Hyperbolic Distributions and Transformations for Clustering Incomplete Data with Extensions to Matrix Variate Normality

HYPERBOLIC DISTRIBUTIONS AND TRANSFORMATIONS FOR CLUSTERING INCOMPLETE DATA WITH EXTENSIONS TO MATRIX VARIATE NORMALITY

BY

NIKOLA POČUČA, M.Sc. (Mathematics and Statistics)

McMaster University, Hamilton, Canada

A THESIS

SUBMITTED TO THE DEPARTMENT OF MATHEMATICS & STATISTICS

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE:

DOCTORATE OF PHILOSOPHY - STATISTICS

Doctor of Philosophy - Statistics (2023) McMaster University

(Mathematics & Statistics) Hamilton, Ontario, Canada

TITLE:	Hyperbolic Distributions and Transformations for Clustering Incomplete Data with Extensions to Matrix Variate Normality
AUTHOR:	Nikola Počuča M.Sc. (Mathematics and Statistics) McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Paul D. McNicholas
NUMBER OF PAGES:	xvii, 138

Za moja ljubav Anna

ABSTRACT

Under realistic scenarios, data are often incomplete, asymmetric, or of high-dimensionality. More intricate data structures often render standard approaches infeasible due to methodological or computational limitations. This monograph consists of four contributions each solving a specific problem within model-based clustering. An R package is developed consisting of a three-phase imputation method for both elliptical and hyperbolic parsimonious models. A novel stochastic technique is employed to speed up computations for hyperbolic distributions demonstrating superior performance overall. A hyperbolic transformation model is conceived for clustering asymmetrical data within a heterogeneous context. Finally, for high-dimensionality, a framework is developed for assessing matrix variate normality within three-way datasets. All things considered, this work constitutes a powerful set of tools to deal with the ever-growing complexity of big data.

ACKNOWLEDGEMENTS

Foremost, I would like to thank my supervisor, Dr. Paul D. McNicholas. For over the last 4 years, Paul has guided me into becoming an independent researcher and created an environment where I could succeed. Even through the COVID epidemic, I have been able to work with many researchers both here at McMaster and worldwide. In addition, I would like to show my appreciation to Dr. Michael P.B. Gallaugher who was my fellow co-author, and senior within the McNicholas research group.

I would like to acknowledge the funds provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grant, E.W.R. Steacie Memorial Fellowship and Canada Research Chairs program, the Ontario Graduate Scholarship, the Department of Mathematics and Statistics, and, McMaster University.

Finally, I would like to thank Dr. Nedialko Nedialakov, and Dr. Franek Frantisek, for serving on my commitee and guidance in the early years of my doctorate on computational workloads. Thank you for making the defence process worthwhile.

PUBLICATIONS

The following articles, at the time of this writing, are to be published or, are in preparation.

- Počuča, N., Gallaugher, M.P., Clark, K.M. and McNicholas, P.D., "Visual Assessment of Matrix-Variate Normality", *Australian and New Zealand Journal of Statistics* (Accepted).
- Počuča, N., Browne, R. P., and McNicholas, P.D., "Mixture 2.1: Model-Based Clustering and Classification, With or Without Missing Data", *The R Journal*, (In preparation).
- Počuča, N., Gallaugher, M.P., and McNicholas, P.D., "Efficient Optimization of Normal Variance-Mean Mixtures", (In preparation).
- Počuča, N., Gallaugher, M.P., and McNicholas, P.D., "The Missing and the Asymmetric: Clustering with Finite Mixtures of S_{U} Johnson Distributions", (In preparation).

GLOSSARY

3PEM Three phase expectation-maximization. 27, 40, 41, 46

ARI Adjusted Rand index. 10, 40, 41, 48

BIC Bayesian information criterion. 10, 24, 34, 39, 78, 79

CMI Conditional mean imputation. xvi, xvii, 25, 26, 72–74, 82–86

DA Deterministic annealing. 23, 24

DD Distance-distance. xvii, 92–100

EM Expectation-maximization. xv, 9, 21–23, 25–27, 32, 36, 37, 47, 49

GHD Generalized hyperbolic distribution. 15, 50, 53, 55, 59, 62

GHPCM Generalized hyperbolic parsimonious model. 34

GIG Generalized inverse Gaussian. 32, 48, 51, 53, 54

GLOC General location model. 25, 46

GPCM Gaussian parsimonious clustering model. xii, 6–9, 20–24

MAR Missing at random. 84, 87

MCAR Missing completely at random. 72, 82–84, 87

MLE Maximum likelihood estimation. 68, 91

MNIST Modified national institute of standards and technology. xvii, 98, 99

MSD Mahalanobis squared distance. 13, 89, 90, 92, 97, 100

MSE Mean squared error. xvii, 40, 41, 85, 86

NVMM Normal variance mean mixture. 14, 15, 17, 31–33, 47, 49, 50, 55, 59, 62, 63

OCLUST Outlier clustering. 102

QQ Quantile-quantile. 12, 89

RDI Random draw imputation. xvi, xvii, 73, 74, 83–86

SAL Shifted asymmetric Laplace. 15

SEM Stochastic expectation-maximization. xii, xv, 36, 37, 48, 49, 51, 53, 55, 57–61, 63

