig}\hat{b}_{i1g}\boldsymbol{B}'^{\frac{1}{2}}\boldsymbol{\upsilon}_i\boldsymbol{\upsilon}_i'\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{P}_g\boldsymbol{S}_g^{-1}\boldsymbol{P}_g'\right) - \mathrm{tr}\left(\sum_{i=1}^{n}\boldsymbol{S}_g^{-1}(\hat{b}_{i1g}\boldsymbol{\mu}_{1g} + \boldsymbol{\beta}_{1g})\boldsymbol{\upsilon}_i'\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{P}_g\right) + C,
$$

with respect to $\boldsymbol{P}_g$. We do this by using a majorizing function $g(\boldsymbol{P}_g)$, which is a function such that, $f(\boldsymbol{P}_g) \leq g(\boldsymbol{P}_g)$ for all $\boldsymbol{P}_g$. Such a function is given in Kiers (1990, 2002), and Browne and Mcnicholas (2014) as having the form $g(\boldsymbol{P}_g) = \text{constant} + \mathrm{tr}(F^{(t)}\boldsymbol{P}_g)$, where $F^{(t)}$ is given by,

$$
\begin{aligned}
F^{(t)} = \sum_{i=1}^{n}\Bigg[&- \hat{z}_{ig}^{(t)}\boldsymbol{S}_g^{-1\,(t+1)}\Big((\hat{b}_{i1g}^{(t)}\boldsymbol{\mu}_{1g}^{(t+1)} + \boldsymbol{\beta}_{1g}^{(t+1)}\Big)\boldsymbol{\upsilon}_i'\boldsymbol{B}^{\frac{1}{2}}\Bigg] + \\
\sum_{i=1}^{n}\Bigg[&- \hat{z}_{ig}^{(t)}\hat{b}_{i1g}^{(t)}\boldsymbol{B}'^{\frac{1}{2}}\boldsymbol{\upsilon}_i\boldsymbol{\upsilon}_i'\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{P}_g\boldsymbol{S}_g^{-1\,(t+1)} - \hat{z}_{ig}^{(t)}\alpha_{ig}^{(t+1)}\boldsymbol{B}'^{\frac{1}{2}}\boldsymbol{\upsilon}_i\boldsymbol{\upsilon}_i'\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{P}_g\Bigg],
\end{aligned}
$$

where α_{ig} is the largest value of the diagonal matrix $w_{i1g}^{-1\,(t)}\boldsymbol{S}_g^{-1\,(t+1)}$. Employing svd we get $-F^{(t)} = L^{(t)}O^{(t)}R'^{(t)}$, where $L^{(t)}$ and $R^{(t)}$ are orthogonal and $O^{(t)}$ is diagonal. The update for $\boldsymbol{\Gamma}_g$ is then $\boldsymbol{\Gamma}_g^{(t+1)} = B^{\frac{1}{2}}R^{(t)}L'^{(t)}$.

3.4.1 Initialization Strategies and Stopping Criteria

Initialization of the algorithm detailed above, which we denote as funGHDDC, can be done with an assortment of approaches. Currently, we have implemented four initialization strategies: k-means, Gaussian parsimonious clustering (GPC), hierarchical clustering, and random. Each of these approaches makes use of the same idea: use some algorithm to generate a set of potential class labels, run a few iterations of the EM algorithm using these class labels, and choose the labels that maximize the log-likelihood. The names of the initialization methods then correspond exactly to the way in which the potential class labels are generated. By default we use a k-means initialization, but in the simulation study we also demonstrate that the GPC initialization performs well.

For stopping, we use an Aitken acceleration-based convergence criterion, which depends on the linear convergence of EM. At the k^{th} iteration, the estimate of the limit is given by,

$$\ell_{\infty}^{(k)} = \ell^{(k)} + \frac{\ell^{(k+1)} - \ell^{(k)}}{1 - a^{(k)}}$$

where,

$$a^{(k)} = \frac{\ell^{(k+1)} - \ell^{(k)}}{\ell^{(k)} - \ell^{(k-1)}}.$$

For a chosen tolerance $\varepsilon > 0$, we stop the algorithm when we have,

$$\left| \ell_{\infty}^{(k)} - \ell_{\infty}^{(k-1)} \right| < \varepsilon.$$

3.5 Simulation Study

In this section we present three simulation studies designed to demonstrate the properties, and advantages of clustering with the funGHDDC algorithm. In the first, data are generated according to the data generative assumptions of the funGHDDC method, where we then demonstrate the parameter recovery potential of the algorithm. In the second simulation study, we compare the performance of some classic selection criteria in choosing the correct number of groups, and the true value for the intrinsic subspace dimension of each group. The third study is comparative, and commences to test the performance of funGHDDC against alternative functional clustering methods.

3.5.1 Parameter Recovery

We show parameter recovery in two separate scenarios. In the first, we have two groups of data, each with a true subspace dimension of $d = 3$. The full dimension of the dataset is set to $p = 100$ and we draw 500 observations from each group. The parameters are chosen as follows:

$$
\begin{aligned}
\boldsymbol{\mu}_1 &= (10, 10, 10) & \boldsymbol{\mu}_2 &= (-10, -10, -10) \\
\boldsymbol{\beta}_1 &= (0.3, 0.6, 0.9) & \boldsymbol{\beta}_2 &= (-0.3, -0.6, -0.9) \\
\boldsymbol{S}_1 &= \text{diag}(2.75, 2.30, 2.10) & \boldsymbol{S}_2 &= \text{diag}(3, 2.5, 2.25) \\
\eta_1 &= 1 & \eta_2 &= 1.1 \\
(\omega_{11}, \lambda_{11}) &= (0.3, -10) & (\omega_{21}, \lambda_{21}) &= (0.3, -10) \\
(\omega_{12}, \lambda_{12}) &= (0.01, -3) & (\omega_{22}, \lambda_{22}) &= (0.01, -3)
\end{aligned}
$$

We have omitted $\boldsymbol{\Gamma}_g$ for each group as its inclusion over-encumbers, and adds little to, the presentation. Means and standard deviations for this parameter are similar to the rest. Those curious can reach out to the author for the complete set of results. The scenario described above is simulated 1000 times, with initialization of the clustering algorithm done with a k-means based approach. The resulting estimate statistics are provided in Table 3.1.

For the second parameter recovery demonstration, we make some alterations. We set the intrinsic dimensions to be different, with $d_1 = 2$ and $d_2 = 4$. We also set the prior probabilities to be $\boldsymbol{\pi} = (0.3, 0.7)$. We set the group parameter values as follows,

$$
\begin{aligned}
\boldsymbol{\mu}_1 &= (10, 10) & \boldsymbol{\mu}_2 &= (0, 0, 0, 0) \\
\boldsymbol{\beta}_1 &= (0, 0) & \boldsymbol{\beta}_2 &= (1, 1, 10, 10) \\
\boldsymbol{S}_1 &= \text{diag}(3, 3) & \boldsymbol{S}_2 &= \text{diag}(1.5, 2, 2, 1.5) \\
\eta_1 &= 1 & \eta_2 &= 1 \\
(\omega_{11}, \lambda_{11}) &= (1, 1) & (\omega_{21}, \lambda_{21}) &= (0.5, -2) \\
(\omega_{12}, \lambda_{12}) &= (0.001, -30) & (\omega_{22}, \lambda_{22}) &= (0.1, -50)
\end{aligned}
$$

Initialization in this case is done using the GPC approach. The results of this parameter recovery simulation are given in Table 3.2.

Table 3.1: Estimation results from the first parameter recovery simulation.

	Component 1		Component 2	
	True	Mean [Std. Dev]	True	Mean [Std. Dev]
$\boldsymbol{\mu}_g$	(10, 10, 10)	(10.25, 9.93, 9.87) [0.584, 0.551, 0.542]	(-10, -10, -10)	(-10.22, -9.78, -10.06) [0.519, 0.621, 0.554]
$\boldsymbol{\beta}_g$	(0.3, 0.6, 0.9)	(0.29, 0.58, 0.94) [0.332, 0.362, 0.384]	(-0.3, -0.6, -0.9)	(-0.28, -0.62, -0.96) [0.271, 0.426, 0.328]
$\boldsymbol{S}_g$	(2.75, 2.30, 2.10)	(2.77, 2.28, 2.11) [0.183, 0.151, 0.141]	(3, 2.50, 2.25)	(2.94, 2.50, 2.26) [0.190, 0.165, 0.143]
η_g	1	1.013 [0.002]	1.1	1.127 [0.002]
$(\omega_{g1}, \lambda_{g1})$	(0.3, -10)	(0.32, -10.11) [0.144, 0.198]	(0.3, -10)	(0.28, -9.98) [0.142, 0.193]
$(\omega_{g2}, \lambda_{g2})$	(0.01, -3)	(0.013, -3.00) [0.041, 0.491]	(0.01, -3)	(0.010, -3.03) [0.042, 0.513]

The results of these two parameter recovery simulations demonstrate the algorithm's ability to recover the parameters in favorable scenarios. That is, with the data generated exactly according to the model assumptions, and with generously chosen sample sizes and replications. We see that variations in the intrinsic subspace dimensions, the distribution of the prior probabilities, the relatively large dataset dimension, and the initialization strategy do not hinder recovery. In addition, we note that for each of the 1000 simulated cases, the algorithm was also able to attain perfect observation classification.

3.5.2 Comparison of Selection Criteria

In this study we check the ability of various selection criteria to choose both the correct number of groups, and the correct values for each of the intrinsic subspace dimensions. Specifically, we test three selection criteria: AIC, BIC, and Hannan-Quinn Information

Table 3.2: Estimation results from the second parameter recovery simulation.

	Component 1		Component 2	
	True	Mean [Std. Dev]	True	Mean [Std. Dev]
$\boldsymbol{\mu}_g$	(10, 10)	(10.33, 9.90) [0.607, 0.511]	(0, 0, 0, 0)	(-0.34, 0.08, 0.35, 0.17) [0.546, 0.625, 0.633, 0.489]
$\boldsymbol{\beta}_g$	(0, 0)	(0.26, -0.12) [0.401, 0.357]	(1, 1, 10, 10)	(0.99, 1.12, 10.24, 10.04) [0.431, 0.515, 0.486, 0.545]
$\boldsymbol{S}_g$	(3, 3)	(3.21, 3.17) [0.283, 0.200]	(1.5, 2, 2, 1.5)	(1.37, 1.86, 2.18, 1.61) [0.231, 0.313, 0.254, 0.207]
η_g	1	1.048 [0.005]	1	1.033 [0.0039]
$(\omega_{g1}, \lambda_{g1})$	(1, 1)	(0.92, 1.07) [0.136, 0.154]	(0.5, -2)	(0.52, -1.99) [0.115, 0.138]
$(\omega_{g2}, \lambda_{g2})$	(0.001, -30)	(0.001, -29.91) [0.003, 0.491]	(0.1, -50)	(0.106, -3.10) [0.055, 0.213]

Criterion (HQC).

Each of these criterion are dependent on the total number of estimated parameter values, k, and both BIC and HQC also depend on the number of sample points, n. For the funGHDDC algorithm, the full set of parameters k is calculated as,

$$k = (G-1) + \sum_{g=1}^{G} \left[(p+3)d_g - d_g(d_g+1)/2 + 6\right]. \tag{3.14}$$

We then have the following formulae for the selection criteria:

$$\begin{aligned} AIC &= 2k - 2\ell(\hat{\boldsymbol{\theta}}), \\ BIC &= k\log n - 2\ell(\hat{\boldsymbol{\theta}}) \text{ and} \\ HQC &= 2k\log\log n - 2\ell(\hat{\boldsymbol{\theta}}). \end{aligned}$$

The simulation procedure proceeds as follows. The true number of groups is chosen to be $G = 2$, and each of these groups has a three dimensional intrinsic subspace. We generate 100 datasets consisting of two groups, each with a three dimensional intrinsic subspace dimension. The group parameters are randomly generated using various distributions of large variance. For each dataset, we fit a model for every possible combination of group values $G = 1, 2, ..., 5$, and intrinsic subspace dimension values $d_g = 2, 3, 4$. For example, $G = 3$, and $d_1 = 2, d_2 = 2, d_3 = 4$ is one possible combination. In total, 363 models are fit for each dataset. For each of these fits, we calculate the three chosen selection criteria. For each of the selection criteria, we chose the model that minimizes its value. The results of this simulation are given in Figure 3.1.

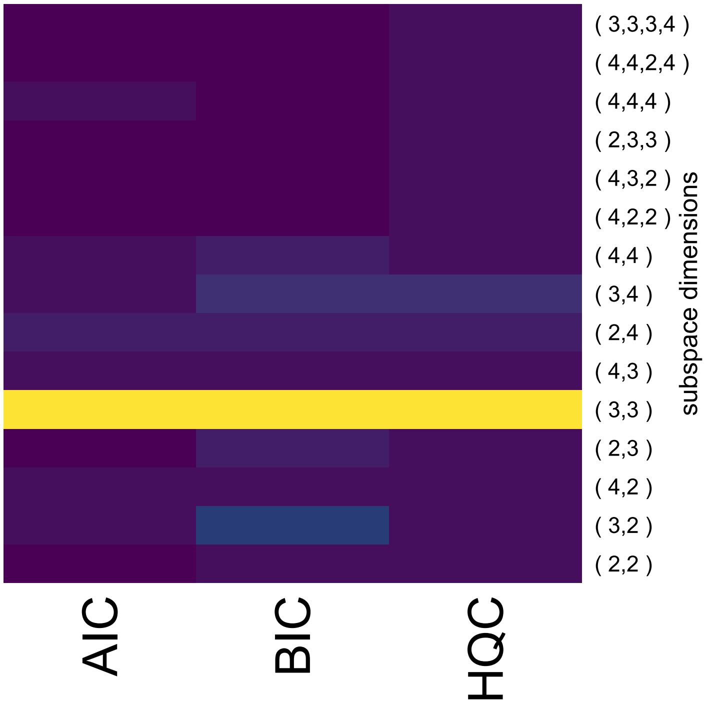


Figure 3.1: A heatmap of the models selected by the three selection criteria. Yellow indicates higher frequency.

Figure 3.1 illustrates that all three criteria do an adequate job of choosing the correct model. This is evidenced by the intense yellow band in Figure 3.1 which corresponding to the frequency with which the correct model as chosen. In particular, of the 100 datasets, HQC chose the correct model most frequently, with 70 successes. It was closely followed by AIC, which chose the correct model 66 times. Finally, BIC chose the correct model 54 times. Interestingly, BIC chooses the wrong number of groups the least, with only 5 such occurrences. Indeed, it seems BIC, when it chooses incorrectly, settles on a model specification that is very close to the true model. In some sense, the variance in model selection of the BIC seems to be smaller than the other two criteria. We conclude that any of these three selection criteria perform sufficiently well for model selection in the context of the funGHDDC algorithm.

3.5.3 Comparison Study

We now proceed to compare the performance of funGHDDC with alternative methods of functional clustering. The comparison is done on simulated datasets, which are constructed as follows: generate 200 observations from each of four separate, 3 dimensional generalized hyperbolic distributions whose parameters are randomly generated. Appended to each of these four sets of observations an additional 98 dimensions of data generated by another generalized hyperbolic distribution with randomly generated parameters constrained to satisfy equation (3.9). The resulting four datasets each consist of two-hundred, 101-dimensional observations. Each of these four datasets is then transformed by a randomly generated orthogonal matrix. Finally, the four separate datasets are brought together to form a single dataset consisting of 800 observations. We assume that these data represent the observed coefficients of functional random variables living in a function space spanned by an orthonormal basis. We procedurally generate 500 such datasets and apply the competing clustering algorithms to each one.

For alternative methods, we choose funHDDC (Bouveyron and Jacques (2011)), the direct predecessor of funGHDDC, and funFEM (Bouveyron et al. (2015)), a best discriminating subspace approach. For further details on these methods, we direct readers to the corresponding papers in which they are developed.

One particular set of simulated functions is depicted in Figure 3.2. The coloring and segmentation of the plot represent the separate groups. The groups vary about an obvious

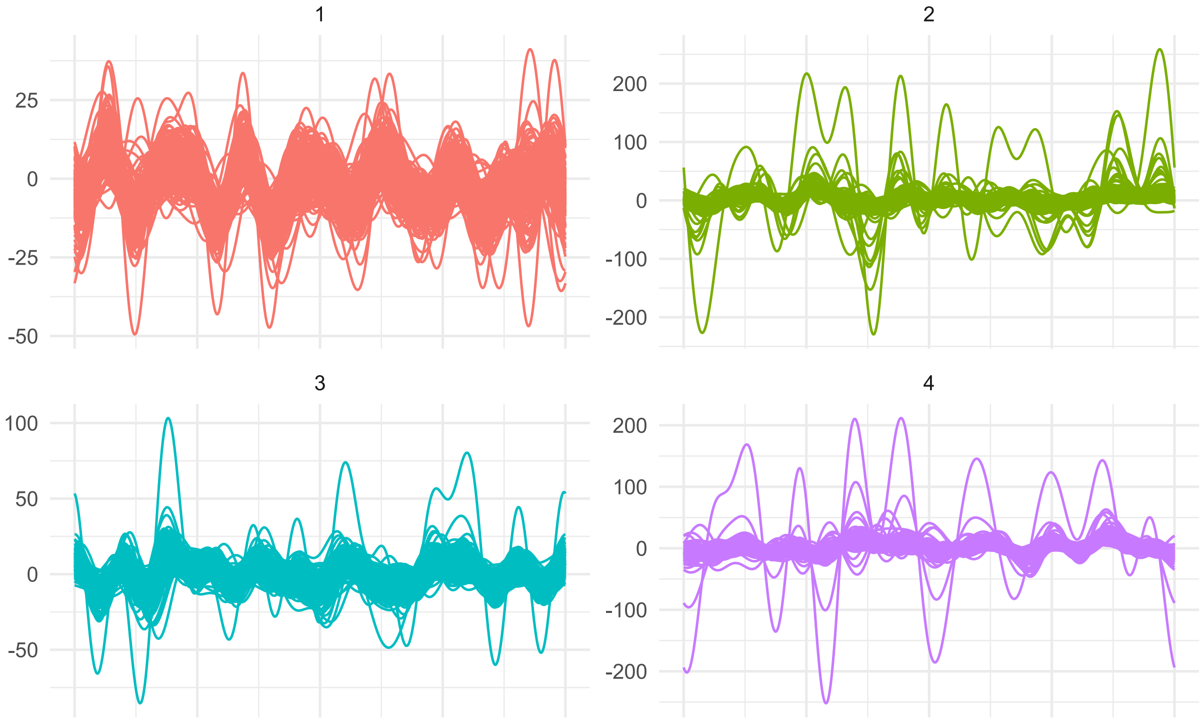


Figure 3.2: One of the simulated functional datasets. Subplots correspond to separate groups

mean function, however each group also has some observations that deviate significantly from this mean, which is a behaviour allowed by the distributional assumption on the coefficients. As for the methods, both funGHDDC and funHDDC are initialized using a k-means approach, while funFEM is initialized by hierarchical clustering. To make the results directly comparable, the true number of groups is set to the true value, $G = 4$ for each algorithm. The success of the classification reported by each of the competing methods is measured by Correct Classification Rate (CCR), which reports the proportion of correctly classified observations to the total number of observations. The simulation results are given in Figure 5.3.

On the left of Figure 5.3 we see three overlaid histograms, each representing the distribution of CCR values generated by a particular clustering method on the simulated datasets.

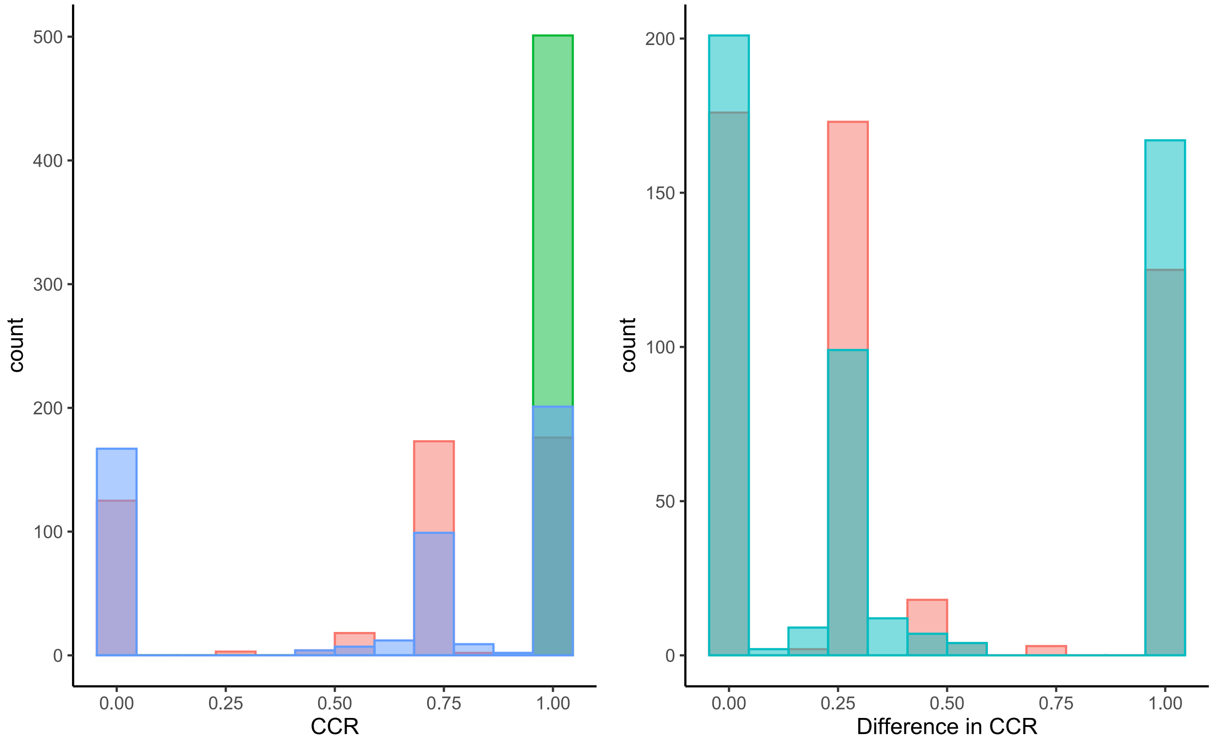


Figure 3.3: [Left]: A histogram of the CCR results from the funHDDC model (blue), the funGHDDC model (green) and the funFEM model (red). Zeros represent cases where the algorithm failed to converge. [Right]: A histogram of the difference in CCR between funGHDDC and funHDDC (aqua) and funGHDDC vs funFEM (red) on each of the simulated datasets.

The first thing to note is that the funHDDC and funFEM histograms are trimodal. The mode at 0 indicates cases where these methods failed to converge. It is important to note that these methods would almost certainly attain convergence on these datasets should the number of groups become a free parameter. Otherwise, it seems that these methods performed adequately, with an average CCR of 0.873 for funHDDC, and 0.836 for funFEM, when the zeros are ignored. As for the funGHDDC algorithm, the numerical results reflect the theory, with near perfect classification exhibited in each case. The second plot in Figure 5.3 graphs two overlaid histograms, each representing the difference in the resulting CCR of funGHDDC and the CCR of one of the competing methods on each of the simulated datasets. We see that almost all of the differences are positive, signifying that funGHDDC consistently outperformed its competitors. We conclude that, in the context of functional basis coefficients exhibiting jointly generalized hyperbolic distributions, funGHDDC is the best functional clustering approach among the tested algorithms.

3.6 Real Data Application

We consider the clustering of three observational datasets. The chosen datasets are the ECG, Wafer, and Symbols datasets. All three datasets are available at Dau et al. (2018). The ECG dataset has been commonly analyzed in the functional clustering context, for example see Jacques and Preda (2014b) and Jacques and Preda (2014a). It is comprised of ECG readings sampled at 96 equally spaced points for two distinct groups.

The Wafer dataset corresponds to 152 measurements made during specialized processing of silicon wafers. Groups are defined by normal vs abnormal observations, based on the results of these measurements, and hence the dataset contains two distinct groups.

Finally, the Symbols dataset is the result of an experiment where people were asked to draw a randomly chosen symbol from 6 possible choices. The observed discrete measurements correspond to the x movement of the writing utensil as the symbol was drawn. This dataset includes six classes, however the classes mostly exhibit highly distinct functional behaviour with the exception of two classes that display an interesting entanglement. We choose to work solely with these two groups. Projections of the three datasets onto a 23 dimensional Fourier basis are plotted in Figure 3.4.

For each of these datasets, we project the discrete observations onto both a Fourier and B-spline basis, each consisting of 23 basis functions. The coefficients of these projections are then fed into each of the competing clustering algorithms. We compare the clustering methods using correct classification rate (CCR), which we define as the proportion of observations that have been correctly classified by the algorithm, and BIC. In contrast to Section 3.5.2, we here define BIC so that larger values are better, to make comparison with the results of other models more intuitive.

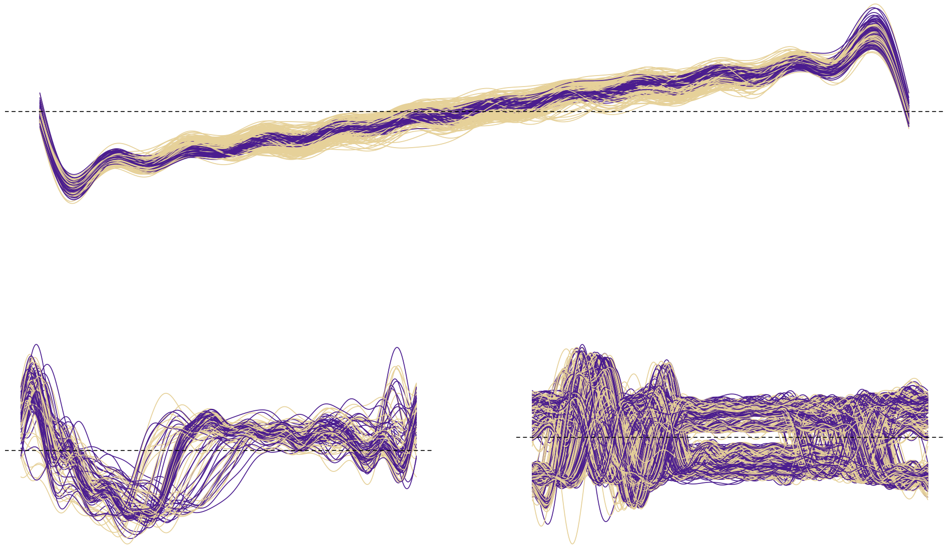


Figure 3.4: The three observational datasets. The top plot corresponds to the Symbols data, bottom left is the ECG data, and bottom right is the Wafer data. Colorings are done according to the true labels, so we see that each dataset is composed of two groups.

Each method was initialized using the default parameter values provided by the associated R function, except in the case that convergence was not attained or the chosen model was a poor fit. In particular, we made the following alterations to the default parameters when required:

i) funHDDC was run with a random initialization for the wafer dataset, when it was projected onto the B-spline basis.

ii) Selection of the subspace dimension was done using BIC for the funHDDC algorithm for the ECG data projected onto the Fourier basis.

iii) The funFEM method was initialized using k-means on both the Wafer and ECG datasets.

Table 3.3: The results of the three observational analyses. Each cell is reported in the format of CCR (BIC). The best CCR for each basis is bolded and best BIC is denoted by an asterisk.

	Fourier			Bspline		
	ECG	Wafer	Symbols	ECG	Wafer	Symbols
funHDDC	71.7 (6509)	65.4 (65484)	73.2 (37942)	65.7 (-446645)	90.1 (-5770441)	N/A
funFEM	75.8 (2216)	65.6 (19136)	86.3 (17341)	77.8 (-2307)	65.2 (-30254)	91.3 (1874)
Funclust	56.7 (1296)	N/A	53.6 (14614)	51.0 (1222)	N/A	51.6 (14638)
funGHDDC	**82.8** (6774)*	**93.0** (67462)*	**95.3** (38389)*	**80.8** (6296)*	**93.5** (57618)*	**97.1** (37831)*

iv) In all cases, we specified that the funFEM algorithm selects the best model from all possible sub-models.

Occasionally, a method would not converge under any parameter specifications, in which case we simply report the value N/A. We can only speculate that in such cases, it is likely that initialization of the class labels results in classes that are highly nonconforming to the assumptions of the model. The results of analyzing these datasets are given in Table 3.3.

In terms of classification rate, we see that funGHDDC is very competitive, producing the best results in each of the six analyses. It is also uniformly chosen as the best model by BIC. We note that when we have the Fourier basis, the BIC values of the funHDDC model closely rival those of the funGHDDC model. Further, we draw attention to the fact that BIC values are fairly stable across basis choice, except in the case of funHDDC, where they vary wildly. This can be explained, in part, by two procedures of the algorithms. First, when the basis is not orthonormal, the coefficients are modified by the basis inner product matrix. This is done to take into account the functional nature of the data, and follows from the underlying theory of functional principal component analysis. In the current scenario, the Fourier basis is orthonormal while the B-spline basis is not. This difference in the treatment of the coefficients across the test bases could explain the BIC deviance here. Additionally, in our own earlier simulation results, we noticed that scaling the coefficients can also cause noticeable changes in the BIC values of the resulting model. It is possible that both of these factors are combining to cause the BIC variation noted in the funHDDC (and to a less extent, the funFEM) results.

3.7 Conclusion

We have introduced a model-based, subspace clustering approach for clustering functional data, funGHDDC. Our work has demonstrated that, in carefully curated scenarios, as well as in observational data analyses, this algorithm can outperform other functional clustering methods and special case counterparts. However, echoing the sentiment of Browne and McNicholas (2015), we do not claim that funGHDDC is a uniformly best approach to functional clustering. Indeed, one heel of our approach is that, although computation for our algorithm is efficient, it is often still slower than competitors. That is, our method appears to be quite widely applicable for clustering functional data, but it is not without drawbacks. Further, there is more to be done regarding the general details of this approach. For example, our model currently assumes that the rotation matrix $\mathbf{\Gamma}$ is full rank. This implies that the span of the chosen basis exactly matches that of the true underlying basis. This would in fact almost never occur, and therefore an adjustment for the model to account for differing basis spans should be considered. With respect to basis choice, currently the choice is rather arbitrary, being guided by simple heuristics. To further improve functional clustering models, one might think of developing a data-driven basis, so that the model assumed by the basis is always guaranteed to cover a good portion, or at least contain, the span of the true underlying basis. This would likely lead to more robust models, and better clustering results overall. Certainly there are many interesting questions still left to explore regarding this model, and functional data clustering in general.

Chapter 4

A Dual Subspace Parsimonious Mixture of Matrix Normal Distributions

4.1 Introduction

An observation $\mathbf{X}$ is said to arise from a finite mixture of G components if the density of the distribution can be expressed as a convex combination of G component densities,

$$\mathbf{X} \sim \sum_{g=1}^{G} \pi_g f(x \mid \boldsymbol{\theta}_g) \quad \text{such that} \quad \sum_{i=1}^{G} \pi_g = 1 \quad \text{and} \quad \pi_g > 0 \ \ , \forall\, g.$$

Model-based clustering is the employment of finite mixture models to identify latent homogeneous subgroups within data. In a typical application, each latent group in the data is assumed to correspond to a unimodal component within the fitted mixture model McNicholas (2016). The first known use of finite mixture models for this purpose is Wolfe (1965), while the idea was popularized by works such as Duda and Hart (1973), Dempster et al. (1977), McLachlan and Peel (2000), and Fraley and Raftery (2002).

One unfortunate issue is the tendency for mixture model inference to suffer from the curse of dimensionality—the number of parameters required to fit the mixture increases rapidly with data dimension Bellman (1954). This issue was the driving force behind research into parsimonious mixture models, in which parameters are provided parsimonious specifications so as to reduce the rate at which the total number increases with dimensionality.

One clever way to foster parsimony is to assume that common parameters exist across the components. This is the approach taken by Banfield and Raftery (1993), which considers fitting the spectral decomposition of the group covariance matrices. In this way, volume, orientation, and shape of each component covariance matrix can be controlled; specifying that some or any of these are also equal across groups provides parsimony. The work of Celeux and Govaert (1995) provides estimation procedures for these models and extends to approach to a more general class of models. Fraley and Raftery (2003) reports the release of publicly available software for fitting models of this class.

Another method for skirting dimensionality issues is subspace clustering, which leverages the *empty space phenomenon* Scott and Thompson (1983) and seeks to identify component-specific, low-dimensional subspaces in which the data are well represented. The mixture of factor analyzers Ghahramani and Hinton (1996), and their specific application to high dimensional data (McLachlan et al. (2003), McNicholas and Murphy (2008)), increase in complexity only to characterize subspaces of interest. Another approach to subspace clustering is proposed in Bouveyron et al. (2007). The approach assumes that trailing covariance eigenvalues are all equal, hence any information within the subspace spanned by the associated eigenvectors is contained in the projection of the data onto the orthogonal complement of the free eigenvectors. As such, the subspace spanned by the free eigenvectors becomes the subspace of interest. Good reviews and references to parsimonious model-based clustering can be found in Bouveyron and Brunet-Saumard (2014) and Bouveyron et al. (2019).

The prevalence of problems considering matrix-valued data observations has increased steadily over recent years, elevated by our continued improvements in computational efficiency and power. Commonly referred to as three-way data, instances typically arise in the form of image data, where the elements of each matrix observation represent some value for each pixel of an image, or longitudinal data, where rows of the matrix correspond to multivariate observations of on a particular subject, and columns represent the change of a particular covariate of interest over time (or vice versa). As with any dataset, it is often of interest to assess the presence of a latent grouping structure in matrix data, as this can often give deeper insight into the patterns and relations inherent therein. For example, clustering of image data may help identify images with similar palette, or representing a similar scene. In the longitudinal case, it may help us find groups of patients who should receive similar treatments, or are exhibiting similar health trends.

The first contribution to model-based clustering of matrix-variate data is provided by Basford and McLachlan (1985). Since then, and in the last decade in particular, attention to developing model-based approaches for clustering matrix-valued data has increased, to great results. Viroli (2011a) derived a general model-based approach under the assumption that the data arise from a finite mixture of matrix normal distributions, as defined in Dawid (1981). Following this, Viroli (2011b) introduced a Bayesian equivalent. Continuing this line of research, Dogru et al. (2016) introduced an analogous extension of the t-distribution to matrix variate data. Concerned with a lack of modelling options for three-way data that provide elliptical distributions with heavier tails than the Gaussian distribution, Tomarchio et al. (2020) introduces two additional distributions for three-way data in the matrix-variate shifted exponential normal (MVSEN) and the matrix-variate tail-inflated normal (MVTIN), which satisfy these conditions. Tomarchio et al. (2021) extend cluster-weighted models to the matrix context, by extending them to work with matrix-variae regression models. Skewness is again considered in the work Melnykov and Zhu (2018a), which proposes to handle skewness in matrix-variate data by assuming a transform to approximate normality exists. Parsimony is inherited by modelling the spectral decomposition of the associated component covariance matrices, or underlying ARMA assumptions regarding the temporal domain (when applicable). The efficacy of this model is demonstrated in Melnykov and Zhu (2018b) through the analysis of crime data. The work of Sarkar et al. (2019) further extends these ideas by introducing parsimony in each of the matrix-normal covariance matrices using the ideas presented in Banfield and Raftery (1993). Gallaugher and McNicholas (2018) also fosters models that account for skewness, and implement this by introducing finite mixture models of matrix variate skew-t and generalized hyperbolic, among others. As one can imagine, the specification of matrix variate distributions results in an abundance of model parameters. This prompts Gallaugher and McNicholas (2019) to investigate parsimonious modelling of matrix variate data through a generalization of the mixture of factor analyzers Ghahramani and Hinton (1996) to matrix variate data.

In a similar vein to this last approach, we propose a parsimonious model for clustering of three-way matrix by extending the subspace clustering approach detailed in Bouveyron et al. (2007) to a finite mixture of matrix normal distributions through a dual-subspace projection. The remainder of the paper proceeds as follows: in Section 4.2, we give a brief overview of the matrix normal distribution and finite mixture models. In Section 4.3.1 we discuss the dual-subspace perspective that serves as the foundation for the proposed model. Section 4.3.3 details our method for parameter estimation, which comprises an Expectation Conditional Maximization algorithm, in which all updates are presented in closed form. Finally, Section 4.3.4 discusses good initialization strategies for the model, including a modification to an algorithm introduced in Bouveyron et al. (2007) for automatically choosing the hyperparameters corresponding to the dimension of the latent data subspaces. In Section 4.4 we demonstrate the parameter recovery and model selection through simulation. We also present two data analyses which demonstrate the capacities of the dual-subspace approach. Section 4.5 ends the chapter with a short summary and some thoughts on future research.

4.2 Background

4.2.1 Matrix Normal Distribution

A $p_1 \times p_2$ matrix $\mathbf{X}_{p_1 \times p_2}$ is said to have a *matrix normal distribution* Dawid (1981) with mean matrix $\mathbf{M}$, and co-variance matrices $\mathbf{\Sigma}_1$, $\mathbf{\Sigma}_2$ if its pdf can be expressed as

$$f(\mathbf{X} \mid \mathbf{M}, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2) = \frac{\exp\left(-\frac{1}{2}\operatorname{tr}\left[\mathbf{\Sigma}_1^{-1}(\mathbf{X}-\mathbf{M})\mathbf{\Sigma}_2^{-1}(\mathbf{X}-\mathbf{M})^T\right]\right)}{(2\pi)^{p_1p_2/2}|\mathbf{\Sigma}_1|^{p_2/2}|\mathbf{\Sigma}_2|^{p_1/2}}. \tag{4.1}$$

When a random variable $\mathbf{X}$ is distributed according to a matrix normal distribution, we denote it as

$$\mathbf{X} \sim MN_{p_1 \times p_2}(\mathbf{M}, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2).$$

The matrix normal distribution is in fact a special form of the multivariate normal, and arises when the co-variance matrix can be decomposed as the kronecker product of two matrices Srivastava et al. (2008). That is,

$$\mathbf{X} \sim MN_{p_1 \times p_2}(\mathbf{M}, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2) \quad \text{iff} \quad \text{vec}(\mathbf{X}) \sim N_{p_1 p_2}(\text{vec}(\mathbf{M}), \mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1).$$

In situations where this covariance structure assumption is met, employing the matrix normal distribution allows for parsimonious modelling, with the number of effective parameters decreasing from,

$$p_1 p_2 + (p_1 p_2)(p_1 p_2 + 1)/2 \quad \text{to} \quad p_1 p_2 + p_1(p_1 + 1)/2 + p_2(p_2 + 1)/2 - 1,$$

which is, however, still quadratic as a function of both p_1 and p_2. It follows that imposing the matrix normal structure on three-way data results in modest model parsimony.

Note that the parameterization of the distribution given in Equation (4.1) implies that we have,

$$E[(\mathbf{X} - \mathbf{M})(\mathbf{X} - \mathbf{M})^T] = \mathbf{\Sigma}_1 \, \text{tr}(\mathbf{\Sigma}_2) \quad \text{and,} \quad E[(\mathbf{X} - \mathbf{M})^T(\mathbf{X} - \mathbf{M})] = \mathbf{\Sigma}_2 \, \text{tr}(\mathbf{\Sigma}_1).$$

In particular, this covariance structure specifies the covariance between rows i and j to be,

$$\text{Cov}\Big(\mathbf{X}_{i\cdot}, \mathbf{X}_{j\cdot}\Big) = \mathbf{\Sigma}_{1ij}\mathbf{\Sigma}_2,$$

while the covariance between columns i and j is specified to be,

$$\text{Cov}\Big(\mathbf{X}_{\cdot i}, \mathbf{X}_{\cdot j}\Big) = \mathbf{\Sigma}_{2ij}\mathbf{\Sigma}_1.$$

Hence, we call $\mathbf{\Sigma}_1$ the "across-column" covariance, while $\mathbf{\Sigma}_2$ is the "across-row" covariance. In light of this structure, we observe that the matrix normal is especially well-suited to the analysis of data for which the rows (columns) are related apriori. We reiterate that repeated measures data Srivastava et al. (2008) is one instance where this structure arises naturally.

4.2.2 Identifiability

A matrix normal model for data, as specified in equation (4.1), is not identifiable. Given $\alpha \neq 0$, the substitutions $\mathbf{\Sigma}_1^* = \alpha\mathbf{\Sigma}_1$, and $\mathbf{\Sigma}_2^* = \frac{1}{\alpha}\mathbf{\Sigma}_2$ leave the distribution unchanged. This identifiability problem for the matrix normal distribution can be addressed using Glanz and Carvalho (2013), which proposes to alleviate the problem through the introduction of a general scale parameter σ^2. We henceforth substitute,

$$\mathbf{\Sigma}_i = \sigma^2 \mathbf{\Psi}_i, \quad i = 1, 2, \tag{4.2}$$

into equation (4.1), where the matrix $\mathbf{\Psi}_i$ is specified to have unit determinant. This allows us to write $\mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1 = \sigma^2(\mathbf{\Psi}_2 \otimes \mathbf{\Psi}_1)$, which results in identifiability of the model parameters.

4.3 Methodology

Utilizing the results of Glanz and Carvalho (2013), the model density is expressed as,

$$f(\mathbf{X} \mid \mathbf{M}, \mathbf{\Psi}_1, \mathbf{\Psi}_2, \sigma^2) = \frac{\exp\left(-\frac{1}{2\sigma^2} \operatorname{tr}\left[\mathbf{\Psi}_1^{-1}(\mathbf{X}-\mathbf{M})\mathbf{\Psi}_2^{-1}(\mathbf{X}-\mathbf{M})^T\right]\right)}{(2\pi\sigma^2)^{p_1 p_2/2}|\mathbf{\Psi}_1|^{p_2/2}|\mathbf{\Psi}_2|^{p_1/2}}. \tag{4.3}$$

Let $I = \{1, 2\}$ be the index set of the model parameter subscripts. For the remainder of this section, all statements made in terms of the index variable i implicitly hold for all $i \in I$.

4.3.1 A Matrix Normal Model for High Dimensional Data

Suppose $\mathbf{X}_{p_1 \times p_2}$ is a random matrix variable having Lebesgue density given by equation (4.3). That is,

$$\mathbf{X} \sim MN_{p_1 \times p_2}(\mathbf{M}, \mathbf{\Psi}_1, \mathbf{\Psi}_2, \sigma^2).$$

Let $\mathbf{\Lambda}_1$ be an orthogonal matrix with dimension p_1. Defining $\dot{\mathbf{X}} := \mathbf{\Lambda}_1^\top \mathbf{X}$, we find that $\dot{\mathbf{X}}$ is again matrix normal distributed, and can be specified by,

$$\dot{\mathbf{X}} \sim MN_{p_1 \times p_2}(\mathbf{\Lambda}_1^\top \mathbf{M}, \mathbf{\Lambda}_1^\top \mathbf{\Psi}_1 \mathbf{\Lambda}_1, \mathbf{\Psi}_2, \sigma^2).$$

Note that this transformation has changed the mean parameter $\mathbf{M}$ and the normalized “across-column" covariance $\mathbf{\Psi}_1$, but left the “across-row" covariance and scale parameter untouched. We now assume that there exists a particular $\mathbf{\Lambda}_1$ such that the "across-column" covariance becomes a diagonal matrix with the form,

$$\mathbf{\Lambda}_1^\top \mathbf{\Psi}_1 \mathbf{\Lambda}_1 = \begin{bmatrix} \mathbf{\Phi}_1 & \\ & \eta_1 \mathbf{I}_{p_1 - q_1} \end{bmatrix} =: \mathbf{\Delta}_1. \tag{4.4}$$

That is, with the column space of $\mathbf{X}$ transformed by $\mathbf{\Lambda}_1^\top$, the "across-column" covariance matrix has become diagonal, with the first q_1 elements given by $\mathbf{\Phi}_1 = \text{diag}(\phi_1, ..., \phi_{q_1})$, and the remaining $p_1 - q_1$ elements given by a single value η_1, which is smaller in magnitude than any element of $\mathbf{\Phi}_1$. Under such a transformation, the column covariances become,

$$\text{Cov}\big(\dot{\mathbf{X}}_{\cdot j}, \dot{\mathbf{X}}_{\cdot k}\big) = \mathbf{\Psi}_{2jk} \mathbf{\Delta}_1. \tag{4.5}$$

Note that $\sigma^2 = 1$ has been implicitly assumed without loss of generality. This assumption continues throughout the remainder of this discussion. Intuitively, Equation (4.5) says that the columns of $\mathbf{X}$ are well approximated by projection into an affine space parallel to the subspace spanned by the first q_1 column vectors of $\mathbf{\Lambda}_1^\top$. Further, outside of this space, the covariance of the columns is spherical.

The assumptions on $\mathbf{\Psi}_1$ also imposes a parsimonious structure on the marginal covariances of the rows of $\dot{\mathbf{X}}$ in the following form,

$$\text{Cov}\big(\dot{\mathbf{X}}_{j \cdot}, \dot{\mathbf{X}}_{k \cdot}\big) = \mathbf{\Delta}_{1jk} \mathbf{\Psi}_2 = \begin{cases} \phi_{1jj} \mathbf{\Psi}_2 & \text{if } j = k, \text{ and } j \leq q_1 \\ \eta_1 \mathbf{\Psi}_2 & \text{if } j = k, \text{ and } j > q_1 \\ \mathbf{0}_{p_1 \times p_2} & \text{if } j \neq k. \end{cases} \tag{4.6}$$

We see that under the constraints imposed on $\mathbf{\Delta}_1$ the final $p_1 - q_1$ rows of $\dot{\mathbf{X}}$ now share a common covariance matrix. Note that this form is not dependent on the original order of the rows (columns) of $\mathbf{X}$, as permutation matrices are also orthogonal.

We may also consider $\mathbf{\Psi}_2$ in the same way. Suppose $\mathbf{\Lambda}_2$ is an orthogonal matrix of dimension p_2. Then $\ddot{\mathbf{X}} := \mathbf{X}\mathbf{\Lambda}_2$ follows a matrix normal distribution, with parameter specification given by,

$$\ddot{\mathbf{X}} \sim MN_{p_1 \times p_2}(\mathbf{M}\mathbf{\Lambda}_2, \mathbf{\Psi}_1, \mathbf{\Lambda}_2^\top \mathbf{\Psi}_2 \mathbf{\Lambda}_2, \sigma^2).$$

Noting that $\ddot{\mathbf{X}}^\top = \mathbf{\Lambda}_2^\top \mathbf{X}^\top$, we see that equation (4.5) now applies to the rows of $\ddot{\mathbf{X}}$ rather than the columns, while equation (4.6) holds for the columns of $\ddot{\mathbf{X}}$ rather than the rows. Finally, $\mathbf{\Lambda}_1^\top \mathbf{X} \mathbf{\Lambda}_2$ is easily seen to be matrix normal as well, and thus we may combine these two subspace projection approaches to generate model parsimony through a dual-subspace projection model specification.

4.3.2 Specification of the Model Likelihood

Following the preceeding discussion, the parameter $\boldsymbol{\Psi}_i$ admits an eigen decomposition, which we denote by $\boldsymbol{\Psi}_i = \mathbf{\Lambda}_i \mathbf{\Delta}_i \mathbf{\Lambda}_i^T$. We suppose $\mathbf{\Delta}_i$ has the form specified in Equation (4.4), so that we may write,

$$\boldsymbol{\Psi}_i = [\mathbf{\Gamma}_i, \mathbf{\Xi}_i] \begin{bmatrix} \mathbf{\Phi}_i & \\ & \eta_i \, \mathbf{I}_{p_i - q_i} \end{bmatrix} [\mathbf{\Gamma}_i, \mathbf{\Xi}_i]^T \, , \tag{4.7}$$

where $\mathbf{\Phi}_i = \operatorname{diag}(\boldsymbol{\phi}_i) = \operatorname{diag}(\phi_{i1}, \ldots, \phi_{iq_i})$, $\mathbf{\Gamma}_i$ is $p_i \times q_i$, and the eigenvalue matrix $\mathbf{\Delta}_i$ is,

$$\mathbf{\Delta}_i = \operatorname{diag}(\boldsymbol{\phi}_i, \eta_i \mathbf{I}_{p_i - q_i}) \, .$$

Defining $\boldsymbol{\theta} := (\mathbf{M}, \sigma^2, \mathbf{\Lambda}_1, \mathbf{\Delta}_1, \mathbf{\Lambda}_2, \mathbf{\Delta}_2, \eta_1, \eta_2)$ the log-likelihood for a single observation can be written,

$$\begin{aligned} \ell(\mathbf{X} \mid \boldsymbol{\theta}) = &- \frac{p_1 p_2}{2} \log 2\pi - \frac{p_1 p_2}{2} \log \sigma^2 - \frac{p_1}{2} \Big(\log|\mathbf{\Phi}_1| + (p_1 - q_1) \log \eta_1 \Big) - \\ & \frac{p_2}{2} \Big(\log|\mathbf{\Phi}_2| + (p_2 - q_2) \log \eta_2 \Big) - \frac{1}{2\sigma^2} \operatorname{tr} \Big[\boldsymbol{\Psi}_1^{-1} \mathbf{R} \boldsymbol{\Psi}_2^{-1} \mathbf{R}^T \Big], \end{aligned} \tag{4.8}$$

where we have used $\mathbf{R}$ to denote the centered observation $\mathbf{X} - \mathbf{M}$. Two constraints are imposed on the model, namely,

$$\det(\mathbf{\Delta}_i) = 1 \quad \text{ and } \quad \mathbf{\Gamma}_i^T \mathbf{\Gamma}_i \ = \mathbf{I}_{p_i}.$$

As discussed in Section 4.2.2, the determinant constraint ensures parameter identifiability, while the eigenvector constraint is the usual one.

Looking at the log-likelihood provides some insight into what exactly our model implies. Specifically, it becomes evident that while our model significantly reduces parameter count, it does not do so by discarding much discriminatory information. While it may assume that the $\boldsymbol{\Delta}_i$ are equally spread along their last $(p_i - q_i)$ principal components, our model fully considers those directions.

With this new assumption the model now contains $p_1 p_2 + \sum_{i=1}^{2} q_i\big(p_i - (q_i - 1)/2\big) + 2$ parameters. When compared against the fully parameterized model, we see that the total number of model parameters has been reduced from a quadratic function of the data dimensions, to a linear one.

4.3.3 A Parsimonious Mixture of Matrix Normal Distributions

Let $S = \{\mathbf{X}_j\}^n$ be a sample of size n drawn from a mixture of matrix normal distributions,

$$p(\mathbf{X}_j) = \sum_{g=1}^{G} \pi_g f\big(\mathbf{X}_j \mid \boldsymbol{\theta}_g\big)$$

where each distribution in the mixture has covariance structure as in Equation (4.7). For a sample S of size n drawn from such a model, the observed likelihood has the form,

$$\mathcal{L}(\boldsymbol{\Theta}; S) = \prod_{j=1}^{n} \sum_{g=1}^{G} \pi_g f\big(\mathbf{X}_j \mid \boldsymbol{\theta}_g\big), \tag{4.9}$$

with $\boldsymbol{\Theta} := (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_g)$. To proceed with estimation, we implement an Expectation-Maximization (EM) algorithm Dempster et al. (1977). This allows the accommodation of a latent unobserved random variables, denoted by $\mathbf{Z} = Z_{jg}$, indicating group membership of the individual observations. Under this assumption, the complete data likelihood, which includes these new latent variables, can then be written as a product of a product, from which we get the complete data log-likelihood,

$$\ell_c(\boldsymbol{\Theta}; S) = \sum_{j=1}^{n} \sum_{g=1}^{G} Z_{jg} \Big[\log \pi_g + \log f\big(\mathbf{X}_j \mid \boldsymbol{\theta}_g\big) \Big]. \tag{4.10}$$

The EM algorithm maximizes Equation (4.9) through two-step, iterative maximization of Equation (4.10). The first step is the Expectation step, or E-step, and consists of replacing the latent variable with its conditional expected value given the data and current parameter values,

$$z_{jg} = \frac{\pi_g f\left(\mathbf{X}_j \mid \boldsymbol{\theta}_g\right)}{\sum_{k=1}^{G} \pi_k f\left(\mathbf{X}_j \mid \boldsymbol{\theta}_k\right)}.$$

Plugging these estimates into Equation (4.10), we get the expected complete-data log-likelihood,

$$\begin{aligned} Q(\boldsymbol{\Theta}; S) = \frac{1}{2} \sum_{g=1}^{G} \Bigg[& n_g \log \pi_g - n_g p_1 p_2 \log \sigma_g^2 - n_g p_2 \log |\boldsymbol{\Psi}_{1g}| \\ & - n_g p_1 \log |\boldsymbol{\Psi}_{2g}| - \sigma^{-2} \mathrm{tr}\Big\{ \mathbf{W}_{2g} \boldsymbol{\Psi}_{1g}^{-1} \Big\} \Bigg] + C \end{aligned} \tag{4.11}$$

which can equivalently be written,

$$\begin{aligned} Q(\boldsymbol{\Theta}; S) = \frac{1}{2} \sum_{g=1}^{G} \Bigg[& n_g \log \pi_g - n_g p_1 p_2 \log \sigma_g^2 - n_g p_2 \log |\boldsymbol{\Psi}_{1g}| \\ & - n_g p_1 \log |\boldsymbol{\Psi}_{2g}| - \sigma^{-2} \mathrm{tr}\Big\{ \mathbf{W}_{1g} \boldsymbol{\Psi}_{2g}^{-1} \Big\} \Bigg] + C, \end{aligned} \tag{4.12}$$

where we have defined $W_{1g} = \sum_{j=1}^{n} z_{jg} \mathbf{R}_{jg}^T \boldsymbol{\Psi}_{1g}^{-1} \mathbf{R}_{jg}$ and $W_{2g} = \sum_{j=1}^{n} z_{jg} \mathbf{R}_{jg} \boldsymbol{\Psi}_{2g}^{-1} \mathbf{R}_{jg}^T$, as well as $n_g = \sum_{j=1}^{n} z_{jg}$ and $\mathbf{R}_{jg} = \mathbf{X}_j - \mathbf{M}_g$. Additionally, C represents the sum over terms not involving model parameters.

Constructing the expected complete-data log-likelihood completes the E-step, and we now turn our attention to the Maximization step, or M-step. In this step, the expected complete-data log-likelihood is maximized with respect the model parameters. Some of the updates for our model depend on the values of other model parameters, so that our algorithm falls under the Expectation Conditional Maximization framework Meng and Rubin (1993). Algorithms of this kind still retain the same monotonicity and convergence guarantees as vanilla EM.

Updates are presented in the order in which they are calculated by the algorithm, and we require an update for each group $g = 1, 2, ..., G$. Beginning with π_g, maximization of

$Q(\boldsymbol{\Theta}; S)$ under the condition $\sum_{g=1}^{G} \pi_g = 1$ yields the update,

$$\hat{\pi}_g = \frac{1}{n} \sum_{j=1}^{n} z_{jg} = \frac{n_g}{n}.$$

Next, maximization with respect to the group mean $\mathbf{M}_g$ gives,

$$\widehat{\mathbf{M}}_g = \frac{\sum_{j=1}^{n} z_{jg} \mathbf{X}_j}{n_g},$$

as an update for the group means. To update the eigenvalues $\boldsymbol{\Delta}_{1g}$, we take the derivative of Equation (4.11) with respect to $\boldsymbol{\Delta}_{1g}$ under the constraint $|\boldsymbol{\Delta}_{1g}| = 1$, yielding,

$$\hat{\boldsymbol{\Delta}}_{1g} = \Big| \operatorname{diag}\Big\{ \boldsymbol{\Lambda}_{1g}^{T} \mathbf{W}_{2g} \boldsymbol{\Lambda}_{1g} \Big\} \Big|^{-1/p_1} \operatorname{diag}\Big\{ \boldsymbol{\Lambda}_{1g}^{T} \mathbf{W}_{2g} \boldsymbol{\Lambda}_{1g} \Big\}.$$

Denoting $c_{1g} = \big| \operatorname{diag}\{ \boldsymbol{\Lambda}_{1g}^{T} \mathbf{W}_{2g} \boldsymbol{\Lambda}_{1g} \} \big|^{-1/p_1}$, our updates for the model “across-column" covariance eigenvalues are then,

$$\hat{\boldsymbol{\Phi}}_{1g} = c_{1g} \operatorname{diag}\Big\{ \boldsymbol{\Gamma}_{1g}^{T} \mathbf{W}_{2g} \boldsymbol{\Gamma}_{1g} \Big\}, \quad \text{and} \quad \hat{\eta}_{1g} = \frac{c_{1g}}{p_1 - q_{1g}} \operatorname{tr}\Big\{ \mathbf{W}_{2g} (\mathbf{I}_{p_1} - \boldsymbol{\Gamma}_{1g} \boldsymbol{\Gamma}_{1g}^{T}) \Big\}.$$

Following a similar procedure with Equation (4.12), our updates for the parameters comprising $\boldsymbol{\Delta}_{2g}$ are given by,

$$\hat{\boldsymbol{\Phi}}_{2g} = c_{2g} \operatorname{diag}\Big\{ \boldsymbol{\Gamma}_{2g}^{T} \mathbf{W}_{1g} \boldsymbol{\Gamma}_{2g} \Big\}, \quad \text{and} \quad \hat{\eta}_{2g} = \frac{c_{2g}}{p_2 - q_{2g}} \operatorname{tr}\Big\{ \mathbf{W}_{1g} (\mathbf{I}_{p_2} - \boldsymbol{\Gamma}_{2g} \boldsymbol{\Gamma}_{2g}^{T}) \Big\},$$

where we have defined $c_{2g} = \big| \operatorname{diag}\{ \boldsymbol{\Lambda}_{2g}^{T} \mathbf{W}_{1g} \boldsymbol{\Lambda}_{2g} \} \big|^{-1/p_2}$.

To derive an update for the eigenvectors, $\boldsymbol{\Gamma}_{1g}$, we note that the objective function Q only depends on $\boldsymbol{\Gamma}_{1g}$ through the trace term. Thus maximization of Q with respect to $\boldsymbol{\Gamma}_{1g}$ is equivalent to solving,

$$\hat{\boldsymbol{\Gamma}}_{1g} = \min_{\boldsymbol{\Gamma}_{1g}} \operatorname{tr}\Big\{ \mathbf{W}_{2g} \boldsymbol{\Gamma}_{1g} \Big(\boldsymbol{\Phi}_{1g}^{-1} - \eta_{1g}^{-1} \mathbf{I}_{p_1} \Big) \boldsymbol{\Gamma}_{1g}^{\top} \Big\}. \tag{4.13}$$

To perform this optimization, we utilize the property that $\mathbf{W}_{2g}$ is a positive definite matrix. Making the substitution, $\mathbf{W}_{2g} := \mathbf{A}_{2g} \mathbf{D}_{2g} \mathbf{A}_{2g}^{\top}$, the optimization problem becomes,

$$\hat{\boldsymbol{\Gamma}}_{1g} = \min_{\boldsymbol{\Gamma}_{1g}} \operatorname{tr}\Big\{ \boldsymbol{\Gamma}_{1g}^{\top} \mathbf{A}_{2g} \mathbf{D}_{2g} \mathbf{A}_{2g}^{\top} \boldsymbol{\Gamma}_{1g} \Big(\boldsymbol{\Phi}_{1g}^{-1} - \eta_{1g}^{-1} \mathbf{I}_{p_1} \Big) \Big\}.$$

The problem is now solved by realizing $\mathbf{\Phi}_{1g}^{-1} - \eta_{1g}^{-1}\mathbf{I}_{p_1}$ is negative definite. Optimization then occurs when $\mathbf{\Gamma}_{1g}^{\top}\mathbf{A}_{2g}\mathbf{D}_{2g}\mathbf{A}_{2g}^{\top}\mathbf{\Gamma}_{1g}$ is as large as possible, a condition satisfied when $\mathbf{\Gamma}_{1g}$ comprises the q_{1g} columns of $\mathbf{A}_{2g}$ corresponding to the q_{1g} largest eigenvalues of $\mathbf{W}_{2g}$. Similarly, $\hat{\mathbf{\Gamma}}_{2g}$ is found to be the first q_{2g} eigenvectors in the spectral decomposition of $\mathbf{W}_{1g}$.

Finally, direct maximization with respect to the scaling parameter σ_g^2 results in the update,

$$\widehat{\sigma}^2 = \frac{1}{n_g p_1 p_2} \sum_{j=1}^{n} \operatorname{tr}\Big\{\mathbf{W}_{2g}\mathbf{\Psi}_{1g}^{-1}\Big\} = \frac{1}{n_g p_1 p_2} \sum_{j=1}^{n} \operatorname{tr}\Big\{\mathbf{W}_{1g}\mathbf{\Psi}_{2g}^{-1}\Big\}.$$

updating this parameter then completes the M-step.

4.3.4 Initialization Strategies, Convergence Criteria, and Hyperparameters

Our optimization algorithm is a version of the Expectation-Maximization algorithm, and as such requires parameter initialization. We implement multiple initialization strategies, most of which are some variation of the *em*EM approach detailed in Biernacki et al. (2003). Generally, this approach works by choosing a relatively large number of starting parameter values and performing a small number of EM iterations for each of them. The best performing set of starting parameters, measured by highest achieved likelihood value, is then chosen to initialize the EM algorithm, which is then run until convergence. Choosing the starting parameters for this initialization strategy can be done in multiple ways. We implement k-means, parsimonious Gaussian clustering Browne and Mcnicholas (2014), hierarchical clustering, random hard, and random soft cluster assignments for this purpose.

Convergence of the algorithm is assessed using an Aitken acceleration-based convergence criterion, which depends on the linear convergence rate of EM. At the k^{th} iteration, the estimate of the limit is given by,

$$\ell_{\infty}^{(k)} = \ell^{(k)} + \frac{\ell^{(k+1)} - \ell^{(k)}}{1 - a^{(k)}} \quad \text{where} \quad a^{(k)} = \frac{\ell^{(k+1)} - \ell^{(k)}}{\ell^{(k)} - \ell^{(k-1)}}.$$

For a chosen tolerance $\varepsilon > 0$, we stop the algorithm when the difference $\ell_\infty^{(k)} - \ell_\infty^{(k-1)}$ falls below this tolerance.

All that remains is to discuss choosing values for the model's hyperparameters—the number of groups G, and the intrinsic subspace dimensions $q = \{(q_{1g}, q_{2g})\}_{g=1}^{G}$. Both can be chosen simultaneously by searching the space of all possible combinations of values for G and q, and picking the best model according to an appropriate model selection criterion. The Bayesian Information Criterion (BIC, Schwarz (1978)) is a commonly used criterion, and we implement it here. For a fitted model $\mathcal{M}$, the corresponding BIC is calculated as,

$$BIC(\mathcal{M}) = k \log n - 2 \log(\widehat{\mathcal{L}_\mathcal{M}}),$$

where $\widehat{\mathcal{L}}_\mathcal{M}$ represents the likelihood evaluated at the parameter values fitted by $\mathcal{M}$, n is the number of data points, and k is the number of free parameters.

One issue with an exhaustive search approach to hyperparameter optimization is that it scales quite poorly with data dimension. Supposing we wish to fit a G-component mixture model, and given that the data have p_1 row, and p_2 column dimensions, there are then $(p_1 p_2)^G$ unique ways to choose the intrinsic subspace dimensions. This number is large, even for modest values of the involved parameters, hence fitting all possible combinations in such a case is an infeasible approach.

To combat this issue, we adapt a method developed in Bouveyron et al. (2007). Here, the authors present a way to estimate the intrinsic subspace dimensions q when each group is projected into the eigenspace of the associated covariance matrix. By estimating q from the data, one skirts the requisite computational burden of exhaustively searching the space of all possible q's. The proposed estimation method for q finds the full set of eigenvalues for each component's covariance matrix, and then, implementing some form of thresholding, chooses the number of "significant" eigenvalues. This number then serves as an estimate of q_g, the dimension of that component's intrinsic subspace. In the case of our model, this procedure requires decomposing both $\mathbf{W}_{1g}$ and $\mathbf{W}_{2g}$, and then applying a threshold condition to each set of computed eigenvalues accordingly.

The threshold condition for determining significance can be implemented in different ways, and is somewhat arbitrary. Bouveyron et al. (2007) suggest one possible way is to take the sequential difference of the sorted (descending) eigenvalues and choose a cutoff value below which the normalized differences can be considered small. We opt for a different approach in our implementation. We instead use the proportion of total variance explained by the retained eigenvalues as a threshold. For example, if we choose a threshold of 0.75, then our estimate for q_{ig} would become the smallest integer such that the sum of the retained eigenvalues exceeds 75% of the total variance in the data, $i = 1, 2$ and $g = 1, ..., G$. A threshold value that gives good performance can then be chosen using BIC.

4.4 Numerical Demonstrations

4.4.1 Parameter Recovery

We now demonstrate the model's ability to recover parameter values under an appropriate data-generation scheme. Our simulation proceeds as follows. We generate a random true parameter set, denoted by $\boldsymbol{\theta}$, from which we proceed to generate $m = 1000$ datasets according to the model details given momentarily. For each dataset, we apply the proposed clustering algorithm. Performance is assessed through a measure of proximity between the estimated parameter values of the model and the true parameter vector $\boldsymbol{\theta}$. Owing to the multiple mixture components and the matrix structure of the data, our model admits many parameters even in fairly simple specifications. Displaying individual recovery results in a set of tables is therefore impractical. Instead, we report the distribution of the estimated mean squared error (MSE) in estimation, which we compute as,

$$\widehat{\text{MSE}} = L^{-1} \parallel \hat{\boldsymbol{\theta}} - \boldsymbol{\theta} \parallel_2^2,$$

where $\hat{\boldsymbol{\theta}}$ is the vectorized set of parameter estimates given by the model, L is the number of entries in the vector $\hat{\boldsymbol{\theta}}$, and $\parallel \cdot \parallel_2$ is the Euclidean norm. To account for the possibility of label switching, we calculate this estimate of the MSE in both possible ways, and then choose the smallest one. The model is said to be recovering the true parameters if the estimated MSE trends to zero as the sample size increases. The simulation method above is applied in three scenarios, corresponding to increasing data dimension. That is, (p_1, p_2) is chosen from one of $(6, 10)$, $(10, 20)$, and $(20, 20)$, which correspond to scenario 1, scenario

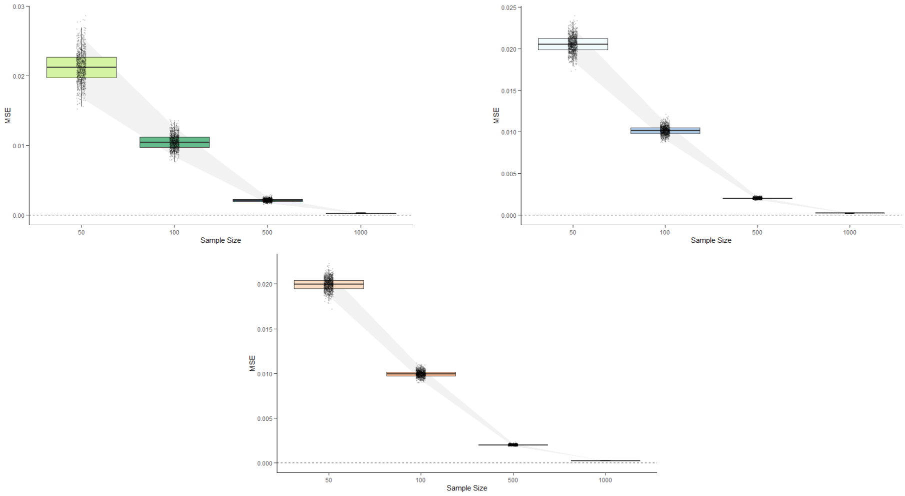


Figure 4.1: Results of parameter recovery simulations. Top left corresponds to p=(6,10), top right corresponds to (10,20), bottom corresponds to (20,20).

2, and scenario 3 respectively. For each simulation scenario we generate data from a four-component mixture model. For each set of components, we specify the associated dual-subspace dimensions (q_{1g}, q_{2g}) to be $(2, 2)$ for component 1, $(2, 3)$ for component 2, $(3, 3)$ for component 3, and $(2, 4)$ for component 4. Finally, we run each scenario for four chosen sample sizes, $N = 50, 100, 500$, and 1000.

Figure 4.1 shows the change in the distribution of the MSE as the sample size increases, for scenario 1 (data dimension 6×10). We see that as the sample size increases both the mean and variance of the distribution of MSE values approaches 0, exactly as we would expect. The plots of the other two scenarios are nearly identical to Figure 4.1, and so are omitted from the main text. Indeed, this shows that our model is effective at recovering the true model parameters when the data have been generated according to the underlying assumptions of matrix normality. Additionally, we note that in all cases, perfect clustering results were achieved.

4.4.2 Model Selection

The viability of the Bayesian Information Criterion (BIC) for choosing the number of components in a finite mixture model is well documented (Aitkin and Rubin (1985), Roeder and Wasserman (1997), Fraley and Raftery (1998), Keribin (2000)). Often BIC is used primarily for this purpose, however here we propose to use it for choosing the latent subspace dimensions as well, via the thresholding method of Bouveyron et al. (2007) discussed in Section 4.3.4. We investigate the viability of the BIC in this scenario empirically through a simple simulation study. For the simulation we generate one true parameter set, which is then manipulated to generate different scenarios for study. The true parameter set parameterizes a four-component mixture model which generates matrix observations of dimension 5×5. The latent subspace dimensions of each component is set to two ($q_{1g} = q_{2g} = 2$), while η_g is chosen so that the proportion of variance explained by the first two principal components is 0.57 in each group. The exact parameter values can be found in Appendix A.1. The means are initially chosen so that the clusters have some overlap, and we push the means further away as we progress through scenarios. That is, we run $k = 4$ scenarios, and in each scenario we set,

$$\mathbf{M}_g^\star = k\mathbf{M}_g,$$

where $\mathbf{M}_g^\star$ is the specified mean matrix of group g, and $\mathbf{M}_g$ is the mean matrix for group g in the original true parameter set. For each value of k, we use the specified parameters to generate a random dataset, which we then fit our model to. For each dataset, we fit the model using all combinations of $\mathcal{G} = \{1, 2, 3, 4, 5, 6, 7\}$ for the components and $\mathcal{T} = \{0.50, 0.55, ..., 0.95\}$ for the thresholds. Each time we run a scenario, the latent subspace for each component is determined randomly. Each of the four scenarios is run $m = 500$ times for sample sizes $N = 100$, 250, and 500. To initialize the algorithm we implement the *em*EM of Biernacki et al. (2003) using 20 starts, each performing 5 iterations of EM. The best set of parameters from this initialization is then used to run the model for 50 iterations. We also keep track of the Adjusted Rand Index (ARI, Hubert and Arabie, 1985) to see how well the best clustering found by the model matches the true labels. The results are provided in Table 4.1.

From the results presented in Table 4.1, we see that BIC does a good job in all scenarios

Scenario	$\bar{G}$	Min BIC	Ave BIC	Max ARI	Min ARI	$\overline{q}_1$	$\overline{q}_2$
M							
N=100	3.96	27395.69	27565.00	0.915	0.577	2.19	2.19
N=250	4	74249.97	74579.51	0.897	0.810	2.20	2.20
N=500	4	147383.99	148151.90	0.896	0.847	2.14	2.14
2**M**							
N=100	4	23751.07	24043.80	1	0.980	2.14	2.15
N=250	4	58434.27	58908.40	1	0.990	2.10	2.11
N=500	4	116480.21	116814.00	1	0.992	2.14	2.14
3**M**							
N=100	4	19855.61	20168.00	1	1	2.23	2.19
N=250	4.04	48783.63	49189.61	1	0.914	2.25	2.22
N=500	4.00	96594.29	97279.02	1	0.999	2.20	2.20
4**M**							
N=100	4	16323.90	16623.93	1	1	2.22	2.20
N=250	4	39942.95	40299.08	1	1	2.23	2.23
N=500	4.04	78618.17	79406.90	1	1	2.22	2.19

Table 4.1: Group selection results for the chosen scenarios. Number of model components and dimension of latent subspace seem to be well estimated using BIC.

at choosing the correct number of groups, faltering only occasionally. The column Min BIC refers to the minimum value of BIC achieved across all 500 runs of the specified scenario. The closeness of this value to the that of the column Ave BIC, which reports the average BIC value across all runs, reflects consistency in the model fitting approach. Pair this consistency with the consistently large maximum ARI value and correct hyperparameter values, and we see that the model is, in some sense, recovering the true latent model. The relatively weak minimum ARI values and decreasing maximum ARI values of the first scenario are a result of component overlap which we alluded to earlier. The columns $\overline{q}_1$ and $\overline{q}_2$ average the value of the latent subspace dimensions across all 500 runs, and all components fit in each of those runs. With a true value of 2 in every case, it seems that the thresholding method of Bouveyron et al. (2007) along with BIC seems to perform well at choosing the latent subspace dimension as well. The performance of this method, however, does not seem to improve with increasing N or decreasing overlap in the components of the model. Across all scenarios, when fitting a model with 4 groups and a threshold value of 0.55, the average run time for $N = 100$ across all scenarios was 31.06 seconds, while increasing to $N = 500$ we found average run time to be 120.14 seconds.

4.4.3 Data Analysis: Landsat Satellite Data

The Landsat data set is taken from the UCI Machine Learning Repository Dua and Graff (2017) and contains multi-spectral values for 3×3 pixel neighbourhoods of a satellite image. Each observation in the dataset corresponds to one 3×3 neighbourhood, and contains 4 values per pixel representing the value at that pixel for different spectral bands. In total, there are 36 values per observation which are arranged into a 4×9 matrix, where the columns correspond to a specific pixel, and rows correspond to a specific spectral band. The structure of the data is then seen to fall under the category of repeated measures data, for which the matrix normal is well-suited. Each observation is also associated with a particular class label, determined by the physical contents of the central pixel of the 3×3 neighbourhood. The classes of interest for our purposes are grey soil (n = 397), damp grey soil (n = 211) and soil with vegetation stubble (n = 237). Attempts to classify this data using unsupervised approaches have been made previously. Namely, Viroli (2011a) restrict attention to only the four measurements on the central pixel, so that each observation is now a vector rather than a matrix, and cluster the results using the `mclust` package Fraley and Raftery (2003). They report that the best model in terms of misclassification rate follows from fitting heteroskedastic components, resulting in a 0.258 MCR. Using

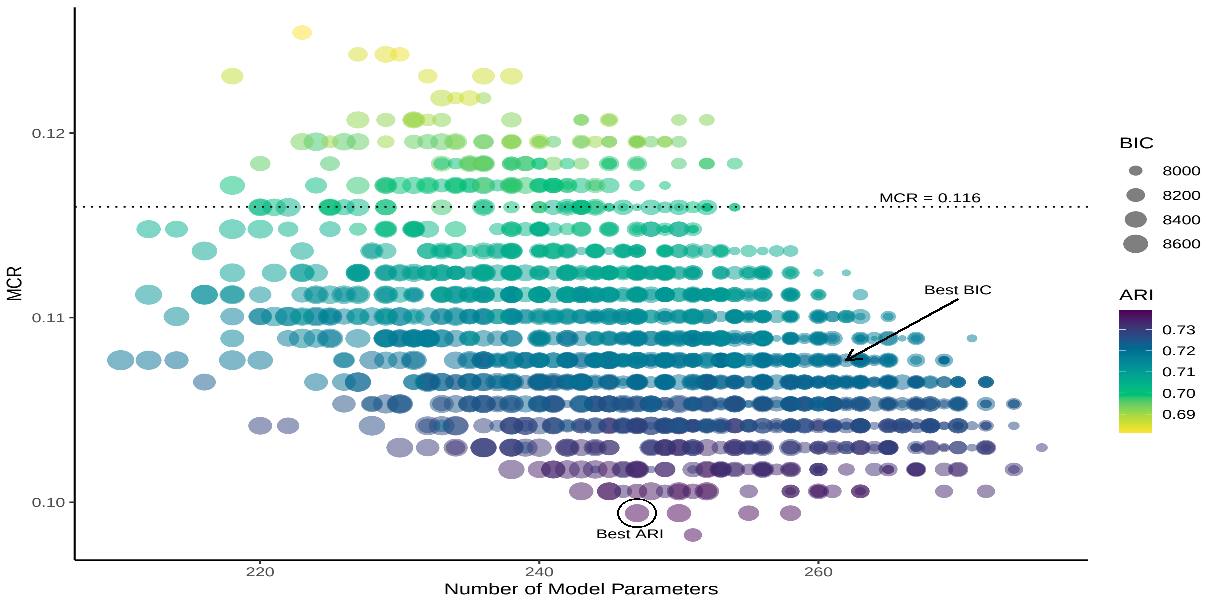


Figure 4.2: The results of all fitted models. Colors corresponds to ARI, size to BIC. The best ARI is marked with a circle, best BIC with an arrow.

the vectorized full data, so that each observation is now a 36 dimensional vector, they achieve a best MCR of 0.283. Finally, using the parsimonious mixture of matrix Gaussian distributions derived therein, Viroli (2011a) reports a best misclassification rate of 0.116, achieved using totally unconstrained model parameters (i.e. with respect to MCR, they found no advantages in parsimonious parameter specification when modelling this data). More recently, Sarkar et al. (2019) also analyzed this dataset using their matrix clustering algorithm which achieves parsimony through assumptions of common parameter values across the eigen decomposition of component “across-row" and “across-column" covariance matrices. Their best fitted model according to BIC achieves an MCR of 0.358, although they note that some models with worse BIC values attained much better clustering performances, with a best ARI of 0.72.

We proceed to model this data using the dual-subspace approach, intending to discover if these data will admit parsimony from an alternative approach. Intuition tells us that parsimony may be attainable in modelling of the rows. This idea stems from the fact that each pixel represents a large geographical area (approximately 80m×80m), as well as the fact that each row represents the same spectral band. Accordingly, for a given spectral band (row) we expect variation to be similar across pixels. This is analogous to assuming the columns reasonably represent iid samples. This assumption is not wholly unreasonable, as groups correspond to homogeneous land topography. Despite intuition, we try all possible values for the intrinsic subspace dimensions. That is, in our model fitting, we try $q_{1g} \in \{1, 2, 3\}$ for the latent subspace dimension of the columns, and $q_{2g} \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ for the latent subspace dimension of the rows. We try each possible combination of elements (q_{1g}, q_{2g}) across the three groups.

We then let BIC determine which choice of q_{1g} and q_{2g} allows the model to best represent each component g. For comparison with Viroli (2011a) we also consider the best performing models in terms of MCR. Additionally, in the interest of utilizing more contemporary metrics, we also provide the ARI (Adjusted Rand Index, Hubert and Arabie (1985)) values for each fit.

Initialization of the models is done with an *em*EM approach utilizing Gaussian parsimonious clustering and random soft classifications, each with 50 initializations.

	Latent Dimensions			MCR	ARI	BIC	Parameters
	q_1	q_2	q_3				
MCR	(2, 8)	(3, 5)	(3, 4)	**0.098**	0.738	8182.1	251
	(3, 8)	(2, 6)	(2, 3)	0.099	**0.739**	8560.0	247
	(2, 8)	(3, 5)	(2, 3)	0.101	0.737	8555.7	**243**
BIC	(3, 8)	(3, 6)	(3, 5)	0.108	0.720	**7901.0**	262
	(3, 8)	(3, 6)	(2, 6)	0.109	0.717	7901.0	264
	(3, 8)	(3, 6)	(2, 5)	0.110	0.715	7903.7	260

Table 4.2: Most parsimonious models for each of the best three MCR values, and top models by BIC, respectively. Best overall values are bolded.

The results of fitting all models are presented graphically in Figure 4.2. We see that all fitted models do an adequate job in terms of classification, coming in under the previously mentioned 0.258 MCR of the `mclust` results, and hanging competitively with the 0.116 MCR presented by the model of Viroli (2011a). We notice that the total number of model parameters is correlated with MCR, but the relationship is not particularly strong. That is, for any given MCR value, models with a wide range of total estimated parameters are seen to achieve it. The only difference between these models is the specification of the latent subspace dimensions for each component, so that Figure 4.2 suggests model performance, as measured by MCR (equivalently ARI), is more strongly associated with good specification of the latent subspace dimensions than with general increases in the total number of model parameters.

The top performing models with respect to MCR and BIC are presented in Table 4.2. We see that the best BIC model achieves the same overall maximum ARI found by Sarkar et al. (2019). Our top ARI, inrrespective of BIC, is 0.739, which is only a slight improvement. We see that MCR noticably favors parsimonious models a bit more than BIC for this dataset. Indeed, all of the best MCR models can be seen to have less total parameters than the BIC models, and in particular, each models the third group (vegetation stubble) fairly parsimoniously. Overall, none of the top models find much parsimony in modelling the first group (grey soil) however, with each model using the maximum number of row dimensions for modelling this group, and most also using the maximum number of column dimension dimensions.

4.4.4 Data Analysis: Fashion-MNIST

The Fashion-MNIST dataset Xiao et al. (2017) became available publicly in the month of August, 2017. It was released by its creators with intentions of supplanting the MNIST handwritten digits as the standard benchmark dataset for machine learning classifiers. Although the handwritten digits are still quite ubiquitous in the field, Fashion-MNIST did see a fair share of success—it was utilized in over 250 academic papers within one year of its release.

The dataset itself consists of 60000 grey scale images, each comprising 748 pixels arranged into a 28 by 28 square matrix. There are 10 different groups in the data, each

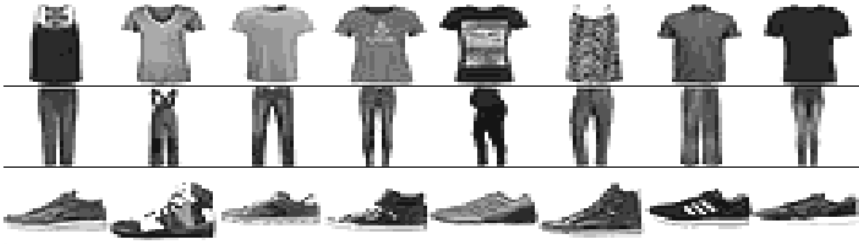


Figure 4.3: A sample of data points from the Fashion-MNIST dataset used in our study.

consisting of 6000 observations. The groups are distinguished by the content of the images, with each group corresponding to a specific article of clothing, hence the "Fashion" moniker. More details can be found in Xiao et al. (2017).

Using the Fashion-MNIST dataset, we task the proposed dual-subspace mixture model with clustering images of t-shirts, pants, and shoes—the basic components of a typical outfit. To do this, we take a sample from the Fashion-MNIST dataset of t-shirt images, pants images, and shoe images, each consisting of $n = 250$ observations. This gives us an operational dataset consisting of $N = 750$ total data points. Some of the images in our operational dataset are presented in Figure 4.3.

The Fashion-MNIST data are fairly high dimensional in the sense that the associated parameter space for q cannot be searched exhaustively in any reasonable amount of time for any interesting values of G. We therefore implement the procedure of estimating q from the data, which was discussed in Section 4.3.4, when fitting the dual-subspace mixture model to Fashion-MNIST.

In terms of fitting the model, our goal is to find a generative model which fits the data well, but also provides a parsimonious representation. We therefore fix the threshold value to be 0.50, and set the possible number of groups to be in $\mathcal{G} = \{3, 4, 5, 6\}$. We then fit a model for each value in G, where each fitted model is initialized with 40 random starts, with the best trajectory according to BIC chosen as the best fitted model for that particular number of groups.

The best model according to BIC for each value of G is presented in Table 4.3, while the confusion matrices for each model are provided in Figure 4.4. Thanks to our choice to hold

G	BIC	Accuracy	OOS Accuracy
3	724177.8	0.903	**0.885**
4	649969.4	0.897	0.868
5	**631428.0**	**0.931**	0.837
6	635905.6	0.929	0.859

Table 4.3: Top model by BIC and its associated accuracy for each value of G. The best performance by a model for each of the three metrics is bolded.

(a) Confusion Matrix for $G = 3$

label	1	2	3
t-shirts	217	24	9
pants	39	0	211
sneakers	1	249	0
q_1	3	3	4
q_2	3	5	2

(b) Confusion Matrix for $G = 4$

label	1	2	3	4
t-shirts	38	202	8	2
pants	3	34	213	0
sneakers	30	0	0	220
q_1	3	3	5	2
q_2	3	3	2	4

(c) Confusion Matrix for $G = 5$

label	1	2	3	4	5
t-shirts	8	1	0	203	38
pants	212	0	0	35	3
sneakers	0	167	78	0	5
q_1	5	2	3	3	3
q_2	2	4	5	3	3

(d) Confusion Matrix for $G = 6$

label	1	2	3	4	5	6
t-shirts	0	8	36	202	2	2
pants	0	212	3	35	0	0
sneakers	80	0	3	0	165	2
q_1	3	5	3	3	2	2
q_2	5	2	3	3	4	1

Figure 4.4: All Confusion Matrices along with the associated fitted values of the hyperparameters q_1 and q_2 for each group.

the threshold value at 0.5, all models produce values of q_1 and q_2 less than 5. Comparing to the data dimension of 28×28, this results in quite noteworthy parameter savings.

In Table 4.3 we have also included the *accuracy*, which reports the proportion of cor-

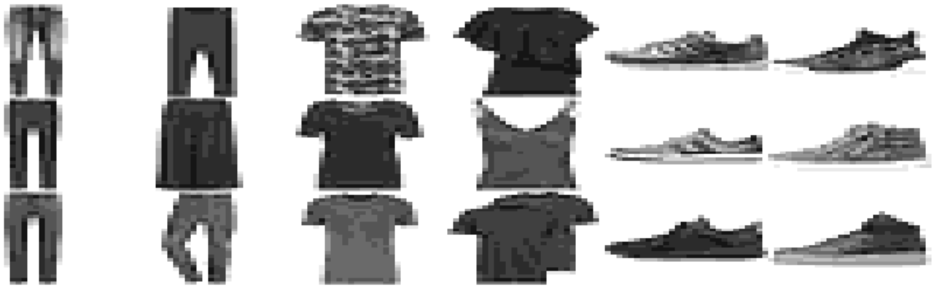


Figure 4.5: Sample data points of each label from its two largest amalgamations. From left to right: component 1 pants, component 4 pants, component 4 t-shirts, component 5 t-shirts, component 2 sneakers, component 3 sneakers.

rect classifications for the sampled data, and *out-of-sample* (OOS) *accuracy*, which is the proportion of correct classifications on the data not used to fit the model. From the table we see that the accuracy corresponding to each model is relatively high, suggesting that the obtained mixture models are identifying approximately correct groupings of the data regardless of the value of G. This is made more granular by inspection of Figure 4.4, and in particular, we see that the models seem to be finding similar solutions. That is, the additional groups in the larger fitted models seem to be formed from partitions of the groups composing the smaller models. Additionally, OOS accuracy results demonstrate that the sample dataset used to fit the models is a good approximation to the population of included images, or alternatively, that the fitted models all generalize fairly well to the population distribution. Overall, all fitted models do a good job at identifying the data groups and generalizing to unseen images, despite their parsimonious specification. In terms of the overall best model, BIC chooses $G = 5$, which also corresponds to the most accurate model.

Looking at Figure 4.4, based on the confusion matrix for the best fitted model by BIC, each image type is seen to be largely collected into a single group while also having a non trivial number of observations sorted into a second group. For example, pants are largely group into the first model component, however, there is a small collection of pants also appearing in the fourth component. We take a look inside these groups to see if the model's distinction between these subsets of each image is associated with any visual differences.

Some sample images for each image (sub)cluster is presented in Figure 4.5.

The first two columns of this figure correspond to sample pants from the first model component and the fourth model component respectively. Based on these images, it seems that the first component corresponds to pants which are dark and have a slim profile. The fourth component, however, has collected pants with a wider profile, and also pants with a bent leg.

The next two columns of Figure 4.5 correspond to t-shirts from the fourth component and fifth component respectively. The fourth component seems to correspond to typical t-shirts, while the fifth component seems to be a catch all for t-shirts that don't fit the mold of the fourth component.

Finally, the last two columns of the figure correspond to sneakers from the second component and the third component respectively. It is quite obvious from these images that the second component corresponds to low-profile sneakers, while the third component corresponds to high-top sneakers.

In conclusion, its seems that, not only has the model been able to distinguish between the true labels of the FMNIST data, but that it has also further separated these groups according to key visual distinctions which correspond to real subgroups within the same general fashion category (e.g. high-top sneakers and regular sneakers). We conclude that the resulting model has achieved the goal of fitting a reasonable approximation to the data distribution of interest while also providing good parsimony.

4.5 Conclusion

In this work, we have defined and investigated a parsimonious, finite mixture of matrix-normal distributions. We have demonstrated that the adaptation of subspace projection to the matrix normal distribution leads parsimony in parameter estimation, reducing quadratic parameter growth to a linear rate. The model formulation is quite flexible, offering the ability to parsimoniously model the rows, the columns, or both, and either of these can be specified with prior knowledge of the data in mind. We have demonstrated the model's applicability in modelling data through two analyses. The first demonstrated competitive performance on a tough dataset, while the second demonstrated the parsimonious benefits that dual-subspace projection can provide. Future work in the area might find a more clever way to choose the threshold for the latent dimensions q, or to estimate q in general. Additionally, modelling of each covariance matrix is somewhat independent for the other, and an extension where different forms of parsimony are found in modelling each covariance matrix may be considered.

Chapter 5

A Joint Latent Factor Analyzer and Functional Subspace Model for Clustering Multivariate Functional Data

5.1 Introduction

Baseball trajectories, air plane flight paths, the motion of a writing utensil or body part—these kinds of measurements are examples of functional data, so-named for their attribute of evolving continuously over some interval of time. Due to technological limitations, the full trajectory of a functional data object is often not observed, but is instead recorded at a discrete set of time points. Despite the fact that these functional data are recorded in the same format as multivariate data, there is potential for information loss if they are analyzed as such (Ramsay and Silverman, 2005). Indeed, a branch of statistics, aptly named Functional Data Analysis, focuses on the development of methods that specifically consider the functional nature of the data to be analyzed. Many of the familiar methods for analyzing univariate and multivariate data have been extended to the functional context. For example, linear models (Cardot et al., 1999, 2003; Chen et al., 2011), graphical models (Zhu et al., 2016; Qiao et al., 2019), PCA (Dauxois et al., 1982; Rice and Silverman, 1991; Silverman, 1996; Jacques and Preda, 2014b), and hypothesis testing (Hall and Keilegom, 2007; Zhang et al., 2011; Fremdt et al., 2013). A good introduction is provided in Ramsay and Silverman (2005), while one may check out the recent review Wang et al. (2016) for a high level overview of more contemporary ideas in the functional data literature.