STPCM Skew-t parsimonious model. 34

VGPCM Variance gamma parsimonious model. 34

LIST OF SYMBOLS

$\mathcal{N}$ Gaussian distribution. 11, 13, 17, 18, 34, 51, 67, 68, 74, 89

∂ Partial derivative with respect to a parameter, Leibniz notation. 16, 52

p Dimension of a multivariate vector. 5–7, 9, 12, 13, 15, 16, 18, 19, 25, 26, 31, 32, 35, 49–51, 53–55, 67–69, 71, 73, 95–97

$\sim$ Distributed by. 11, 13, 16, 17, 32, 34, 37, 50, 51, 55, 64, 65, 68, 74, 89, 90, 95

S_{U} System unbounded, one of three Johnson's transformations. xii, xiii, xvi, 17–19, 67–69, 72–76, 78–82, 86, 87

Unif Uniform distribution. 64, 65

vec Vectorize a matrix according to column-major order. 11, 89, 92, 94, 95

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In the chaos of modern data, there is often clarity when one uses the correct tool to make sense of its intricacies. As Peter F. Drucker elucidates,

"You can't manage what you don't measure," (Drucker, 2006),

Too often this idea is ignored when making decisions or dealing with highly complex problems. As a consequence, data-driven decision making is becoming evermore popular leading to the rise of what many contemporaries call *data science* (McAfee *et al.*, 2012). As with every popular movement, there are contrarians who seek to emphasize the drawbacks. In the majority of cases, every methodological innovation often increases what is known as *technical debt* (Sculley *et al.*, 2014). This refers to the idea that as we develop novel methods, we are bound to incur massive ongoing maintenance costs at a system level. The accrescent complexity of such technologies must be countered with sound methods that eliminate such issues. Furthermore,

these methods must be motivated, and grounded within realism.

Under realistic scenarios, data is often heterogeneous. A population of individuals may contain a mixture of sub-populations each with their own distinct identity or culture. This heterogeneity often violates most statistical assumptions and fundamentally limits analysis. Such limitations of homogeneous statistical assumptions gives rise to intrinsic biases for data-driven decision making. In addition, multivariate datasets often contain incomplete entries for some observations. The standard practice for these incomplete observations is to disregard them in favour of fully-complete datasets. If we consider both the presence of heterogeneity and incomplete entries, the result is a highly complex dataset plagued with issues of unintelligibility. This monograph consists of methodologies aimed to alleviate such issues. In addition, an extension to assessing the normality of higher-order data is also developed.

The work is organized as follows. Chapter 2 provides a background on all methodological developments for proceeding chapters. A historical summary and definitions of model-based clustering is discussed. A series of hyperbolic distributions are defined with some additional extensions to hyperbolic transformation systems. Finally, we conclude with the concept of matrix-variate distributions and their relationship to their multivariate counterparts. Chapter 3 revolves around the development and use of a software package for multivariate clustering and classification. The software combines previously developed methods and implements a novel approach for clustering multivariate datasets with, or without missing data. Be it symmetric or skewed, the software allows a user to handle a variety of multivariate datasets. Chapter 4 pertains to a stochastic alternative for estimating mixtures of hyperbolic distributions.

A series of computational investigations are performed and demonstrate the effectiveness of a stochastic algorithm against the standard approach. Chapter 5 is concerned with modelling asymmetric multivariate datasets with hyperbolic transformations. Often overlooked, these hyperbolic systems outperform more popular transformation methods. In addition, imputation methods for missing data are also developed and investigated for such models. Chapter 6 branches off from previous contributions and introduces a novel approach for evaluating the normality of three-way data. In tandem with existing hypothesis tests, a new visual method is developed and combined to form a powerful framework for assessing matrix-variate normality. Finally, Chapter 7 ends with future directions of research and possible extensions of developed methodologies.

CHAPTER 2

BACKGROUND

The following sections outline the introduction of statistical methodology necessary to understand the proceeding contributions. Model-based clustering is primarily discussed with a focus on finite mixture models, parsimony, and asymmetric approaches. In addition, matrix-variate normality and its structure is elucidated as well as frameworks for assessing multivariate normality are summarized.