Our research is specifically interested in the extension of mixture models and model-based clustering to the functional data context. Functional data often arise from clearly defined groups, such as longitudinal biological measurements of healthy and sick individuals. Functional mixture models would then be employed to help identify these groups when the labels are latent or unobserved. The main challenge in using this approach for functional modelling stems from the fact that a random variable taking functional values does not in general admit a probability density function (Lin et al., 2018). Despite this fact, workable approximations for a density have been theorized. Delaigle and Hall (2010) show that by projecting the functional random variable into the eigen basis associated with its covariance operator, one can attribute to it a surrogate density by assuming a distribution on the resulting functional principal component scores (see also: Bongiorno and Goia, 2017). This result is utilized by Jacques and Preda (2013) to create *Funclust*, a methodology which assumes a group specific Gaussian distribution on the first principal components. The advantage of this approach over previous approaches such as Chiou and Li (2007), is that it allows the number of retained terms in the Karhunen-Loeve expansion (KLx) to be group specific, and allows the variance matrices to be non-spherical. In a similar vein to this, Bouveyron and Jacques (2011) also assume a Gaussian distribution on the group specific KLx coefficients, however, rather than modelling only the first few principal components as in Jacques and Preda (2013), this method models all computable principal components under parsimonious assumptions that make the method amount to an extension of Bouveyron et al. (2007) to the function case. In Jacques and Preda (2014b) the authors utilize the multivariate extension of the KLx derived in Saporta (1981) to extend the work of Jacques and Preda (2013) to the case of multivariate functional observations. Likewise, Schmutz et al. (2020) extend Bouveyron and Jacques (2011) to the multivariate case using the same machinery. Other good methods exist for clustering univariate (see: James and Sugar, 2003; Sangalli et al., 2010; Bongiorno and Goia, 2016; Zambom et al., 2019) and multivariate (see: Singhal and Seborg, 2005; Tokushige et al., 2007; Kayano et al., 2010; Ieva et al., 2013) functional data.

The method introduced in this chapter also makes use of the surrogate density of Delaigle and Hall (2010) and employs this alongside a direct sum decomposition of the function space to develop a model for multivariate functional data that treats it as p dependent univariate functional random variables. Under this framework we may utilize the properties of the matrix normal distribution to allow distinct parsimonious modelling of the univariate function spaces and the associated coefficient spaces. That is, we assume a latent subspace structure for the functional principal components, as in Bouveyron et al. (2007), while we assume a latent factor structure on the associated latent coefficients. This formulation allows for interpretation at the component function level, which is useful when the multivariate functional data arise as multiple continuous observations on a single entity which are interesting in their own right.

5.2 Background

5.2.1 Multivariate Functional Principal Component Analysis

We now review how the method of principal component analysis extends to the multivariate function setting. There are a few ways to approach the extraction of functional principal components, however we will focus on the variety of functional principal component analysis (FPCA), and the associated multivariate extension, that is rooted in the works of Dauxois et al. (1982), Jacques and Preda (2014a), and Jacques and Preda (2014b).

Let $(\Omega, \mathcal{A}, Q)$ be a probability space and $(\mathcal{T}, \mathcal{F}, \lambda)$ a measure spaces—where $\mathcal{T}$ is a compact interval, $\mathcal{F}$ its associated Borel σ-algebra, and λ a finite measure—and $L_p^2(\mathcal{T})$ the associated space of p-dimensional square-integrable functions on $\mathcal{T}$. Let $\mathbf{X} : \mathcal{T} \times \Omega \to \mathbb{R}^p$ be a second-order continuous-time stochastic process taking values in $L_p^2(\mathcal{T})$ with mapping,

$$(t, \omega) \mapsto \mathbf{X}(t, \omega) = \Big(\mathbf{X}_1(t, \omega), \ldots, \mathbf{X}_p(t, \omega)\Big)^\top, \tag{5.1}$$

and imbue this process with a continuous mean function $\mu(t) := \mathbb{E}_Q \mathbf{X}(t)$ and autocovariance function $V(s, t) := \mathbb{E}_Q\big[(\mathbf{X}(s) - \mu(s))(\mathbf{X}(t) - \mu(t))^\intercal\big]$, for all $s, t \in \mathcal{T}$. That is, $\mathbf{X}$ is a p-dimensional function-valued random variable, where $p \geq 1$. In the sequel, we use

$\mathbf{X}(t,\cdot)$ and $\mathbf{X}(t)$ interchangeably to represent the random variable at t, $\mathbf{X}(\cdot,\omega)$ and $\mathbf{x}$ interchangeably to represent observed paths, and we use $\mathcal{P}=\{1,2,...,p\}$ to index the spatial dimensions.

The random variable $\mathbf{X}$ establishes an integral operator $\mathcal{V}: L^2_p(\mathcal{T})\to L^2_p(\mathcal{T})$ defined by,

$$(\mathcal{V}\phi)(s)=\int_{\mathcal{T}} V(s,t)\phi(t)d\lambda, \quad \phi\in L^2_p(\mathcal{T}).$$

Under the specified conditions on $\mathbf{X}$, Mercer's Theorem (Mercer (1909), Hsing and Eubank (2015)) states that the covariance function admits the following representation,

$$V(s,t)=\sum_{j\in\mathbb{N}}\sigma_j^2\psi_j(s)\psi_j(t)^{\mathrm{T}},$$

where (σ_j,ψ_j) are the eigenpairs of the operator $\mathcal{V}$, so-called because they satisfy $\mathcal{V}\psi_j=\sigma_j\psi_j$. We also have for all j that $\sigma_j>0$ and $\langle\psi_j,\psi_k\rangle=\mathbf{I}\{j=k\}$, where $\langle\cdot,\cdot\rangle$ is the L^2_p-space inner product defined by,

$$\langle f,g\rangle=\int_{\mathcal{T}}\sum_{i\in\mathcal{P}} f_i(t)g_i(t)\,d\lambda, \quad f,g\in L^2_p(\mathcal{T}). \tag{5.2}$$

From the results of Wang (2008) we may express $\mathbf{X}$ using the (multivariate) Karhunen-Loeve expansion (KLx) as,

$$\mathbf{X}(t)=\mu(t)+\sum_{j\in\mathbb{N}}\big\langle\mathbf{X}-\mu,\psi_j\big\rangle\psi_j(t), \quad t\in\mathcal{T}, \tag{5.3}$$

where $\tilde{C}_j=\langle\mathbf{X}-\mu,\psi_j\rangle$ is the length of the projection of the centered process onto the the jth eigenfunction, known as the jth principal component score. Note that each principal component is a function of a random variable and are hence itself a random variable. It is apparent that constructing a methodology that utilizes the results of Delaigle and Hall (2010) will require representation of the associated data in the KLx form of Equation (5.3). To do this, we first note that the assumptions on $\mathbf{X}$ are equivalent to assuming $\mathbf{X}$ exhibits mean square continuity so that it satisfies,

$$\mathbb{E}_Q\Big[\big\|\mathbf{X}(t_i)-\mathbf{X}(t)\big\|^2\Big]\xrightarrow{i\uparrow\infty}0, \quad \forall t\in\mathcal{T}, \tag{5.4}$$

where (t_i) is some sequence of elements in $\mathcal{T}$ converging to t, and $\|\cdot\|^2$ is the norm induced by the inner product of Equation (5.2). Aside from mean square continuity of the p-dimensional $\mathbf{X}$, Equation (5.4) additionally implies that each subprocess of $\mathbf{X}$, denoted $\mathbf{X}_i$, is also mean square continuous with values lying in $L^2(\mathcal{T})$ for all i in $\mathcal{P}$ and for each ω in Ω. From this it follows that the full realizations $\mathbf{X}(\cdot, \omega)$ can be equivalently thought to live in $\oplus_{i\in\mathcal{P}} L^2(\mathcal{T})$. This formulation is important because it allows us the liberty of modelling $\mathbf{X}$ through joint modelling of its subprocesses $\mathbf{X}_i$. Indeed, we suppose that each component process $\mathbf{X}_i$ produces realizations in a finite dimensional subspace of $L^2(\mathcal{T})$, and without loss of generality, we suppose that space is in the span of the b-dimensional basis $\mathbf{\Phi} := \{\phi_j\}_{j\in\mathcal{B}}$ with index set $\mathcal{B} := \{1, 2, ..., b\}$. We use $H(\mathcal{T})$ to denote the space spanned by the basis $\mathbf{\Phi}$. Define the vector of basis functions evaluated at t in $\mathcal{T}$ by $\mathbf{\Phi}(t) := \big(\phi_j(t)\big)^{\mathrm{T}}_{j\in\mathcal{B}}$ and let C_i be the coefficients such that $\mathbf{X}_i(t) = C_i^\top \mathbf{\Phi}(t)$. We then define the coefficient matrix $\mathbf{C}$ as,

$$\mathbf{C} := \begin{bmatrix} C_1^\top \\ C_2^\top \\ \vdots \\ C_p^\top \end{bmatrix}.$$

With this notation we may then write,

$$\mathbf{X}(t) = \mathbf{C}\mathbf{\Phi}(t), \tag{5.5}$$

to represent the stochastic process $\mathbf{X}$ succinctly in matrix form. Additionally, we may also write the process using a vectorized representation, viz,

$$\mathbf{X}(t) = \big[\mathbf{I}_p \otimes \mathbf{\Phi}(t)\big]^{\mathrm{T}} \mathrm{vec}\big\{\mathbf{C}^{\mathrm{T}}\big\}. \tag{5.6}$$

Under these assumptions we may represent the mean and auto covariance function as,

$$\begin{aligned} \mu(t) &= \big[\mathbf{I}_p \otimes \mathbf{\Phi}(t)\big]^{\mathrm{T}} \mathrm{vec}\big\{\mathbb{E}_Q \mathbf{C}^{\mathrm{T}}\big\}, \qquad \text{and,} \\ V(s,t) &= \big[\mathbf{I}_p \otimes \mathbf{\Phi}(s)\big]^{\mathrm{T}} \mathbf{\Sigma} \big[\mathbf{I}_p \otimes \mathbf{\Phi}(t)\big], \end{aligned} \tag{5.7}$$

where we have define $\mathbf{\Sigma} = \mathbb{E}_Q \mathrm{vec}\big\{(\mathbf{C} - \mathbb{E}_Q\mathbf{C})^{\mathrm{T}}\big\}\mathrm{vec}\big\{(\mathbf{C} - \mathbb{E}_Q\mathbf{C})^{\mathrm{T}}\big\}^{\mathrm{T}}$.

Functional principal component analysis consists of finding the functions in $L_p^2(\mathcal{T})$ that solve,

$$\int_{\mathcal{T}} V(s,t)\psi(t)\,d\lambda = \sigma\psi(s).$$

Substituting in the expression for $V(s,t)$ given in Equation (5.7) the functional principal component problem becomes,

$$\int_{\mathcal{T}} \big[\mathbf{I}_p \otimes \mathbf{\Phi}(s)\big]^{\mathrm{T}} \mathbf{\Sigma} \big[\mathbf{I}_p \otimes \mathbf{\Phi}(t)\mathbf{\Phi}(t)^{\mathrm{T}}\big]\mathbf{e}\,d\lambda = \\ \sigma\big[\mathbf{I}_p \otimes \mathbf{\Phi}(s)\big]^{\mathrm{T}}\mathbf{e},$$

where we have also substituted $\psi(t) = \big[\mathbf{I}_p \otimes \mathbf{\Phi}(t)\big]^{\mathrm{T}}\mathbf{e}$. Pulling constant terms out of the integral, we find the equivalent expression,

$$\big[\mathbf{I}_p \otimes \mathbf{\Phi}(s)\big]^{\mathrm{T}} \mathbf{\Sigma} \big[\mathbf{I}_p \otimes \mathbf{W}\big]\mathbf{e} = \sigma\big[\mathbf{I}_p \otimes \mathbf{\Phi}(s)\big]^{\mathrm{T}}\mathbf{e}, \tag{5.8}$$

where $\mathbf{W}$ is the $b \times b$ symmetric matrix of inner products between the basis functions of $\mathbf{\Phi}$, defined by,

$$\mathbf{W}_{ij} = \langle \phi_i, \phi_j \rangle, \quad i,j \in \mathcal{B}.$$

Equation (5.8) must hold for all s in $\mathcal{T}$, so that we may finally write,

$$\big(\mathbf{I}_p \otimes \mathbf{W}^{\frac{1}{2}}\big)\mathbf{\Sigma}\big(\mathbf{I}_p \otimes \mathbf{W}^{\frac{1}{2}}\big)\mathbf{u} = \sigma\mathbf{u}, \tag{5.9}$$

where we have defined $\mathbf{e} = \big(\mathbf{I}_p \otimes \mathbf{W}^{-\frac{1}{2}}\big)\mathbf{u}$. To compute these eigenpairs in the case of observed data $\mathcal{S} = \{x_i\}_{i=1}^n$, we use the sample estimators,

$$\hat{\mu}(t) = \frac{1}{n}\sum_{i\in\mathcal{S}} x_i(t), \quad \text{and,}$$
$$\hat{V}(s,t) = \frac{1}{n-1}\sum_{i\in\mathcal{S}} \big[x_i(s) - \hat{\mu}(s)\big]\big[x_i(t) - \hat{\mu}(t)\big]^{\mathrm{T}},$$

where $x_i(t) = \big[\mathbf{I}_p \otimes \mathbf{\Phi}(t)\big]^{\mathrm{T}}\hat{c}_i$. The estimated coefficients $\hat{c}_i$ are typically found through least-squares. We may then plug these estimates into the equations above, at which point

Equation (5.9) can be solved using regular PCA methods. The coefficients for the jth eigenfunction are given by $\hat{\mathbf{e}}_j = \left(\mathbf{I}_p \otimes \mathbf{W}^{-\frac{1}{2}}\right)\hat{\mathbf{u}}_j$ and the associated score is computed as $\tilde{c}_j = \hat{c}_j^{\mathrm{T}}(\mathbf{I}_p \otimes \mathbf{W})\hat{\mathbf{e}}_j$. Note that the matrix $\mathbf{W}^{-\frac{1}{2}}$ accounts for the function metric, so as to make the solutions eigenfunctions rather than eigenvectors, that is,

$$1 := \left\langle \psi_j, \psi_j \right\rangle = \mathbf{e}_j^{\mathrm{T}}\left(\mathbf{I}_p \otimes \mathbf{W}\right)\mathbf{e}_j.$$

5.2.2 Matrix Normal Distribution

A $p_1 \times p_2$ matrix $\mathbf{X}_{p_1 \times p_2}$ is said to have a *matrix normal distribution* (Dawid, 1981) with $p_1 \times p_2$ dimensional mean matrix $\mathbf{M}$, and covariance matrices $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ having dimension $p_1 \times p_1$ and $p_2 \times p_2$ respectively, if its associated pdf can be expressed as,

$$\begin{aligned} f(\mathbf{X} \,|\mathbf{M}, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2) = \\ \frac{\exp\left(-\frac{1}{2}\operatorname{tr}\left[\mathbf{\Sigma}_1^{-1}(\mathbf{X}-\mathbf{M})\mathbf{\Sigma}_2^{-1}(\mathbf{X}-\mathbf{M})^T\right]\right)}{(2\pi)^{p_1p_2/2}|\mathbf{\Sigma}_1|^{p_1/2}|\mathbf{\Sigma}_2|^{p_2/2}}. \end{aligned} \tag{5.10}$$

When a random variable $\mathbf{X}$ is distributed according to a matrix normal distribution, we denote it by

$$\mathbf{X} \sim \mathcal{N}_{p_1 \times p_2}(\mathbf{M}, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2).$$

The matrix normal distribution arises as a special case of the multivariate normal, and occurs when the specified covariance matrix can be decomposed as the kronecker product (Srivastava et al., 2008). That is,

$$\begin{gathered} \mathbf{X} \sim \mathcal{N}_{p_1 \times p_2}(\mathbf{M}, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2) \\ \textit{iff} \\ \operatorname{vec}(\mathbf{X}) \sim \mathcal{N}_{p_1p_2}(\operatorname{vec}(\mathbf{M}), \mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1). \end{gathered} \tag{5.11}$$

By assuming the full covariance matrix can be specified as a kronecker product of two lower dimensional covariance matrices, the total number of parameters needed to specify the full covariance matrix is reduced, bringing parsimony. The total number of model parameters is still a quadratic function of both the row and column dimension however, so the gains are modest.

Identifiability

A matrix normal model for data, as specified in Equation (5.10), is not identifiable. Given $\alpha \neq 0$, the substitutions $\mathbf{\Sigma}_1^* = \alpha \mathbf{\Sigma}_1$, and $\mathbf{\Sigma}_2^* = \frac{1}{\alpha}\mathbf{\Sigma}_2$ leave the distribution unchanged. This identifiability problem for the matrix normal distribution can be addressed using Glanz and Carvalho (2013), which proposes to alleviate the problem through the introduction of a general scale parameter σ^2. The suggested substitution is,

$$\mathbf{\Sigma}_i = \sigma \mathbf{\Psi}_i, \quad i = 1, 2, \tag{5.12}$$

where the matrix $\mathbf{\Psi}_i$ is specified to have unit determinant. This allows us to write $\mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1 = \sigma^2(\mathbf{\Psi}_2 \otimes \mathbf{\Psi}_1)$, which results in identifiability of the model parameters. In our work, we instead absorb σ^2 into $\mathbf{\Psi}_1$, i.e. we define $\mathbf{\Sigma}_1 = \sigma^2 \mathbf{\Psi}_1$. This specification is used to keep the estimation procedure for the proposed model fairly simple.

5.3 Methodology

In this section, we delve into the details of our proposed methodology. We begin by specifying the context in which our functional data analysis will take place. This involves recognizing functional data as path realizations of some multivariate stochastic process. We follow up that discussion by relating our model formulation to the previous work of Jacques and Preda (2014b) and Schmutz et al. (2020). Finally, we give full details of our model specification, and discuss the implications thereof.

5.3.1 A Model-Based Approach for Clustering Functional Data

Let $\mathbf{X}$ and $\mathbf{\Phi}$ be as defined in Section 5.2.1. We suppose that the coefficent matrix $\mathbf{C}$ is distributed according to a matrix normal distribution,

$$\mathbf{C} \sim \mathcal{N}_{p\times b}(\mathbf{M}, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2). \tag{5.13}$$

Our intention in defining a distribution on the coefficient matrix $\mathbf{C}$ is to employ a model-based approach for finding homogeneous groups within a set of sample paths from $\mathbf{X}$. The work of Delaigle and Hall (2010), however, enlightens us to the fact that distributional assumptions on $\mathbf{X}$ must be theoretically justified, as $\mathbf{X}$ does not generally admit a density. The alternative they provide, and the direction we follow here, is that one may impose a surrogate density on $\mathbf{X}$ through a joint distribution on its principal component scores. We now show that in the setting specified by Equation (5.13), defining a distribution on the coefficient matrix $\mathbf{C}$ is equivalent to defining a distribution on its associated principal component scores.

By the distributional assumptions on $\mathbf{C}$, the autocovariance function $V(s,t)$ may be written as,

$$\begin{aligned} V(s,t) &= \big[\mathbf{I}_p \otimes \mathbf{\Phi}(s)\big]^{\mathrm{T}}\big[\mathbf{\Sigma}_1 \otimes \mathbf{\Sigma}_2\big]\big[\mathbf{I}_p \otimes \mathbf{\Phi}(t)\big] \\ &= \mathbf{\Sigma}_1 \otimes \big[\mathbf{\Phi}(s)^{\mathrm{T}}\mathbf{\Sigma}_2\mathbf{\Phi}(t)\big]. \end{aligned}$$

With the autocovariance function formulated in this manner, Equation (5.9) of Section 5.2.1 becomes,

$$\mathbf{\Sigma}_1 \otimes \big(\mathbf{W}^{1/2}\mathbf{\Sigma}_2\mathbf{W}^{1/2}\big)u_j = \omega_j u_j. \tag{5.14}$$

Note that since the matrix $\mathbf{\Sigma}_1 \otimes \big(\mathbf{W}^{1/2}\mathbf{\Sigma}_2\mathbf{W}^{1/2}\big)$ has spectral decomposition of the form $(\mathbf{\Gamma}_1 \otimes \mathbf{\Gamma}_2)(\mathbf{\Delta}_1 \otimes \mathbf{\Delta}_2)(\mathbf{\Gamma}_1 \otimes \mathbf{\Gamma}_2)$, the $pb \times 1$ vectors u_j that solve Equation (5.14) can each be expressed as a Kronecker product of a $p \times 1$ and a $b \times 1$ dimensional vector. Using this result, we discover that the principal components of $\mathbf{X}$ are obtained through the following transformation of the coefficients,

$$\tilde{C} = \big(\mathbf{\Gamma}_1 \otimes \mathbf{\Gamma}_2\big)^{\mathrm{T}}\mathrm{vec}\big\{\mathbf{W}^{\frac{1}{2}}(\mathbf{C}-\mathbf{M})^{\mathrm{T}}\big\}. \tag{5.15}$$

The principal components are a linear transformation of the Gaussian distributed coefficients, hence they are themselves Gaussian. Thus our distributional assumptions on the coefficients $\mathbf{C}$ imply the existence of a distribution on the principal components $\tilde{C}$, fulfilling the stipulations made in Delaigle and Hall (2010) for defining a surrogate density on $\mathbf{X}$.

5.3.2 A Latent Factor Model for Parsimonious Principal Components

Recall Equation (5.14), and observe that the problem of finding the eigen-pairs of the autovariance function $V(s,t)$ has been decomposed into distinct parts: an FPCA problem pertaining to $\mathbf{W}^{\frac{1}{2}}\boldsymbol{\Sigma}_2\mathbf{W}^{\frac{1}{2}}$ and a regular PCA problem pertaining to $\boldsymbol{\Sigma}_1$. Noting that the autocovariance function of the subprocess $\mathbf{X}_i$ is,

$$V_i(s,t) = \boldsymbol{\Phi}(s)^{\mathrm{T}}\boldsymbol{\Sigma}_{1ii}\boldsymbol{\Sigma}_2\boldsymbol{\Phi}(t),$$

we see that the FPCA problem corresponds to projection of each subprocess into its associated eigenspace. We call the space spanned by the resulting eigenfunctions $\boldsymbol{\Upsilon}(t) := \boldsymbol{\Gamma}_2^{\mathrm{T}}\mathbf{W}^{-\frac{1}{2}}\boldsymbol{\Phi}(t)$ the intrinsic functional subspace. We can represent the data in the intrinsic functional subspace with the following transformation,

$$\begin{aligned}\mathbf{X}(t) - \mu(t) &= \left(\mathbf{I}_p \otimes \mathbf{W}^{-\frac{1}{2}}\boldsymbol{\Phi}(t)\right)^{\mathrm{T}}\left(\boldsymbol{\Gamma}_1 \otimes \boldsymbol{\Gamma}_2\right) \\ &\qquad \left(\boldsymbol{\Gamma}_1 \otimes \boldsymbol{\Gamma}_2\right)^{\mathrm{T}}\mathrm{vec}\left\{\mathbf{W}^{\frac{1}{2}}(\mathbf{C}-\mathbf{M})^{\mathrm{T}}\right\} \\ &= (\mathbf{C}-\mathbf{M})\mathbf{W}^{\frac{1}{2}}\boldsymbol{\Gamma}_2\boldsymbol{\Upsilon}(t).\end{aligned}$$

Let the projected coefficients be represented by $C^{\star} = C\mathbf{W}^{\frac{1}{2}}\boldsymbol{\Gamma}_2$, and define $\mathbf{M}^{\star}$ analogously. Then $\tilde{\boldsymbol{C}}^{\star} := \mathbf{C}^{\star} - \mathbf{M}^{\star}$ are the subprocess principal components and are distributed according to,

$$\tilde{\boldsymbol{C}}^{\star} \sim \mathcal{N}_{p\times b}(\mathbf{0}, \boldsymbol{\Sigma}_1, \boldsymbol{\Delta}_2), \tag{5.16}$$

where $\boldsymbol{\Delta}_2$ is a $b \times b$ diagonal matrix with unit determinant. Under the matrix normal assumption, the eigenvalues associated with each subprocess $\mathbf{X}_i$ are proportional to $\boldsymbol{\Delta}_2$, with proportionality constant given by the ith diagonal element of $\boldsymbol{\Sigma}_1$. Model parsimony is achieved by assuming only the first d eigenvalues are important,

$$\boldsymbol{\Delta}_2 = \begin{bmatrix}\boldsymbol{\Omega}_2 & \\ & \eta_2\mathbf{I}_{b-d}\end{bmatrix},$$

where $\mathbf{\Omega}_2 = \text{ diag}\{\omega_1, \omega_2, ..., \omega_d\}$ and η_2 is a scalar with smaller magnitude than any entry of $\mathbf{\Omega}_2$. This particular model for the eigenvalues is akin to that of the subspace clustering approach detailed in Bouveyron et al. (2007). Additionally, the latent coefficient mean is

$$\mathbf{M}^{\star} = \begin{bmatrix} m_1 & m_2 & \dots & m_d & \mathbf{0}_{p\times(b\text{-}d)} \end{bmatrix}.$$

The formulation of $\mathbf{M}^{\star}$ also follows from the works of Bouveyron et al. (2007) and Sharp and Browne (2021), in which the mean of the coefficients in the latent intrinsic functional subspace is restricted to have non-zero values only in the components corresponding to the d-dimensional intrinsic subspace. Intuitively the formulation on $\mathbf{\Delta}_2$ and $\mathbf{M}^{\star}$ supposes the existence of a d-dimensional subspace around which the data accumulate in a noisy fashion. This formulation is often referred to as a latent subspace model.

When represented in the base-space, we assume that each set of same index subprocess principal components admit a latent factor structure. That is,

$$\tilde{\boldsymbol{C}}^{\star} = \left[\ \mathbf{\Lambda}_1\mathbf{u} + \epsilon_1 \ \middle|\ \ \mathbf{\Lambda}_2\mathbf{u} + \epsilon_2 \ \middle|\ \ \dots \ \ \middle|\ \mathbf{\Lambda}_b\mathbf{u} + \epsilon_b \ \right],$$

where $\mathbf{\Lambda}_j$ is a $p \times q$ matrix of factor loadings for each j in $\mathcal{B}$. With the matrix normal distribution, each of the latent loading matrices $\mathbf{\Lambda}_j$ will be proportional to a single loading matrix, say $\mathbf{\Lambda}_1$, hence we may write the full coefficient model as,

$$\big(\mathbf{C} - \mathbf{M}\big)\mathbf{W}^{\frac{1}{2}} = \mathbf{\Lambda}_1\mathbf{U}\mathbf{\Delta}_2^{1/2}\mathbf{\Gamma}_2^{\mathrm{T}} + \boldsymbol{\varepsilon}, \tag{5.17}$$

where we have defined,

$$\begin{aligned} \mathbf{U} &\sim \mathcal{N}_{q\times b}(\mathbf{0}, \mathbf{I}_q, \mathbf{I}_b), \text{ and} \\ \boldsymbol{\varepsilon} &\sim \mathcal{N}_{p\times b}\big(\mathbf{0}, \mathbf{\Xi}_1, \mathbf{\Gamma}_2\mathbf{\Delta}_2\mathbf{\Gamma}_2^{\mathrm{T}}\big). \end{aligned}$$

With this model specification, we then find that the coefficients are distributed according to a matrix normal distribution, viz,

$$\mathbf{C} \sim \mathcal{N}_{p\times b}\big(\mathbf{M}, \mathbf{\Lambda}_1\mathbf{\Lambda}_1^{\mathrm{T}} + \mathbf{\Xi}_1, \mathbf{W}^{\text{-}\frac{1}{2}}\mathbf{\Gamma}_2\mathbf{\Delta}_2\mathbf{\Gamma}_2^{\mathrm{T}}\mathbf{W}^{\text{-}\frac{1}{2}}\big). \tag{5.18}$$

Intuitively, the q latent factors serve to identify subprocesses of $\mathbf{X}$ whose variation in terms of each base-space eigenfunction is similar. The factor weights $\mathbf{\Lambda}_j := \mathbf{\Delta}_{2jj}\mathbf{\Lambda}_1$ identify the strength and direction of this variation for basis function υ_j.

In Equation (5.17), the random matrix $\mathbf{U}$ is a latent variable whose first and second moments will need to be estimated in the model fitting process. To that end, we state the conditional distribution of this latent variable given the observed coefficients, which one may show is given by,

$$\begin{aligned}\mathbf{U} \mid \mathbf{C}\mathbf{W}^{\frac{1}{2}} = \mathbf{c} \sim& \\ &\mathcal{N}_{q\times b}\Big(\boldsymbol{\beta}_1(\mathbf{c}-\mathbf{M})\boldsymbol{\Gamma}_2\boldsymbol{\Delta}_2^{-\frac{1}{2}},\, \mathbf{I}_q - \boldsymbol{\beta}_1\boldsymbol{\Lambda}_1,\, \mathbf{I}_b\Big),\end{aligned} \tag{5.19}$$

where we have defined $\boldsymbol{\beta}_1 = \boldsymbol{\Lambda}_1^{\top}\boldsymbol{\Sigma}_1^{-1}$. The moments of interest are then given by,

$$\begin{aligned}\mathbb{E}\big[U \mid \mathbf{c}\big] &= \boldsymbol{\beta}_1(\mathbf{c}-\mathbf{M})\boldsymbol{\Gamma}_2\boldsymbol{\Delta}_2^{-\frac{1}{2}}, \quad \text{and} \\ \mathbb{E}\big[UU^{\mathrm{T}} \mid \mathbf{c}\big] &= b\big(\mathbf{I}_q - \boldsymbol{\beta}_1\boldsymbol{\Lambda}_1\big).\end{aligned}$$

5.3.3 A Mixture of Joint Latent Factor Analyzer and Functional Subspace Models

Suppose data arise from a functional random variable as defined in Equation (5.5). In particular, let $\mathcal{G} := \{1, 2, ..., G\}$ be an index set and suppose the existence of a random variable $\mathbf{Z} = (Z_g)_{g\in\mathcal{G}}$ which is distributed according to a multinoulli distribution with parameter vector $\boldsymbol{\pi} = (\pi_g)_{g\in\mathcal{G}}$ such that $\pi_g > 0$ for each g, and $\sum_{g\in\mathcal{G}}\pi_g = 1$. A single draw from $\mathbf{Z}$ will produce a vector $\mathbf{z} = (z_g)_{g\in\mathcal{G}}$ where only one of the z_g's takes the value 1, while the rest take the value zero. Associate with each element of $\mathcal{G}$ a set of parameters $\boldsymbol{\theta}_g = \{\mathbf{M}_g, \boldsymbol{\Lambda}_{1g}, \boldsymbol{\xi}_{1g}, \boldsymbol{\Gamma}_{2g}, \boldsymbol{\Omega}_{2g}, \eta_{2g}\}$. By linking the location of the value 1 in a roll of $\mathbf{Z}$ with the corresponding element of $\mathcal{G}$, we may equate a roll of $\mathbf{Z}$ with random selection from our collection of parameters $\{\boldsymbol{\theta}_g\}_{g\in\mathcal{G}}$. Indeed, we suppose that generating an observation from $\mathbf{X}$ proceeds in this manner: first roll $\mathbf{Z}$ to choose a set of parameters $\boldsymbol{\theta}_g$. With $\boldsymbol{\theta}_g$ specified, draw a single observation for the matrix normal distribution,

$$\mathcal{N}_{p\times b}\big(\mathbf{M}_g, \boldsymbol{\Lambda}_{1g}\boldsymbol{\Lambda}_{1g}^{\mathrm{T}} + \boldsymbol{\Xi}_{1g}, \mathbf{W}^{\frac{1}{2}}\boldsymbol{\Gamma}_{2g}\boldsymbol{\Delta}_{2g}\boldsymbol{\Gamma}_{2g}^{\mathrm{T}}\mathbf{W}^{\frac{1}{2}}\big), \tag{5.20}$$

which we will denote by C. An observation from $\mathbf{X}$ is then given by the curve $C\boldsymbol{\Phi}(t)$ which is drawn as t traverses through the values of $\mathcal{T}$.

Generating the random coefficients in this manner, i.e. by randomly choosing their distribution using a set of G densities each time an observation is drawn, imbues the coefficients with a mixture distribution. The density of a mixture distribution can be written as,

$$p(\mathbf{C}) = \sum_{g \in \mathcal{G}} \pi_g f(\mathbf{C}; \boldsymbol{\theta}_g),$$

where π_g is the prior probability of choosing $\boldsymbol{\theta}_g$, while $f(\cdot; \boldsymbol{\theta}_g)$ is the density of the matrix normal distribution given in Equation (5.20). Our methodology proceeds under the assumption that the coefficients of the functional random variable $\mathbf{X}$ are generated in this manner, with $\mathcal{G}$ known apriori, while the collection of parameters, $\boldsymbol{\theta}_{g \in \mathcal{G}}$, is considered unknown. It is also possible to handle the case of unknown $\mathcal{G}$ through the aid of model selection tools, which is discussed in Section 5.4.3.

Note that since the parameter vectors $\boldsymbol{\theta}_g$ need only parameterize the density discussed in Section 5.3.2 and are otherwise arbitrary, the mixture model specification implicitly allows the values of the hyperparameters q and d to vary across mixture components. We use q_g and d_g respectively to refer to the specific values of these parameters for a given component, and we use $\mathbf{q} = (q_g)_{g \in \mathcal{G}}$ and $\mathbf{d} = (d_g)_{g \in \mathcal{G}}$ to refer to collection of these hyperparameters for a particular mixture model.

Occasionally, it will be necessary to discuss the mixture from the perspective of a single component. For that purpose, we define the notation $\mathbf{X}_g$ and $\mathbf{X}_g(t)$ which we use to denote the functional random variable, and the random variable at time t, constructed by drawing coefficients exclusively from the distribution with density $f(\cdot; \boldsymbol{\theta}_g)$.

5.4 Parameter Estimation of a Functional Mixture

Suppose we have a sample of data, $\{\mathbf{X}_i\}_{i \in \mathcal{S}}$, with index set $\mathcal{S} = \{1, 2, ..., n\}$ which arise as a set of independently observed paths generated according to the random variable described in Section 5.3.3. Using the basis $\boldsymbol{\Phi}$ we project these data to obtain a set of coefficients $\{C_i\}_{i \in \mathcal{S}}$. Note that each C_i is a matrix with dimension $p \times b$. We denote the associated vectorization of each matrix observation by $\mathbf{c}_i$. Under our model, the likelihood for these data can be expressed generally as,

$$\mathcal{L}(\boldsymbol{\Theta}; \mathcal{S}) = \prod_{i \in \mathcal{S}} \sum_{g \in \mathcal{G}} \pi_g f(C_i \mid \boldsymbol{\theta}_g). \tag{5.21}$$

Equation (5.21) displays the likelihood function of a mixture distribution, which is difficult to optimize directly. When faced with a mixture objective function it is common to assume the existence of a latent variable $\mathbf{Z}$, as described in Section 5.3.3, whose value indicates mixture component membership for each of the observed data points. Assuming the value of this random variable is observed alongside each data point, our data then consists of a set of tuples $(C_i, z_i)_{i \in \mathcal{S}}$, where $z_i = (z_{ig})_{g \in \mathcal{G}}$ is the observation of $\mathbf{Z}$ corresponding to the ith observation. Assuming we have access to this “complete" dataset, the model complete-data log-likelihood becomes,

$$\begin{aligned}
\ell_c(\boldsymbol{\Theta}; \mathcal{S}, \mathbf{Z}) &= \sum_{i \in \mathcal{S}} \sum_{g \in \mathcal{G}} z_{ig} \log \Big[\pi_g f\big(C_i \mid \boldsymbol{\theta}_g\big) \Big] \\
= \sum_{i \in \mathcal{S}} \sum_{g \in \mathcal{G}} z_{ig} \Bigg[& \log \pi_g - \frac{b}{2} \log |\boldsymbol{\Sigma}_{1g}| \\
& - \frac{p}{2} \Big(\log |\boldsymbol{\Omega}_{2g}| + (b - d_g) \log \eta_{2g} \Big) \\
& - \frac{1}{2} \operatorname{tr} \Big\{ \boldsymbol{\Sigma}_{1g}^{-1} \mathbf{R}_{ig} \boldsymbol{\Gamma}_{2g} \boldsymbol{\Delta}_{2g} \boldsymbol{\Gamma}_{2g}^{\mathrm{T}} \mathbf{R}_{ig}^{\top} \Big\} \Bigg],
\end{aligned} \tag{5.22}$$

where we have defined $\boldsymbol{\Theta} = (\theta_g)_{g \in \mathcal{G}}$ the vector of all component parameters, and $\mathbf{R}_{ig} = C_i - \mathbf{M}_g$. It will be convenient to re-express the trace term of Equation (5.22) in the following form,

$$\operatorname{tr} \Big\{ \mathbf{H}_g \boldsymbol{\Lambda}_{2g} \big(\boldsymbol{\Omega}_{2g}^{-1} - \eta_{2g}^{-1} \mathbf{I}_b \big) \boldsymbol{\Lambda}_{2g}^{\mathrm{T}} \Big\},$$

where $\mathbf{H}_g := \sum_{i \in \mathcal{S}} z_{ig} \mathbf{R}_{ig}^{\mathrm{T}} \boldsymbol{\Sigma}_{1g}^{-1} \mathbf{R}_{ig}$ and $\boldsymbol{\Lambda}_{2g}$ consists of the first d_g eigenvectors of $\boldsymbol{\Gamma}_{2g}$. The objective function ℓ_c therefore only depends on $\boldsymbol{\Gamma}_{2g}$ through $\boldsymbol{\Lambda}_{2g}$.

Our model also involves the random variable $\mathbf{U}$, which is present latently in Equation (5.22) as a component of the C_i for each data point. Let U_i denote the true value drawn from $\mathbf{U}$ for observation i. Assuming momentarily that the U_i are observed, the group specific density for the transformed coefficients becomes,

$$\mathbf{C}_i \mid \mathbf{U}_i = U_i, Z_{ig} = 1 \sim \mathcal{N}_{p\times b}\left(\mathbf{M}_g, \boldsymbol{\Xi}_{1g}, \boldsymbol{\Gamma}_{2g}\boldsymbol{\Delta}_{2g}\boldsymbol{\Gamma}_{2g}^{\mathrm{T}}\right).$$

It follows that the model complete-data log-likelihood in the case that both $\mathbf{Z}$ and $\mathbf{U}$ are observed for each observation can be written,

$$\begin{aligned}
&\ell_c(\boldsymbol{\Theta}; \mathcal{S}) \\
&\quad = \sum_{i\in\mathcal{S}}\sum_{g\in\mathcal{G}} z_{ig}\Bigg[\log \pi_g - \frac{b}{2}\log|\boldsymbol{\Xi}_{1g}| - \frac{p}{2}\log|\boldsymbol{\Delta}_{2g}| \\
&\qquad\quad - \frac{1}{2}\operatorname{tr}\Big\{\boldsymbol{\Xi}_{1g}^{-1}\big(\underline{\mathbf{R}}_{ig} - \boldsymbol{\Lambda}_{1g}U_i\big)\big(\underline{\mathbf{R}}_{ig} - \boldsymbol{\Lambda}_{1g}U_i\big)^{\top}\Big\}\Bigg],
\end{aligned} \tag{5.23}$$

where we define the notation $\underline{\mathbf{R}}_{ig} := \mathbf{R}_{ig}\boldsymbol{\Gamma}_{2g}\boldsymbol{\Delta}_{2g}^{-\frac{1}{2}}$. From the perspective of either Equation (5.22) or Equation (5.23) (we will need both), the formulation of our model includes both observed and latent variables, making the expectation-maximization (EM) algorithm (Dempster et al., 1977) a natural choice for parameter estimation.

The EM algorithm is an iterative algorithm composed of two general steps: an expectation step (E-Step) and a maximization step (M-step). Initialization is necessary, and can be achieved by specifying an initial value for either the latent variable $\mathbf{Z}$ for each data point or the model parameters $\boldsymbol{\Theta}$. At iteration k, the E-step updates our estimate of the unobservable latent variables based on our previous estimate of the model parameters $\boldsymbol{\Theta}^{(k-1)}$. The M-step then proceeds to update our estimate for the model parameter vector $\boldsymbol{\Theta}$ given the estimated value of the latent variables found in the E-step. This iterative process proceeds for some predetermined number of steps or until some convergence criterion is satisfied.

All parameter updates for the proposed model are demonstrated to have a closed form, however some of the updates can only be computed conditional on the value of other parameters. Thus, we will need to proceed with an Expectation-Conditional Maximization (ECM) algorithm for estimation (Meng and Rubin, 1993). Due to the latent Gaussian portion of the model, our estimation procedure must also perform multiple cycles per iteration, and is therefore more specifically an Alternating Expectation-Conditional Maximization algorithm (AECM). This particular flavor of EM was developed in Meng and Van Dyk (1997) and maintains the monotonicity and convergence guarantees of the vanilla EM algorithm.

In this section we will derive both the E-step and the M-step for each cycle of our AECM algorithm. All parameter updates are presented in order of computation for each cycle. Following these derivations, we will discuss appropriate initialization strategies and convergence criteria for the model.

5.4.1 First Cycle

The formulation of our model involves both the latent variable $\mathbf{Z}$, which determines component membership for the observed data, and $\mathbf{U}$, the value of the latent Gaussian matrix. In the first cycle of our AECM algorithm, we ignore $\mathbf{U}$ and suppose our latent data is comprised of draws from $\mathbf{Z}$ alone. Since the value of $\mathbf{Z}$ is not actually observed for any of the data points, the process of estimating the values of $\mathbf{Z}$ associated with the observed data then composes the E-step for the first cycle of our algorithm. We use the expected value of the conditional distribution of the random variable $\mathbf{Z}$, given the observed data, as our estimate. Based on our formulated mixture model, the random variable $\mathbf{Z}_i = (Z_{ig})_{g\in\mathcal{G}} \mid C_i$ associated with the ith observation follows a multinoulli distribution with class probability vector given by

$$\mathbf{p}_i = \left(\frac{\pi_g f(C_i \mid \boldsymbol{\theta}_g)}{\sum_{h\in\mathcal{G}} \pi_h f(C_i \mid \boldsymbol{\theta}_h)} \right)_{g\in\mathcal{G}}.$$

At iteration k, given the estimate of the model parameter $\boldsymbol{\Theta}$ found in the previous iteration, we find the conditional expected value to be,

$$E\left(Z_{ig} \mid C_i, \boldsymbol{\Theta}\right) = \frac{\pi_g f\left(C_i \mid \boldsymbol{\theta}_g\right)}{\sum_{h\in\mathcal{G}} \pi_h f\left(C_i \mid \boldsymbol{\theta}_h\right)} =: \hat{z}_{ig}. \tag{5.24}$$

Substituting the $\hat{z}_{ig}$ into Equation (5.22) then gives us the first cycle expected complete-data log-likelihood,

$$\begin{aligned}
&Q_1(\boldsymbol{\Theta}; \mathcal{S}) = \\
&\sum_{g \in \mathcal{G}} \Bigg[n_g \log \pi_g - \frac{b n_g}{2} \log |\boldsymbol{\Sigma}_{1g}| \\
&\qquad - \frac{p n_g}{2} \Big(\log |\boldsymbol{\Omega}_{2g}| + (b - d_g) \log \eta_{2g} \Big) \\
&\qquad - \frac{1}{2} \operatorname{tr} \Big\{ \mathbf{H}_g \boldsymbol{\Lambda}_{2g} \big(\boldsymbol{\Omega}_{2g}^{-1} - \eta_{2g}^{-1} \mathbf{I}_b \big) \boldsymbol{\Lambda}_{2g}^{\mathrm{T}} \Big\} \Bigg],
\end{aligned} \tag{5.25}$$

where we have defined $n_g = \sum_{i \in \mathcal{S}} \hat{z}_{ig}$. We may now initiate the M-step, which is comprised of maximizing Equation (5.25) with respect to elements of $\boldsymbol{\Theta}$.

M-Step

In the first cycle M-step we update the parameters π_g, $\mathbf{m}_{gj}$, $\boldsymbol{\Lambda}_{2g}$, and $\boldsymbol{\Delta}_{2g}$. Thus, $\boldsymbol{\Sigma}_{1g}$ remains fixed at its estimated value from the previous iteration. Updates are provided in running order, as determined by the algorithm. If a parameter has not been provided an explicit update formula, yet appears in the update of another parameter, it takes its value calculated in the previous iteration when used in that expression.

The algorithm first updates the prior group probabilities as,

$$\hat{\pi}_g = \frac{\sum_{i \in \mathcal{S}} \hat{z}_{ig}}{\sum_{g \in \mathcal{G}} \sum_{i \in \mathcal{S}} \hat{z}_{ig}} = \frac{n_g}{n}.$$

Direct maximization of Equation (5.25) with respect to the mean parameters $\boldsymbol{\mu}_{gj}$ gives the updates,

$$\hat{\mathbf{m}}_{gj} = \frac{\sum_{i \in \mathcal{S}} \hat{z}_{ig} C_i \boldsymbol{\Lambda}_{2g(\cdot,j)}}{n_g}, \quad j = 1, 2, ..., d_g, \tag{5.26}$$

for each $g \in \mathcal{G}$, where $\boldsymbol{\Lambda}_{2g(\cdot,j)}$ is the jth column of $\boldsymbol{\Lambda}_{2g}$. To update the eigenvalues $\boldsymbol{\Delta}_{2g}$, we perform differentiation on Equation (5.25) with respect to $\boldsymbol{\Delta}_{2g}$ under the constraint $|\boldsymbol{\Delta}_{2g}| = 1$ yielding,

$$\hat{\boldsymbol{\Delta}}_{2g} = \Big| \operatorname{diag}\Big\{ \boldsymbol{\Gamma}_{2g}^{\mathrm{T}} \mathbf{H}_g \boldsymbol{\Gamma}_{2g} \Big\} \Big|^{-1/b} \operatorname{diag}\Big\{ \boldsymbol{\Gamma}_{2g}^{\mathrm{T}} \mathbf{H}_g \boldsymbol{\Gamma}_{2g} \Big\}.$$

Denoting $v_{2g} = \big|\operatorname{diag}\{\boldsymbol{\Gamma}_{2g}^{\mathrm{T}}\mathbf{H}_g\boldsymbol{\Gamma}_{2g}\}\big|^{-1/b}$, our updates for the components of $\boldsymbol{\Delta}_{2g}$ are then,

$$\begin{aligned} \hat{\boldsymbol{\Omega}}_{2g} &= v_{2g}\operatorname{diag}\Big\{\mathbf{H}_g\boldsymbol{\Lambda}_{2g}\boldsymbol{\Lambda}_{2g}^{\mathrm{T}}\Big\}, && \text{and} \\ \hat{\eta}_{2g} &= \frac{v_{2g}}{b-d_g}\operatorname{tr}\Big\{\mathbf{H}_g\Big(\mathbf{I}_b - \boldsymbol{\Lambda}_{2g}\boldsymbol{\Lambda}_{2g}^{\mathrm{T}}\Big)\Big\}. \end{aligned}$$

Finally, to derive an update for the eigenvectors $\boldsymbol{\Lambda}_{2g}$ we note that the eigenvectors that maximize Q_1 equivalently solve,

$$\hat{\boldsymbol{\Lambda}}_{2g} = \min_{\boldsymbol{\Lambda}_{2g}}\operatorname{tr}\Big\{\mathbf{H}_g\boldsymbol{\Lambda}_{2g}\big(\boldsymbol{\Omega}_{2g}^{-1} - \eta_{2g}^{-1}\mathbf{I}_b\big)\boldsymbol{\Lambda}_{2g}^{\mathrm{T}}\Big\}.$$

We now utilize that the fact that $\mathbf{H}_g$ is positive definite and hence admits a spectral decomposition $\mathbf{H}_g := \mathbf{A}_{1g}\mathbf{D}_{1g}\mathbf{A}_{1g}^{\mathrm{T}}$. Substituting this representation into the optimization problem produces,

$$\hat{\boldsymbol{\Lambda}}_{2g} = \min_{\boldsymbol{\Lambda}_{2g}}\operatorname{tr}\Big\{\boldsymbol{\Lambda}_{2g}^{\mathrm{T}}\mathbf{A}_{1g}\mathbf{D}_{1g}\mathbf{A}_{1g}^{\mathrm{T}}\boldsymbol{\Lambda}_{2g}\big(\boldsymbol{\Omega}_{2g}^{-1} - \eta_{2g}^{-1}\mathbf{I}_b\big)\Big\}.$$

The problem is now solved by realizing that $\boldsymbol{\Omega}_{2g}^{-1} - \eta_{2g}^{-1}\mathbf{I}_b$ is negative definite, hence optimization occurs when $\boldsymbol{\Lambda}_{2g}^{\mathrm{T}}\mathbf{A}_{1g}\mathbf{D}_{1g}\mathbf{A}_{1g}^{\mathrm{T}}\boldsymbol{\Lambda}_{2g}$ is as large as possible—a condition satisfied when $\boldsymbol{\Lambda}_{2g}$ comprises the d_g columns of $\mathbf{A}_{2g}$ corresponding to the first d_g largest eigenvalues of $\mathbf{H}_g$.

5.4.2 Second Cycle

After completion of the first cycle, we are left without updates for the parameters $\boldsymbol{\Lambda}_{1g}$ and $\boldsymbol{\Xi}_{1g}$—the parameters which compose $\boldsymbol{\Sigma}_{1g}$. We now outline a second cycle for updating these parameters. In this second cycle, we consider both $\mathbf{Z}$ and $\mathbf{U}$, so that representation of the objective function is more suitably done with Equation (5.23). Rearranging to obtain an explicit function of the latent variables and the parameters of interest, we discover the expression,

$$
\begin{aligned}
&\ell_c(\boldsymbol{\Theta};\mathcal{S}) = \\
&\quad \sum_{i\in\mathcal{S}}\sum_{g\in\mathcal{G}} z_{ig}\Bigg[\frac{b}{2}\log\left|\boldsymbol{\Xi}_{1g}^{-1}\right| \\
&\qquad\quad - \frac{1}{2}\,\mathrm{tr}\Bigg\{\boldsymbol{\Xi}_{1g}^{-1}\underline{\mathbf{R}}_{ig}\underline{\mathbf{R}}_{ig}^{\mathrm{T}} - \boldsymbol{\Xi}_{1g}^{-1}\boldsymbol{\Lambda}_{1g}\mathbf{U}_i\underline{\mathbf{R}}_{ig}^{\mathrm{T}} \\
&\qquad\quad - \boldsymbol{\Xi}_{1g}^{-1}\underline{\mathbf{R}}_{ig}\mathbf{U}_i^{\top}\boldsymbol{\Lambda}_{1g}^{\top} + \boldsymbol{\Xi}_{1g}^{-1}\boldsymbol{\Lambda}_{1g}\mathbf{U}_i\mathbf{U}_i^{\top}\boldsymbol{\Lambda}_{1g}^{\top}\Bigg\}\Bigg].
\end{aligned}
$$

The estimated component memberships $\hat{z}_{ig}$ are updated in identical fashion to the first cycle, while updating our belief regarding the latent matrix Gaussian $\mathbf{U}_i$ requires,

$$
\begin{aligned}
\hat{U}_{ig} &:= E\big[\mathbf{U}_i \mid C_i, z_{ig}\big]\underline{\mathbf{R}}_{ig}^{\mathrm{T}} = \hat{\boldsymbol{\beta}}_{1g}\underline{\mathbf{R}}_{ig}\underline{\mathbf{R}}_{ig}^{\mathrm{T}} \\
\hat{V}_{ig} &:= E\big[\mathbf{U}_i\mathbf{U}_i^{\top} \mid C_i, z_{ig}\big] = \mathbf{I}_{q_g} - \hat{\boldsymbol{\beta}}_{1g}\boldsymbol{\Lambda}_{1g} + \hat{\boldsymbol{\beta}}_{1g}\underline{\mathbf{R}}_{ig}\underline{\mathbf{R}}_{ig}^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{1g}^{\mathrm{T}}.
\end{aligned}
\tag{5.27}
$$

Since $\boldsymbol{\beta}_{1g}$ is a function of parameters to be estimated in this cycle, we have added a hat to its instances within Equation (5.27) to indicate its role here is as a constant. Plugging Equations (5.24) and (5.27) into ℓ_c, we discover the second cycle expected complete-data log-likelihood,

$$
\begin{aligned}
&Q_2(\boldsymbol{\Theta};\mathcal{S}) = \\
&\quad \sum_{g\in\mathcal{G}}\frac{1}{2}\Bigg[bn_g\log\left|\boldsymbol{\Xi}_{1g}^{-1}\right| - \,\mathrm{tr}\Big\{\boldsymbol{\Xi}_{1g}^{-1}\mathbf{R}_g\Big\} \\
&\qquad\quad - 2\,\mathrm{tr}\Big\{\boldsymbol{\Xi}_{1g}^{-1}\boldsymbol{\Lambda}_{1g}\hat{\boldsymbol{\beta}}_{1g}\mathbf{R}_g\Big\} + \,\mathrm{tr}\Big\{\boldsymbol{\Xi}_{1g}^{-1}\boldsymbol{\Lambda}_{1g}\hat{V}_g\boldsymbol{\Lambda}_{1g}^{\top}\Big\}\Bigg],
\end{aligned}
$$

where we have defined $\mathbf{R}_g = \sum_{i\in\mathcal{S}}\hat{z}_{ig}\underline{\mathbf{R}}_{ig}\underline{\mathbf{R}}_{ig}^{\mathrm{T}}$, as well as $\hat{V}_g = \sum_{i\in\mathcal{S}}\hat{z}_{ig}\hat{V}_{ig}$. This completes the second cycle E-step.

Second M-Step

The second cycle M-step proceeds with maximization of Q_2 with respect to $\boldsymbol{\Lambda}_{1g}$ and $\boldsymbol{\Xi}_{1g}$, for each g in $\mathcal{G}$. Holding $\boldsymbol{\Xi}_{1g}$ fixed and differentiating with respect to $\boldsymbol{\Lambda}_{1g}$ we obtain the update,

$$\hat{\mathbf{\Lambda}}_{1g} = \mathbf{R}_g \hat{\boldsymbol{\beta}}_{1g}^{\mathrm{T}} \hat{V}_g^{-1}.$$

Substituting this estimate into Q_2 and taking the derivative with respect to $\mathbf{\Xi}_{1g}^{-1}$ we obtain,

$$\hat{\mathbf{\Xi}}_{1g} = b^{\text{-}1} \operatorname{diag}\Big\{ \big(\mathbf{I}_p - \hat{\mathbf{\Lambda}}_{1g} \hat{\boldsymbol{\beta}}_{1g} \big) \mathbf{R}_g \Big\}.$$

5.4.3 Initialization Strategies, Convergence Criteria, and Hyperparameters

Our optimization algorithm is a version of the Expectation-Maximization algorithm, and as such requires parameter initialization. We implement multiple initialization strategies, most of which are derived from approaches detailed in Biernacki et al. (2003). In particular, we find the SEMmax approach mentioned therein to work fairly well for our model, and we generally employ this initialization strategy in our studies unless otherwise specified. We note that the SEMmean strategy is problematic for our model, as we implement an automatic strategy for choosing the latent subspace dimension d_g (see the end of this Section) which often leads to inconsistent parameter dimensions across iterations. We therefore do not implement the SEMmean initialization.

To initialize the parameters for the SEMmax strategy, we usually choose random $\mathbf{Z}$ to initialize cycle 1, while for cycle 2 we compute the maximum likelihood estimate of $\hat{\mathbf{\Sigma}}_{1g}$ and then set the inital value for the factor loadings, $\mathbf{\Lambda}_{1g}^{\text{init}}$, to be the first q_g eigenvectors of this estimate, each multiplied by the square root of the associated eigenvalue. We then initialize $\mathbf{\Xi}_1$ as $\hat{\mathbf{\Sigma}}_{1g} - \mathbf{\Lambda}_{1g}^{\text{init}}$.

Convergence of the algorithm can be assessed using an Aitken acceleration-based convergence criterion, which depends on the linear convergence rate of EM. At the k^{th} iteration, the estimate of the limit is given by,

$$\ell_{\infty}^{(k)} = \ell^{(k)} + \frac{\ell^{(k+1)} - \ell^{(k)}}{1 - a^{(k)}} \quad \text{where} \quad a^{(k)} = \frac{\ell^{(k+1)} - \ell^{(k)}}{\ell^{(k)} - \ell^{(k-1)}}.$$

For a chosen tolerance $\varepsilon > 0$, we stop the algorithm when the difference $\ell_{\infty}^{(k)} - \ell_{\infty}^{(k-1)}$ falls below this tolerance.

All that remains is to discuss choosing values for the model's hyperparameters—the number of groups G, the latent subspace dimensions d_g and the latent factor dimensions q_g, for each g in $\mathcal{G}$. All can be chosen simultaneously by searching the space of all possible combinations and picking the best model according to an appropriate model selection criterion (Section 5.5). One issue with an exhaustive search approach to hyperparameter optimization is that it scales quite poorly with data dimension. Supposing we wish to fit a G component mixture model, and given that the data have p row, and b column dimensions, there are then $(pb)^G$ unique combinations of hyperparameter values. This number is large, even for modest values of the involved parameters, hence fitting all possible combinations in such a case is an infeasible approach.

To combat this issue, we adapt a method developed in Bouveyron et al. (2007) for choosing the latent subspace dimension d_g. By estimating d_g from the data, one skirts the requisite computational burden of exhaustively searching the space of all possible d_g's. The proposed estimation method for d_g finds the full set of eigenvalues for $\mathbf{\Sigma}_{2g}$, and then, implementing some form of thresholding, chooses the number of "significant" eigenvalues. This number then serves as an estimate of d_g, the dimension of that component's intrinsic subspace. In the case of our model, this procedure requires decomposing $\mathbf{H}_g$ and then applying a threshold condition to the set of computed eigenvalues accordingly.

The threshold condition for determining significance can be implemented in different ways, and is somewhat arbitrary. Bouveyron et al. (2007) suggest one possible way is to take the sequential difference of the sorted (descending) eigenvalues and choose a cutoff value below which the normalized differences can be considered small. We opt for a different approach in our implementation. We instead use the proportion of total variance explained by the retained eigenvalues as a threshold. For example, if we choose a threshold of 0.75, then our estimate for d_g would become the smallest integer such that the sum of the retained eigenvalues exceeds 75% of the total variance in the data, for $g \in \mathcal{G}$. A threshold value that gives good performance can then be chosen using a model selection criterion.

5.4.4 Comparison to Existing Methods

There are two works in the literature that propose methodologies which aim to cluster multidimensional functional data through a mixture of Gaussians, namely, the *funHDDC* model of Schmutz et al. (2020), and the *Funclust* model of Jacques and Preda (2014b). In this section we briefly outline the construction of each model, and contrast them to the proposed model.

Both models consider a functional random variable akin to that defined in Equation (5.5),

$$\mathbf{X}(t) = [\mathbf{I}_p \otimes \mathbf{\Phi}(t)]^{\mathrm{T}} \mathbf{c}^{\mathrm{T}},$$

where $\mathbf{c}$ is the pb-dimensional random coefficient vector. The models are unified in their assumption that $\mathbf{c}$ has a finite mixture distribution constructed from G Gaussian components. Hence, for each element g of $\mathcal{G}$ we have a corresponding random variable,

$$\mathbf{X}_g(t) = [\mathbf{I}_p \otimes \mathbf{\Phi}(t)]^{\mathrm{T}} \mathbf{c}_g^{\mathrm{T}},$$

where $\mathbf{c}_g \sim \mathcal{N}_{pb}(\boldsymbol{\mu}_g, \mathbf{\Sigma}_g)$. Difference between the models is found in the specification of the component covariance matrix $\mathbf{\Sigma}_g$, which is motivated by the resulting interpretation of the functional principal components. Specifically, for each g in $\mathcal{G}$, if we suppose that the covariance matrix in Equation (5.9) has spectral decomposition given by $\mathbf{\Gamma}_g \mathbf{\Delta}_g \mathbf{\Gamma}_g$, then the *Funclust* and *funHDDC* models can be seen to differ in their specification of the diagonal matrix of eigenvalues $\mathbf{\Delta}_g$.

For the *Funclust* model of Jacques and Preda (2014b), it is assumed that for each g in $\mathcal{G}$ there is a value $k_g \leq pb$ and a diagonal $k_g \times k_g$ matrix $\mathbf{\Omega}_g$ such that $\mathbf{\Delta}_g$ can be written as,

$$\mathbf{\Delta}_g = \begin{bmatrix} \mathbf{\Omega}_g & \\ & \end{bmatrix}, \tag{5.28}$$

where an omitted entry is implied to have the value 0. Although Equation (5.28) may initially appear strange, the resulting interpretation for the functional principal components is quite natural. Indeed, this formulation is equivalent to assuming that the k_g term truncated KLx sufficiently distinguishes $\mathbf{X}_g(t)$ from the other $G-1$ processes, which ought to be true for large enough k_g. Hence, after MFPCA, the gth mixture component is assumed to have a k_g-dimensional multivariate Gaussian distribution with covariance matrix $\boldsymbol{\Omega}_g$.

In a similar vein, the *funHDDC* model specifies that $\boldsymbol{\Delta}_g$ will have the form,

$$\boldsymbol{\Delta}_g = \begin{bmatrix} \boldsymbol{\Omega}_g & \\ & \eta_g \mathbf{I}_{pb-k_g} \end{bmatrix}. \tag{5.29}$$

It is immediately obvious from this formulation that the *funHDDC* model is a direct extension of the *Funclust* model, where now the information from the trailing $p-k_g$ principal components is not completely discarded. Additionally, Equation (5.29) also conveys that the *funHDDC* model extends the latent subspace model developed in Bouveyron et al. (2007), discussed in Section 5.3.2, to the case of functional data. Intuitively, for each g in $\mathcal{G}$, this model assumes that there exists a k_g dimensional subspace of $H(\mathcal{T})$ around which the observations of the gth component accumulate in a noisy fashion. It follows that under the *funHDDC* model, the gth functional principal components are distributed according to $\tilde{\mathbf{c}}_g \sim \mathcal{N}_{pb}(\boldsymbol{\mu}_g, \boldsymbol{\Delta}_g)$.

Recall that the proposed model does not solve Equation (5.9) for the full functional principal components, but instead opts to invoke properties of the assumed kronecker product structure to transform the MPFCA problem to an FPCA problem corresponding to the components of $\mathbf{X}$ (Equation 5.14). Despite this deviance in approach, the models are comparable. Indeed, since $\boldsymbol{\Lambda}_{1g}\boldsymbol{\Lambda}_{1g} + \boldsymbol{\Xi}_{1g}$ is still a positive definite matrix it admits a spectral decomposition which we denote by $\boldsymbol{\Gamma}_{1g}\boldsymbol{\Delta}_{1g}\boldsymbol{\Gamma}_{1g}^{\mathrm{T}}$. Hence, whenever we specify an $\boldsymbol{\Sigma}_{1g}$ having the factor analyzer form, we are equivalently defining an $\boldsymbol{\Delta}_g$ of the form $\boldsymbol{\Delta}_{1g} \otimes \boldsymbol{\Delta}_{2g}$ in the vectorized interpretation. It follows that the proposed model could coincide with the *Funclust* model or the *funHDDC* model whenever the specification of $\boldsymbol{\Sigma}_{1g}$ results in an $\boldsymbol{\Delta}_g$ whose elements can be permuted to form a matrix of the form given in Equation (5.28) or Equation (5.29) respectively, for each g in $\mathcal{G}$. Of course, the proposed model always results in a fully specified $\boldsymbol{\Delta}_g$ and hence never coincides with the *Funclust* model, which assumes trailing eigenvalues are 0. As for the *funHDDC* model, note that we may write

$\mathbf{\Delta}_{1g} \otimes \mathbf{\Delta}_{2g}$ as,

$$\begin{bmatrix} \mathbf{\Delta}_{1g} \otimes \mathbf{\Omega}_{2g} & \\ & \eta_{2g}\mathbf{\Delta}_{1g} \otimes \mathbf{I}_{b-d_g} \end{bmatrix}.$$

From this formulation we see that the proposed model formulation technically transposes to the *funHDDC* model by defining $\eta_g = \delta\eta_{2g}$, where δ is the smallest element of $\mathbf{\Delta}_{1g}$, and amalgamating the other eigenvalues into $\mathbf{\Omega}_g$. However, this correspondence forces the latent dimension k_g of the *funHDDC* model to be $pb - (b - d_g)$ which is approximately $(p-1)b$. In this case, the ratio of the latent data dimension to the full data dimension is $(p-1)/p$, which is quite large. It follows that the *funHDDC* model is generally not well-suited for modelling data generated according to the proposed model as the benefits of its parsimonious specification are lost.

There are, however, ways to specify the models in which true overlap exists. For example, consider the case in which $\mathbf{\Delta}_{1g}$ has the same form as $\mathbf{\Delta}_{2g}$, where we now specify its components as the $q_g \times q_g$ matrix $\mathbf{\Omega}_{1g}$ and η_{1g} the error variance parameter. It is always possible to do this by specifying $\mathbf{\Lambda}_{1g} = \mathbf{\Gamma}_{1g}\big(\mathbf{\Omega}_{1g} - \eta_{1g}\mathbf{I}_{q_g}\big)^{\frac{1}{2}}$ and $\mathbf{\Xi}_{1g} = \eta_{1g}\mathbf{I}_p$. With this additional structure we have,

$$\mathbf{\Delta}_{1g} \otimes \mathbf{\Delta}_{2g} = \begin{bmatrix} \mathbf{\Omega}_g & \\ & \eta_g\mathbf{I}_{pb-k_g} \end{bmatrix}, \tag{5.30}$$

where $\mathbf{\Omega}_g$ is written in terms of the proposed model parameters as,

$$\begin{bmatrix} \mathbf{\Omega}_{1g} \otimes \mathbf{\Omega}_{2g} & & \\ & \eta_{2g}\mathbf{\Omega}_{1g} \otimes \mathbf{I}_{b-d_g} & \\ & & \eta_{1g}\mathbf{\Omega}_{2g} \otimes \mathbf{I}_{p-q_g} \end{bmatrix}, \tag{5.31}$$

while $\eta_g = \eta_{1g}\eta_{2g}$. It follows that the latent data dimension k_g then takes the value $d_gp + q_gb - d_qq_g$. With this specification, the ratio of the latent data dimension for the *funHDDC* model is a more modest $d_g/b + q_g/p - d_gq_g/pb$.

When both $\mathbf{\Sigma}_{1g}$ and $\mathbf{\Sigma}_{2g}$ are supposed to exhibit the latent subspace model structure, the proposed model gains an additional interpretation as a sort of generalized latent subspace model. Here, we can consider the eigenfunctions corresponding to the eigenvalues

(q,d)		D_1	D_2	D_3	D_4
(1,1)	BIC	0.944	0.956	0.996	0.974
	ICL	0.944	0.956	0.996	0.974
(3,3)	BIC	0.932	0.968	0.924	0.990
	ICL	0.932	0.968	0.924	0.990
(5,5)	BIC	1.000	0.998	1.000	1.000
	ICL	1.000	0.998	1.000	1.000

Table 5.1: The percentage of times each criterion chose the correct model for each of the simulated scenarios.

of $\mathbf{\Omega}_{1g} \otimes \mathbf{\Omega}_{2g}$ to span the latent subspace which accounts for much of the variance in the paths $\mathbf{X}(t)$, while the remaining directions account for noise directions, but now with varying levels of variation.

5.5 Investigative Analyses

In this section we employ the proposed model in a number of application studies for the purpose of demonstrating its clustering capabilities. This includes simulation studies regarding parameter recovery and model selection, two comparison studies alongside established functional clustering methodologies, and two data analyses. To facilitate discussion of proposed model in this section, we introduce the name *Multivariate Functional Subspace and Factor Analyzer* model, from which we get the acronym MFSF. Both will be used in the sequel to refer to the proposed model.

5.5.1 Model Selection Simulation

The MFSF model contains three hyperparameters—G, $\mathbf{q}$ and $\mathbf{d}$—which cannot be estimated from the likelihood. We intend to employ the Bayesian Information Criterion (BIC) to facilitate the choice of these hyperparameters for the MFSF model in practical applications. The BIC (Schwarz, 1978) is a criterion for model selection that attempts to counteract the likelihood's propensity to increase with the number of model parameters through the addition of a penalization term. The BIC is defined as,

$$BIC(m) = k_m \log n - 2 \log p(\mathbf{X} \mid m),$$

where m represents the fitted model and k is the total number of free parameters in the model. For the proposed model k is computed as $(2p + b + 0.5) \sum_{g \in \mathcal{G}} d_g - \frac{1}{2} \sum_{g \in \mathcal{G}} d_g^2 + \frac{1}{2} \sum_{g \in \mathcal{G}} q_g(q_g - 1) + (p + 1)G$. By the work of Keribin (2000), we know that the BIC is asymptotically consistent for choosing the correct number of mixture components, i.e. G, while Steele and Raftery (2010) demonstrates that it performs well in a number of practical situations.

We also consider the Integrated Completed Likelihood of Biernacki et al. (2000), which we define here as,

$$ICL(m) = k_m \log n - 2 \log p(\mathbf{X}, \hat{\mathbf{Z}} \mid m),$$

to align with our definition of BIC, with $\hat{\mathbf{Z}}$ being the MAP estimates of the latent class labels. Use of this criterion has become commonplace in the mixture model literature, and in particular, was shown to perform adequately in this context by Steele and Raftery (2010).

In the present section we seek to provide evidence that these two criteria can also be used for selecting the hyperparameters d_g and q_g of the MFSF model. We bring this evidence using a simulation study. In an effort to make the simulation study informative, we devise multiple experimental settings, each of which will be carried out for 500 replications. These settings are differentiated by changing simulation hyperparameters. For this study, we choose to vary the dimension of the data, and the true values of d_g and q_g. For data dimension, we choose four scenarios: $D_1 = (10 \times 10)$, $D_2 = (10 \times 20)$, $D_3 = (20 \times 10)$, and $D_4 = (20 \times 20)$. For the values of the model hyperparameters we choose three settings: $(q_g, d_g) = (1, 1)$, $(q_g, d_g) = (3, 3)$, and $(q_g, d_g) = (5, 5)$. These are set to the same value of each group in the mixture model. The number of components, G, is set to 2 in all scenarios. This setup results in a simulation study with $4 \times 3 = 12$ experimental conditions.

For each of the experimental conditions, we repeat 500 replications of the simulation study. Each replication proceeds as follows: we first generate a random set of parameters for a mixture model satisfying the assumptions of the proposed methodology, according to the specifications outlined in Appendix B.3. Using these parameters, we then generate a dataset on which to fit the model. On this dataset we fit the model 25 times, once for each unique combination of $d = 1, 2, 3, 4, 5$ and $q = 1, 2, 3, 4, 5$ for the model hyperparameters. For example, $\mathbf{q} = (2, 2)$ and $\mathbf{d} = (5, 5)$ corresponds to one of these 25 settings. We calculated the BIC and ICL for each of these fits, and then choose the model resulting in the best value of each. The value of $\mathbf{q}$ and $\mathbf{d}$ for the chosen model of each criterion is then returned, and if this value matches the value used to generate the data we say that the criterion chose correctly, otherwise we say it did not. The results of carrying out this simulation study are provided in Table 5.1.

Here we see that across all scenarios, both BIC and ICL do a fairly good job at identifying the structure of the true underlying model. We also see that there is no difference between the performance of the BIC and the ICL in this simulation scenario. The reason for this is likely due to the fact that the models are generated according to the data, hence, we expected low entropy in the resulting model fits when the correct model is specified. As the difference between our formulation of the BIC and ICL is simply the entropy of the estimated matrix $\hat{\mathbf{Z}}$, this result is not overly surprising. We conclude that either criterion should perform well as a model selection tool for our method.

5.5.2 Parameter Recovery

Our first analysis will assess how well our methodology can estimate parameters the true model parameters used to generate a sample dataset. Such an assessment is typically referred to as parameter recovery, and it serves to reassure us that the model works as expected in the context for which it is intended. Of course, since we are dealing with data samples and iterative numerical algorithms, the true parameters are almost never recovered exactly. We therefore need to choose a measure which informs about the quality of the parameters recovered by the model from a given sample. In this study, we choose to assess recovery performance with the mean squared error (MSE). Letting $\boldsymbol{\Theta}_v := \big(\ \text{vec}\{\theta_1\},\ ...,\ \text{vec}\{\theta_g\}\big)^{\mathrm{T}}$ be the vectorization of the model parameters, the mean squared error between the true and estimated parameters is given by,

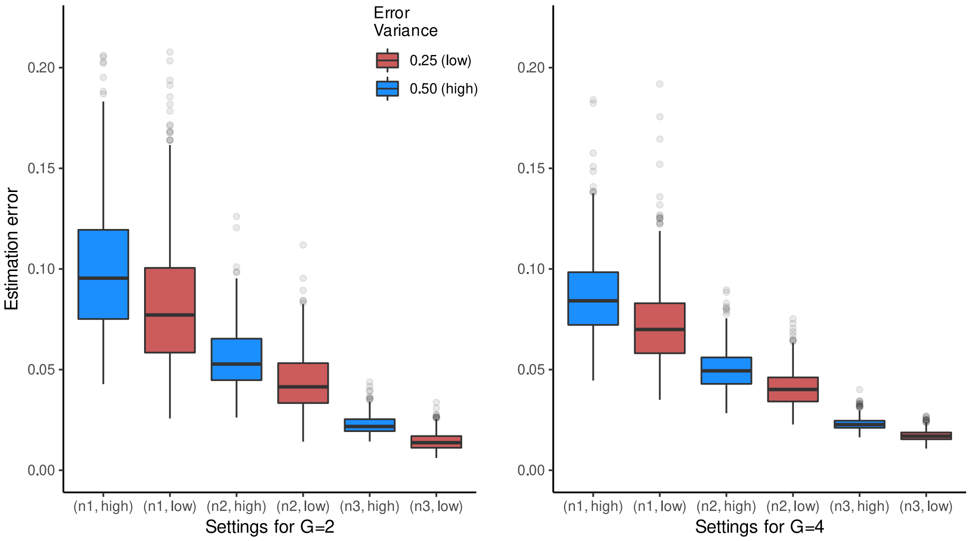


Figure 5.1: Distributions of the parameter estimation error for each condition of the parameter recovery experiment, presented as boxplots. On the left are the plots related to the two component mixtures, and on the right are the four component mixture results. Color indicates the level of the error variance, with blue representing the larger value.

$$L^{-1}\left\|\hat{\mathbf{\Theta}}_v - \mathbf{\Theta}_v\right\|^2,$$

where L is the dimension of the vector $\mathbf{\Theta}_v$. The lower the value of the mean squared error between the estimated parameters and the true parameters, the better the estimation process is deemed to have performed.

To commence with the study we generate 12 different collections of datasets, each of which follows from a unique set of experimental conditions. Specifically, we manipulate three different simulation properties for this purpose: G, the number of mixture components, n, the size of the sample taken from each component, and $\mathbf{\Xi}_{1g}$, the scaling matrix for the factor analyzer errors of each component. The particular values we used for these properties are given in Table 5.2.

There are an abundance of ways we could choose to vary the value of $\mathbf{\Xi}_{1g}$ to specify

Attribute	Description	Values
G	Number of components	2, 4
n	Sample size	$n_1 = 50$, $n_2 = 100$, $n_3 = 500$
η_1	error variance scale	0.25, 0.5

Table 5.2: Values used to create the 12 different experimental settings for the parameter recovery simulation study.

different conditions. We settle for a simple implementation in which $\mathbf{\Xi}_{1g}$ is spherical with proportionality constant η_1. We specify this matrix to be the same across all groups in each of the conditions. Different conditions are achieved by changing the value of η_1. Under this specification, the error term of the jth factor analyzer, ε_j, has variance $\eta_1 \mathbf{\Gamma}_{2g} \mathbf{\Delta}_{2g} \mathbf{\Gamma}_{2g}^{\mathrm{T}}$, for possible g. For specifics regarding the specification of parameters not changed across simulation conditions, please see Appendix B.3. In short, these parameters were generated randomly once, and fixed across conditions.

There are 12 unique combinations of the values in Table 5.2, and for each combination we generate a collection of 500 datasets, which we then use to fit the MFSF model. Each time the model is fit, the em-EM strategy of Biernacki et al. (2003) is used for initialization. This comprises running multiple short (in terms of total iterations) instances of EM from randomly chosen positions, with the parameter set corresponding to the best run (in terms of highest likelihood) then being used to run the algorithm to convergence. Here we choose to use 30 random starts, each run for 20 iterations. Of course, it would be much faster and simpler to use the true parameter values to initialize the algorithm each time, which ensures that we check if a local maximum exists in their vicinity. However, by refraining from this approach we allow the model to converge to a solution close to the true parameters only if there is a maximum in their vicinity, and if that maximum happens to be the best among the randomly chosen search space. This second approach more closely reflects the conditions of applying the model in practice, hence it is preferred here.

By varying some aspects of the underlying mixture model parameters we hope to show that the model performs well generally when data are drawn according to its underlying assumptions. Further than that, however, we also aspire to provide a high level analysis of how a change in each of these aspects impacts the performance of the model. The first goal can be validated through assessment of the properties of the distribution of MSE values resulting from running the model on each of 12 collections of datasets, while the second goal can be assessed by comparing the differences in the distribution of MSE values across these different conditions. The results of our parameter recovery simulation are reported in Figure 5.1.

By looking at the blue boxplots and the red boxplots in each subfigure separately, we see that as the sample size increases, estimation error, as measured my MSE, decreases. This hints that the algorithm is providing consistent parameter estimation. Although this pattern is exhibited by both colored sets of plots, it is notable that for a given sample size, the corresponding blue boxplot has a larger central tendency than the corresponding red boxplot. This suggests that larger error variance inherent in the data results in more estimation error for each sample size, on average. Finally, we notice that the boxplots associated with four group mixture models tend to have smaller variation than their two component counterparts. This makes intuitive sense, as the MSE calculations regarding the four component mixtures include twice as many terms as the two component calculations. Indeed, we can consider each four component calculation as an average of two two component calculations. Overall, we conclude that the MFSF model has demonstrated adequate performance in recovering the true model parameters by providing patterns in the results which give evidence of consistency.

5.5.3 Comparison Study I

In this section, we compare MFSF to other existing methodologies that focus on unsupervised clustering of vector-valued functional data. The methodologies chosen for this comparison are *funHDDC* (Schmutz et al., 2020) and *Funclust* (Jacques and Preda, 2014b), both of which are model-based, and also the nonparametric methodology of Martino et al. (2017), which is an adaptation of the k-means algorithm to the functional context.

We compare the algorithms using an extension of *Scenario B* from the simulation study section of Schmutz et al. (2020). Similar to the original authors' intentions, our motivation

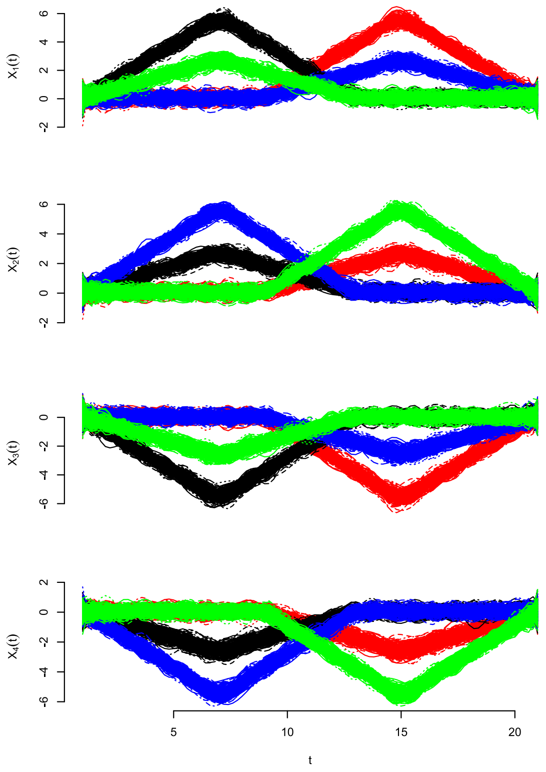


Figure 5.2: An instance of data generated according to our extension of the Schmutz et al. (2020) *Scenario B*. Each row of plots shows data generated from a particular dimension, with colors indicating group membership.

for choosing *Scenario B* lies in the fact that the data should not particularly favor any of the chosen methodologies.

$$\begin{aligned}
\text{Group 1:}\quad & X_1(t) = U + (1-U)h_1(t) + \epsilon(t),\\
& X_2(t) = U + (0.5-U)h_1(t) + \epsilon(t),\\
\text{Group 2:}\quad & X_1(t) = U + (1-U)h_2(t) + \epsilon(t),\\
& X_2(t) = U + (0.5-U)h_2(t) + \epsilon(t),\\
\text{Group 3:}\quad & X_1(t) = U + (0.5-U)h_1(t) + \epsilon(t),\\
& X_2(t) = U + (1-U)h_2(t) + \epsilon(t),\\
\text{Group 4:}\quad & X_1(t) = U + (0.5-U)h_2(t) + \epsilon(t),\\
& X_2(t) = U + (1-U)h_1(t) + \epsilon(t),
\end{aligned}$$

Originally, *scenario B* was comprised of the following 4 group generative model, where the time domain is chosen as $t \in [1, 21]$, $h_1(t) = (6-|t-7|)_+$ and $h_2(t) = (6-|t-15|)_+$ are the generating basis functions, U is a uniform random variable on $[0, 0.1]$, and $\epsilon(t)$ is white noise independent of U with variance 0.25. Each group is therefore composed of a two dimensional vector-valued function $X(t) = \big(X_1(t), X_2(t)\big)^{\mathrm{T}}$. We extend this generating model to four dimensions by now specifying that the stochastic function of each group takes the form $X(t) = (X_1(t), X_2(t), X_3(t), X_4(t))^{\mathrm{T}}$, where $X_3(t)$ for each group has the form $-X_1(t)$, while $X_4(t)$ has the form $-X_2(t)$, although they are distinguished from these random variables through an independent draw from the random variable U. Assuming we observe these functions at the specified time points, the observations are then returned to functional form through projection onto a B-spline basis consisting of 35 functions. Some data generated according to this adapted scenario, which we call *Scenario B+*, are displayed in Figure 5.2.

Using datasets generated from *Scenario B+*, we compare the chosen methodologies in both their ability to identify the correct number of latent groups, and their ability to find the correct groups. Each iteration of our simulation study is orchestrated as follows: we first generate a dataset according to *Scenario B+*. We then proceed to stage 1, in which we run each of the algorithms on this dataset and provide them with the knowledge that the true number of groups is four. We then assess how well each was able to recover the

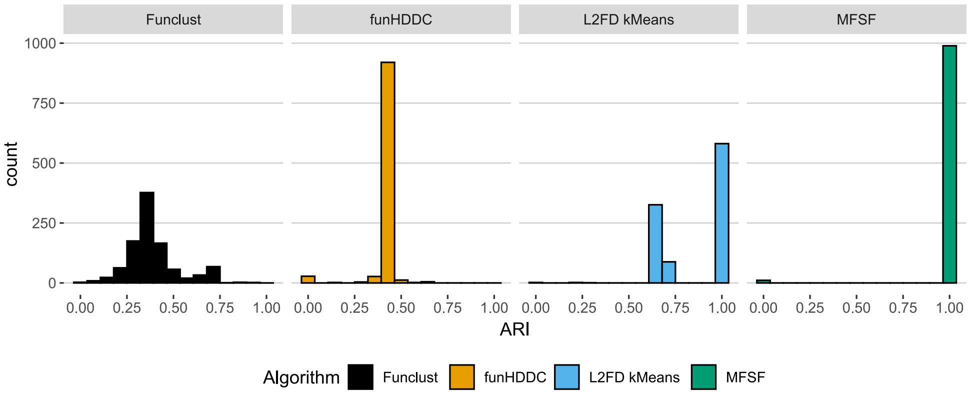


Figure 5.3: Histograms of the results for each of the clustering methods.

true underlying groups in the data. This assessment is completed using the Adjusted Rand Index (ARI) which is a common tool for measuring the agreement between two different clustering (Hubert and Arabie, 1985). In stage 2, we let each of the algorithms choose the number of latent groups from the set of possible values $\{1, 2, 3, 4, 5\}$. For the model-based approaches, this choices is made using BIC. For the functional k-means approach, this choice is made using the silhouette method (Rousseeuw, 1987), as suggested by Martino et al. (2017). Our study is completed by carrying out this process 1000 times.

In an effort to keep the playing field level, each model-based method is initialized in an identical fashion. The em-EM initialization strategy proposed in Biernacki et al. (2003) is the method of choice. We specify 20 random starts, each run for 20 iterations in each case. Because the *funHDDC* results presented in Schmutz et al. (2020) for *Scenario B* were found using a k-means initialization, and because this initialization is the default for the `R` function, we also include this initialization for the *funHDDC* algorithm. If this initialization out performs the results using the em-EM initialization in terms of BIC, then we uses those results instead. Due to limitations in the provided package, we do not have control over any aspects of initializing the functional k-means algorithm, hence the default is used throughout.

The results of stage 1 are depicted graphically in Figure 5.3 as histograms. The mean and standard error of the result data are provided in Table 5.3, along with the average

Algorithm	Avg. ARI	ARI Std. Err.	Avg. Time (secs)
funHDDC	0.40	0.07	249.7
Funclust	0.38	0.14	218.1
MFSF	0.99	0.10	228.8
L2FD K Means	0.85	0.18	5.7

Table 5.3: Summary statistics of results from stage 1 of the simulation study for each of the algorithms.

computation time of each algorithm. Here we use the label L2FD for the results of the functional k-means methodology. This label is chosen to reflect the fact that the L_2 function space metric is used when we run the algorithm (others are possible, see Martino et al., 2017). From Figure 5.3 we see that both the MFSF and L2FD k-means approach perform the best in stage 1 of the study, with each reporting a large number of perfect clusterings across the simulated *Scenario B+* datasets. The remaining two model-based approaches, *Funclust* and *funHDDC*, did not perform as well. Of particular interest are the results of the *funHDDC* algorithm, which reports a fairly low standard error. This suggests that *funHDDC* is finding a very similar model specification each time, regardless of the variation in the dataset and random starts. This could suggest that the implementation of the em-EM algorithm in the `funHDDC` R function is using the same seed in each iteration, and therefore always producing the same "random" starts, or that a global maximum corresponding to the true groups is a steep peak and therefore difficult to find. Given that the *Funclust* methodology, which is quite similar to *funHDDC*, produces more variation in its results but obtains a similar mean ARI to that of *funHDDC*, we are inclined to believe it is the former of these issue.

In terms of computation time, we see that the functional k-means methodology outperforms the model-based approaches by a large margin, however, these times include the em-EM initialization time for the model-based algorithms. Implementing an alternative initialization strategy could speed up any of these algorithms, but this may come at the cost of fitted model quality. Overall, the computation times of the three model-based approaches are fairly comparable, but *Funclust* is the fastest on average by about 10 seconds.

Algorithm	1	2	3	4	5
funHDDC	0	0	1	99	900
MFSF	0	0	19	831	150
Funclust	110	860	30	0	0
L2FD K Means	0	3	77	632	288

Table 5.4: The number of times each possible value of G was chosen by each of the algorithms over the course of the simulation. The numbers at the top denote the tested values of G, and the numbers in each column represent the number of times that value of G was chosen by each of tghe algorithms.

Stage 2 results are reported in Table 5.4. We again see that MFSF and functional k-means are the best performers, each choosing the correct number of groups in a majority of their runs. Interestingly, *Funclust* serially underestimates the number of groups, while *funHDDC* serially overestimates. In fact, *Funclust* never chooses the correct number of groups, and *funHDDC* only underestimates 1 time out of the 1000 trials. This seems a bit odd, given the closeness of the two methodologies. Both MFSF and functional k-means overestimate the number of groups quite a bit more often than they underestimate, suggesting that the problem of *Funclust* may be the lack of flexibility in its modelling of the trailing eigenvalues.

5.5.4 Comparison Study II

As we discussed in Section 5.4.4, in the special case that $\mathbf{\Sigma}_1$ exhibits both the factor analyzer and latent subspace structure, the MFSF methodology and the *funHDDC* methodology of Schmutz et al. (2020) overlap. In this second study, we investigate the impact of this overlap as it relates to model performance. To do this, we generate five parameter sets corresponding to five different models, denoted by $M1-M5$. We start with the model $M3$, which is chosen to be comprised of parameters that satisfy both the assumptions of MFSF and of *funHDDC*. This makes $M3$ the overlap model. Details on how $M3$ is generated can

Attribute	Description	Values
$p \times b$	Data dimension	6×11, 18×33
n	Sample size	50, 250
ρ	Cluster overlap	3.5, 5, 7

Table 5.5: Values used for various properties of the parameter settings for comparison study between the *funHDDC* methodology and our own.

be found in Appendix B.4. Data generated from $M3$ should be easily handled by both algorithms. Next, we apply a deterministic perturbation to the parameters of $M3$ so that the resulting parameters still satisfy the assumptions of *funHDDC*, but no longer satisfy the assumptions of MFSF. We use this perturbation at two different strengths to get the models $M1$ and $M2$, with $M1$ corresponding to the parameters resulting from the perturbation with the larger strength. To get the parameters of $M4$ and $M5$ we proceed in a similar fashion, this time perturbing the parameters of $M3$ so that they satisfy only the modelling assumptions of MFSF. In this case, $M5$ corresponds to the stronger perturbation. Based on this construction, the five models $M1 - M5$ can be thought to form a sort of interpolation between the assumptions of *funHDDC* and MFSF. For complete details regarding how the model parameters are generated for this simulation study see Appendix B.4.

Our simulation intends to assess the performance of each model on data generated according to the parameters of $M_1 - M_5$. If the overlap is important, each model will do well only on data generated according to its own assumptions. We also include the functional k-means algorithm in our analysis as a sort of baseline model. Its results should reflect how well separated the clusters of functions are geometrically. A poor performance by the kmeans algorithm should imply high cluster overlap.

In an attempt to make the analysis more insightful, we carry out the simulation under different experimental settings. These settings are differentiated by variation in data dimension, sample size, and cluster overlap. The number of groups G remains fixed at 2, and the hyperparameters q_g and d_g remaining fixed at 2 and 3 respectively across all settings. In terms of data dimension, we specify a low dimensional (6×11) and a high dimensional (18×33) setting. For sample size, we specify both a small ($n = 50$) and a moderate ($n = 250$) setting. We devise cluster overlap to be the value $\rho := \|\mathbf{M}_1 - \mathbf{M}_2\|$, which is the Euclidean distance between the vectorized group means. We specify three values of ρ—3.5, 5, and 7. These conditions are summarized in Table 5.5. This results in 12 unique experimental conditions for testing each model $M1 - M5$.

For each model $M1 - M5$, and for each unique combination of experimental settings from Table 5.5, we generate 100 datasets on which to fit the competing models. We assess the performance of the models using ARI and model complexity. The first, ARI, gives us an idea of how well the models are able to identify the latent groups of the data, while model complexity (number of parameters) gives us an idea of how many parameters the model needs to achieve this performance. This second performance metric does not apply to the functional k-means methodology, so model complexity values relating to this methodology are not reported.

For initialization of the model-based methods, we again turn to the em-EM strategy, this time using 30 starting points. For our algorithm, we use the automatic selection method discussed in Section 5.4.3 for choosing the latent d_g, and we use BIC to choose the q_g. When implementing the *funHDDC* algorithm, we use the default hyperparameter selection method, which is based on Cattell's scree test (Schmutz et al., 2020). As in Section 5.5.3, we use the default initialization strategy and the L_2 distance for the functional k-means method.

In general, we notice that all three simulation hyperparameters appear to impact the performance of the tested models. For example, increasing the cluster overlap can be seen to reduce each model's ability to correctly ascertain the latent groups, as evidenced by decreasing average ARI for each fixed model and sample size. The notion that increasing the sample size positively benefits model performance is clearly supported by these results as well. This is most noticeable in the performance of the *funHDDC* algorithm in Table 5.6, which often faltered for the small sample size $n = 50$, but performed quite well when provided a bit more data to work with in the form of $n = 250$. Finally, the increased challenge presented by high dimension data can be seen by comparing the results of Table 5.6 and Table 5.7, especially for models $M1$ and $M2$, on which every model performed poorly.