2.1 Model-based Clustering

The first notion of model-based clustering can be traced back to Tiedeman (1955) with a standard Gaussian example (see section 9.1 of McNicholas, 2016a) . Let G be the number of distinct groups within a general population. For each gth group, let observations be drawn from some corresponding Gaussian distribution. Upon

removing the group identities to which each observation belongs to; the result is a mixture of data with unknown density. Here, Tiedeman defines clustering as the reconstruction of the original G densities and their types. However, Wolfe (1965) formally considers a cluster through a measure of similarity. As McNicholas (2016a) elucidates, similarity is often difficult to define as a coherent, quantifiable measure. If one considers an appropriate finite mixture model which has the flexibility, and parametrization that is necessary to fit the data; then a cluster can be defined as a uni-modal component of said mixture model. For some heterogeneous dataset, the type of selection for this uni-modal component bears the full structure and interpretation of said cluster. For the Gaussian case as in Tiedeman (1955), each cluster is defined as a symmetric curve with identifiably distinct parameters (mean and variance). However, by relaxing the Gaussian assumption, one can conceive of a variety of cluster shapes and structures. Through this perspective, the framework of model-based clustering can capture any heterogeneous representation of data where the only limitation is the interpretability of the type of uni-modal selection.

2.1.1 Gaussian Finite Mixture Model

One of the most popular methods for model-based clustering is the finite mixture model. Consider a random variable $\boldsymbol{\mathcal{X}}$ which is characterized by a G component finite mixture model. For a set of parameters $\boldsymbol{\Theta} = (\pi_1, \ldots, \pi_G, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_G)$, a p-dimensional random vector $\boldsymbol{\mathcal{X}}$ follows a finite mixture distribution for all $\boldsymbol{x} \subset \boldsymbol{\mathcal{X}}$ if its density can be written as

$$f(\boldsymbol{x}; \boldsymbol{\Theta}) = \sum_{g=1}^{G} \pi_g f_g(\boldsymbol{x}; \boldsymbol{\theta}_g).$$

Here, $\pi_g > 0$ is the gth mixing proportion with $\sum_{g=1}^{G} \pi_g = 1$, G is the number of components, and f_g is the density of the gth component parametrized by $\boldsymbol{\theta}_g$. Usually, the density for each component is taken to be the same so that $f_g = f$ for all g. Suppose that $\boldsymbol{\mathcal{X}}$ follows a multivariate Gaussian distribution, then $\boldsymbol{\theta}_g = \{\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}$, and

$$f(\boldsymbol{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}_g|^{1/2}} \exp\left\{ -\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_g)^\top \boldsymbol{\Sigma}_g^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_g) \right\}.$$

Here, the parameters $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ represent the mean and variance of the gth component (cluster). In the context of clustering, $\boldsymbol{\mu}_g \neq \boldsymbol{\mu}_{g'}, \ \forall g \neq g' \in \{1, \ldots, G\}$. This constraint implies that all components must have different means in order for the model to be identifiable. However, this constraint is not necessary for $\boldsymbol{\Sigma}_g$, as you can take specific parameters to be the same across components. This method of imposing constraints for $\boldsymbol{\Sigma}_g$ leads to the family of 14 Gaussian parsimonious clustering models (GPCMs, Banfield and Raftery, 1993).

2.1.2 Gaussian Parsimonious Clustering Models

Referring to the idea of constraining covariances across components, the concept of parsimony was first introduced by Banfield and Raftery (1993) for a Gaussian finite mixture model. Subsequently, the cornerstone work by Celeux and Govaert (1995) gave estimations for the majority, but not all parsimonious models. Finally, Browne and McNicholas (2014) developed estimation procedures for the remaining models completing the work spanning several decades. By decomposing the covariance matrix $\boldsymbol{\Sigma}_g$, and then further imposing constraints on the resulting elements, the family of 14

GPCMs is defined as follows. Let the eigenvalue decomposition of $\boldsymbol{\Sigma}_g$ be written as

$$\boldsymbol{\Sigma}_g = \lambda_g \boldsymbol{D}_g \boldsymbol{A}_g \boldsymbol{D}_g^\top, \tag{2.1}$$