$(p \times b) = (6 \times 11)$				Avg. ARI		Avg. Complexity	
	ρ	n	MFSF	funHDDC	LF2D	MFSF	funHDDC
$M1$	7	50	0.999	0.266	0.912	214.8	1233.5
	5	50	0.986	0.015	0.363	215.6	1860.7
	3.5	50	0.754	0.008	0.098	207.2	2132.4
	7	250	0.999	0.991	0.991	222.9	3282.0
	5	250	0.992	0.952	0.816	224.6	3402.2
	3.5	250	0.931	0.848	0.258	224.8	3248.3
$M2$	7	50	0.999	0.320	0.895	219.0	1121.1
	5	50	0.983	0.023	0.412	216.4	1783.2
	3.5	50	0.823	0.008	0.108	213.0	2007.0
	7	250	1.000	0.988	0.994	224.5	3081.9
	5	250	0.990	0.936	0.865	225.9	3143.6
	3.5	250	0.933	0.645	0.236	225.0	2674.4
$M3$	7	50	1.000	0.293	0.962	263.4	1028.3
	5	50	0.991	0.028	0.414	264.1	1688.9
	3.5	50	0.934	0.005	0.137	264.1	1929.8
	7	250	1.000	0.962	0.992	278.4	2652.0
	5	250	0.997	0.931	0.865	279.9	2503.2
	3.5	250	0.970	0.321	0.258	280.1	1880.6
$M4$	7	50	1.000	0.247	0.890	261.8	1363.5
	5	50	0.990	0.020	0.279	265.3	2000.1
	3.5	50	0.943	0.007	0.086	263.0	2254.5
	7	250	1.000	0.989	0.983	278.6	3061.1
	5	250	0.995	0.942	0.801	279.0	2960.0
	3.5	250	0.960	0.400	0.248	277.4	2561.6
$M5$	7	50	1.000	0.210	0.710	263.3	1673.5
	5	50	0.998	0.039	0.251	265.1	2222.1
	3.5	50	0.981	0.032	0.082	261.8	2312.9
	7	250	1.000	0.990	0.963	278.5	3316.9
	5	250	0.998	0.950	0.731	278.4	3391.4
	3.5	250	0.993	0.591	0.240	278.2	2958.3

Table 5.6: Average ARI and model complexity results from the competing algorithms on low dimensional datasets (6×11). When considering only these results, it seems that ρ and n have the greatest impact on model performance.

In terms of the competing algorithms, both are able to perform well on all models $M1 - M5$ for the low dimensional datasets. This suggests that the overlap is not such an important distinction when data dimension is not high. It may be that for lower dimensions, the structure of the noise for each model is well approximated by the structure

$(p \times b) = (18 \times 33)$			Avg. ARI		Avg. Complexity	
ρ	n	MFSF	funHDDC	LF2D	MFSF	funHDDC
M17	50	0.152	0.000	0.020	607.3	57049.6
5	50	0.050	0.000	0.015	623.0	57051.3
3.5	50	0.022	0.000	0.009	631.4	57049.3
7	250	0.081	0.093	0.443	549.2	128336.6
5	250	0.017	0.000	0.130	512.6	234839.5
3.5	250	0.020	0.000	0.040	520.9	235161.3
M27	50	0.330	0.003	0.021	715.8	57050.0
5	50	0.140	0.003	0.014	703.7	57057.8
3.5	50	0.050	-0.002	0.005	697.7	57051.5
7	250	0.330	0.192	0.700	725.4	119879.3
5	250	0.042	0.002	0.149	595.1	227179.0
3.5	250	0.003	0.001	0.041	585.2	226021.0
M37	50	0.844	0.000	0.038	1245.1	57050.0
5	50	0.354	0.000	0.017	1241.3	57049.3
3.5	50	0.138	0.000	0.001	1236.5	57072.3
7	250	1.000	0.105	0.837	1389.9	112860.4
5	250	0.998	0.002	0.159	1391.4	204289.5
3.5	250	0.984	0.010	0.050	1388.9	187345.4
M47	50	1.000	0.028	0.017	1257.0	47099.1
5	50	1.000	0.021	0.006	1255.7	47543.3
3.5	50	0.996	0.032	0.005	1258.1	47983.7
7	250	1.000	0.709	0.228	1387.4	121578.4
5	250	1.000	0.017	0.091	1387.4	230709.7
3.5	250	1.000	0.008	0.026	1387.5	229497.0
M57	50	1.000	-0.008	0.030	1260.6	46171.6
5	50	1.000	-0.009	0.023	1258.3	47111.1
3.5	50	1.000	-0.007	0.013	1256.2	47096.3
7	250	1.000	0.074	0.160	1387.2	210630.9
5	250	1.000	0.017	0.054	1387.2	231167.0
3.5	250	1.000	0.003	0.026	1387.3	234235.0

Table 5.7: Average ARI and model complexity results for each of the algorithms on the high dimensional datasets (18 × 33). Overall, high dimensional data seems to be troublesome for all models, as performance is mostly poorer when compared to the equivalent conditions in Table 5.6.

of the noise of the other, so that the difference in formulation does not have a severe impact on clustering performance. The story appears to be different in the high dimensional data scenario. According to Table 5.7, *funHDDC* did not able to perform well, on average, for any of the experimental settings. There are many potential reasons for this. For example, it could be that the chosen sample sizes are too low to properly inform the model at for the chosen data dimensions, or the initialization procedure was possibly not given enough opportunities to explore the likelihood landscape. In any case, since $M1 - M3$ are generated according to the modelling assumptions of *funHDDC*, it is very likely that this performance is indicative of some problem of implementation, and not of the methodology itself. The results of the MFSF methodology presented in Table 5.7, more closely resemble the results one would expect based on the simulation design. In particular, on data from models $M1$ and $M2$, MFSF performs poorly, achieving an average ARI of just 0.152 and 0.330 in the best case of each model, respectively. These are the models corresponding specifically to *funHDDC* modelling assumptions. However, Table 5.7 also shows that datasets from models $M3 - M5$ are clustered more favorably by MFSF, as the algorithm demonstrates excellent performance, on average, for almost all conditions corresponding to these models. Hence, the results of the MFSF algorithm suggest that the overlap is potentially important in these higher dimensional settings.

Recall that in Section 4.4 we mentioned that the models MFSF and *funHDDC* were such that even low values of the MFSF hyperparameters q and d correspond to relatively large values of the *funHDDC* hyperparameter k. Indeed, $\mathbf{q} = (2, 2)$ and $\mathbf{d} = (3, 3)$ were the values used for the MFSF hyperparameters in all models $M1 - M5$ in this study. These correspond to a value of $k = 34$ for the low dimensional setting and a value of $k = 114$ for the high dimensional setting for each of the mixture components. In the first case k is large relatively, accounting for more than half of the total data dimension $(66/34 < 2)$. Hence the *funHDDC* model is not parsimonously specified for these data. This is reflected in the average complexity of the fitted *funHDDC* model in Table 5.6. In the case of the high dimensional data, k is quite a bit smaller relatively, but now it is large absolutely. As a result, the average complexity of the *funHDDC* model also reaches large absolute values, sometimes even reaching values more than 100 times larger than the complexity of MFSF on the same data. Under such specifications, the parsimony of the model is lost and performance suffers.

5.5.5 Energy Sector Data

For more than a century the vast majority of the world's energy demands have been met by oil, gas, and coal. Continued dependence on these materials, collectively known as fossil fuels, is fueling two ballooning crises. First, the combustion of fossil fuels produces toxic emissions that pollute the Earth's air supply, negatively impacting public health and leading to an estimated 10.2 million premature deaths annually (Vohra et al., 2021). On top of their inherent toxicity, these emissions are also considered to be major agitators of the global climate crisis, which sees global average temperatures rising steadily toward dangerous levels. Second, the earth's repository of fossil fuels is finite and quickly becoming exhausted. In their 2020 review, the British multinational energy company BP noted that, we have 132, 50, and 49.8 years left respectively of coal, oil, and natural gas reserves, given 2019 production levels. Thus, when it comes to energy production, business as usual is no longer a viable option. With these crises looming, focus heightens on implementing sustainable, scalable, and clean energy solutions, with preference toward those that are based on renewable, rather than non-renewable, resources. In 2016 this sentiment lead to the Paris Agreement, which seeks to guide countries to minimize harmful emissions and reduce the increase in global temperatures. The agreement has been signed by all 197 members of the United Nations Framework Convention on Climate Change.

With so much discussion and attention surrounding energy production, we design an analysis that endeavours to develop better understanding of the global energy sector landscape. We begin by gathering data on 15 energy sector indicators from the World Bank's *World Development Indicators* database (https://databank.worldbank.org/source/world-development-indicators). The name and description of each chosen indicator is presented in Table 5.8. Of the 15 indicators, 6 are provided by the International Energy Agency (IEA). We have highlighted these indicators in Table 5.8 with an asterisk. All indicators are fully observed for 97 countries from the year 1993 to 2015, the year before signing of the Paris Agreement began. A full list of all countries included in the dataset can be found in Appendix B.1. Additionally, Figure 5.4 presents a map of the included countries, with colors assigned by our best fitted model.

The 15 included indicators fall into four general categories. The first category is titled

	Series	Description
Energy Production (Dirty)	Coal* (% of total)	Share of electricity generated by all coal and brown coal, both primary and derived fuels. Peat is also included in this category.
	Gas* (% of total)	The share of electricity generated by natural gas, excluding natural gas liquids.
	Oil* (% of total)	The share of total electricity generated by crude oil and petroleum products.
Energy Production (Clean)	Hydroelectric* (% of total)	Electrical energy from hydropower is derived from turbines being driven by flowing water in rivers, with or without man-made dams forming reservoirs.
	Nuclear* (% of total)	The share of electricity produced by nuclear power plants in total electricity production.
	Renewable* (not Hydro) (% of total)	The share of electricity produced by geothermal, solar, tide, wind, waste, primary solid biofuels, liquid biofuels, and charcoal. Hydro is not included.
	Renewable Consumption (% of total final energy consumption)	The share of renewables energy in total final energy consumption.
Rents	Coal Rents (% of GDP)	Coal rents are the difference between the value of both hard and soft coal production at world prices and their total costs of production.
	Forest Rents (% of GDP)	Forest rents are roundwood harvest times the product of regional prices and a regional rental rate.
	Mineral Rents (% of GDP)	The difference between the value of production for a stock of minerals at world prices and their total costs of production. Minerals included in the calculation are tin, gold, lead, zinc, iron, copper, nickel, silver, bauxite, and phosphate.
	Natural Gas Rents (% of GDP)	The difference between the value of natural gas production at regional prices and total costs of production.
	Oil Rents (% of GDP)	The difference between the value of crude oil production at regional prices and total costs of production.
Adjusted Savings	CO2 Damage (% of GNI)	Cost of damage due to carbon dioxide emissions from fossil fuel use and the manufacture of cement, estimated to be US$40 per ton of CO2 times the number of tons of CO2 emitted.
	Natural Resource Depletion (% of GNI)	Depletion of natural resources, which covers net forest depletion, energy depletion, and mineral depletion, reflects the decline in asset values associated with the extraction and harvest of natural resources - this is analogous to depreciation of fixed assets.
	Particulate Emission Damage (% of GNI)	The damage due to exposure of a country's population to ambient concentrations of particulates measuring less than 2.5 microns in diameter (PM2.5). Calculated as foregone labor income due to premature death.

Table 5.8: Descriptions for each of the energy sector variables used in our analysis. An asterisk denotes a measure provided by the International Energy Agency (www.iea.org/statistics. All rights reserved.)

	Fourier			B-Spline		
G	BIC	q	d	BIC	q	d
2	-342.41	4	2	10457.67	5	1
3	-11645.75	3	{1,2}	-3639.37	3	1
4	-18902.78	3	{1,2}	-12443.45	5	1
5	**-23311.65**	3	{1,2}	-16190.85	5	{1,2}
6	-21444.45	3	{1,2}	-16460.58	2	{1,2}

Table 5.9: Best model by BIC for each potential number of groups and for each possible basis, with corresponding latent factor dimensions and subspace dimensions.

Dirty Energy Production and includes the share of total electricity produced through means of fossil fuels. The title alludes to the dangers of an economy hinged to continued use of these fuels. These indicators give insight into the level of dependence each specific country's energy sector has on fossil fuels. The second category of indicators is labelled as Clean Energy Production. This includes both renewable energy sources, and non-renewable sources that result in low, or less dangerous emissions. These indicators highlight which countries are adopting cleaner alternatives, what those cleaner alternatives are, and the rate at which adoption has been taking place. The third category is Economic Rents pertaining to natural resources. These are calculated as the price of a commodity minus its average production cost, multiplied by the extracted haul of that resource for the particular country. These are reported as a percentage of GDP, and they give an indication of the importance of these materials, not just in terms of producing energy to run the economy, but also in terms of generating revenue on the global marketplace. The final group is titled Adjusted Savings. These attempt to measure some of the economic costs associated with the production of energy through fossil fuel combustion. Higher values of these variables imply higher damages and costs, and are therefore worse.

The collected set of indicators are transformed into functional data using basis projection, and we carry out this process using the `fda` package (Ramsay et al., 2009). We project the data onto both B-spline and Fourier bases, each comprised of 14 basis functions. We use MFSF to cluster the resulting functional data, intending to identify groups

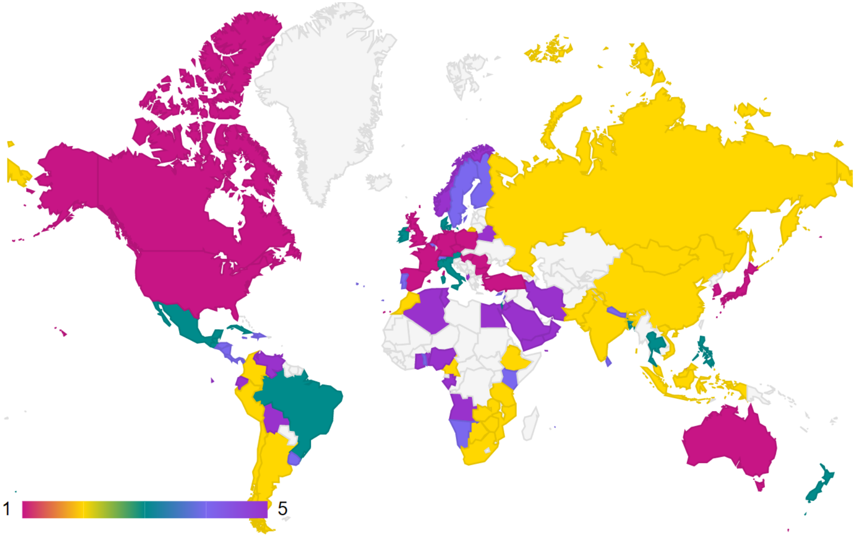


Figure 5.4: A plot illustrating the found clusters, identified by color. Countries in gray are those for which data were not available. We note that many countries in the same group share a border or a continent. This suggests that energy sectors are locally similar, which may reflect that countries in the same geographic vicinity are likely to have similar distributions of natural resources.

of countries with similarly structured energy sectors. When fitting, we choose the hyperparameter space as follows: we allow $G \in \{2, 3, 4, 5, 6\}$ and $q \in \{1, 2, 3, 4, 5\}$ for each group. We choose the latent subspace dimension d using the thresholding approach, implying that its value is bounded above by the number of basis functions. For computational reasons, we fix q across model components. For each unique combination of hyperparameters, we generate 1000 sets of random parameters and use them to initialize a stochastic EM algorithm, which is subsequently run for 500 iterations. We retain the parameters associated with the maximum achieved likelihood value and use them to initialize a regular EM algorithm which is run until convergence. By way of this process, the best fitted model for each choice of G is given in Table 5.9.

Inspecting the BIC values displayed in Table 5.9, we note that for each specified G, the Fourier basis produces better BIC values than the corresponding B-spline basis, while B-spline projection seems to result in a larger proportion of variance being explained by

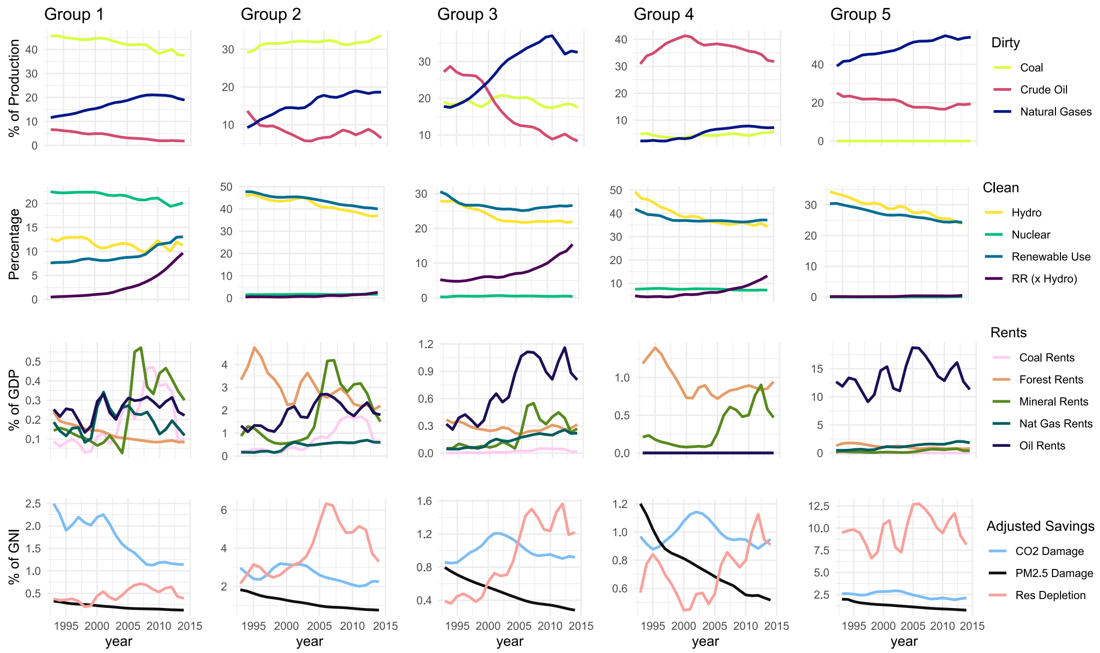


Figure 5.5: Plots of the univariate components of the mean function fitted to each of the five groups in the best BIC model. Partially based on IEA data from the IEA, www.iea.org/statistics. All rights reserved; as modified by Alex Sharp.

the first eigen-function. Overall, projecting the data into the Fourier space and fitting a five component mixture with latent factor dimension $q = 3$ for each group results in the best fit. The groups assigned by this model are represented visually in Figure 5.4. A table listing the countries that belong to each group can be found is provided in Appendix B.1. As a first preliminary analysis of the assigned groupings, Figure 5.4 seems to suggest that the clustering algorithm is identifying trends in the data relating countries both geographical and by the size of their economy. We posit that these relations could certainly be represented latently within energy production and economic rents, or that is, correlated with resource abundance and therefore usage.

To get a sense of the general properties of each fitted group, we analyze the mean functions, presented in Figure 5.5. Here, columns comprise the mean function for a specified group, while rows correspond to related sets of energy sector indicators, as defined

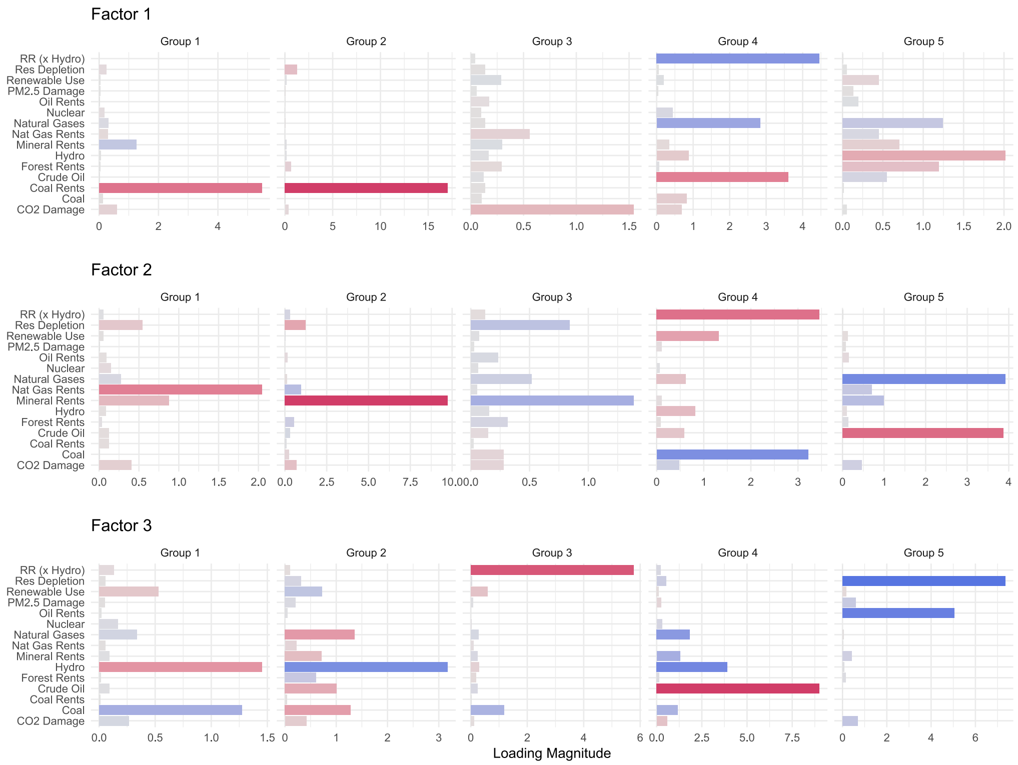


Figure 5.6: The fitted factor loadings for each group. Columns correspond to groups, while rows correspond to factors. The values presented are the results of a varimax rotation on each of the loading matrices separately.

by the partitions of Table 5.8. Glancing over dirty energy production, we notice global trends of increasing natural gas dependence and decreasing oil use, with the rates of these changes varying across the groups. Group 3, which consists of countries such as Brazil, Mexico, and Italy, seems to have the fastest rate of natural gas adoption, while group 5, which is comprised of countries such as Iran, Norway, and Saudi Arabia, boast the highest overall percentage of energy generated from natural gas, at over 40 percent. This may be an indication that the world overall was concerned with CO2 emissions ahead of the Paris Agreement as energy generated by natural gas produces lower overall emissions than its other dirty energy counterparts. This shift toward natural gas could be seen as an intermediate step toward transitioning away from fossil fuels, as the reduced emissions buy countries time to develop the infrastructure needed to support cleaner methods.

One clear distinction across groups is coal useage. Group 1, which consists of Canada, America, most of Europe, and Australia, reports the highest percentage of energy produced by coal. As we move across from group 1, across the plot from left to right, coal use progressively decreases in an almost systematic, stepwise fashion, with group 5 exhibiting almost zero coal dependency for energy production. There are clear distinctions across the groups in terms of renewables as well. Group 1 seems to be the only group with a non-negligible portion of energy produced through means of nuclear power. This group also appears to possess the steepest increase in use of renewables, although this is closely rivaled by the uptake of renewables in group 4. The title of most energy produced through renewables on average, however, belongs to group 3. Hydro appears to be the largest source of clean energy for all groups besides 1, although Hydro also touts a decreasing average trend across all groups. Overall, group 2, which is composed of countries such as Argentina, China, India, Russia, and South Africa, is the only one in which the dominate method of energy production is a clean resource (hydro).

The factor loadings used to model the coefficients of the data on the first eigenvector are presented in Figure 5.6 for each group. Recall that the loading matrix is the same up to a scalar multiple across all eigenvectors, so that this analysis holds for the coefficients on all eigenfunctions up to a rescaling of the x-axis. For the first three groups, the first latent factor is synonymous with a single variable. For group 1 and 2 that variable is `Coal Rents`, whereas for group 3 the variable is `CO2 Damage`. For group 4, `Renewables`, `Natural Gases` and `Crude Oil` have the largest loadings on the first latent factor. The first two are in opposition of the third, implying that this latent factor represents a measure of adoption of cleaner fuels. The first latent factor of group 5 is a bit harder to unpack. We first notice that there are large, opposite sign loadings on `Hydro` and `Natural Gases` so that this factor measures some tension between these two methods of energy production. Additionally, we see non-negligible loadings on `Forest Rents` and `Mineral Rents`, with these sharing the same sign as `Hydro`. We also see non-negligible loadings on `Nat Gas Rents` and `Crude Oil` having the same sign as the loading on `Natural Gases`. Recalling that the countries comprising Group 5 produce no energy from coal on average, we conclude that this is an overall cleanliness factor for the energy and resource sector.

Group	Mean Features	Significant Factors	Notable Members
1	• highest coal and nuclear energy production • Resource rents not strong contributors to GDP • Lowest PM2.5 Damage and Resource Depletion	• Coal rents • Natural gas rents • Hydro and coal tradeoff	Australia, Canada, Germany, Japan, South Korea
2	• Hydro is largest energy source • Increasing coal use • High forest and mineral rents; strong rents overall • Second worst in resource depletion and CO2 damage	• Coal Rents • Mineral Rents • Hydro and fossil fuel tradeoff	Argentina, China, India, Russia, South Africa
3	• Fastest increase in natural gas use, greatest reduction in oil use. • Best renewable use outside of hydropower • Oil is only significant rent • Increasing resource depletion, low PM2.5 damage	• CO2 damage • Resource extraction (heavily influenced by minerals) • Renewable resources versus coal	Brazil, Italy, Mexico, New Zealand, Thailand
4	• Highest oil production, lowest natural gas • High rate of renewable adoption; second best nuclear and hydro • Forest and mineral only non-negligible rents • Largest relative change in PM2.5 damage	• Oil versus renewable resources • Coal versus renewable resources • Oil versus hydro	Costa Rica, Dominican Republic, Kenya, Sweden, Uruguay
5	• Highest natural gas use, lowest coal use • Overall lowest percentage of energy by clean methods • Extremely high oil rents • Very high resource depletion	• Hydro and non-fuel rents versus fossil fuels and fuel rents (cleanliness of sector) • Natural gas and crude oil tradeoff • Oil extraction	Albania, Iran Nigeria, Saudi Arabia, Venezuela

Table 5.10: A summary of the characteristics of each group found by the best BIC model.

For groups 1, 2, and 3, the second factor is mostly comprised of a single rent, whereas for groups 4 and 5, the second factor represents the exchange of a dirty energy source for a cleaner one. For all but group 5, factor 3 represents a shift in energy production. For group 1 and 4, this takes the form of exchanging `Coal` for `Hydro`, for group 2 it takes the form of `Hydro` versus fossil fuels, and for group 3 its a disproportionate tradeoff between energy produced by renewable resources and `Coal`. Finally for group 5, the third factor is comprised of large, same sign loadings on `Oil Rents` and `Resource Depletion`. Therefore, we can interpret this latent factor as representing the state of a countries oil industry.

Collecting the findings of our analysis into Table 5.10, we make some concluding remarks regarding the state of each energy sector group in terms of the aforementioned global crises. We emphasize that these comments are based solely on the high level data summary presented by the model, and we realize that real world circumstances surrounding energy production are far more complex than what we present here. Accordingly, our comments should not be interpreted as praise or condemnation of any particular country's business, nor should they be interpreted as comprehensive advice that we believe countries should actually act on.

Group 1 reports the highest average dependency on coal among the five groups, and this source stands out as this groups largest contributed to energy production on average across the entire period of study. The glimmer of positivity here is that these countries also showed increasing adoption of renewable fuels, as well as boasted the most well established nuclear sector. Members of this group should continue to build on these promising trends, and use the gains from investing in these cleaner sources to reduce dependency on coal.

Among all five groups, Group 2 was the only one in which the largest contributor to energy production was a renewable resource throughout the entire period of study. This is a great achievement, but some concerns linger. Although it was the largest contributor, the percentage of energy generated by hydropower presented a decreasing average trend, falling by 10% over the course of the 22 year period covered by the dataset. Over this same time period, the contribution of coal was seen to rise about 5% on average. Additionally, outside of hydropower, the countries comprising Group 2 seem to be relatively uninterested in renewable energy sources. Whether this is a function of resource availability, indifference, or insufficient infrastructure is unclear, however, it is an issue that these countries will need to address if the stipulations of the Paris Agreement are to be met. Even ignoring global concerns, Group 2 also presented some of the worst trends in the adjusted savings category, suggesting that it is also in the personal interest of these countries to make an effort to alleviate fossil fuel dependence. However, this choice may be made harder by the fact that Group 2 reported some of the most significant fossil fuel rents.

Group 3 presents itself as a set of countries whose energy sectors are in a state of transition. This is observed in plummeting oil use parallel to rising natural gas use, and is additionally supported by exponential-like increases in energy contribution from renewables. A slight blemish is that these increases are matched by decreased utility of hydropower, which results in a relative decrease in clean energy consumption by this group over the period of study. On the whole, these countries seem to be on a good path toward successfully overhauling their energy sectors for more sustainable production.

On the surface, Group 4's consistently high oil use and increasing resource depletion seem to paint a picture of countries rooted firmly in the fossil fuel industry, however a closer look reveals a different story. Like Group 2, this group also reports high contributions in energy production from hydropower, but unlike Group 2, there is a clear interest in other renewable sources of energy outside of hydro. Further, low average oil rents throughout the period of study suggests these countries may not resist adoption of alternative energy solutions as they become increasing viable. Outside of oil, Group 4 already seems relatively unattached to fossil fuels, with some of the lowest average contributions from both coal and natural gas among the five groups, along with negligible rents. Finally, relatively low CO2 and PM2.5 damages suggest these countries are well on their way to achieving a clean energy revolution.

Group 5 is the least dependent on coal of all groups, being nearly equal to 0% of overall production for the entire period of study. Natural gas use in this group is also noteworthy—in 1993 natural gas in this group was, on average, higher than is ever achieved by any other group, and the trend steadily rises as time passes. Despite negigible coal use, extremely high resource depletion numbers coupled with declining clean energy use paints a picture that these countries are not preparing well for the future and have energy sectors completely entrenched in the fossil fuels industry. This is likely perpetuated by obscenely large oil rents, suggesting that these countries may be wary to move on from fossil fuels because

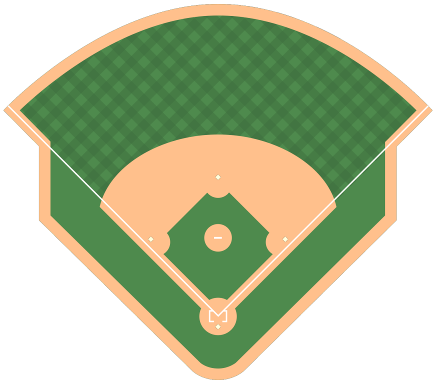


Figure 5.7: A typical baseball field. The white diamonds and the encompassing sandy clay area compose the infield, while the large section of green grass beyond it comprises the outfield. The brown circle with the white bar in the middle of the diamond represents the mound from which the pitcher throws, while the box at the bottom of the diamond is where the batter stands.

they are proving to be very lucrative. However, owing to the fact that we will run out of oil globally within the current generation, these countries would be wise to start investing their money in energy sources of the future.

5.5.6 Baseball Pitch Trajectory Data

Baseball is a sport played between two teams, wherein each team alternates between fielding and batting. The game has no time limit, rather, it progresses as the fielding team produces “outs" against the batting team. When three outs are achieved, teams switch roles, with the batting team moving to the field, and the fielding team taking up batting. Once each team has both fielded and batted nine times, the game finishes. The game proceeds in segments, wherein a player on the fielding team, called the pitcher, pitches (throws) a ball toward a player on the opposing team, who must then try to swing and hit the ball. The pitcher is penalized if the throw is errant, while the batter is penalized if the throw is good but not struck. A penalty against the pitcher is called a “ball" while a penalty against the batter is called a “strike." If the pitcher manages to rack up three strikes

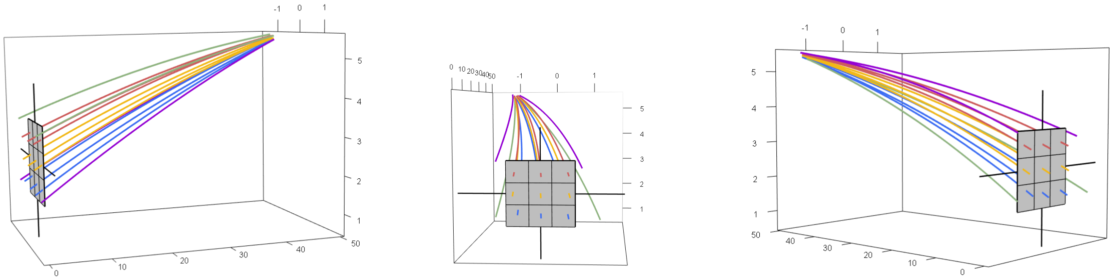


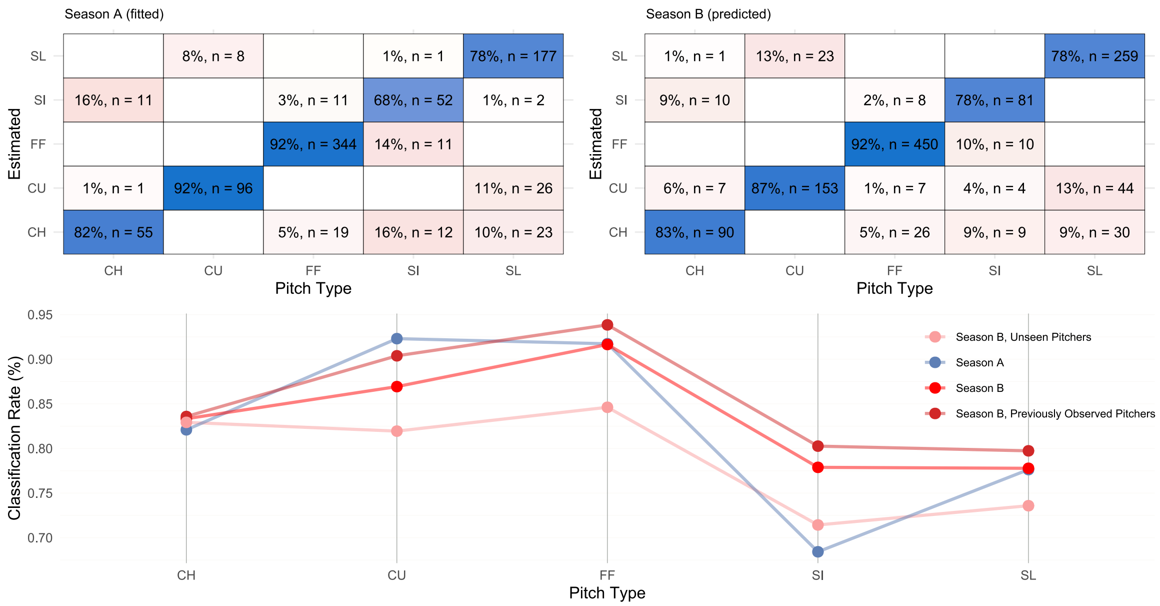
Figure 5.8: An example observation from our constructed dataset. The plot shows the average fastball trajectory of MLB pitcher Jacob deGrom to each of the 13 zones defined by the variable zone within the dataset. All axis measurements are in feet; the aspect ratio has been adjusted for concise presentation.

against the current batter, the pitcher's team accumulates an out, and the batter must return to the bench. The batter is replaced with a new player from the batting team, the number of balls and strikes are reset to zero, and play continues. Points are scored by the batting team, and are accrued by batters hitting a pitched ball and proceeding to run the bases (see Figure 5.7) before getting tagged with the ball by the fielding team. The team with the most points at the end of the game is deemed the winner. Given that the batting team can (generally) only score points by making contact with a pitched ball, one quickly realizes that good pitching is integral to winning a baseball game. Good pitchers are those who have an assortment of different pitches—particular ways of throwing the ball—that are difficult for the opposing batter to hit, identify, or adjust to. There are many types of pitches used in baseball, but there are five main types that find ubiquity at the major league level: four-seam fastball (FF), curveball (CU), slider (SL), two-seam fastball (FT/SI), and changeup (CH). For a short discussion on the role and proprieties of each pitch, see https://www.mlb.com/glossary/pitch-types. We note that the type and role of a given pitch is intimately related to its trajectory. That is, despite natural variations across realizations of any particular pitch type, which occur due to differences in physical stature of the pitcher or nuisances in the delivery of the pitch, each one is quite easily identified through visual inspection of its associated trajectory. Interest now lies in assessing whether or not MFSF can do the same, that is, identify similar pitches through trajectory analysis.

Since 2006, Major League Baseball (MLB) has been collecting pitch trajectory data for every pitch thrown in every game played during both the regular season and the playoffs. These data have been made publicly available, and an R function for scraping the data can be found in Appendix B.2. For our analysis, we collect all pitches thrown from two different seasons, which we label Season A (2019) and Season B (2018), by right-handed pitchers. Left-handed pitchers are not included in our analysis due to the reflected nature of their pitch trajectories with respect to right-handers. One could potentially include southpaws if all included trajectories were reflected to appear as thrown from the same hand (left or right).

Unfortunately, MLB does not provide the raw trajectory data, rather, a three-dimensional constant acceleration model for space curves is fit, and trajectory data are released publicly in the form of parameter values estimated by the fitted model. Details regarding the fitting procedure can be found in Nathan (2008). Utilizing the model formulation provided in Nathan (2008), we use the coefficients recorded in our dataset and generate 1000 points along each trajectory. Each pitch in the dataset then represented by a 3×1000 matrix. These are then projected onto a 29 dimensional B-spline basis, and the resulting 3×29 matrix of coefficients are extracted. The resulting matrices do not exhibit large enough row dimensions to properly take advantage of the properties of MFSF. We therefore seek an alternative representation for these data that result in increased row dimension. To do this, we appeal to the context of the data as pitch trajectories.

For every recorded pitch, MLB also provides copious contextual meta-data, such as which batter faced the pitch, who threw the pitch and much more (a complete list can be found at https://baseballsavant.mlb.com/csv-docs). One of the provided meta-data variables, titled `zone`, is of particular interest. As previously mentioned, a baseball pitch is always thrown by a pitcher on the mound in the direction of a batter who stands at the bottom of the playing diamond directly next to a special base called “homeplate" (Figure 5.7). The variable `zone` records the location of the pitched ball as it crosses the front of homeplate based on predefined zones. That is, the vertical plane in front of homeplate, where the batter stands, is separated into 13 different zones, and pitch locations are distinguished using these. A visual representation of these zones, and a set of fastball pitch trajectories, one to each zone, is presented in Figure 5.8.



Season A (fitted)

Estimated	CH	CU	FF	SI	SL
SL		8%, n = 8		1%, n = 1	78%, n = 177
SI	16%, n = 11		3%, n = 11	68%, n = 52	1%, n = 2
FF			92%, n = 344	14%, n = 11	
CU	1%, n = 1	92%, n = 96			11%, n = 26
CH	82%, n = 55		5%, n = 19	16%, n = 12	10%, n = 23

Season B (predicted)

Estimated	CH	CU	FF	SI	SL
SL	1%, n = 1	13%, n = 23			78%, n = 259
SI	9%, n = 10		2%, n = 8	78%, n = 81	
FF			92%, n = 450	10%, n = 10	
CU	6%, n = 7	87%, n = 153	1%, n = 7	4%, n = 4	13%, n = 44
CH	83%, n = 90		5%, n = 26	9%, n = 9	9%, n = 30

Figure 5.9: [Top Left]: The confusion matrix for the model chosen by BIC. [Top Right]: The confusion matrix for predictions by the chosen model on unseen data from Season B. [Bottom]: A break down of classification rate by pitch type, for Season A and Season B. Also included is the prediction accuracy on the subgroup of Season B corresponding to pitchers who also threw in Season A (n=848) and the subgroup of those who did not (n=364).

Each trajectory presented in Figure 5.8 is calculated as the average trajectory, within each zone, of fastballs thrown by one particular pitcher (in this case, New York Mets' star pitcher Jacob deGrom). In a way, the set of trajectories displayed in Figure 5.8 could be said to represent Jacob deGrom's fastball more holistically than any particular, single realized trajectory does. We therefore proceed as follows: we group observations in the dataset by pitcher name and pitch type, and we keep each group that consists of at least 100 observations. This results in 1068 unique pitcher-name/pitch-type pairings. We further partition the trajectories of each retained pairing according to the variable `zone`. We then take the average of the trajectories in each subset of the partition, which produces thirteen, 3 dimensional trajectories that together form a 39 dimensional space curve describing the associated pitcher-name/pitch-type pairing. We note that this space curve has an interpretable embedding in 3-space, of which an example is plotted in Figure 5.8. These data are transformed to functional data using the approach previously detailed, so that each curve pertains to a 39×29 matrix of coefficients on a B-spline basis. Our hope for cluster analysis on these data is to find groups that are homogeneous with respect to pitch type.

Before fitting the model, we must determine appropriate hyperparameter values. The total number of different pitches in the dataset is known a priori, hence we set $G = 5$ accordingly. We allow the latent subspace dimension d to be chosen using the thresholding method, with threshold set to 0.9. To find a good value of q for each component, we adopt an approach reminiscent of the *em*EM algorithm of Biernacki et al. (2003). The allowable values of the latent factor dimension q is $\mathcal{Q} = \{3, 4, 5, 6\}$, and for each unique 5-tuple $\mathbf{q} = (q_1, ..., q_5)$ of elements in $\mathcal{Q}$, we fit the model with the latent factor dimension of group i specified by q_i. For each $\mathbf{q}$ we initialize 20 runs of stochastic EM using random soft memberships and run each one for 100 iterations, saving the parameter values that result in the best BIC. The 5-tuple $\mathbf{q}$ which produces the overall best BIC value is then considered the best choice for the latent factor dimensions.

With hyperparameter values set, we now focus on fitting the model. We initialize 100 stochastic EM algorithms using randomly generated soft membership matrices $\mathbf{Z}$, and run each for 500 iterations. We initialize one additional model using the best parameter set from the selection of $\mathbf{q}$. The set of parameters resulting in the lowest BIC is then used to initialize an AECM algorithm which is run until convergence. Although both $\mathbf{q}$ and G are fixed, BIC is still used for selection at this stage to account for potential variation across the estimation of d.

The resulting best model achieves a BIC of 2158103, and reports a correct classification rate of 0.852 and an ARI of 0.747. The top left plot of Figure 5.9 depicts the confusion matrix for this model. Each component of the model approximately corresponds to a single pitch type, as hoped. For each component, the associated latent factor and subspaces dimension are $(3, 5)$ for the CH component, $(3, 6)$ for the CU component, $(4, 7)$ for the FF component, $(6, 6)$ for the SI component, and $(3, 7)$ for the SL component. The model has therefore represented the data with satisfying amounts of parsimony, and the heavy diagonal in the top left plot of Figure 5.9 suggests that this representation produces a workable approximation.

More specifically, Figure 5.9 shows that the model does a good job at representing fastballs (FF), curveballs (CU) and changeups (CH), however, it does relatively poorly at identifying sinkers (SI) and sliders (SL). Specifically, sliders are frequently identified as curveballs or changeups, while sinkers are often guessed to be fastballs or changeups. These misclassifications are reasonable in the baseball context. Sliders are quite similar to curveballs, as both pitches trace a curved trajectory, while sinkers are essentially fastballs thrown just a bit slower, producing an earlier dip. Changeups are a still slower version of the sinker, albeit with a small degree more lateral movement.

In the interest of assessing the generality of our fitted model, we use it to predict pitch type labels on the data accrued for Season B. The resulting confusion matrix is given in the top right portion of Figure 5.9. Overall, the model produces numbers on these unseen pitches similar to those reported on the pitches used for fitting. The one exception is 78% of sinker labels correctly predicted, marking a 10% increase over what would have been expected based on the fitted model. We need to recognize, however, that Season B contains data on some of the same pitchers who are included in Season A. Although the Season A and Season B datasets are temporally partitioned and therefore share no common observations, we suppose that the same pitch thrown by the same pitcher is likely to be correlated across time. It follows that some of the observations in our Season B cannot be truly considered as independent from those of Season A. We account for this issue by including the graphic presented at the bottom of Figure 5.9. This plot presents the same information as the twin confusion matrices, however, it additionally decomposes the results on the Season B pitches into those pitches thrown by pitchers who are also present in the Season A dataset (n=848), and those who are not (n=364). The results on previously unseen pitchers is closer to what one would expect from prediction, although we note that even in this case the prediction on sinkers is better than what was fitted to the original data.

5.6 Conclusion

In this work, we have defined and investigated a parsimonious, finite mixture model for clustering observations of multivariate functional random variables. We have demonstrated that modelling multivariate functions as a collection of dependent univariate functions on the same domain, combined with the model-based assumption of a matrix normal distribution on the functional principal components, results in a model which flexibly models the function space and the coefficient space. Indeed, this formulation facilitates the specification of different levels and styles of parsimony in each of these spaces for each mixture component, according to the nuances of the data under investigation. An analysis of multiple country energy sectors demonstrates the interpretability this brings to modelling, as we were able to discuss the contributions of separate univariate functional covariates in the context of the energy sector whole. An analysis of baseball pitch trajectories showed that the model also attains good performance in the case of a large spatial dimension and moderate basis dimension, as it was able to cluster the trajectories into groups which roughly corresponded to the different pitch types, and it did so with a satisfying amount of parameter parsimony. For future work, we have two suggestions. Previous work allows one to choose the subspace dimension d_g for each component without the computational burden of a space search, however a search was still required for the latent factor dimension hyperparameter q_g. As far as we know, there are no methods for automatic choice of the latent dimension in factor analysis which can be seamlessly implemented alongside the maximum likelihood estimation of the parameters. Discovering a convenient way of choosing q_g across the components would further ease computational burden of model fitting, potentially allowing a good model to be fit in a single run (using SEM initialization and assuming the number of model components is known). Additionally, one may look at alternative specifications of parsimony for either $\mathbf{\Sigma}_1$ or $\mathbf{\Sigma}_2$ which may produce a model formulation which is more natural to some particular context.

Chapter 6

Maximum Contribution to the Likelihood: Increasing the Precision of Estimation with the Stochastic Expectation-Maximization Algorithm

6.1 Introduction

The expectation-maximization (EM) algorithm (Dempster et al., 1977) is a well known method for obtaining maximum likelihood estimates of parameters for latent variable models. The algorithm is generally composed of two steps, an expectation (E) step, and a maximization (M) step. Depending on the chosen data model, one or both steps may be complex or intractable. The stochastic EM algorithm (Celeux and Diebolt, 1985; Diebolt and Ip, 1995; Celeux et al., 1995) is a methodology which aims to deal with cases of a problematic E-step. In so-doing, it may also occassionally simplify the M-step (Celeux et al., 1995). For a given parameter value, the stochastic EM algorithm imputes the latent variables by drawing from their conditional distribution given the observed data. One may then approach the M-step as if the data were fully observed. This turns out to be equivalent to estimating the value of the expectation in the E-step using Monte Carlo integration based on a single sample (McLachlan and Krishnan, 2007). This imputation step has been dubbed the S-step (Celeux and Diebolt, 1988; Diebolt and Ip, 1995), alluding to its stochastic nature.

By replacing the deterministic E-step with the stochastic S-step, one sacrifices some of the desirable convergence features of the EM algorithm (Dempster et al., 1977; Wu, 1983). For example, stochastic EM does not guarantee monotonic increase in the observed data log-likelihood, nor does it converge to a limit point, local or otherwise. However, the S-step does bring the ability to escape from local maxima and saddle points, making the algorithm less sensitive to initial conditions. Along with the simplicity of the S-step, this property makes the stochastic EM algorithm a fairly attractive approach for estimating latent variable models. For example, it has recently been employed in the estimation of a large-scale, full-information item factor analysis model (Zhang et al., 2020), a latent regression item response theory model (Chen et al., 2022), a Markov-modulated diffusion risk model (Baltazar-Larios and Esparza, 2022), and a shape invariant model for co-clustering of time-dependent data (Casa et al., 2021).

Under standard regularity conditions it was shown by Nielsen (2000) that the chain produced by the stochastic EM algorithm is an ergodic and irreducible Markov chain. He further showed that the stationary distribution is asymptotically Gaussian and centered at the maximum likelihood estimate (Similar results were shown in Celeux and Diebolt, 1985, 1987, for a more restricted class of models). These results also hold for the related Monte Carlo EM (Wei and Tanner, 1990), which differs from stochastic EM by allowing the S-step to utilize multiple draws from the conditional distribution of the latent variables.

From Nielsen (2000), we immediately get that the chain itself is a consistent and asymptotically Gaussian estimator. However, convergence of the stochastic EM algorithm to a stationary distribution, rather than a limit point, means the last element of the algorithm chain is not necessarily the optimal choice for point estimation. A more efficient estimator, and the most commonly used in application, is the average of the chain's tail. This estimator was investigated by Ip (1995) for exponential family models, while Nielsen (2000) continued the investigation under more general conditions. The latter showed that, asymptotically in the sample size and with the chain tail having fixed length, the tail average is consistent and has an asymptotically Gaussian distribution under suitable normalization.

An alternate estimation scheme chooses the element of the chain with the largest likelihood value. This approach appears to be first mentioned in Diebolt and Ip (1995), where it is noted as a potential alternative to the tail average. In Biernacki et al. (2003), this estimator is put to the test in the context of Gaussian mixture models. Therein, however, it is used as an initialization scheme, in that the value produced by a stochastic EM algorithm chain with the largest likelihood value is used as the starting point of a traditional EM algorithm.

Our first step is to demonstrate that the estimator obtained by selecting the element of the chain with the highest likelihood possesses a precision that is the square of the precision of the tail average, assuming that the model parameter is scalar-valued. Furthermore, we establish that the estimator follows an asymptotic Laplace distribution.

The purpose of the present chapter is to propose an estimator, based on this alternate scheme, which achieves greater precision than the tail average. We first prove that the estimator obtained by selecting the element of the chain with the highest likelihood possesses a precision that is the square of the precision of the tail average, assuming that the model parameter is scalar-valued. Furthermore, we establish that the estimator follows an asymptotic Laplace distribution in this same context. We then show that this estimator suffers from the curse of dimensionality in that, as the model parameter dimension increases, this estimator becomes bounded away from the maxmimum likelihood estimate with high probability. We therefore recommend against the use of this estimator in practical applications. However, the desirable behaviours of this estimator in the scalar-valued parameter case motivate a new estimator. The proposed estimator utilizes the marginal algorithm chains and the profile likelihood to extend the scalar-valued model parameter properties of the largest loglikelihood estimator, namely its increased precision, to the high-dimensional parameter case. We prove that the proposed estimator is consistent and has asymptotically Laplace marginals, under suitable normalization.

6.2 Preliminaries

6.2.1 Notation

The context in which the EM algorithm operates starts with the assumption that there exists a random variable Y whose values represent complete information regarding some process or object of interest. This random variable can only be partially observed, and we label the observable part with X. We denote the latent unobservable data as Z, so that we have $Y = (X, Z)$. We suppose X has density $f(x \mid \theta)$, with respect to a σ-finite dominating measure denoted by dx, and support $\mathcal{X}$. Similarly, Z is assumed to have support $\mathcal{Z}$ and conditional density $k(z \mid x, \theta)$ given X, with respect to the dominating measure dz. Finally, Y has support $\mathcal{Y} = \mathcal{X} \times \mathcal{Z}$ and density $g(y \mid \theta)$ with respect to dominating measure $dy = dx \times dz$. We then have

$$f(x \mid \theta) = \int_{\mathcal{Z}} g(y \mid \theta) dz, \quad k(z \mid x, \theta) = g(y \mid \theta) f(x \mid \theta)^{-1}.$$

We denote the parameter space of the model by $\Theta \subseteq \mathbb{R}^p$, and we let $\theta_0 \in \Theta$ denote the true underlying value of the parameter. Score and information functions for the random variables of interest are identified by a subscript, e.g. $s_x(\theta)$ denotes the score of the observed data X, and $\mathbf{I}_{z|x}(\theta) = E(s_{z|x}(\theta) s_{z|x}(\theta)^{\mathrm{T}} \mid X = x)$ denotes the information of the latent data Z given $X = x$.

We use $X_{1:n} = (X_1, \ldots, X_n)$ to denote a random sample of the observed data, while $x_{1:n}$ denotes a realization. The related objects $Y_{1:n}$, $y_{1:n}$, $Z_{1:n}$ and $z_{1:n}$ are defined similarly. Additionally, we will sometimes use $f(x_{1:n})$ to denote the sequence of evaluations $(f(x_1), \ldots, f(x_n))$ in the interest of maintaining compact notation.

We let $\ell_x(\theta)$ denote the observed data log-likelihood, $\sum_{i=1}^n \log f_x(x_i \mid \theta)$, with the dependence on n being suppressed. For a stochastic EM algorithm based on the sample $x_{1:n}$, we let $\tilde{\theta}_{n,k}$ denote the kth element of the resulting chain, while $\tilde{\theta}_{n,k:t} = (\tilde{\theta}_{n,k}, \ldots, \tilde{\theta}_{n,k+(t-1)})$ denotes the chain section of length t which begins at index k. We use $\tilde{\theta}_n$ to denote the random variable distributed according to the stationary distribution of the stochastic EM Markov chain. Finally, $\hat{\theta}_n$ denotes the maximizer of $\ell_x(\theta)$.

6.2.2 The Stochastic EM Algorithm and Parameter Estimation

Given a sample $x_{1:n}$, interest lies in estimating the parameter θ associated with the distribution of X. The EM algorithm is typically employed in cases where direct estimation of θ by maximization of the observed data log-likelihood $\ell_x(\theta)$ is difficult, but maximization over the associated complete-data log-likelihood is comparatively simpler. Of course, in practice the complete-data likelihood cannot be computed because it requires knowledge of the unobserved values $z_{1:n}$. The EM algorithm circumvents this by instead considering

$$Q(\theta \mid \hat{\theta}) = E\left\{\log g(y_{1:n} \mid \hat{\theta}) \mid x_{1:n}, \hat{\theta}\right\} = E\left[\sum_{i=1}^{n} \log g\big\{(x_i, Z_i) \mid \theta\big\} \mid x_{1:n}, \hat{\theta}\right],$$

where $\hat{\theta}$ is the current best guess for the unknown parameter θ, and the expectation is taken with respect to the conditional distribution $\prod_{i=1}^{n} k(z_i \mid x_i, \hat{\theta})$. Computation of $Q(\theta \mid \hat{\theta})$ composes the so-called E-step of the EM algorithm. Despite the supposed simplificiations inherited by shifting efforts to the complete-data model, the E-step can still present difficulties due to the presence of integration. Stochastic EM allows one to avoid a problematic E-step by instead performing the following S-step: At iteration k, given current parameter estimate $\tilde{\theta}_{n,k}$, complete the observation x_i with a draw from the conditional density $k(\cdot \mid x_i, \tilde{\theta}_{n,k})$, for each x_i. Denote the simulated value associated with x_i by $\tilde{z}_i$. The completed pseudo-sample produced by the S-step is then utilized at the M-step by maximizing $n^{-1}\sum_{i=1}^{n} \log g\big\{(x_i, \tilde{z}_i) \mid \theta\big\}$. The execution of both stochastic EM steps produces a new parameter value, denoted by $\tilde{\theta}_{n,k+1}$.

Construction of a point estimate from the resulting chain, $\tilde{\theta}_{n,1:t}$, is typically done in one of two ways. The tail average, denoted by $\bar{\theta}_{n,m}$, is the point estimator which averages the last $m \leq t$ elements of the chain. Alternatively, one may take the element of the chain which produces the largest value of observed log-likelihood function $\ell_x(\theta)$. Equivalently, one may express this estimator in terms of the *likelihood disparity function* (LDF) which is defined as $D_x(\theta) = -2\{\ell_x(\theta) - \ell_x(\hat{\theta}_n)\}$. The LDF is just the likelihood ratio test statistic viewed as a function of θ rather than the observed data $x_{1:n}$. The estimator is then expressed as $\tilde{\theta}_{n,\min t} = \operatorname{argmin}_{k=1,\ldots,t} D_x(\tilde{\theta}_{n,k})$. This latter perspective is more amenable to the derivation of theoretical results, and hence is the perspective taken in the sequel.

6.2.3 Framework

Conditonal on the observed data, parameter estimators obtained from the stochastic EM algorithm still exhibit randomness due to the stochastic nature of the S-step. In order to understand the behaviour of these estimators, then, we must establish the nature of the randomness due to the algorithm as well as the data models. We begin with a modelling assumption.

Assumption 1. *We assume the models of X, Y, and Z satisfy the conditions outlined in Nielsen (2000). We additionally assume ergodicty of the Markov Chain $\tilde{\theta}_{n,k}$.*

Details on the data models can be found in Section 2.3 of Nielsen (2000), while conditions for ergodicity of the Markov chain $\tilde{\theta}_{n,k}$ are outlined in Theorem 1 of Nielsen (2000) and elaborated on in the subsequent remarks. Of particular importance to the present analysis is regularity of the observed data model. Specifically, it is assumed that the observed data log-likelihood $\ell_x(\theta)$ behaves asymptotically as a quadratic function in the vicinity of its maximum. More formally, for $\theta_n = \hat{\theta}_n + n^{-1/2}h$ we have

$$D_x(\theta_n) = h^{\mathrm{T}} I_x(\theta_0)h + r_n(h), \tag{6.1}$$

where $r_n(h)$ goes to zero uniformly over compact sets containing the maximum likelihood estimate as $n \to \infty$ and θ_0 is the true underlying value of the parameter.

Under Assumption 1, Nielsen (2000) showed that the stochastic behaviour of the algorithm can be specified. Specifically, for the Markov chain $\tilde{\theta}_{n,k}$, we can specify when the stationary distribution of the chain exists for finite sample sizes, and what the stationary distribution will be in the limit a $n \to \infty$. Operating within the confines of these assumptions makes deriving results regarding the asymptotic behaviour of estimators obtained from stochastic EM achievable. We present these results formally as Proposition 2.

Proposition 2 (Nielson, 2000)**.** *Suppose Assumption 1 holds. Then the following properties hold regarding the Markov chain $\tilde{\theta}_{n,k}$ and the stationary distribution $\tilde{\theta}_n$ of the stochastic EM algorithm.*

(i) Suppose $\tilde{\theta}_{n,k} = \hat{\theta}_n + (n^{-1/2})h + o(n^{-1/2})$ and $\tilde{Z}_i \sim k(z_i \mid x_i, \tilde{\theta}_{n,k})$. Then, for almost all observed sample sequences, the transition probabilities of the stochastic EM Markov chain

converge continously to those of a Gaussian autoregressive process of order 1,

$$n^{1/2}(\tilde{\theta}_{n,k+1} - \hat{\theta}_n) \to_d N\left(F(\theta_0)^{\mathrm{T}}h,\, \mathbf{I}_y(\theta_0)^{-1}\mathbf{I}_z(\theta_0)\mathbf{I}_y(\theta_0)^{-1}\right),$$

where $\tilde{\theta}_{n,k+1}$ *is the estimate generated by the algorithm based on the simulated* $\tilde{z}_{1:n}$*,* $\mathbf{I}_z = E(I_{z|x}(\theta_0) \mid X = x)$*, and* $F(\theta_0) = I_z(\theta_0)I_y(\theta_0)^{-1}$ *is the expected fraction of missing information.*

(ii) For almost all observed sample sequences, $n^{1/2}(\tilde{\theta}_n - \hat{\theta}_n)$ *is tight conditional on the sample and,*

$$n^{1/2}(\tilde{\theta}_n - \hat{\theta}_n) \to_d N\left(0,\, \mathbf{I}_x(\theta_0)^{-1}\{I - (I + F(\theta_0))^{-1}\}\right),$$

so that the stationary distribution is asymptotically normal and root-n consistent for $\hat{\theta}_n$*.*

(iii) For almost all samples,

$$n^{1/2}(\tilde{\theta}_n - \theta_0) \to_d N\left(0,\, \mathbf{I}_x(\theta_0)^{-1}\{2I - (I + F(\theta_0))^{-1}\}\right).$$

The proof of Proposition 2, as with all proofs, can be found in Appendix C. Proposition 2 serves as the starting point of our investigation of the estimator $\tilde{\theta}_{n,\min t}$. In addition, we will also rely on the fact that there exists a sequence of integers k_n such that the total variation distance between $\tilde{\theta}_{n,k_n}$ and $\tilde{\theta}_n$ is less than $1/n$. For any n, this allows us to approximately draw from the stationary distribution after finitely many algorithm iterations. In particular, a chain starting from the k_nth iteration still converges in distribution to a chain drawn from the Gaussian autoregressive process in part (i), and the normalized random variable $n^{1/2}(\tilde{\theta}_{n,k_n} - \hat{\theta}_n)$ still converges in distribution to the Gaussian distribution in part (ii) of Proposition 2. Hence, our investigation will focus specifically on the estimator $\tilde{\theta}_{n,\min t} = \mathrm{argmin}_{k=k_n,\ldots,k_n+t-1} D_x(\tilde{\theta}_{n,k})$, as this is the one we will be employing in practice.

6.3 Asymptotics Regarding the Minimum Likelihood Discrepancy Function Estimator

Interest lies in deriving properties of the estimator $\tilde{\theta}_{n,\min t}$, with the idea of understanding in what contexts (if any) its qualities make it preferable to the tail average $\bar{\theta}_{n,t}$. Our investigation proceeds by first considering the related random variable $m_{nt} = \min_{k=k_n,\ldots,k_n+t-1} D_x(\tilde{\theta}_{n,k})$, which is the minimum LDF value associated with the chain of length t starting at the k_nth chain component. The estimator $\tilde{\theta}_{n,\min t}$ is the chain value associated with m_{nt}, hence the level of concentration about 0 observed in the distribution of m_{nt} gives insight into the precision of $\tilde{\theta}_{n,\min t}$.

Due to the Markov behaviour of the algorithm chain, the sequence $D_x(\tilde{\theta}_{n,k_n:t})$ is correlated. It follows that the extreme value theorem (See, e.g. Leadbetter et al., 1983), does not directly apply to m_{nt}. However, there is a so-called condition D (see Leadbetter et al., 1983, ,pg. 53) such that, if a correlated sequence satisfies D condition, one can employ an result analogous to the extreme value theorem. Intuitively, when a correlated sequence of interest satisfies D, one can use an associated i.i.d. sequence to derive results regarding the distribution of extremes related to the correlated sequence. Our first result confirms that the sequence $D_x(\tilde{\theta}_{n,k_n:t})$ satisfies this condition.

Proposition 3. *The sequence of LDF values, $D_x(\tilde{\theta}_{n,k_n:t})$, satisfies condition D.*

The proof of Proposition 3, as well as details regarding condition D, can be found in Appendix C. Proposition 3 tells us that the random variable m_{nt} will converge in distribution to the same extreme value distribution as the minimum of t i.i.d. draws from $D_x(\tilde{\theta}_n)$ converges to, as t approaches infinity. We can therefore determine the distribution of m_{nt} by first determining the distribution of $D_x(\tilde{\theta}_{n,k_n})$.

Using Equation (6.1) and Proposition 2, the observed data LDF can be written as $D_x(\tilde{\theta}_{n,k_n}) = Z^{\mathrm{T}}\Delta Z + \xi_n + o_p(1)$, where ξ_n is a random variable such that $\xi_n \rightarrow_P 0$ as $n \rightarrow \infty$, Δ is the diagonal matrix comprised of the eigenvalues of $\mathbf{I}_x(\theta_0)^{-1/2}\{\mathbf{I} - (\mathbf{I} + F(\theta_0))^{-1}\}\mathbf{I}_x(\theta_0)^{1/2}$, and Z is a standard multivariate normal random variable. It is apparent from this representation that the observed data LDF converges to a linear combination of p independent Gamma distributions as the sample size n approaches infinity. Our next Lemma is a first step to utlizing this behaviour.

Lemma 4. *Let $X = \sum_{i=1}^{p} W_i$ be a linear combination of p independent random variables with distribution $W_i \sim Gamma(\alpha_i, \beta_i)$ where $\beta_1 \leq \ldots \leq \beta_p$. Define $m_t = \min_{j=1,..,t}\{X_j\}$*

as the minimum of t independent draws from X. Then, as $t \to \infty$ we have,

$$c_t^{-1} m_t \to_d \textit{Weibull}\left(\prod_{i=1}^{p}(\beta_1/\beta_i)^{\alpha_i/\alpha^\star}, \alpha^\star\right),$$

where $\alpha^\star = \sum_i \alpha_i$, and c_t are the normalizing constants such that $-c_t^{-1}\max\{-Y_1, \ldots, -Y_t\}$ converges in distribution as $t \to \infty$ with $Y_j \sim \textit{Gamma}(\alpha^\star, \beta_1)$ independently.

Remark. The proof of Lemma 4 relies on the closure of maximum domains of attraction under tail equivalence. For minima of i.i.d. draws from non-negative random variables X and Y, we consider X and Y to be tail-equivalent if there exists a constant c such that $\lim_{u \to t_0} S_X(t)/S_Y(t) = c$, where S_X and S_Y are the respective survival functions, e.g. $S_X(t) = 1 - F_X(t)$, and t_0 is their common right endpoint. With X and Y defined as in Lemma 4, we show tail equivalance of $-X$ and $-Y$ with $c = \prod_{i=1}^{p}(\beta_1/\beta_i)^{\alpha_i}$. Since $tS_{-Y}(c_t x) \to (-x)^{\alpha^\star}$ for appropriate normalizing constants c_t, by tail equivalence we also have $tS_{-X}(c_t x) \to c(-x)^{\alpha^\star}$. As a corollary, the normalizing constants c_t used to normalize the minimum of draws from a single Gamma distribution can be used to normalize the minimum of a linear combination of independent Gamma distributions as well.

It remains to give an explicit form for the sequence of constants c_t. By Embrechts et al. (1997), $-c_t \max\{-Y_1, \ldots, -Y_t\}$ converges in distribution for $c_t = F_{-Y}^{\leftarrow}(1 - t^{-1})$, where $F^{\leftarrow}(t)$ denotes the generalized inverse of a non-decreasing function F, which is defined as $F^{\leftarrow}(t) = \inf\{x \mid F(x) \geq t\}$. Since $F_{-Y} = S_Y(-y)$ for $y \in \mathbb{R}^-$, we are interested in the generalized inverse of $S_Y(y)$ as y approaches 0 from the positive side. The Gamma cdf is continuous and strictly increasing, so $S_Y(-y)$ admits an inverse, although it cannot generally be written in closed form. We therefore use the property that the lower incomplete gamma function $\gamma(\alpha, -\phi^{-1}y)$ is approximately equal to $\alpha^{\star\text{-}1}(-\beta_1^{-1}y)^{\alpha^\star}$ as $-y \downarrow 0$ to find that the normalizing constants can be written $c_t = \left[\alpha^\star \beta_1^{\alpha^\star}\Gamma(\alpha^\star)\right]^{1/\alpha^\star} t^{\text{-}1/\alpha^\star}$ for t sufficiently large.

We now use the results of Lemma 4 to derive the asymptotic distribution of m_{nt} for a suitable normalization.

Theorem 5. *Conditional on the observed data sample and for almost all samples,*

$$t^{2/p} m_{nt} \to_d \textit{Weibull}\left(2\left\{\frac{p}{2}\Gamma\left(\frac{p}{2}\right)\right\}^{2/p}\prod_{i=1}^{p}\lambda_i^{1/p}, \frac{p}{2}\right),$$

where $\lambda_1, \ldots, \lambda_p$ *are the eigenvalues of* $\mathbf{I}_x(\theta_0)^{-1/2}(I - [I + F(\theta_0)]^{-1})\mathbf{I}_x(\theta_0)^{1/2}$.

Remark. Theorem 5 is not a corollary of Lemma 4 as it at first appears. This is because it requires additionally taking $n \to \infty$ to get Gamma samples, alongside sending $t \to \infty$ to apply the extreme value theory results.

Remark. The eigenvalues $\lambda_{1:p}$ of the covariance matrix $\mathbf{I}_x(\theta_0)^{-1/2}(I - [I + F(\theta_0)]^{-1})\mathbf{I}_x(\theta_0)^{1/2}$ are the same as those of $I - (I + F(\theta_0))^{-1}$, from which it follows that $0 < \lambda_i < 1/2$ for each $i = 1, \ldots, p$. In the subsequent discussion, we assume that the minimum eigenvalue $\lambda_{(1)}$ is always positive. This restricts attention to model specifications under which the missing data is not identified in a redundant manner. We then have $\liminf_{p\to\infty} \prod_{i=1}^{p} \lambda_{(i)}^{1/p} \geq \lim_{p\to\infty} \lambda_{(1)} > 0$.

We now investigate the behaviour of the asymptotic distribution of m_{nt} on intervals of the form $(0, \delta)$, $\delta > 0$, as the parameter dimension, p, increases. Let $\epsilon, \delta > 0$ and consider the interval $(0, \delta)$. For a fixed dimension p, the asymptotic probability that the (normalized) minimum LDF value occurs in this interval is,

$$1 - \exp\left(-\left[\frac{\delta}{2\left\{\Gamma\left(\frac{p}{2}\right)\frac{p}{2}\right\}^{2/p}\left\{\prod_{i=1}^{p}\lambda_i\right\}^{1/p}}\right]^{p/2}\right) \leq 1 - \exp\left(-\left[\frac{\delta}{2\lambda_{(1)}\left\{\Gamma\left(\frac{p}{2}\right)\frac{p}{2}\right\}^{2/p}}\right]^{p/2}\right).$$

Observing that $\lim_{p\to\infty}\left\{\Gamma\left(\frac{p}{2}\right)\frac{p}{2}\right\}^{2/p} = \infty$, we find that the scale parameter grows infinitely in the parameter dimension p. It follows that there exists a p such that,

$$\frac{\delta}{2\lambda_{(1)}\left[\Gamma\left(\frac{p}{2}\right)\frac{p}{2}\right]^{2/p}} < \epsilon.$$

Since ϵ is arbitrary, we conclude that the probability that the minimum is in the interval $(0, \delta)$ can be made arbitrarily small by increasing p. Since δ is arbitrary, this result holds for any positive real δ, so that the minimum LDF value escapes away from 0 with high probability as the parameter dimension p increases. Additionally, $m_{nt} = O_P(t^{2/p})$ so that the rate of convergence is also slowed considerably with increasing parameter dimension.

In terms of the estimator $\tilde{\theta}_{n,\min t}$, these results imply that its distribution puts diminishing weight on open balls centered at $\hat{\theta}_n$ as p increases. Thus, as p increases, the distribution becomes anti-modal at $\hat{\theta}_n$. The distribution can therefore be visualized as a ripple, with the crest of the ripple moving away from $\hat{\theta}_n$ as p increases. Interestingly, $p = 1$ would then correspond to the case in which the ripple crest resides at $\hat{\theta}_n$. It follows that $\tilde{\theta}_{n,\min t}$ could be a precise estimator in this scenario. Investigating this idea begins with the following corrollary to Theorem 5.

Corollary 6. *When the model parameter θ is a scalar, we have*

$$t^2 m_{nt} \to_d \textit{Weibull}\left(\frac{\pi}{2}\left\{1 - \frac{1}{1+F(\theta_0)}\right\}, \frac{1}{2}\right),$$

as $n, t \to \infty$ for almost all observed data samples and conditional on the sample.

The shape parameter of the Weibull distribution in Corollary 6 is less than 1, which means the density exhibits an asymptote at 0. Further, the fraction of missing information, $F(\theta_0)$, is on the unit interval, hence the scale parameter is in the interval $(0, \pi/4)$. The associated density is therefore concentrated near 0. It follows that the minimum approches 0 quickly in probability. More importantly, the estimator $\tilde{\theta}_{n,\min t}$ approaches the maximum likelihood estimate $\hat{\theta}_n$ in probability quickly as the chain length increases. Our goal now is to quantify what is meant by quickly. To do this, we obtain the asymptotic distribution of $\tilde{\theta}_{n,\min t}$ in n and t.

Theorem 7. *Suppose θ is a scalar, so that Corollary 6 holds. Then, as both $n, t \to \infty$ the normalized random variable $tn^{1/2}(\tilde{\theta}_{n,\min t} - \hat{\theta}_n)$ satifies,*

$$tn^{1/2}(\tilde{\theta}_{n,\min t} - \hat{\theta}_n) \to_d \textit{Laplace}\left(0, \mathbf{I}_x(\theta_0)^{-1/2}\frac{\{2\pi F(\theta_0)\}^{1/2}}{2\{1+F(\theta_0)\}^{1/2}}\right),$$

so that it is asymptotically Laplace distributed with location parameter 0 and scale parameter $(\pi/2)^{1/2}[F(\theta_0)/\{1+F(\theta_0)\}]^{1/2}\mathbf{I}_x(\theta_0)^{-1/2}$.

Remark. The proof of Theorem 7 relies on Corollary 6, and hence only works when $p = 1$. In this context the random variable of interest $tn^{1/2}(\tilde{\theta}_{n,\min t} - \hat{\theta}_n)$ can be written as $\operatorname{sgn}\left\{\tilde{\theta}_{n,\min t} - \hat{\theta}_n\right\}\{t^2 m_{nt}\}^{1/2}$. The asymptotic symmetry of the stationary distribution about $\hat{\theta}_n$ means $\operatorname{sgn}\left\{\tilde{\theta}_{n,\min t} - \hat{\theta}_n\right\}$ is asymptotically binomial and independent of m_{nt}. Deriving the distribution of the product is then straightforward.

It is interesting that the limiting distribution of the estimator is Laplace, however the key property we are interested in is the rate of convergence with respect to chain length, which we now justify. In Nielson's investigation of the tail average (Nielsen, 2000), he found that for a fixed tail length t, the random variable $tn^{1/2}(\bar{\theta}_{n,t} - \hat{\theta}_n)$ converges in distribution to a Gaussian distribution with variance given by,

$$\sigma_t^2 = n^{-1}t^{-1}\mathbf{I}_x(\theta_0)^{-1}\frac{F(\theta_0)}{1+F(\theta_0)}\left(1 + 2\frac{F(\theta_0)}{1-F(\theta_0)} - 2t^{-1}\frac{F(\theta_0)(1-F(\theta_0)^t)}{(1-F(\theta_0))^2}\right).$$

Unfortunately, the conditions underpinning his investigation (as well as ours) are too general to guarantee convergence of the tail average to the expected value of the stationary distribution as the length of the chain becomes infinite. Nielson was therefore unable to give proper asymptotic results regarding convergence of $\bar{\theta}_{n,t}$ in terms of t. Nevertheless, for comparison sake we will say that, for large n, we have $n^{1/2}(\bar{\theta}_{n,t} - \hat{\theta}_n) = O_p(t^{-1/2})$, approximately. On the other hand, Theorem 7 gives us that $n^{1/2}(\tilde{\theta}_{n,\min t} - \hat{\theta}_n) = O_p(t^{-1})$ for n sufficient large. Hence, $\tilde{\theta}_{n,\min t}$ converges at square the rate of $\bar{\theta}_{n,t}$ in t asymptotically in n, making it the more precise estimator in terms of chain length. Intuitively, when computed from the same algorithm chain, we expect $\tilde{\theta}_{n,\min t}$ to be closer to $\hat{\theta}_n$ than $\bar{\theta}_{n,t}$ on average. We demonstrate the difference this makes in practice with an example using right-censored data.

6.3.1 Demonstrative Simulation: Right-Censored Data

We demonstrate and compare the two estimators using a right-censored data experiment. We consider a sample from an exponential random variable X, where right censoring of the observations at time $t = t_0$ occurs. The observed data then take the form $x = (x_1, ..., x_{n-r}, t_0, ..., t_0)$, where r is the total number of censored data points. The associated complete-data loglikelihood is, $\ell(\theta) = -n\log\theta - \theta^{-1}\sum_{j=1}^{n-r} x_j - \theta^{-1}\sum_{j=1}^{r} X_{n-r+j}$. The observed data information and the expected fraction of information are given respectively by,

$$\mathbf{I}_x(\theta_0) = \big(1 - e^{\text{-}t_0/\theta_0}\big)\theta_0^{\text{-}2}, \quad F(\theta_0) = e^{\text{-}t_0/\theta_0}.$$

It follows that the variances of the respective estimators are approximately,

$$\mathrm{var}(\tilde{\theta}_{n,\min t}) \approx \frac{\theta_0^2 e^{-t_0/\theta_0}}{nt^2\big(1 - e^{-t_0/\theta_0}\big)\big(1 + e^{-t_0/\theta_0}\big)}, \quad \mathrm{var}(\bar{\theta}_{n,t}) \approx \frac{\theta_0^2 e^{-t_0/\theta_0}}{nt\big(1 - e^{-t_0/\theta_0}\big)\big(1 + e^{-t_0/\theta_0}\big)}.$$

These variances depend on the true parameter value θ_0, the length of observation time, t_0, the chain length t, and the sample size n; their form is nearly identical, with the only difference being the additional multiplicative factor of t in the denominator of $\mathrm{var}(\tilde{\theta}_{n,\min t})$. To demonstrate the impact of this additional factor, we generate a right censored dataset from an exponential distribution as described, using a true value $\theta_0 = 2$ and a cut off value of $t_0 = 2.4$. The dataset consists of 1000 observations, 305 of which are censored. The true value of the maximum likelihood estimate for this dataset is $\hat{\theta}_n = 1.9833$. We initialize a stochastic EM algorithm at $\tilde{\theta}_{n,0} = 1.9$ and generate chains of length $t = 1000$. We do this 1000 times. We then calculate parameter estimates from these chains using both the minimum log-likelihood ratio estimate, $\tilde{\theta}_{n,\min t}$, and the tail average, $\bar{\theta}_{n,t}$. Note that the minimum log-likelihood ratio estimate does not require knowledge of $\hat{\theta}_n$, because in practice it is instead obtained by the maximum likelihood value.

The empirical distributions are provided in Figure 6.1. The first two plots are given on the same scale, while the third plot has an adjusted scale for clear presentation of the densities structure. The leftmost plot is the empirical distribution of the tail average estimator $\bar{\theta}_{n,t}$, while the central plot corresponds to empirical distribution of the minimum log-likelihood ratio estimator $\tilde{\theta}_{n,\min t}$. The stark contrast in the concentration of the distributions about the maximum likelihood estimate accentuates the difference in the performance of these estimators in the scalar-valued parameter case, and highlights the significance of the increased convergence rate of $\tilde{\theta}_{n,\min t}$.

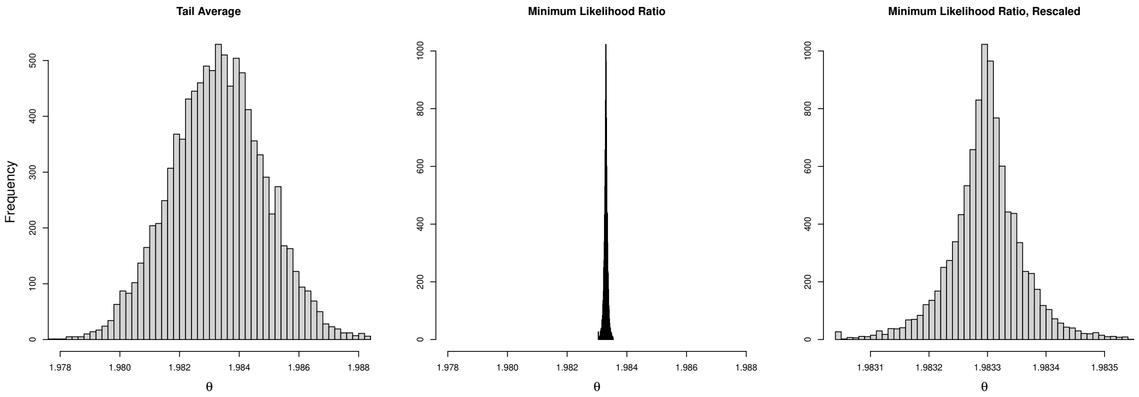


Figure 6.1: The simulated distributions from the right censored simulation. On the left, we have the distribution of the tail average estimator, and in the middle we have the distribution of the minimum log-likelihood ratio estimator on the same scale. The rightmost plot is a repeat of the central one with an adjusted scale to illustrate the Laplace resemblance.

6.4 Maximum Contribution via the Profile Likelihood

The results of Section 6.3 suggest that the estimator $\tilde{\theta}_{n,\min t}$ should be avoided in high dimensional problems, because the minimum LDF value on which it relies becomes bounded away from 0 with high probability. Intuitively, this occurs because the squared, component-wise differences between $\tilde{\theta}_{n,t}$ and $\hat{\theta}_n$, which comprise the dominant term of the LDF, accumulate with p. The more dimensions we have, the less likely a random value drawn from the chain will be close to $\hat{\theta}_n$ in all dimensions at the same time. We hope to address this issue by instead assessing the closeness of each component separately.

Suppose $\tilde{\theta}_{n,1:t}$ is the chain of values generated by a stochastic EM algorithm, with p large. Although each value in $\tilde{\theta}_{n,1:t}$ is likely to produce a large squared deviation from $\hat{\theta}_n$ in at least one component, it is also likely the case that, for each component of $\hat{\theta}_n$, denoted $\hat{\theta}_{ni}$, there exists a value in $\tilde{\theta}_{n,1:t}$ which has a small squared deviation from $\hat{\theta}_n$ in that particular component. Hence, if we could cherry-pick from each marginal chain, $\tilde{\theta}_{ni,1:t}$, the value which is closest to $\hat{\boldsymbol{\theta}}_{ni}$, we could construct an estimate that is closer to $\hat{\theta}_n$ than any value in $\tilde{\theta}_{n,1:t}$, and in particular, $\tilde{\theta}_{n,\min t}$. By relying on different members of $\tilde{\theta}_{n,1:t}$ to estimate different components of $\hat{\theta}_n$, we skirt the issue of requiring one particular memeber of $\tilde{\theta}_{n,1:t}$ to provide a sufficiently small LDF value. For such an approach to work, an appropriate definition of closeness is required. Up to this point we have used the observed data likelihood for this purpose, however it is no longer sufficient. Indeed, unless the log-likelihood decomposes into a linear combination of terms, each of which is a function of only one component of the model parameter, there is no way to determine the contribution of each parameter component to the total likelihood value. Therefore, the likelihood cannot be used to assess closeness of parameter components to the corresponding components of $\hat{\theta}_n$.

We instead consider the profile likelihood, whose specific purpose is to evaluate a subset of the model parameter. For a specific value θ of the model parameter, let θ_i be the ith component and θ_{-i} be all components except the ith, the latter of which has associated parameter space Θ_{-i}. The profile log-likelihood at θ_i is then defined as $\ell_{\mathrm{p}}(\theta_i) = \sup_{\theta \in \Theta_{-i}} \ell_x(\theta_i, \theta)$; in words, it is the maximum value the likelihood can achieve given that the ith component of the model parameter is fixed to be θ_i. We further define the profile likelihood discrepancy function as $D_{\mathrm{p}}(\theta_i) = -2(\ell_x(\hat{\theta}_n) - \ell_{\mathrm{p}}(\theta_i))$. By construction, the profile LDF is minimized by each component $\hat{\theta}_{ni}$ of $\hat{\theta}_n$. Regularity of the observed data model then allows us to conclude that the profile LDF exhibits the properties we seek. Subsequently, we propose the estimator $\tilde{\theta}_{n,\mathrm{p}\,t} = (\tilde{\theta}_{\mathrm{p}\,n1,t}, \ldots, \tilde{\theta}_{np,\mathrm{p}\,t})$ such that,

$$\tilde{\theta}_{ni,\mathrm{p}\,t} = \underset{k=k_n,\ldots,k_n+t-1}{\operatorname{argmin}} D_{\mathrm{p}}(\tilde{\theta}_{ni,k}).$$

The estimator $\tilde{\theta}_{n,\mathrm{p}\,t}$ estimates each component $\hat{\theta}_{ni}$ with the value in the marginal chain $\tilde{\theta}_{ni,k_n:t}$ that maximizes the profile likelihood. We now show that this estimator is equiped with improved convergence properties over its predecessors $\tilde{\theta}_{n,\min t}$ and $\bar{\theta}_{n,t}$. We begin with a proposition, which follows immediately from Proposition 2.

Proposition 8. *Let $\tilde{\theta}_{ni,k_n}$ be the ith component of $\tilde{\theta}_{n,k_n}$, $i = 1, \ldots, p$. Then, the marginal chain $\sqrt{n}(\tilde{\theta}_{ni,k_n:t} - \hat{\theta}_{ni})$ converges in distribution to a sample of the same length from the stationary marginal vector autoregressive process with autoregressive parameter $F(\theta_0)_i^{\mathrm{T}}$ and innovation variance $\{\mathbf{I}_y(\theta_0)^{-1}\mathbf{I}_z(\theta_0)\mathbf{I}_y(\theta_0)^{-1}\}_{ii}$. It follows that,*

(i) For all observed sample sequences and conditional on the sample,

$$\sqrt{n}(\tilde{\theta}_{ni,k_n} - \hat{\theta}_{ni}) \to_d N(0, \mathbf{I}_x^{-1}(\theta_0)_{ii} - \mathbf{I}_x^{-1}(\theta_0)_{i\cdot}^{\mathrm{T}}(I + F(\theta_0))_{\cdot i}^{-1}).$$

(ii) Unconditionally,

$$\sqrt{n}(\tilde{\theta}_{ni,k_n} - \theta_{0i}) \rightarrow_d N(0, 2\mathbf{I}_x^{-1}(\theta_0)_{ii} - \mathbf{I}_x^{-1}(\theta_0)_{i\cdot}^{\mathrm{T}}(I + F(\theta_0))_{\cdot i}^{-1}).$$

The idea is to utilize Proposition 8 as a starting point for obtaining results regarding $\tilde{\theta}_{n,\mathrm{p}\,t}$, in a similar vein to our utilization of Proposition 2 in Section 6.3. In particular, since each component of $\tilde{\theta}_{n,\mathrm{p}\,t}$ is identical in form to $\tilde{\theta}_{n,\min t}$ when $p = 1$, if we can further show that $D_{\mathrm{p}}(\theta)$ has a similar form to $D_x(\theta)$, we should be able to achieve results equivalent to Corollary 6 and Theorem 7 in this profile context.

Proposition 9. *Under the assumed regularity conditions on the observed data model, for a parameter component value of the form $\theta_{ni} = \hat{\theta}_{ni} + n^{-1/2}h$ the profile LDF exhibits the approximation,*

$$D_{\mathrm{p}}(\theta_{ni}) = n\mathbf{I}_x^{-1}(\theta_0)_{ii}(\theta_{ni} - \hat{\theta}_{ni})^2 + o_p(1),$$

for almost all observed data samples, and conditional on the sample.

Proposition 9 shows that the profile LDF is asymptotically quadratic in the model parameter, in a fashion reminiscent of the observed data LDF. Based on this result, and the results of Section 6.3, the forthcoming corollary concerning the minimum profile LDF value over the marginal chains, denoted by $m_{nit} = \min_{k=k_n,\ldots,k_n+t-1} D_{\mathrm{p}}(\tilde{\theta}_{ni,k})$, follows almost immediately.

Corollary 10. *Assume Proposition 8 and Proposition 9 hold. Then, for all $i = 1, \ldots, p$,*

(i) The profile LDF value associated with the randon variable $\tilde{\theta}_{ni,k_n}$ converges in distribution,

$$D_{\mathrm{p}}(\tilde{\theta}_{ni,k_n}) \rightarrow_d \mathit{Gamma}\left(\frac{1}{2}, 2[1 - \{\mathbf{I}_x^{-1}(\theta_0)_{ii}\}^{-1}\mathbf{I}_x^{-1}(\theta_0)_{i\cdot}^{\mathrm{T}}\{I + F(\theta_0)\}_{\cdot i}^{-1}]\right),$$

as $n \to \infty$;

(ii) the minimum profile LDF associated with the chain $\tilde{\theta}_{ni,k_n:t}$ converges in distribution,

$$t^2 m_{nit} \rightarrow_d \mathit{Weibull}\left(\frac{\pi}{2}[1 - \{\mathbf{I}_x^{-1}(\theta_0)_{ii}\}^{-1}\mathbf{I}_x^{-1}(\theta_0)_{i\cdot}^{\mathrm{T}}\{I + F(\theta_0)\}_{\cdot i}^{-1}], \frac{1}{2}\right),$$

as $n, t \to \infty$.

Remark. In part (ii), the variable term of the scale parameter can be written as $1 - \{\mathbf{I} - F(\theta_0)\}^{-1}_{ii} + \sum_{k \neq i} \mathbf{I}_x(\theta_0)^{-1}_{ik}\{\mathbf{I} + F(\theta_0)\}^{-1}_{ki}$. This has a lower bound of 0, which is achieved as $F(\theta_0) \to 0$. Likewise, as $F(\theta_0) \to \mathbf{I}$ we have $\{\mathbf{I} + F(\theta_0)\}^{-1}_{ki} \to 0$ and $\{\mathbf{I} + F(\theta_0)\}^{-1}_{ii} \to 1/2$, after which we find the upper bound of the entire term to be $1/2$. It follows that the scale parameter lies in the interval $(0, \pi/4)$. Couple this with the shape parameter value of $1/2$, and we find that this asymptotic distribution is concentrated near 0 for any nontrivial $F(\theta_0)$. This asymptotic distribution therefore has the same parameter space as the asymptotic distribution in Corollary 6.

Remark. Since $P(t^{-2}m_{nit} \leq t^{-\epsilon}x) \to 0$ for all $\epsilon > 0$ and all $x \in \mathbb{R}^+$ as $t \to \infty$, we have $m_{nit} = o_p(t^{-2+\epsilon})$ for all $\epsilon > 0$, and therefore $m_{nit} = O_p(t^{-2})$.

The speed with which m_{nit} approaches 0 in probability, for each i, suggests that $\tilde{\theta}_{n,\mathrm{p}\,t}$ should be a more efficient for $\hat{\theta}_n$ than $\tilde{\theta}_{n,\min t}$ and $\bar{\theta}_{n,t}$ in terms of chain length. The next theorem aims to substantiate this claim.

Theorem 11. *For the estimator $\tilde{\theta}_{ni,\mathrm{p}\,t}$, we have,*

$$tn^{1/2}\mathbf{I}_x^{-1}(\theta_0)^{-1/2}_{ii}(\tilde{\theta}_{ni,\mathrm{p}\,t} - \hat{\theta}_n) \to_d \mathit{Laplace}\left(0, \frac{(2\pi)^{1/2}}{2}[1 - \{\mathbf{I}_x^{-1}(\theta_0)_{ii}\}^{-1}\mathbf{I}_x^{-1}(\theta_0)^{\mathrm{T}}_{i\cdot}\{I + F(\theta_0)\}^{-1}_{\cdot i}]^{1/2}\right),$$

as $n, t \to \infty$.

Theorem 11 verifies that the proposed estimator is $O_p(t^{-1})$, which is the rate of stochastic boundedness that we set out to achieve. It follows that the performance of $\tilde{\theta}_{n,\mathrm{p}\,t}$ for $p \geq 2$ relative to $\bar{\theta}_{n,t}$ should be similar to that of $\tilde{\theta}_{n,\min t}$ demonstrated in Section 6.3.1. In the next section, we demonstrate the implications this result has in applications.

One concern regarding the estimator $\tilde{\theta}_{n,\mathrm{p}\,t}$ is its reliance on the profile likelihood. Computation of the profile likelihood is not always feasible, and in such cases, the proposed estimator cannot be used. A heuristic alternative, which can in theory always be obtained, is to evaluate the LDF at all possible combinations of the components of the marginal chains, and choose the combination which gives the smallest value. That is, for each $i = 1, \ldots, p$ choose an element j_i from $\{1, \ldots, t\}$ and evaluate the observed data LDF at $\hat{\theta} = (\tilde{\theta}_{n1,j_1}, \ldots, \tilde{\theta}_{np,j_p})$. Assuming a chain of length t, it would require t^p evaluations of the LDF to check all possible estimates of this form. Even with state of the art computational hardware, this rapidly becomes infeasible for large t and p, especially in cases where evaulation of the likelihood is difficult.

Additionally, one may choose instead to use an integrated likelihood (see e.g. Berger et al., 1999; Severini, 2007), which is typically easier to obtain in practical applications than is the profile likelihood. Generally, integrated likelihood functions do not exhibit the properties which make likelihoods useful for frequentist analyses. To instead formulate the proposed estimator in terms of the integrated likelihood, it is likely that additional assumptions, and careful selection of the weighting function $\pi(\theta_{-i} \mid \theta_i)$, would be needed in order to obtain results similar to those derived here. See Severini (2007) for more details.

6.5 Numerical Experiments

We demonstrate behaviours of interest for stochastic EM estimators by way of numerical experimentation. We choose to do this in the context of robust regression, wherein we assume a random variable of the form,

$$T = \mu + \sigma R; \quad R = ZU^{-1/2},$$

where, $Z \sim N(0,1)$, $U \sim \Gamma(\nu/2, \nu/2)$, $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$. A regression model follows from the assumption that μ has the linear representation $\mu = x^{\mathrm{T}}\beta$, where x is a known vector of covariates and β is the corresponding parameter of regression coefficients. For a fixed x, the random variable T exhibits a t-distribution, with $T \sim t(\mu = x^{\mathrm{T}}\beta, \sigma, \nu)$. To keep the example simple, we assume that the parameters σ and ν are known.

Given a sample $(x_i, t_i)_{i=1}^n$, the robust regression model can be estimated using the stochastic EM algorithm by designating U to be the missing data. Under the assumption that $U = u$ is observed, we find that the dependent variable is normally distributed, $T|(U = u) \sim N(0, \sigma^2/u)$. When the response is observed, the missing data has conditional distribution $U|(T = t) \sim \text{Gamma}(\frac{\nu+1}{2}, 0.5(\nu + r^2))$ with $r = \sigma^{-1}(t - x^{\mathrm{T}}\beta)$. The stochastic EM algorithm proceeds as follows: for a given estimate $\hat{\beta}$ the S-step draws the missing values $\tilde{u}_{1:n}$ from the corresponding conditional distributions using $\hat{r}_{1:n} = \sigma^{-1}(t_{1:n} - x_{1:n}^{\mathrm{T}}\hat{\beta})$.

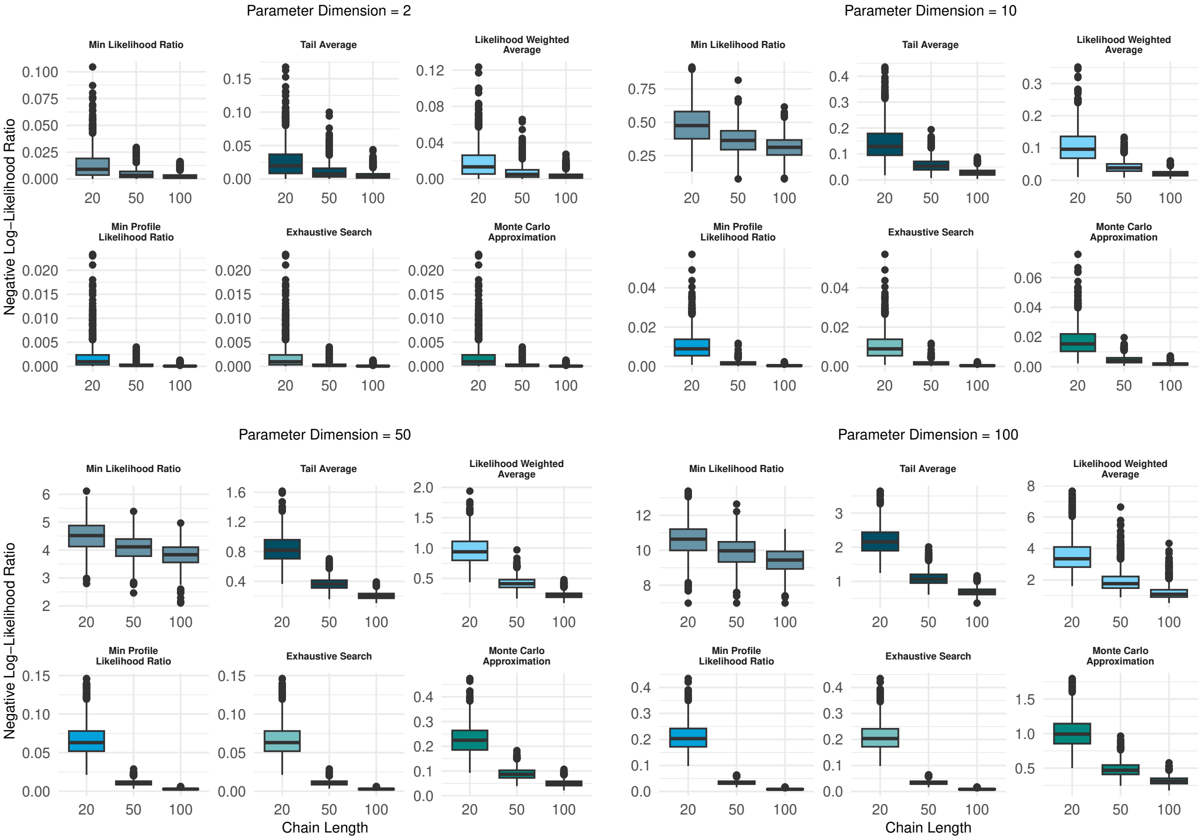


Figure 6.2: Boxplots of the negative log likelihood-ratio values associated with each estimator, for each simulation scenario. Each subplot corresponds to a different value of the parameter dimension, while the plots within each subplot illustrate the distribution of the negative log likleihood-ratio values across different chain lengths t.

The M-step updates the regression parameters as $\hat{\beta} = (X^{\mathrm{T}}\tilde{W}X)^{-1}X^{\mathrm{T}}\tilde{W}t_{1:n}$, where X is the $n \times p$ matrix with the observed covariate x_i as its ith row, and $\tilde{W}$ is the diagonal matrix with $\tilde{u}_i$ as its ith diagonal element. After t iterations, we have a chain $\hat{\beta}_{1:t}$, which is then used to compute point estimates for the model parameter β.

Estimators of interest for this numerical experiment are $\tilde{\theta}_{n,\min t}$, $\bar{\theta}_{n,t}$ and $\tilde{\theta}_{n,\mathrm{p}\,t}$. We include 3 additional estimators: one which serve as an alternative approach to averaging, and two which serve as approximations to $\tilde{\theta}_{n,\mathrm{p}\,t}$ which do not include use of the profile liklelihood. For alternative to the average, we investigate a weighted average where the normalized likelihood values of the chain elements are used as the weights. For approximations to $\tilde{\theta}_{n,\mathrm{p}\,t}$, we use the two approaches mentioned at the end of Section 6.4: the exhaustive search through the chain elements, and an integrated likelihood approach. The second uses a Monte Carlo approximation to an integrated likelihood, e.g. for a chain estimate component $\tilde{\beta}_{ik}$ we compute $\bar{\ell}_x(\tilde{\beta}_{ik}) = t^{-1} \sum_{j=1}^{t} \ell_x(\tilde{\beta}_{ik}, \tilde{\beta}_{-i,j})$. The estimate of each component of β is then the one corresponding to the largest value of this metric.

We choose two factors to vary in the simulation: the parameter dimension p, and the chain length t. Specifically, we look at all combinations of $p = 1, 10, 50, 100$ and $t = 20, 50, 100$. For each pair of simulation factors, (p, t), we generate a single dataset, which is then used as the starting point for 1000 independent runs of the stochastic EM algorithm. Each resulting chain is used to compute the value of the estimators included in the study. Across all simulations the sample size is set to $n = 500$, while the nusiance parameters are fixed at $\sigma = 2$ and $\nu = 2.5$.

Data generation begins with the $n \times p$ data matrix X, which has each row drawn according to a standard Gaussian. The regression parameters, β, are drawn componentwise and independently from a uniform distribution on $[-2, 2]$. Finally, we use the covariates and model parameters to generate the response values, $t_{1:n}$.

These data are used to generate 1000 estimate chains through independent runs of stochastic EM . Each chain is started at the maximum likelihood estimate so that it can be considered to have been sampled from the stationary distribution of the algorithm. To guard against the potential for bias, we give each chain an initial burn-in period of 20 iterations.

The results of the simulation study are given in Figure 6.2, which contains four subplots, each corresponding to a particular value for the parameter dimension p. Within each subplot are six sets of boxplots, one for each estimator included in the simulation. Each boxplot corresponds to one particular value for the chain length, and plots the associated distribution of negative log likelihood-ratio values produced by the estimator over the 1000 chains generated for that scenario.

A general trend can be seen to occur across all estimators: for a fixed value of the parameter dimension p, increasing t shifts the central tendency of the LDF value distribution toward 0. Of particular interest is the rate at which this shift occurs. The minimum LDF estimator, for example, exhibits slow convergence, especially for larger values of the parameter dimension. This is evidenced by the relatively unchanged boxplots across increasing values of t. In contrast, the proposed estimator demonstrates rapid convergence of the central tendency towards 0 as chain length increases, regardless of p. For large values of p, the proposed estimator achieves, in just 20 iterations, an average LDF value considerably smaller than what the other estimators achieve in 50 or 100 iterations. Furthermore, a comparison of the boxplots of the proposed estimator with those of the exhaustive search estimator reveals striking similarities, suggesting the two approaches are nearly equivalent in practice. Since the true value of the maximum likelihood estimate was directly used in the computation of the exhaustive search estimator, this seems a satisfactory performance.

The heuristic estimators have also turned in good results, with both outperforming the minimum LDF estimator as parameter dimension increases. However, the weighted likelihood average estimator has not shown much, if any, improvement over the tail average estimator, particularly for larger parameter dimensions. On the other hand, the Monte Carlo integrated likelihood estimator outperformed the tail average estimator in terms of accuracy and precision, supporting the earlier notion that this estimator may be a worthwhile alternative to the proposed estimator.

Chapter 7

Conclusion

This thesis has offered a small contribution to the field of functional finite mixture models with a suite of model-based clustering methodologies which accomodate both real-valued and vector-valued functions of a single variable. Each methodology rests on the notion that the joint density of the Karhunen-Loeve expansion coefficients serves as a surrogate for the density of the function-valued random variable to which they correspond. The tabiya of the proposed methodologies is reached by assuming this density exhibits the finite mixture architecture. The first methodology then specifed a joint generalized hyperbolic distribution on the principal components of a real-valued random function. The second methodology extends the parsimonious dual-subspace parameter specification for the matrix normal distribution of Chapter 4, to a more general latent factor analyzer specification for the development of a finite mixture model for clustering high-dimensional space curves. Addtionally, an estimatior to be used in tandem with SEM was introduced, and its properties were investigated. By taking the idea of maximum likelihood literally and harnessing the extreme value theory of weakly correlated sequences, the estimator was able to achieve greater precision than topical approaches. In summary, the methodologies presented herein not only contribute to the existing body of literature but also pave the way for future research in this exciting and challenging field.

References

Aitkin, M. and Rubin, D. B. (1985). Estimation and hypothesis testing in finite mixture models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):67–75.

Baek, J., McLachlan, G. J., and Flack, L. K. (2010). Mixtures of factor analyzers with common factor loadings: Applications to the clustering and visualization of high-dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1298–1309.

Baltazar-Larios, F. and Esparza, L. J. R. (2022). Statistical Inference for Partially Observed Markov-Modulated Diffusion Risk Model. *Methodology and Computing in Applied Probability*, 24(2):571–593.

Banfield, J. and Raftery, A. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, 49:803–821.

Basford, K. E. and McLachlan, G. J. (1985). The mixture method of clustering applied to three-way data. *Journal of Classification*.

Bellman, R. (1954). The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60(6):503–515.

Berger, J. O., Liseo, B., and Wolpert, R. L. (1999). Integrated likelihood methods for eliminating nuisance parameters. *Statistical Science*, 14(1):1 – 28.

Bickel, P. J. and Levina, E. (2008). Regularized estimation of large covariance matrices. *Ann. Statist.*, 36(1):199–227.

Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725.

Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3):561–575. Recent Developments in Mixture Model.

Bongiorno, E. and Goia, A. (2017). Some insights about the small ball probability factorization for hilbert random elements. *Statistica Sinica*, 27:1949–1965.

Bongiorno, E. G. and Goia, A. (2016). Classification methods for hilbert data based on surrogate density. *Computational Statistics & Data Analysis*, 99:204–222.

Bouveyron, C. and Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71:52 – 78.

Bouveyron, C., Celeux, G., Murphy, T., and Raftery, A. (2019). *Model-Based Clustering and Classification for Data Science: With Applications in R*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Bouveyron, C., Côme, E., and Jacques, J. (2015). The discriminative functional mixture model for a comparative analysis of bike sharing systems. *The Annals of Applied Statistics*, 9(4):1726 – 1760.

Bouveyron, C., Girard, S., and Schmid, C. (2007). High-dimensional data clustering. *Computational Statistics & Data Analysis*, 52(1):502–519.

Bouveyron, C. and Jacques, J. (2011). Model-based Clustering of Time Series in Group-specific Functional Subspaces. *Advances in Data Analysis and Classification*, 5(4):281–300.

Browne, R. P. and Mcnicholas, P. D. (2014). Estimating common principal components in high dimensions. *Adv. Data Anal. Classif.*, 8(2):217–226.

Browne, R. P. and McNicholas, P. D. (2015). A mixture of generalized hyperbolic distributions. *Canadian Journal of Statistics*, 43(2):176–198.

Cardot, H., Ferraty, F., and Sarda, P. (1999). Functional linear model. *Statistics & Probability Letters*, 45:11–22.

Cardot, H., Ferraty, F., and Sarda, P. (2003). Spline estimators for the functional linear model. *Statistica Sinica*, 13(3):571–591.

Casa, A., Bouveyron, C., Erosheva, E., and Menardi, G. (2021). Co-clustering of time-dependent data via the shape invariant model. *Journal of Classification*, 38:626–649.

Celeux, G., Chauveau, D., and Diebolt, J. (1995). On Stochastic Versions of the EM Algorithm. Research Report RR-2514, INRIA.

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82.

Celeux, G. and Diebolt, J. (1987). *The EM and SEM algorithms for mixtures: Statistical and numerical aspects*. PhD thesis, INRIA.

Celeux, G. and Diebolt, J. (1988). *A random imputation principle: the stochastic EM algorithm*. PhD thesis, INRIA.

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

Chen, D., Hall, P., and Müller, H.-G. (2011). Single and multiple index functional regression models with nonparametric link. *The Annals of Statistics*, 39(3):1720 – 1747.

Chen, Y., von Davier, M., Weng, H., and Xie, Z. (2022). Variable selection in latent regression irt models via knockoffs: An application to international large-scale assessment in education.

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

Dau, H. A., Keogh, E., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., and Hexagon-ML (2018). The ucr time series classification archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Dauxois, J., Pousse, A., and Romain, Y. (1982). Asymptotic theory for the principal component analysis of a vector random function: Some applications to statistical inference. *Journal of Multivariate Analysis*, 12(1):136–154.

Dawid, A. P. (1981). Some matrix-variate distribution theory: Notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274.

Delaigle, A. and Hall, P. (2010). Defining probability density for a distribution of random functions. *The Annals of Statistics*, 38(2):1171 – 1193.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Diebolt, J. and Ip, E. H. (1995). A stochastic em algorithm for approximating the maximum likelihood estimate.

Dogru, F. Z., Bulut, Y. M., and Arslan, O. (2016). Finite mixtures of matrix variate t distributions. *gazi university journal of science*, 29:335–341.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Duda, R. and Hart, P. (1973). Pattern classification and scene analysis. In *A Wiley-Interscience publication.*

Egorov, V. A. and Nevzorov, V. B. (1975). On the rate of convergence of linear combinations of absolute order statistics to the normal law. *Theory of Probability & Its Applications*, 20(1):203–211.

Embrechts, P., Mikosch, T., and Klüppelberg, C. (1997). *Modelling extremal events: for insurance and finance*. Springer-Verlag.

Fraley, C. and Raftery, A. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97:611–631.

Fraley, C. and Raftery, A. (2003). Enhanced model-based clustering, density estimation, and discriminant analysis software: Mclust. *Journal of Classification*, 20:263–286.

Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588.

Fremdt, S., Steinbach, J. G., Horvath, L., and Kokoszka, P. (2013). Testing the equality of covariance operators in functional samples. *Scandinavian Journal of Statistics*, 40(1):138–152.

Gallaugher, M. and McNicholas, P. (2019). Mixtures of skewed matrix variate bilinear factor analyzers. *Advances in Data Analysis and Classification*, 14.

Gallaugher, M. P. B. and McNicholas, P. (2018). Finite mixtures of skewed matrix variate distributions. *Pattern Recognit.*, 80:83–93.

Ghahramani, Z. and Hinton, G. E. (1996). The em algorithm for mixtures of factor analyzers.

Ghahramani, Z. and Hinton, G. E. (1997). The em algorithm for mixtures of factor analyzers. Technical report.

Glanz, H. and Carvalho, L. (2013). An expectation-maximization algorithm for the matrix normal distribution. *Journal of Multivariate Analysis*, 167.

Hall, P. and Keilegom, I. V. (2007). Two-sample test in functional data analysis starting from discrete data. *Statistica Sinica*, 17(4):1511–1531.

Hastie, T., Buja, A., and Tibshirani, R. (1995). Penalized discriminant analysis. *Ann. Statist.*, 23(1):73–102.

Hsing, T. and Eubank, R. (2015). *Theoretical Foundations of Functional Data Analysis, with an Introduction to Linear Operators*. John Wiley & Sons Ltd, West Sussex, UK.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Ieva, F., Paganoni, A. M., Pigoli, D., and Vitelli, V. (2013). Multivariate functional clustering for the morphological analysis of electrocardiograph curves. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(3):401–418.

Ip, E. H. (1995). *A stochastic EM estimator in the presence of missing data: Theory and applications*. PhD thesis, Stanford University.

Jacques, J. and Preda, C. (2013). Funclust: A curves clustering method using functional random variables density approximation. *Neurocomputing*, 112:164–171. Advances in artificial neural networks, machine learning, and computational intelligence.

Jacques, J. and Preda, C. (2014a). Functional data clustering: a survey. *Advances in Data Analysis and Classification*, 8(3):231–255.

Jacques, J. and Preda, C. (2014b). Model-based clustering for multivariate functional data. *Computational Statistics & Data Analysis*, 71:92–106.

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

Kayano, M., Dozono, K., and Konishi, S. (2010). Functional cluster analysis via orthonormalized gaussian basis expansions and its application. *J. Classif.*, 27(2):211–230.

Keribin, C. (2000). Consistent estimation of the order of mixture models. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 62(1):49–66.

Kiers, H. A. (2002). Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems. *Computational Statistics & Data Analysis*, 41(1):157–170. Matrix Computations and Statistics.

Kiers, H. A. L. (1990). Majorization as a tool for optimizing a class of matrix functions. *Psychometrika*, 55(3):417–428.

Kim, N.-H. and Browne, R. (2018). Subspace clustering for the finite mixture of generalized hyperbolic distributions. *Advances in Data Analysis and Classification*.

Leadbetter, M. R., Lindgren, G., and Rootzen, H. (1983). *Extremes and Related Properties of Random Sequences and Processes*. Springer Series in Statistics. Springer Verlag.

Lin, Z., Müller, H.-G., and Yao, F. (2018). Mixture inner product spaces and their application to functional data analysis. *The Annals of Statistics*, 46(1):370 – 400.

Martino, A., Ghiglietti, A., Ieva, F., and Paganoni, A. (2017). A k-means procedure based on a mahalanobis type distance for clustering multivariate functional data. *Statistical Methods & Applications*, 28.

Mclachlan, G., Bean, R., and Ben-Tovim Jones, L. (2007). Extension of the mixture of factor analyzers model to incorporate the multivariate distribution. *Computational Statistics & Data Analysis*, 51:5327–5338.

McLachlan, G. and Krishnan, T. (2007). *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics. Wiley.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley.

McLachlan, G., Peel, D., and Bean, R. (2003). Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics & Data Analysis*, 41(3):379–388. Recent Developments in Mixture Model.

McNeil, A. J., Frey, R., and Embrechts, P. (2015). *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton University Press, Princeton, NJ, USA.

McNicholas, P. and Murphy, T. (2008). Parsimonious gaussian mixture models. *Statistics and Computing*, 18:285–296.

McNicholas, P. D. (2016). Model-based clustering. *Journal of Classification*, 33:331 – 373.

Melnykov, V. and Zhu, X. (2018a). On model-based clustering of skewed matrix data. *Journal of Multivariate Analysis*, 167.

Melnykov, V. and Zhu, X. (2018b). Studying crime trends in the usa over the years 2000–2012. *Advances in Data Analysis and Classification*, 13.

Meng, X.-L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278.

Meng, X.-L. and Van Dyk, D. (1997). The em algorithm—an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):511–567.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209:415–446.

Moschopoulos, P. (1985). The distribution of the sum of independent gamma random variables. *Annals of the Institute of Statistical Mathematics*, 37:541–544.

Nathan, A. (2008). Analysis of pitchf/x pitched baseball trajectories.

Nielsen, S. F. (2000). The stochastic em algorithm: Estimation and asymptotic results. *Bernoulli*, 6(3):457–489.

Parsons, L., Haque, E., and Liu, H. (2004). Subspace clustering for high dimensional data: A review. *SIGKDD Explor. Newsl.*, 6(1):90–105.

Pesevski, A., Franczak, B., and McNicholas, P. (2017). Subspace clustering with the multivariate-t distribution. *Pattern Recognition Letters*, 112.

Qiao, X., Guo, S., and James, G. M. (2019). Functional graphical models. *Journal of the American Statistical Association*, 114(525):211–222.

Ramsay, J. and Silverman, B. (2005). *Functional Data Analysis*. Springer Series in Statistics. Springer New York.

Ramsay, J. O., Hooker, G., and Graves, S. (2009). *Functional Data Analysis with R and MATLAB*. Springer Publishing Company, Incorporated, 1st edition.

Rice, J. A. and Silverman, B. W. (1991). Estimating the mean and covariance structure nonparametrically when the data are curves. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(1):233–243.

Roeder, K. and Wasserman, L. (1997). Practical bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, 92(439):894–902.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Sangalli, L. M., Secchi, P., Vantini, S., and Vitelli, V. (2010). k-mean alignment for curve clustering. *Computational Statistics & Data Analysis*, 54(5):1219–1233.

Saporta, G. (1981). *Méthodes exploratoires d'analyse de données temporelles*. Theses, Université Pierre et Marie Curie - Paris VI.

Sarkar, S., Zhu, X., Melnykov, V., and Ingrassia, S. (2019). On parsimonious models for modeling matrix data. *Computational Statistics & Data Analysis*, 142:106822.

Scheffé, H. (1947). A useful convergence theorem for probability distributions. *The Annals of Mathematical Statistics*, 18(3):434–438.

Schmutz, A., Jacques, J., Bouveyron, C., Cheze, L., and Martin, P. (2020). Clustering multivariate functional data in group-specific functional subspaces. *Computational Statistics*.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Scott, D. and Thompson, J. (1983). Probability density estimation in higher dimension. *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface.*

Severini, T. A. (2007). Integrated likelihood functions for non-bayesian inference. *Biometrika*, 94(3):529–542.

Sharp, A. and Browne, R. P. (2021). Functional data clustering by projection into latent generalized hyperbolic subspaces. *Advances in Data Analysis and Classification.*

Silverman, B. W. (1996). Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics*, 24(1):1 – 24.

Singhal, A. and Seborg, D. (2005). Clustering multivariate time-series data. *Journal of Chemometrics*, 19:427 – 438.

Srivastava, M., von Rosen, T., and von Rosen, D. (2008). Models with a kronecker product covariance structure: Estimation and testing. *Mathematical Methods of Statistics*, 17:357–370.

Steele, R. and Raftery, A. (2010). Performance of bayesian model selection criteria for gaussian mixture models 1. *Frontiers of Statistical Decision Making and Bayesian Analysis.*

Tokushige, S., Yadohisa, H., and Inada, K. (2007). Crisp and fuzzy k-means clustering algorithms for multivariate functional data. *Computational Statistics*, 22:1–16.

Tomarchio, S., McNicholas, P., and Punzo, A. (2021). Matrix normal cluster-weighted models. *Journal of Classification.*

Tomarchio, S., Punzo, A., and Bagnato, L. (2020). Two new matrix-variate distributions with application in model-based clustering. *Computational Statistics & Data Analysis*, 152:107050.

Viroli, C. (2011a). Finite mixtures of matrix normal distributions for classifying three-way data. *Statistics and Computing*, 21:511–522.

Viroli, C. (2011b). Model based clustering for three-way data structures. *Bayesian Analysis*, 6(4):573 – 602.

Vohra, K., Vodonos, A., Schwartz, J., Marais, E. A., Sulprizio, M. P., and Mickley, L. J. (2021). Global mortality from outdoor fine particle pollution generated by fossil fuel combustion: Results from geos-chem. *Environmental Research*, 195:110754.

Wang, J.-L., Chiou, J.-M., and Müller, H.-G. (2016). Functional data analysis. *Annual Review of Statistics and Its Application*, 3(1):257–295.

Wang, L. (2008). *Karhunen-Loeve Expansions and their Applications*. PhD thesis.

Wei, G. C. G. and Tanner, M. A. (1990). A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704.

Wolfe, J. H. (1965). A computer program for the maximum likelihood analysis of types. *Technical Bulletin 65-15, U.S Naval Personnel Research Activity.*

Wu, C. F. J. (1983). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95 – 103.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

Zambom, A. Z., Collazos, J. A. A., and Dias, R. (2019). Function data clustering via hypothesis testing k-means. *Computational Statistics*, 34:527–549.

Zhang, J.-T., Liang, X., and Xiao, S. (2011). On the two-sample behrens-fisher problem for functional data. *Journal of Statistical Theory and Practice*, 4.

Zhang, S., Chen, Y., and Liu, Y. (2020). An improved stochastic em algorithm for large-scale full-information item factor analysis. *British Journal of Mathematical and Statistical Psychology*, 73(1):44–71.

Zhu, H., Strawn, N., and Dunson, D. B. (2016). Bayesian graphical models for multivariate functional data. *Journal of Machine Learning Research*, 17(204):1–27.

APPENDICES

Appendix A

Supplementary Material for Functional Data Clustering by Projection in Latent Generalized Hyperbolic Subspaces

A.1 Parameters Used in Model Selection Simulation

The parameters used to generate observations are as follows. The mean parameters are,

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{M}_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \mathbf{M}_4 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The covariance parameters are the same across groups and across dimensions and are specifed as,

$$\mathbf{\Phi}_{ig} = \begin{bmatrix} 1.5157166 & 0 \\ 0 & 1.5157166 \end{bmatrix}, \quad \text{and} \quad \eta_{ig} = 0.7578583,$$

where $i = 1, 2$.

Appendix B

Supplementary Materials for A Joint Factor Analyzer and Functional Subspace Model for Clustering Multivariate Functional Data

B.1 Countries Included in the Energy Sector Analysis

We gather complete data on the following 97 countries:

Africa	Asia	Europe	North America	Oceania	South America
Algeria	Bahrain	Albania	Canada	Australia	Argentina
Angola	Bangladesh	Austria	Costa Rica	Indonesia	Bolivia
Benin	Brunei	Belarus	Cuba	New Zealand	Brazil
Botswana	China	Belgium	Dominican Republic		Chile
Cameroon	India	Bulgaria	Guatemala		Colombia
Cote d'Ivoire	Indonesia	Cyprus	Haiti		Ecuador
Egypt	Iran	Czech Republic	Honduras		El Salvador
Ethiopia	Israel	Denmark	Jamaica		Peru
Gabon	Japan	Finland	Mexico		Uruguay
Ghana	Jordan	France	Nicaragua		Venezuela
Kenya	Korea	Germany	Panama		
Mauritius	Lebanon	Hungary	Trinidad and Tobago		
Morocco	Malaysia	Ireland	United States		
Mozambique	Mongolia	Italy			
Namibia	Nepal	Netherlands			
Nigeria	Oman	North Macedonia			
South Africa	Pakistan	Norway			
Tanzania	Philippines	Poland			
Togo	Russian Federation	Portugal			
Tunisia	Saudi Arabia	Romania			
Zambia	Singapore	Spain			
Zimbabwe	Sri Lanka	Sweden			
	Thailand	Switzerland			
	Turkey	Turkey			
	Vietnam	United Kingdom			
	Yemen				

Table B.1: Countries used in the energy sector analysis, sorted by geographical location and listed in alphabetical order.

The following table lists the countries assigned to each group in alphabetical order.

Group 1	Group 2	Group 3	Group 4	Group 5
Australia	Argentina	Austria	Belgium	Albania
Bulgaria	Botswana	Bangladesh	Costa Rica	Algeria
Canada	Cameroon	Brazil	Cyprus	Angola
Czech Republic	Chile	Cuba	Dominican Republic	Bahrain
France	China	Denmark	El Salvador	Belarus
Germany	Colombia	Guatemala	Finland	Benin
Hungary	Ethiopia	Ireland	Haiti	Bolivia
Japan	India	Israel	Honduras	Brunei
South Korea	Indonesia	Italy	Jamaica	Cote d'Ivoire
Netherlands	Malaysia	Mexico	Kenya	Ecuador
North Macedonia	Mongolia	New Zealand	Lebanon	Egypt
Poland	Morocco	Philippines	Mauritius	Gabon
Romania	Mozambique	Thailand	Namibia	Ghana
Spain	Pakistan		Nepal	Iran
Turkey	Peru		Nicaragua	Jordan
United Kingdom	Russian Federation		Panama	Nigeria
United States	South Africa		Portugal	Norway
	Tanzania		Singapore	Oman
	Vietnam		Sri Lanka	Saudi Arabia
	Zambia		Sweden	Trinidad and Tobago
	Zimbabwe		Switzerland	Tunisia
			Togo	Venezuela
			Uruguay	Yemen

Table B.2: Countries used in the energy sector analysis sorted by the best BIC model grouping and listed in alphabetical order.

B.2 Code for Scraping Pitch Data

```
> get_statcast = function (start_date, end_date)
+ {
+   if (!is.character(start_date) | !is.character(end_date)) {
+     stop("Please wrap your dates in quotations in the 'yyyy-mm-dd' format.")
+   }
+   if (as.Date(start_date) <= "2015-03-01") {
+     warning("Some metrics such as Exit Velocity and Batted Ball Events have
+             only been compiled since 2015.")
+   }
+   if (as.Date(start_date) <= "2008-03-25") {
+     stop("The data are limited to the 2008 MLB season and after.")
+   }
+   if (as.Date(start_date) > as.Date(end_date)) {
+     stop("The start date is later than the end date.")
+   }
+   year <- substr(start_date, 1, 4)
+   days <- seq.Date(as.Date(start_date), as.Date(end_date),
+                   by = "day")
+   start_days <- as.character(days[(1:length(days))%%7 == 1])
+   end_days <- as.character(days[(1:length(days))%%7 == 0])
+   res <- list()
+   n <- max(length(start_days), length(end_days))
+   res <- foreach(i = 1:n) %do% {
+     if (i == n)
+       end_days[i] <- end_date
+     url <- paste0("https://baseballsavant.mlb.com/statcast_search/csv?all=true",
+                  "&hfPT=&hfAB=&hfBBT=&hfPR=&hfZ=
+                    &stadium=&hfBBL=&hfNewZones=&hfGT=R%7CPO%7CS%7C&hfC&hfSea=",
+                  year,
+                  "%7C&hfSit=&hfOuts=&opponent=&pitcher_throws=&batter_stands=&",
+                  "hfSA=&player_type=pitcher&hfInfield=&team=&position=&",
+                  "hfOutfield=&hfRO=&home_road=&game_date_gt=",
+                  start_days[i],
+                  "&game_date_lt=",
+                  end_days[i],
+                  "&hfFlag=&hfPull=&metric_1=&hfInn=&min_pitches=0&",
+                   "min_results=0&group_by=name&sort_col=pitches&",
+                   "player_event_sort=h_launch_speed&sort_order=desc&min_abs=0",
+                   "&type=details")
+     suppressMessages( suppressWarnings( readr::read_csv(url,na = "null") ) ) %>%
+         select( game_year, game_date, game_pk, pitcher_name=player_name, inning,
+                 inning_topbot, strikes, balls, outs_when_up, p_throws,
+                 pitch_number, pitch_type, pitch_name, release_speed,
+                 release_pos_x, release_pos_y, release_pos_z, plate_x,
+                 plate_z, vx0, vy0,vz0, ax, ay, az,
+                 launch_speed, launch_angle, effective_speed,
+                 release_spin_rate, release_extension,
+                 launch_speed_angle, zone, type, at_bat_number, stand,
+                 events, description, bb_type,
+                 hit_location, hc_x, hc_y, hit_distance_sc
+         )
+   }
+   res_data <- do.call("rbind", res) %>%
+     arrange( game_year,game_date, game_pk, inning, desc(inning_topbot),
+              at_bat_number, pitch_number) %>%
+     as.data.frame()
+
+   return(res_data)
+ }
```

B.3 Parameter Specification for the Model Selection and Parameter Recovery Simulation

In our parameter recovery simulation, we specify three simulation parameters which we choose to vary the value of across different implementations. Every other model parameter which is not mentioned here is fixed across these implementations. In this section we give a brief overview of how these parameters were generated. In particular, a clever specification for these parameters eluded us, so we instead proceeded to generate the parameters randomly. The generation process was the same for each group, and proceeded in the following manner. The mean matrix $\mathbf{M}_g^{\star}$ was generated from a matrix normal distribution specified as $\mathcal{N}_{p\times d_g}(\mathbf{0}, \mathbf{I}_p, 4\mathbf{I}_{d_g})$. Next, we created a $2p \times p$ matrix filled with iid samples from a standard normal distribution. We then estimate the covariance matrix of these data and set $\mathbf{\Lambda}_{1g}$ to be the first q_g eigenvectors found in the corresponding spectral decomposition. Let $\mathcal{U}_1 \sim \text{Unif}(50, 100)$ and $\mathcal{U}_2 \sim \text{Unif}(0.5, 5)$ be two uniform random variables. Let $\mathbf{\Omega}_g$ be a $d_g \times d_g$ diagonal matrix with diagonal elements comprised of iid draws from $\mathcal{U}_1$. Let η_g be the result of a single draw from $\mathcal{U}_2$. We proceeded to construct a b-dimensional diagonal matrix $\mathbf{\Delta}_g$ from these by specifying the diagonal to be $\mathbf{\Omega}_g$ followed by $p - d_g$ copies of η_g. We then set,

$$\begin{aligned} \mathbf{\Omega}_{2g} &= |\mathbf{\Delta}_g|^{\text{-}1/b}\, \mathbf{\Omega}_g, \quad \text{and,} \\ \eta_{2g} &= |\mathbf{\Delta}_g|^{\text{-}1/b}\, \eta_g. \end{aligned}$$

The associated matrix of eigenvalues, $\mathbf{\Gamma}_{2g}$, was generated randomly from a uniform distribution over the $b \times b$ orthogonal matrices. This completes the parameter generation process.

B.4 Parameter Specification for Comparative Analysis II

As mentioned in Chapter 5.4.4, MFSF and the *funHDDC* model overlap when we specify our factor loadings and specific variances to have the form $\mathbf{\Lambda}_{1g} = \mathbf{\Gamma}_{1g}(\mathbf{\Omega}_{1g} - \eta_{1g}\mathbf{I}_{q_g})^{\frac{1}{2}}$ and $\mathbf{\Xi}_{1g} = \eta_{1g}\mathbf{I}_p$ respectively, where $\mathbf{\Omega}_{1g}$ is a diagonal $q_g \times q_g$ matrix and η_{1g} is a positive real number which is less than any of the entries of $\mathbf{\Omega}_{1g}$. Under such a scenario we will have $\mathbf{\Sigma}_{1g} = \mathbf{\Lambda}_{1g}\mathbf{\Lambda}_{1g}^{\mathrm{T}} + \mathbf{\Xi}_{1g} = \mathbf{\Gamma}_{1g}\mathbf{\Delta}_{1g}\mathbf{\Gamma}_{1g}^{\mathrm{T}}$, where $\mathbf{\Delta}_{1g}$ has the subspace clustering form given in Equation (5.28) with $\mathbf{\Omega}_{1g}$ in place of $\mathbf{\Omega}_g$ and η_{1g} replacing η_g. Hence, $\mathbf{\Sigma}_{1g}$ has both the factor analyzer and subspace clustering form. The resulting group covariance matrix then has the subspace clustering form with $\mathbf{\Omega}_g$ given by Equation (5.31) and η_g given by $\eta_{1g}\eta_{2g}$. Under such parameter specification, the MFSF and *funHDDC* overlap.

For our comparative analysis, this model specification serves as basis for $M3$, the situation in which parameter specification satisfies the requirements of both the *funHDDC* algorithm as well as MFSF. With this starting point, we devise a way to deterministically perturb these parameters so that they satisfy only one of the competing models, rather than both. To do this, we need to identify a defining characteristic of each model that is not important for the other. For the *funHDDC* model, that characteristic is the constant value of the trailing eigenvalues, while for MFSF it is the presence of the kronecker product form. We begin with the former. All subsequent discussion will pertain to a particular, but arbitrary, group g of the model. We hence drop the subscript g in the remainder.

The overlap model is characterized by the dual specification of a latent factor model and a latent subspace model through $\mathbf{\Sigma}_1$. However, as we noted previously in Chapter 5.4.4, when $\mathbf{\Sigma}_1$ does not exhibit the latent subspace form, then the *funHDDC* model no longer holds. Hence, our goal is to find a transformation that can be applied to $\mathbf{\Sigma}_1$ which will weaken or remove its latent subspace structure, but preserve its latent factor structure. One obvious way to do this is to alter the specific variances $\mathbf{\Xi}_1$. In particular, the latent subspace structure requires $\mathbf{\Xi}_1$ to be spherical, so altering its diagonal values so that they differ from one another will subjugate $\mathbf{\Sigma}_1$ to deviance from this model. To this end, let $\{\delta_{1i}\}$ denote the trailing $p-q$ eigenvalues of $\mathbf{\Sigma}_1$. Under the latent subspace model $\delta_{1i} = \eta_1$ for each i. We let the following linear relationship define the δ_{1i},

$$\delta_{1i} = \eta_1 + a\left[\frac{\omega_{1q} - \eta_1}{p-q-1}\right](p-q-i), \quad i = 1, 2, ..., p-q,$$

where ω_{1q} is the smallest element of $\mathbf{\Omega}_1$, and a is a value between 0 and 1. We see that when $a = 0$, we recover the latent subspace structure, while $a = 1$ results in equally spacing the δ_{1i} along the line between the point ω_{1q} and η_1. A graphical example of how this changes the eigenvalues of $\mathbf{\Sigma}_1$ for different values of a is provided in Figure B.1. In this figure, $q = 2$, and hence the first two points plotted in the Figure correspond to $\mathbf{\Omega}_1$. In the case that $a = 0$, which corresponds to the black line, we get a constant line at η_1, recovering the latent subspace structure. As a increases the eigenvalues are lifted above η_1 at different rates, causing them to take different values and eliminating the latent subspace structure.

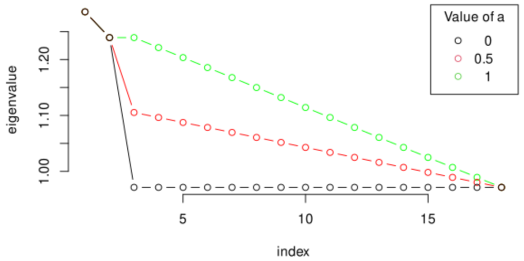


Figure B.1: A depiction of the trailing eigenvalues for different values of the parameter a. We see that as a increases, the eigenvalues move up like a drawbridge. This eliminates the subspace structure from $\mathbf{\Sigma}_1$.

For our simulation, the value $a = 0.5$ corresponds to $M4$ and the value $a = 1$ corresponds to $M5$.

A central component of MFSF is the assumption that the model covariance matrix is formed as the kronecker product of two lower dimensional matrices. When we consider parameter specifications that overlap with the *funHDDC* model, this causes the $\mathbf{\Omega}$ matrix to have the form given in Equation (5.28). From this structure, we see that $\mathbf{\Omega}$ will always have repeated eigenvalues, thanks to the terms involving $\eta_1\mathbf{\Omega}_2$ and $\eta_2\mathbf{\Omega}_1$. This property is a direct result of the kronecker product assumption, hence, if we transform $\mathbf{\Omega}$ so that no repeated values appear, the resulting model will no longer satisfy the MFSF modelling assumptions. Note that under the *funHDDC* assumptions, $\mathbf{\Omega}$ is arbitrary (aside from being diagonal with nonegative entries), so these assumptions will still be satisfied.

Let (ω_{ij}) be the sorted vector of the repeated eigenvalues of $\mathbf{\Omega}$ under $M3$, where i indexes the unique eigenvalues, and j indexes the repetitions of each, and let $\{\omega_i\}$ be the corresponding set of unique values. We specify the relationship between the eigenvalues using a linear model. If ω_i belongs to $\eta_1\mathbf{\Omega}_2$ then,

$$\omega_{ij} = \omega_{i-1} + a\left[\frac{\omega_{i-1} - \omega_i}{p - q}\right](p - q - j)$$

where a is again some value between 0 and 1. If ω_i belongs to $\eta_2\mathbf{\Omega}_1$, then,

$$\omega_{ij} = \omega_{i-1} + a\left[\frac{\omega_{i-1} - \omega_i}{b-d}\right](b-d-j).$$

This approach works in exactly the same manner as the previous. By increasing a, we raise the set of repeated eigenvalues like a drawbridge to connect them with the preceding eigenvalue. By doing this, we remove all repetitions in the eigenvalues, and hence remove the kronecker structure. Setting the value of a to 0.5 corresponds to model $M2$ and setting the value of a to be 1 corresponds to model $M1$.

In our study, for each of the 12 scenarios, we generated one parameter set according to the specification $M3$ and then modified these according to the rules described above to obtain the parameters for models $M1 - M5$. Lacking clever ideas for choosing the particular values of these parameters ourselves, we resigned to generating them randomly. Generation proceeded in the following manner. The mean matrix for the first group, denoted by $\mathbf{M}_1$, was generated from $\mathcal{N}_{p\times b}(\mathbf{0}, \mathbf{I}_p, \mathbf{I}_b)$, which is the standard matrix normal distribution. The mean of the other parameter group, denoted by $\mathbf{M}_2$, was determined by adding a random pb-dimensional vector of length ρ to $\mathbf{M}_1$, where ρ is the value such that $\|\mathbf{M}_1 - \mathbf{M}_2\| = \rho$, which is specified by each of the experimental conditions. Define the random variables $\mathcal{U}_1 \sim \text{Unif}(5, 5.5)$ and $\mathcal{U}_2 \sim \text{Unif}(0.5, 5)$. Let $\mathbf{\Omega}^\star_g$ be a $q_g \times q_g$ diagonal matrix with elements composed of iid draws from $\mathcal{U}_1$. Define $\eta^\star_g$ as a single draw from $\mathcal{U}_2$. Construct a p-dimensional diagonal matrix $\mathbf{\Delta}^\star$ from these by specifying the diagonal as $\mathbf{\Omega}^\star$ followed by $p - q_g$ copies of $\eta^\star$. We then set,

$$\begin{aligned}\mathbf{\Omega}_{1g} &= |\mathbf{\Delta}^\star|^{\text{-}1/p}\,\omega^\star, \quad \text{and,}\\ \eta_{1g} &= |\mathbf{\Delta}^\star|^{\text{-}1/p}\,\eta^\star\end{aligned}$$

from which we can then construct $\mathbf{\Gamma}_{1g}$ and $\mathbf{\Xi}_{1g}$. The parameter $\mathbf{\Delta}_{2g}$ is generated similarly, but with p replaced by b and q_g replaced with d_g. For simplicity, we specify that the eigenvector matrix is equal to identity for each group. In each of the 12 experimental conditions, the hyperparameters q_g and d_g are set to 2 and 3 respectively. This results in a value of k for the *funHDDC* model of 34 for the low dimensional settings, and 114 for the high dimensional settings.

Appendix C

Proofs for Maximum Contribution to the Likelihood

Proposition C.1 (Nielsen (2000))**.** *Suppose the following properties hold regarding the Markov chain* $\tilde{\theta}_{n,k}$ *and stationary distribution* $\tilde{\theta}_n$ *of the stochastic* EM *algorithm.*

(i) Suppose $\tilde{\theta}_{n,k} = \hat{\theta}_n + (n^{-1/2})h + o(n^{-1/2})$ *and* $\tilde{Z}_i \sim k(z_i \mid x_i, \tilde{\theta}_{n,k})$*. Then, for almost all observed sample sequences, the transition probabilities of the stochastic* EM *Markov chain converge continously to those of a Gaussian autoregressive process of order 1,*

$$n^{1/2}(\tilde{\theta}_{n,k+1} - \hat{\theta}_n) \to_d N\left(F(\theta_0)^{\mathrm{T}}h,\, \mathbf{I}_y(\theta_0)^{-1}\mathbf{I}_z(\theta_0)\mathbf{I}_y(\theta_0)^{-1}\right),$$

where $\tilde{\theta}_{n,k+1}$ *is the estimate generated by the algorithm based on the simulated* $\tilde{z}_{1:n}$*,* $\mathbf{I}_z = E(I_{z|x}(\theta_0) \mid X = x)$*, and* $F(\theta_0) = I_z(\theta_0)I_y(\theta_0)^{-1}$ *is the expected fraction of missing information.*

(ii) For almost all observed sample sequences, $n^{1/2}(\tilde{\theta}_n - \hat{\theta}_n)$ *is tight conditional on the sample and,*

$$n^{1/2}(\tilde{\theta}_n - \hat{\theta}_n) \to_d N\left(0,\, \mathbf{I}_x(\theta_0)^{-1}\{I - (I + F(\theta_0))^{-1}\}\right),$$

so that the stationary distribution is asymptotically normal and root-n consistent for $\hat{\theta}_n$*.*

(iii) For almost all samples,

$$n^{1/2}(\tilde{\theta}_n - \theta_0) \to_d N\left(0,\, \mathbf{I}_x(\theta_0)^{-1}\{2I - (I + F(\theta_0))^{-1}\}\right).$$

Proof. The proof of this proposition can be found in Nielsen (2000), as the proofs of Lemma 3 for part (*i*), and Theorem 2 for parts (*ii*) and (*iii*). □

To prove the next proposition, we first introduce Condition D which can be found in Chapter 3 of Leadbetter et al. (1983). To do that, we need some additional notation. For a sequence of random variables $\{X_i\}$ and the sets of indices $I = \{i_1, \ldots, i_m\}$ and $J = \{j_1, \ldots, j_{m'}\}$, we define the joint distribution of the random variables with indices in I as $F_I(x_{i_1}, \ldots, x_{i_m})$, and for indices in I and J as $F_{I \cup J}(x_{i_1}, \ldots, x_{i_m})$. In addition, when for the index I all the x_i are equal to a common value u, we use $F_I(u)$ to denote $F_I(u, \ldots, u)$.

Condition C.2 (D, Leadbetter et al. (1983))**.** The condition D will be said to hold if, for sets of integers $I = \{i_1, \ldots, i_m\}$ with $i_1 < \cdots < i_p$ and $J = \{j_1, \ldots, j_{m'}\}$ with $j_1 < \cdots < j_{p'}$ such that $j_1 - i_m \geq \ell$, and real u, we have

$$|F_{I \cup J}(u) - F_I(u) F_J(u)| \leq g(\ell), \tag{C.1}$$

where $g(\ell) \to 0$ as $\ell \to \infty$.

Proposition C.3. *The sequence of likelihood values, $R_x(\tilde{\theta}_{n,k_n:t})$, satisfies condition D.*

Proof. We prove in the general case of an ergodic Markov chain Y_t taking values in $\mathbb{R}^p$, and the associated stochastic process $X_t = R(Y_t)$, where R is some deterministic, continuous, many-to-one function taking values in $\mathbb{R}$. We then link the result to the context of the proposition.

Let μ be the stationary initial distribution of Y_t. It follows that X_t is marginally distributed according to $\mu \circ R^{-1}$, where R^{-1} is the inverse image of R. That is, $P\{X_t \in A\} = P\{Y_t \in R^{-1}(A)\}$ for any A in the σ-algebra associated with X_t. We then also have that $P(X_{t+k} \in A \mid Y_t) = P\{Y_{t+k} \in R^{-1}(A) \mid Y_t\}$ and hence by the ergodicity of Y_t and the continuity of R, $P(X_{t+k} \in A \mid Y_t) \to P(X_t \in A)$ as $k \to \infty$.

To prove condition D, we must first obtain the joint cdf of $X_{i_1}, \ldots, X_{i_m}$ for arbitrary, finite index set $I = \{i_1, \ldots, i_m\}$. We start by finding the distribution of $X_{t+1} \mid X_t$, the conditional distribution given the previous observation.

Since $R(Y_t)$ is many-to-one, knowing the value of X_t provides less information than knowing Y_t. Let $x_t = R(y_t)$ be the observed value. We define C_0 as the preimage of x_t, $C_0 = \{y \mid R(y) = x_t\}$. The distribution of $X_{t+1} \mid X_t$ is then,

$$\begin{aligned} P(X_{t+1} \in A \mid X_t) &= \int_{C_0} P(X_{t+1} \in A \mid y_t) P(Y_t \in C_0)^{-1} dP(y_t) \\ &= \int P(X_{t+1} \in A \mid Y_t = y_t) dP_0(y_t) \end{aligned}$$

where P_0 is the distribution of Y_t restricted to C_0. Extending to arbitary k, we then have,

$$P(X_{t+k} \in A \mid X_t) = \int P(X_{t+k} \in A \mid Y_t = y_t) dP_0(y_t).$$

By the dominated convergence theorem, we also get that $P(X_{t+k} \in A \mid X_t) \to P(X_t \in A)$ as $k \to \infty$.

Since X_t is not a Markov chain, the distribution of $X_{t+k} \mid X_t$ is not sufficient for specifying the joint distribution of the arbitrary chain $X_{i_1}, \ldots, X_{i_m}$. To that end, we continue to derive the distribution of $X_{t+2} \mid X_{t+1}, X_t$ which we then extend to $X_{t+k_2} \mid X_{t+k_1}, X_t$, and finally to a finite, arbitrary number of conditioning terms.

For the random variable $X_{t+2} \mid X_{t+1}, X_t$ we must consider all paths (y_{t+1}, y_t) that can produce the observed values (x_{t+1}, x_t). That is, $X_{t+1} = x_{t+1}$ implies we must have $y_{t+1} \in C_1 = \{y \mid R(y) = x_{t+1}\}$ and simultaneously $X_t = x_t$ means we must have $y_t \in C_0$. The distribution of X_{t+2} given the previous two iterates, X_{t+1} and X_t is characterized by,

$$\begin{aligned} P(X_{t+2} \in A \mid X_{t+1}, X_t) &= \int_{C_1} \int_{C_0} P(X_{t+2} \in A \mid y_{t+1}) P(Y_{t+1} \in C_1 \mid y_t)^{-1} P(Y_t \in C_0)^{-1} dP(y_{t+1}, y_t) \\ &= \int_{C_1} P(X_{t+2} \in A \mid y_{t+1}) dP_{10}(y_{t+1}), \end{aligned}$$

where $P_{10}(y_{t+1})$ represents the distribution of Y_{t+1} on C_1, given that Y_t was observed to be in C_0.

For the more general case $X_{t+k_2} \mid X_{t+k_1}, X_t$ with $k_1 < k_2$ we have,

$$P(X_{t+k_2} \in A \mid X_{t+k_1}, X_t) = \int_{C_1} P(X_{t+k_2} \in A \mid y_{t+k_1}) dP_{k_10}(y_{t+k_1}),$$

where we have defined C_{k_1} and P_{k_10} as previously, replacing 1 with k_1 everywhere appropriate.

Let $K = \{k_1, \ldots, k_m\}$ be an index set with $k_1 < \cdots < k_m$ and suppose we observe values of the random variables $X_{t+k_{m-1}}, \ldots, X_t$. We know that the observed path $(y_{t+k_{m-1}}, \ldots, y_t)$ must lie in $C_{x_{t+k_{m-1}}} \times \cdots \times C_{x_t}$. Hence, in order to find the distribution of $X_{t+k_m} \mid X_{t+k_{m-1}}, \ldots, X_t$ we need to integrate $X_{t+k_m} \mid (y_{t+k_{m-1}}, \ldots, y_t)$ with respect to all possible paths $(y_{t+k_{m-1}}, \ldots, y_t) \in C_{x_{t+k_{m-1}}} \times \cdots \times C_{x_t}$. The distribution can be characterized by,

$$\begin{aligned}
&P(X_{t+k_m} \in A \mid x_{t+k_{m-1}}, \ldots, x_t) = \\
&\quad \int_{C_{k_{m-1}}} \cdots \int_{C_0} P(X_{t+k_m} \in A \mid y_{t+k_{m-1}}) \prod_{i=0}^{m-2} P(y_{t+k_{i+1}} \in C_{k_{i+1}} \mid y_{t+k_i})^{-1} P(Y_t \in C_0)\, dP(y_{t+k_{m-1}}, \ldots, y_t) \\
&\quad = \int_{C_{k_{m-1}}} P(X_{t+k_m} \in A \mid y_{t+k_{m-1}}) dP_{k_{m-1}\cdots 0}(y_{t+k_{m-1}}),
\end{aligned}$$

where we have defined $P_{k_{m-1}\cdots 0}(y_{t+k_{m-1}})$ as the distribution of $Y_{t+k_{m-1}}$ on $C_{k_{m-1}}$ given that the chain of previous values were observed to have been in $C_{x_{t+k_{m-2}}} \times \cdots \times C_{x_t}$, which can be obtained by evaluating all of the integrals that do not depend on X_{t+k_m}.

Now consider two sets of indices, $I = \{i_1, \ldots, i_m\}$ and $J = \{j_1, \ldots, j_{m'}\}$ with $i_1 < \ldots < i_m < j_1 < \ldots < j_{m'}$, and $j_1 - i_m \geq \ell$. For brevity, we write X_{t+i_k} as X_{i_k}, where observation order is implied by the ordering on the elements of the index set. We also use X_I to denote the collection of random variables corresponding to the index I, and x_I the collection of their observed values. We do likewise for the index set J with X_J and x_J respectively.

Then, for any $u \in \mathbb{R}$ we have,

$$\begin{aligned}
F_I(u) &= P(X_I \leq u) \\
&= P(X_{i_m} \leq u, \ldots, X_{i_1} \leq u) \\
&= \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} dP(x_I) \\
&= \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} dP(x_{i_m}, \ldots, x_{i_1}) \\
&= \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} dP(x_{i_m} \mid x_{i_{m-1}}, \ldots, x_{i_1}) \ldots dP(x_{i_1}),
\end{aligned}$$

where $dP(x_{i_m} \mid x_{i_{m-1}}, \dots, x_{i_1}) \dots dP(x_{i_1})$ represents the disintegration of the joint distribution of X_I, which we know exists due to the implicit disintegrability of the joint distribution of the associated elements of the Markov chain Y_t and the continuity of R.

Likewise, for X_J we have,

$$\begin{aligned} F_J(u) &= P(X_J \leq u) \\ &= P(X_{j'_m} \leq u, \dots, X_{j_1} \leq u) \\ &= \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} dP(x_J) \\ &= \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} dP(x_{j'_m}, \dots, x_{j_1}) \\ &= \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} dP(x_{j'_m} \mid x_{j_{m'-1}}, \dots, x_{j_1}) \dots dP(x_{j_1}). \end{aligned}$$

However,

$$\begin{aligned} F_{I \cup J}(u) &= P(X_J \leq u \wedge X_I \leq u) \\ &= \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} dP(x_J \mid x_I) dP(x_I) \\ &= \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} dP(x_{j'_m} \mid x_{j_{m'-1}}, \dots, x_{j_1}, x_I) \dots dP(x_{j_1} \mid x_I) dP(x_I). \end{aligned}$$

We see that the difference between $F_{I \cup J}$ and $F_I F_J$ is that, for each $j \in J$, the conditional distribution of X_j contained in expression for $F_{I \cup J}$ depends on X_I. We therefore need to show that $P(x_J \mid x_I) \to P(x_J)$ as $\ell \to \infty$.

To this end, let $j_\star$ be an arbitrary element of J, and consider the conditional distribution $P(x_{j_\star} \mid x_{j_{\star-1}}, \dots, x_{j_1}, x_I)$. Since $dP(x_{j_\star} \mid x_{j_{\star-1}}, \dots, x_{j_1}, x_I)$ is uniquely determined by the values of $P\{X_{j_\star} \in (-\infty, u) \mid x_{j_{\star-1}}, \dots, x_{j_1}, x_I\}$ for $u \in \mathbb{R}$, it is sufficient to consider the behaviour of these probabilities as ℓ increases.

Recall that we express the probability of interest as,

$$
\begin{aligned}
&P\{X_{j_\star} \in (-\infty, u) \mid x_{j_{\star-1}}, \ldots, x_{j_1}, x_I\} \\
&\quad = \int_{C_{j_\star-1}} P\{X_{j_\star} \in (-\infty, u) \mid y_{j_\star-1}\} dP_{j_\star-1\cdots i_1}(y_{j_\star-1}) \\
&\quad = \int_{C_{j_\star-1}} P\{X_{j_\star} \in (-\infty, u) \mid y_{j_\star-1}\} \int_{C_{j_\star-2}} \cdots \int_{C_{i_1}} dP_{j_\star-1}(y_{j_\star-1} \mid y_{j_\star-2}) \cdots dP_{i_2}(y_{i_2} \mid y_{i_1}) dP_{i_1}(y_{i_1}),
\end{aligned}
$$

where $P_k(y_k \mid y_{k-1})$ denotes the conditional distribution of y_k given y_{k-1} on the set C_k. By the ergodicity of the Markov chain Y_t we have $\lim_{\ell\to\infty} P(Y_{j_1} \mid y_{i_m}) \to P(Y_{j_1})$, and hence we have,

$$
\begin{aligned}
&\lim_{\ell\to\infty} dP_{j_\star-1\cdots i_1}(y_{j_\star-1}) \\
&\qquad = \lim_{\ell\to\infty} \int_{C_{j_\star-2}} \cdots \int_{C_{i_1}} dP_{j_\star-1}(y_{j_\star-1} \mid y_{j_\star-2}) \cdots dP_{j_1}(y_{j_1} \mid y_{i_m}) \cdots dP_{i_2}(y_{i_2} \mid y_{i_1}) dP_{i_1}(y_{i_1}) \\
&\qquad = \int_{C_{j_\star-2}} \cdots \int_{C_{i_1}} \lim_{\ell\to\infty} dP_{j_\star-1}(y_{j_\star-1} \mid y_{j_\star-2}) \cdots dP_{j_1}(y_{j_1} \mid y_{i_m}) \cdots dP_{i_2}(y_{i_2} \mid y_{i_1}) dP_{i_1}(y_{i_1}) \\
&\qquad = \int_{C_{j_\star-2}} \cdots \int_{C_{j_1}} dP_{j_\star-1}(y_{j_\star-1} \mid y_{j_\star-2}) \cdots dP_{j_1}(y_{j_1}) \int_{C_{i_m}} \cdots \int_{C_{i_1}} dP_{i_m}(y_{i_m} \mid y_{i_m-1}) \cdots dP_{i_1}(y_{i_1}) \\
&\qquad = \int_{C_{j_\star-2}} \cdots \int_{C_{j_1}} dP_{j_\star-1}(y_{j_\star-1} \mid y_{j_\star-2}) \cdots dP_{j_1}(y_{j_1}) \\
&\qquad = dP_{j_\star-1\cdots j_1}(y_{j_\star-1}),
\end{aligned}
$$

where the limit and integration can be exchanged because C_k is compact for each $k = j_\star - 1, \ldots, i_1$, and integration is with respect to probability measures on these compact spaces. Hence, we have that,

$$
P\{X_{j_\star} \in (-\infty, u) \mid x_{j_{\star-1}}, \ldots, x_{j_1}, x_I\} \to P\{X_{j_\star} \in (-\infty, u) \mid x_{j_{\star-1}}, \ldots, x_{j_1}\} \quad \text{as } \ell \to \infty.
$$

Since $j_\star$ was arbitary, this holds for all $j \in J$. We then have that,

$$\begin{aligned}
&\lim_{\ell\to\infty} F_{I\cup J}(u) \\
&\qquad = \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} \lim_{\ell\to\infty} dP(x_{j'_m} \mid x_{j_{m'-1}}, \dots, x_{j_1}, x_I) \dots dP(x_{j_1} \mid x_I) dP(x_I) \\
&\qquad = \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} dP(x_{j'_m} \mid x_{j_{m'-1}}, \dots, x_{j_1}) \dots dP(x_{j_1} \mid) \int_{-\infty}^{u} \cdots \int_{-\infty}^{u} dP(x_I) \\
&\qquad = F_J(u) F_I(u),
\end{aligned}$$

where we may take the limit inside integration using Scheffé (1947). Since u is arbitrary, condition D is satisfied. In particular, since $g(\ell)$ does not depend on u we can define $g(\ell) = \sup_u |F_{I\cup J}(u) - F_I(u)F_J(u)|$.

The proposition is now proved by working under a model for which ergodicity of the chain $\tilde{\theta}_{n,k_n:t}$ is satisfied, as laid out in Nielsen (2000), which we assumed to get Proposition C.1, and by defining $X_t = R_x(\tilde{\theta}_{n,t})$.

□

Lemma C.4. *Let $X = \sum_{i=1}^{p} W_i$ be a linear combination of p independent random variables with distribution $W_i \sim$ Gamma(α_i, β_i) where $\beta_1 \le \ldots \le \beta_p$. Define $m_t = \min_{j=1,..,t}\{X_j\}$ as the minimum of t independent draws from X. Then, as $t \to \infty$ we have,*

$$c_t^{-1} m_t \to_d \textit{Weibull}\left(\prod_{i=1}^{p} (\beta_1/\beta_i)^{\alpha_i/\alpha^\star}, \alpha^\star\right),$$

where $\alpha^\star = \sum_i \alpha_i$, and c_t are the normalizing constants such that $-c_t^{-1}\max\{-Y_1, \ldots, -Y_t\}$ converges in distribution as $t \to \infty$ with $Y_j \sim$ Gamma$(\alpha^\star, \beta_1)$ independently.

Proof. Our goal will be to prove the result for $p = 1$, and then use tail equivalence (see Embrechts et al., 1997) to extend the result to $p \ge 2$.

To begin we define the random variable $Y \sim \text{Gamma}(\alpha^\star, \beta_1)$, where $\alpha^\star = \sum_{i=1}^{p} \alpha_i$ and $\beta_1 = \min_{i=1,...,p}\{\beta_i\}$. Our interest is then in the asymptotic distribution of $\min\{Y_1, ..., Y_t\}$. Since $\min\{Y_1, ..., Y_t\} = -\max\{-Y_1, ..., -Y_t\}$ it is sufficient to work with the maximum.

The random variable $-Y$ has finite right end point 0, so it is natural to check that the corresponding distribution function is in the maximum domain of attraction of the Type III extreme value distribution, i.e. the reversed Weibull. A sufficient condition is that the survival function satisfies $S_{-Y}(y^{-1}) = y^{-\alpha^\star} L(y)$, where L is a slowly varying function and $\alpha \in \mathbb{R}$ (Leadbetter et al., 1983). This is equivalent to the statement that the survival function is regulary varying with index $-\alpha$. If a function $g(y)$ is a regularly varying function with index α then it satisfies,

$$\lim_{y\uparrow\infty} \frac{g(\lambda y)}{g(y)} = \lambda^{\alpha},$$

for all $\lambda > 0$. Set $g(y) = S_{-Y}(y^{-1})$, with $y \in \mathbb{R}^-$. Then,

$$\begin{aligned}\lim_{y\to\infty} \frac{g(\lambda y)}{g(y)} &= \lim_{y\to\infty} \frac{S_{-Y}(\{\lambda y\}^{-1})}{S_{-Y}(y^{-1})} \\ &= \lim_{y\to\infty} \frac{F_Y(-\lambda^{-1}y^{-1})}{F_Y(-y^{-1})},\end{aligned}$$

where F_Y is the cdf of Y. Substitution gives,

$$\lim_{y\to\infty} \frac{\Gamma(\alpha^\star)^{-1}\beta_1^{\alpha^\star}\int_0^{-\lambda^{-1}y^{-1}} u^{\alpha^\star-1}e^{-\beta_1 u}\,du}{\Gamma(\alpha^\star)^{-1}\beta_1^{\alpha^\star}\int_0^{-y^{-1}} u^{\alpha^\star-1}e^{-\beta_1 u}\,du} = \lim_{y\to\infty} \frac{\gamma(\alpha^\star, -\beta_1\lambda^{-1}y^{-1})}{\gamma(\alpha^\star, -\beta_1 y^{-1})},$$

where $\gamma(a,x)$ is the lower incomplete gamma function. Using the property that for any $a > 0$,

$$\frac{\gamma(a,x)}{x^a} \xrightarrow{x\downarrow 0} a^{-1},$$

we reexpress the limit as,

$$\lim_{y\to\infty} \frac{\gamma(\alpha^\star, -\beta_1\lambda^{-1}y^{-1})}{(-\beta_1\lambda^{-1}y^{-1})^{\alpha^\star}} \cdot \frac{(-\beta_1 y^{-1})^{\alpha^\star}}{\gamma(\alpha^\star, -\beta_1 y^{-1})} \cdot \lambda^{-\alpha^\star} = \frac{1}{\alpha^\star}\left(\frac{1}{\alpha^\star}\right)^{-1}\lambda^{-\alpha^\star} = \lambda^{-\alpha^\star}.$$

Therefore $\max\{-Y_1, ..., -Y_t\}$ is in the maximum domain of attraction of the reversed Weibull distribution, which has cdf $G(y;\alpha) = e^{-(-y)^{\alpha^\star}}$. We therefore have that, $P(c_t^{-1}(M_t - d_t) < y) \to G(y;\alpha^\star)$ with normalizing sequences $c_t > 0$ and d_t, which can be chosen as

$c_t = -F^{\leftarrow}_{-Y}(1 - t^{-1})$ and $d_t = 0$ (These are, however, not unique. See: Leadbetter et al., 1983). We may then write,

$$\begin{aligned} P(c_t^{-1} M_t < y) &= 1 - P(c_t^{-1} M_t \geq y) \\ &= 1 - P(c_t^{-1} m_t \leq -y), \end{aligned}$$

so that $P(c_t^{-1} m_t \leq -y) \to 1 - G(-y; \alpha^\star)$. We see that $1 - G(-y; \alpha^\star) = 1 - e^{-y^{\alpha^\star}}$ is the cdf of the Weibull distribution with shape parameter $\alpha^\star$ and scale parameter 1. This proves the statement for $p = 1$.

To extend to the proof to $p \geq 2$ we need both $S_{\text{-}X}(x) = F_X(-x)$ and $f_{\text{-}X}(x) = f_X(-x)$, for $x \in \mathbb{R}^-$. These can be obtained from Moschopoulos (1985) as,

$$\begin{aligned} f_X(x) &= c \sum_{k=0}^{\infty} \delta_k x^{\alpha^\star + k - 1} e^{-x/\beta_1} / \left(\Gamma(\alpha^\star + k) \beta_1^{\alpha^\star + k} \right) \mathbb{1}\{x \in \mathbb{R}^+\}, \\ F_X(x) &= c \sum_{k=0}^{\infty} \delta_k \frac{\gamma(\alpha^\star + k, x\beta_1^{-1})}{\Gamma(\alpha^\star + k)}, \end{aligned}$$

where $\beta_1 = \min_i\{\beta_i\}$, $c = \prod_{i=1}^{p} (\beta_1/\beta_i)^{\alpha_i}$, $\alpha^\star = \sum_{i=1}^{p} \alpha_i$, and the coefficients are described recursively as,

$$\delta_{k+1} = (k+1)^{-1} \sum_{j=1}^{k+1} j \left(\sum_{i=1}^{p} \alpha_i [1 - \beta_1/\beta_i]^k / k \right) \delta_{k+1-j}, \quad k = 0, 1, \ldots \tag{C.2}$$

with $\delta_0 = 1$.

Now, consider the gamma random variable Y with shape parameter $\alpha^\star$ and scale parameter β_1 as in the proof of the $p = 1$ case. Our goal is to show that $-Y$ and $-X$ are tail-equivalent, from which it will follow that they belong to their distributions belong to the same maximum domain of attraction. Recall $-Y$ is in the maximum domain of attraction of the reversed Weibull distribution with parameter $\alpha^\star$, and additionally notice that the distributions of $-X$ and $-Y$ both have right endpoint equal to 0. Now,

$$\begin{aligned}
\lim_{x\to 0}\frac{S_{\text{-}X}(x)}{S_{\text{-}Y}(x)} &= \lim_{x\to 0}\frac{c\sum_{k=0}^{\infty}\delta_k\gamma(\alpha^\star+k,-x\beta_1^{-1})/\Gamma(\alpha^\star+k)}{\gamma(\alpha^\star,-x\beta_1^{-1})/\Gamma(\alpha^\star)} \\
&= c+\lim_{x\to 0}\frac{c\sum_{k=1}^{\infty}\delta_k(-x)^{\alpha^\star+k}/\left[\alpha^\star+k)\Gamma(\alpha^\star+k)\right]}{(-x)^{\alpha^\star}/\left[\alpha^\star\Gamma(\alpha^\star)\right]} \\
&\leq c+\lim_{x\to 0}c\sum_{k=1}^{\infty}|\delta_k|\frac{\alpha^\star\Gamma(\alpha^\star)}{(\alpha^\star+k)\Gamma(\alpha^\star+k)}(-x)^k. \qquad \text{(C.3)}
\end{aligned}$$

To determine the limit on the right-hand side, we find an upper bound for $|\delta_k|$. For any k we have that,

$$\left|\sum_{i=1}^{p}\alpha_i[1-\beta_1/\beta_i]^k/k\right| \leq \alpha^\star b^k/k,$$

where $b=\max_{i\neq 1}(1-\beta_1/\beta_i)$. Using Equation C.2 we may then write,

$$|\delta_{k+1}| \leq \frac{\alpha^\star}{(k+1)}\sum_{j=1}^{k+1}b^j|\delta_{k+1-i}|,$$

and by induction we have,

$$|\delta_{k+1}| \leq \frac{\Gamma(\alpha^\star+k+1)b^{k+1}}{\Gamma(\alpha^\star)(k+1)!}.$$

Plugging this into the second term of Equation C.3 we find,

$$\begin{aligned}
\lim_{x\to 0}\sum_{k=1}^{\infty}|\delta_k|\frac{\alpha^\star\Gamma(\alpha^\star)}{(\alpha^\star+k)\Gamma(\alpha^\star+k)}(-x)^k &\leq \lim_{x\to 0}\sum_{k=1}^{\infty}\frac{\Gamma(\alpha^\star+k)b^k}{\Gamma(\alpha^\star)k!}\frac{\alpha^\star\Gamma(\alpha^\star)}{(\alpha^\star+k)\Gamma(\alpha^\star+k)}(-x)^k \\
&= \lim_{x\to 0}\alpha^\star\sum_{k=1}^{\infty}\frac{(-bx)^k}{k!(\alpha^\star+k)} \\
&\leq \sum_{k=1}^{\infty}\lim_{x\to 0}\frac{(-bx)^k}{k!} \\
&= 0.
\end{aligned}$$

However, the sum is nonnegative, so we have

$$\lim_{x\to 0}\frac{S_{\text{-}X}(x)}{S_{\text{-}Y}(x)} = c,$$

and therefore the dfs of $-X$ and $-Y$ are tail equivalent. By the first part of the proof we may write,

$$F^t_{\text{-}Y}(c_t x) \to e^{-x^{\alpha^\star}}.$$

Taking the logarithm of this expression and using the fact that $S_{-Y}(c_t x) \to 0$ we can write,

$$tS_{\text{-}Y}(c_t x) \to x^{\alpha^\star}.$$

Then, from tail equivalence we have

$$tS_{\text{-}X}(c_t x) \to (c^{1/\alpha^\star}x)^{\alpha^\star},$$

which then gives the result. □

Theorem C.5. *Conditional on the observed data sample and for almost all samples,*

$$t^{2/p}m_{nt} \to_d \mathit{Weibull}\left(2\left\{\frac{p}{2}\Gamma\left(\frac{p}{2}\right)\right\}^{2/p}\prod_{i=1}^{p}\lambda_i^{1/p}, \frac{p}{2}\right),$$

where $\lambda_1, \ldots, \lambda_p$ *are the eigenvalues of* $\mathbf{I}_x(\theta_0)^{\text{-}1/2}(I - [I + F(\theta_0)]^{-1})\mathbf{I}_x(\theta_0)^{1/2}$.

Proof. Let $\{F_n\}$ be the sequence of cdfs associated with the sequence of random variables $\tilde{\theta}_{n,k_n}$ and $\{H_n\}$ the sequence of cdfs associated with the sequence of random variables $R_x(\tilde{\theta}_{n,k_n})$. Since $F_n \to F$ with F continuous, $F_n \to F$ uniformly. By the continuity of R_x it follows that $H_n \to H$ uniformly, where H is a linear combination of independent Gamma distributions.

Let $\{c_t\}$ be the sequence of constants such that $1 - (1 - H_{(}c_t x))^t \to G(x)$, chosen according to e.g. Embrechts et al. (1997), where G is the cdf of the associated extreme value distribution, and x is a continuity point of G. To prove the assertion it is sufficient to show that,

$$\lim_{(n,t)\to(\infty,\infty)} pr(c_t^{-1}m_{nt} \le x) = 1 - e^{-x^{p/2}}, \tag{C.4}$$

for all $x \in \mathbb{R}^+$. The limit in Equation C.4 must be shown to hold for all possible paths of n and t to their respective limit point. To do this, we first consider properties of the individual limits with respect to each argument. With respect to t we have that,

$$\lim_{t\to\infty} pr(c_t^{-1} m_{nt} \leq x) = \lim_{t\to\infty} 1 - (1 - H_n(c_t x))^t, \tag{C.5}$$

exists for all t trivially since $H_n(c_t x)$ is bounded and c_t is nonincreasing in t (hence, $H_n(c_t x)$ is nonincreasing). The three possible limits are 0, 1, and $G(x)$, where G is some extreme value distribution. Hence, the limit exists for all n. Now consider,

$$\lim_{n\to\infty} 1 - (1 - H_n(c_t x))^t. \tag{C.6}$$

By the uniform convergence of $\{H_n\}$, the limit in Equation (C.6) converges uniformly with limit point $1-(1-H(c_t x))^t$. It follows by the Moore-Osgood theorem, that the multivariate limit exists and the limit is equal to the limit of the iterated limit,

$$\lim_{t\to\infty} \lim_{n\to\infty} pr(c_t^{-1} m_{nt} \leq x).$$

We have already shown that this limit evaluates to,

$$\lim_{t\to\infty} \lim_{n\to\infty} pr(c_{nt}^{-1} m_{nt} \leq x) = \lim_{t\to\infty} pr(c_t^{-1} m_t \leq x) = 1 - e^{-x^{p/2}},$$

using Lemma C.4. To get the final result, we use Embrechts et al. (1997) to specify the normalizing constants as $c_t = \{\Gamma(p/2)p/2(2\lambda_p)^{p/2}\}^{2/p} t^{-2/p}$ and then rearrange the components of c_t which are not a function of t into the distribution.

□

Corollary C.6. *When the model parameter θ is a scalar, we have*

$$t^2 m_{nt} \to_d \textit{Weibull}\left(\frac{\pi}{2}\left\{1 - \frac{1}{1+F(\theta_0)}\right\}, \frac{1}{2} \right),$$

as $n, t \to \infty$ for almost all observed data samples and conditional on the sample.

Proof. This follows directly from Theorem C.5 by setting $p = 1$. □

To aid the proof of Theorem 2, we provide a helpful definition and lemma.

Definition C.1 (Symmetric Random Variable)**.** We say that a random variable X is symmetric, has a symmetric distribution, or is symmetric about x, if there exists x such that $\mathrm{pr}(X > x + y) = \mathrm{pr}(X < x - y)$ for all y such that the probabilities are well-defined.

Lemma C.7. *Let X be a random variable symmetric about the value μ. Then, the random variable sgn$\left\{\mathrm{argmin}_{j=1,..,t}(X_i - \mu)^2\right\}$ is independent of $\min_{j=1,..,t}(X_i - \mu)^2$, where sgn$\{\cdot\}$ the function defined on $\mathbb{R}$ such that sgn$\{x\} = -1$ for $-\infty < x < 0$, sgn$\{x\} = 1$ for $0 < x < \infty$, and sgn$\{0\} = 0$.*

Proof. Without loss of generality, assume $\mu = 0$. By the construction provided in the proof of Lemma 5 in Egorov and Nevzorov (1975), if X is symmetric, there exists a random variable G such that G is independent of $|X|$ and,

$$X \overset{d}{=} G|X|,$$

where $pr(G = 1) = pr(G = -1) = 1/2$. In particular, for an i.i.d. sample $X_{1:t}$ drawn from X and an i.i.d. sample $G_{1:t}$ drawn from G we have,

$$\underset{j=1,\dots,t}{\mathrm{argmin}}\, X_i^2 \overset{d}{=} \underset{j=1,\dots,t}{\mathrm{argmin}}(G_i|X_i|)^2$$

and the result follows since the sign associated with the right-hand side is drawn from G independently. That is, $\mathrm{sgn}\left\{\mathrm{argmin}_{j=1,..,t} X_i^2\right\}$ is independent of $|\mathrm{argmin}_{j=1,..,t} X_i^2| = (\min_{j=1,..,t} X_i^2)^{1/2}$. □

Theorem C.8. *Suppose the parameter dimension p is 1, so that Corollary C.6 holds. Then, as both $n, t \to \infty$ the normalized random variable $tn^{1/2}(\tilde{\theta}_{n,\min t} - \hat{\theta}_n)$ satifies,*

$$tn^{1/2}(\tilde{\theta}_{n,\min t} - \hat{\theta}_n) \to_d \mathit{Laplace}\left(0, \mathbf{I}_x(\theta_0)^{-1/2}\frac{\{2\pi F(\theta_0)\}^{1/2}}{2\{1 + F(\theta_0)\}^{1/2}}\right),$$

so that it is asymptotically Laplace distributed with location parameter 0 and scale parameter $(\pi/2)^{1/2}[F(\theta_0)/\{1 + F(\theta_0)\}]^{1/2}\mathbf{I}_x(\theta_0)^{-1/2}$.

Proof. Starting from the definition of m_{nt}, a bit of algebra allows us to write,

$$\begin{aligned}(t^2 m_{nt})^{1/2} &= \{t^2 n \mathbf{I}_x(\theta_0)(\tilde{\theta}_{n,\min t} - \hat{\theta}_n)^2 + o_p(1)\}^{1/2} \\ &= tn^{1/2}|\tilde{\theta}_{n,\min t} - \hat{\theta}_n| + o_p(1),\end{aligned}$$

where we get the second line using the knowledge that $t^2 m_{nt} = O_p(1)$ and the series expansion $(x/y+1)^{1/2} = (1 + 1/2(x/y) + \cdots)$. We therefore have,

$$tn^{1/2}\mathbf{I}_x(\theta_0)^{1/2}(\tilde{\theta}_{n,\min t} - \hat{\theta}_n) = s_{nt}(t^2 m_{nt})^{1/2} + o_p(1),$$

where $s_{nt} = \operatorname{sgn}\left\{n^{1/2}(\tilde{\theta}_{n,\min t} - \hat{\theta}_n)\right\}$. Applying the continuous mapping theorem to Corollary C.6 we have,

$$(t^2 m_{nt})^{1/2} \to_d \text{Weibull}\,(\sigma, 1)\,, \quad \sigma = \frac{\{2\pi F(\theta_0)\}^{1/2}}{2\{1+F(\theta_0)\}^{1/2}}.$$

By similar arguments to the proof of Lemma C.7, we also have that,

$$s_{nt} \to_d \text{Bernoulli}(1/2).$$

Additionally, Lemma C.7 also gives us that s_{nt} and m_{nt} are asymptotically independent, hence,

$$s_{nt}(t^2 m_{nt})^{1/2} \to_d ZX,$$

where $Z \sim \text{Bernoulli}(1/2)$, $X \sim \text{Weibull}(\sigma, 1)$, and $Z \perp X$. The characteristic function of this random variable is,

$$\begin{aligned} E(e^{itZX}) &= \int_{\mathbb{R}^+} \sum_{z\in\{\text{-}1,1\}} e^{itzx}\frac{1}{2\sigma}e^{-x/\sigma}\,dx \\ &= \frac{1}{2\sigma}\int_{\mathbb{R}^+} e^{-(\sigma^{-1}+it)x}\,dx + \frac{1}{2\sigma}\int_{\mathbb{R}^+} e^{(it-\sigma^{-1})x}\,dx \\ &= \frac{1}{2(1+it\sigma)} + \frac{1}{2(1-it\sigma)} \\ &= (1+\sigma^2 t^2)^{-1} \end{aligned}$$

which is the characteristic function of a Laplace$(0,\sigma)$ distribution. □

Proposition C.9. *Let $\tilde{\theta}_{ni,k_n}$ be the ith component of $\tilde{\theta}_{n,k_n}$, $i = 1, \ldots, p$. Then, the marginal chain $\sqrt{n}(\tilde{\theta}_{ni,k_n:t} - \hat{\theta}_{ni})$ converges in distribution to a sample of the same length from the stationary marginal vector autoregressive process with autoregressive parameter $F(\theta_0)_i^{\mathrm{T}}$ and innovation variance $\{\mathbf{I}_y(\theta_0)^{-1}\mathbf{I}_z(\theta_0)\mathbf{I}_y(\theta_0)^{-1}\}_{ii}$. It follows that,* *(i) For all observed sample sequences and conditional on the sample,*

$$\sqrt{n}(\tilde{\theta}_{ni,k_n} - \hat{\theta}_{ni}) \to_d N(0, \mathbf{I}_x^{-1}(\theta_0)_{ii} - \mathbf{I}_x^{-1}(\theta_0)_{i\cdot}^{\mathrm{T}}(I + F(\theta_0))_{\cdot i}^{-1}).$$

(ii) Unconditionally,

$$\sqrt{n}(\tilde{\theta}_{ni,k_n} - \theta_{0i}) \to_d N(0, 2\mathbf{I}_x^{-1}(\theta_0)_{ii} - \mathbf{I}_x^{-1}(\theta_0)_{i\cdot}^{\mathrm{T}}(I + F(\theta_0))_{\cdot i}^{-1}).$$

Proof. This follows immediately from Proposition C.1. □

Proposition C.10. *Under the assumed regularity conditions on the observed data model, for a parameter component value of the form $\theta_{ni} = \hat{\theta}_{ni} + n^{\text{-}1/2}h$ the profile log-likelihood ratio exhibits the approximation,*

$$R_{\mathrm{p}}(\theta_{ni}) = n\mathbf{I}_x^{-1}(\theta_0)_{ii}(\theta_{ni} - \hat{\theta}_{ni})^2 + o_p(1),$$

for almost all observed data samples, and conditional on the sample.

Proof. Let $\theta = \hat{\theta}_n + n^{\text{-}1/2}h$ with $h \in \mathbb{R}^p$. Recall that regularity of the observed data model gives us that,

$$R_x(\theta_n) = h^{\mathrm{T}} I_x(\theta_0)h + r_n(h), \tag{C.7}$$

where, for any $M > 0$ and θ_0 in the interior of Θ, we have that $\sup_{|h|\leq M}|r_n(h)|$ converges to 0 as $n \to \infty$. The profile loglikelihood ratio follows by fixing the value $\theta_{ni} = \hat{\theta}_{ni} + n^{\text{-}1/2}h$ and maxmizing Equation C.7 with respect to the remaining parameters over their parameter space, which we define as Θ_{-i}.

Define $c_n = \sup_{|h^\star|\leq|h|}|r_n(h^\star)|$. Then,

$$\min_{\theta\in\Theta_{-i}} h^{\mathrm{T}} I_x(\theta_0)h - c_n \leq \min_{\theta\in\Theta_{-i}} h^{\mathrm{T}} I_x(\theta_0)h \leq \min_{\theta\in\Theta_{-i}} h^{\mathrm{T}} I_x(\theta_0)h + c_n,$$

and since $c_n \to 0$ as $n \to \infty$ it follows that the contribution of $r_n(h)$ to the minimization is asymptotically negigible, is the sense that $\min_{\theta\in\Theta_{-i}} h^{\mathrm{T}} I_x(\theta_0)h + r_n(h) \to \min_{\theta\in\Theta_{-i}} h^{\mathrm{T}} I_x(\theta_0)h$ as $n \to \infty$, so it can be ignored.

Computation of the profile likelihood at $\theta_{ni} = \hat{\theta}_{ni} + n^{-1/2}h$ can therefore be written asymptotically as the following quadratic programming problem,

$$\begin{aligned} &\text{minimize} \quad h^{\mathrm{T}}\mathbf{I}_x(\theta_0)h \\ &\text{subject to} \quad Ah = d, \end{aligned}$$

where,

$$\begin{aligned} A &= (0, \ldots, 0, 1, 0, \ldots, 0)^{\mathrm{T}} \\ d &= n^{1/2}(\theta_i - \hat{\theta}_{ni}). \end{aligned}$$

Using Lagrange's method, the solution is given by the system of equations,

$$\begin{bmatrix} \mathbf{I}_x(\theta_0) & A \\ A^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} h \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ d \end{bmatrix},$$

where λ is the Lagrange multipler. The solution for h is given by,

$$\begin{aligned} \hat{h} &= \mathbf{I}_x^{-1}(\theta_0)A\big(A^{\mathrm{T}}\mathbf{I}_x^{-1}(\theta_0)A\big)^{-1}d \\ &= \frac{d}{\mathbf{I}_x^{-1}(\theta_0)_{ii}}\mathbf{I}_x^{-1}(\theta_0)_{\cdot i}, \end{aligned}$$

where $\mathbf{I}_x^{-1}(\theta_0)_{cdoti}$ is the ith column of $\mathbf{I}_x^{-1}(\theta_0)$ and $\mathbf{I}_x^{-1}(\theta_0)_{ii}$ is the ith diagonal element. The value of the profile likelihood ratio is then,

$$\begin{aligned} R_p\big(\hat{h}\big) &= \hat{h}^{\mathrm{T}}\mathbf{I}_x(\theta_0)\hat{h} \\ &= \frac{d^2}{\big(\mathbf{I}_x^{-1}(\theta_0)_{ii}\big)^2}\mathbf{I}_x^{-1}(\theta_0)_{\cdot i}^{\mathrm{T}}\mathbf{I}_x(\theta_0)\mathbf{I}_x^{-1}(\theta_0)_{\cdot i} \\ &= \frac{\mathbf{I}_x^{-1}(\theta_0)_{ii}}{\big(\mathbf{I}_x^{-1}(\theta_0)_{ii}\big)^2}d^2 \\ &= n\mathbf{I}_x^{-1}(\theta_0)_{ii}(\theta_i - \hat{\theta}_{ni})^2, \end{aligned}$$

which gives the result. □

Corollary C.11. *Assume Proposition C.9 and Proposition C.10 hold. Then, for all* $i = 1, \ldots, p$, *(i) The profile likelihood value associated with the randon variable $\tilde{\theta}_{ni,k_n}$ converges in distribution,*

$$R_{\mathrm{p}}(\tilde{\theta}_{ni,k_n}) \to_d \textit{Gamma}\left(\frac{1}{2}, 2[1 - \{\mathbf{I}_x^{-1}(\theta_0)_{ii}\}^{-1}\mathbf{I}_x^{-1}(\theta_0)_{i\cdot}^{\mathrm{T}}\{I + F(\theta_0)\}_{\cdot i}^{-1}]\right),$$

as $n \to \infty$;

(ii) the minimum profile loglikelihood associated with the chain $\tilde{\theta}_{ni,k_n:t}$ converges in distribution,

$$t^2 m_{nit} \to_d \textit{Weibull}\left(\frac{\pi}{2}[1 - \{\mathbf{I}_x^{-1}(\theta_0)_{ii}\}^{-1}\mathbf{I}_x^{-1}(\theta_0)_{i\cdot}^{\mathrm{T}}\{I + F(\theta_0)\}_{\cdot i}^{-1}], \frac{1}{2}\right),$$

as $n, t \to \infty$.

Proof. For, (i), we use Proposition C.10 to write,

$$R_{\mathrm{p}}(\tilde{\theta}_{ni,k_n}) = n\mathbf{I}_x^{-1}(\theta_0)_{ii}(\tilde{\theta}_{ni,k_n} - \hat{\theta}_{ni})^2 + o_p(1).$$

By Proposition C.9, we have,

$$\sqrt{n}(\mathbf{I}_x^{-1}(\theta_0)_{ii})^{-1/2}(\tilde{\theta}_{ni,k_n} - \hat{\theta}_{ni}) \to_d \mathrm{N}(0, (\mathbf{I}_x^{-1}(\theta_0)_{ii})^{-1}(\mathbf{I}_x^{-1}(\theta_0)_{ii} - \mathbf{I}_x^{-1}(\theta_0)_{i\cdot}^{\mathrm{T}}(I + F(\theta_0))_{\cdot i}^{-1})). \tag{C.8}$$

the result follows by squaring the sequence of random variables in Equation C.8 and applying the continuous mapping theorem.

The proof for (ii) then follows by applying the proof of Theorem C.5 to the random variable $m_{nit} = \min_{k=k_n,\ldots,k_n+t-1} R_{\mathrm{p}}(\tilde{\theta}_{ni,k})$. □

Theorem C.12. *For the estimator $\tilde{\theta}_{ni,\mathrm{p}\,t}$, we have,*

$$tn^{1/2}(\mathbf{I}_x^{-1}(\theta_0)_{ii})^{-1/2}(\tilde{\theta}_{ni,\mathrm{p}\,t} - \hat{\theta}_n) \to_d \textit{Laplace}\left(0, \frac{(2\pi)^{1/2}}{2}[1 - \{\mathbf{I}_x^{-1}(\theta_0)_{ii}\}^{-1}\mathbf{I}_x^{-1}(\theta_0)_{i\cdot}^{\mathrm{T}}\{I + F(\theta_0)\}_{\cdot i}^{-1}]^{1/2}\right),$$

as $n, t \to \infty$.

Proof. Starting with the identity $\min R_{\mathrm{p}}(\tilde{\theta}_{ni,k_n}) = n(\mathbf{I}_x^{-1}(\theta_0)_{ii})^{-1}(\tilde{\theta}_{ni,\mathrm{p}\,t} - \hat{\theta}_n)^2 + o_p(1)$, we find after a few algebraic manipulations that,

$$tn^{1/2}(\mathbf{I}_x^{-1}(\theta_0)_{ii})^{-1/2}(\tilde{\theta}_{ni,\mathrm{p}\,t} - \hat{\theta}_n) \overset{d}{=} s_{nt}(t^2 m_{nit})^{1/2} + o_p(1). \tag{C.9}$$

Now apply the proof of Theorem C.8 to the right-hand side of Equation C.9 to get the result. □