where $\lambda_g = \det(\boldsymbol{\Sigma}_g)^{1/p}$, det() is the matrix determinant operator, $\boldsymbol{D}_g$ is the matrix of eigenvectors, and $\boldsymbol{A}_g$ the diagonal matrix of normalized eigenvalues. The parameter λ_g is usually referred as the volume of component g, $\boldsymbol{D}_g$ is the orientation, and $\boldsymbol{A}_g$ its shape. To yield the family of 14 GPCMs, impose constraints on (2.1) across components as follows. The spherical family of GPCMs can be derived by allowing $\boldsymbol{\Sigma}_g = \lambda \boldsymbol{I} \ \forall g$. Here, $\boldsymbol{I}$ is a $p \times p$ dimensional identity matrix. Referred to as the EII model, this structure implies that all components share the same volume and spherical shape. The second model of this family VII imposes constraints along direction and orientation, such that only volume can vary between clusters. As a result, the covariance is of the form $\boldsymbol{\Sigma}_g = \lambda_g \boldsymbol{I} \ \forall g$. This model structure implies that all clusters share the same spherical shape but varying volume size. Moving on to the second family of GPCMs, the diagonal family relaxes the constraint on shape, but imposes equal orientation. Specifically, $\boldsymbol{D}_g = \boldsymbol{I} \ \forall g$. Beginning with the EEI model, Let $\boldsymbol{\Sigma}_g = \lambda \boldsymbol{A} \ \forall g$. This imposes constraints that volume and shape be equal for all clusters. The VEI model imposes $\boldsymbol{\Sigma}_g = \lambda_g \boldsymbol{A}$. Here, volume is allowed to vary while shape is kept fixed. The EVI model implies $\boldsymbol{\Sigma}_g = \lambda \boldsymbol{A}_g$, where volume is fixed across clusters, but shape is allowed to vary. Finally the VVI model implies $\boldsymbol{\Sigma}_g = \lambda_g \boldsymbol{A}_g$, where both shape and volume is relaxed. To conclude, the diagonal family of GPCMs relaxes the constraint on shape and volume, resulting in parsimony along each of the components of (2.1). The EEE model implies $\boldsymbol{\Sigma}_g = \lambda \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^\top \ \forall g$, imposing equal volume, shape, and orientation across clusters. The VEE model implies $\boldsymbol{\Sigma}_g = \lambda_g \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^\top$, allowing volume

to vary. The EVE model implies $\boldsymbol{\Sigma}_g = \lambda \boldsymbol{D}\boldsymbol{A}_g\boldsymbol{D}^\top$, allowing only shape to vary. The EEV model implies $\boldsymbol{\Sigma}_g = \lambda \boldsymbol{D}_g\boldsymbol{A}\boldsymbol{D}_g^\top$, allowing orientation to vary. The VVE model implies $\boldsymbol{\Sigma}_g = \lambda_g \boldsymbol{D}\boldsymbol{A}_g\boldsymbol{D}^\top$, only constraining orientation. The EVV model implies $\boldsymbol{\Sigma}_g = \lambda \boldsymbol{D}_g\boldsymbol{A}_g\boldsymbol{D}_g^\top$, constraining only volume. The VEV model constrains only shape implying $\boldsymbol{\Sigma}_g = \lambda_g \boldsymbol{D}_g\boldsymbol{A}\boldsymbol{D}_g^\top$. Finally, the VVV model relaxes all constraints and has the form of (2.1). For convenience, Table 2.1 summarizes the nomenclature and characteristics of each model. Within these 14 families, the most familiar is that of VII and VVV. According to assumption (1) of Celeux and Govaert (1995), under certain conditions, the VII model is analogous to performing K-means clustering (MacQueen *et al.*, 1967) where K is the number of groups. In addition, VVV is the standard assumption that most Gaussian mixture models impose with no restrictions. One of the more interesting results of parsimony is the interpretation of cluster shapes. Figure 2 of Murphy and Murphy (2020) visualizes the resultant model fits for a particular selection of constraints. For example, the VII model clusters are spheres of differing volumes. If VII is selected as the top performing model, then the data exhibits behaviour where each variate determines cluster shape equally. Each of the 14 parsimonious models contain similar interpretations which are useful for explaining phenomena within the context of an application. In summary, imposing parsimony within model-based clustering and classification is continuing area of research. McNicholas (2016b) provides a concise review of this area; for a more detailed view, see McNicholas (2016a). Extension to the GPCMs include the parsimonious Gaussian mixture models (McNicholas and Murphy, 2008). Further parsimony is introduced by considering a latent factor-based approach (Ghahramani *et al.*, 1996). Other extensions include extending parsimony to skewed distributions to model asymmetric

components (Vrbik and McNicholas, 2014; Tortora *et al.*, 2016).

Table 2.1: Nomenclature, covariance structure, and number of free covariance parameters for each member of the GPCM family.

Model	Volume	Shape	Orientation	$\boldsymbol{\Sigma}_g$	Free Covariance Parameters
EII	Equal	Spherical	-	$\lambda \boldsymbol{I}$	1
VII	Variable	Spherical	-	$\lambda_g \boldsymbol{I}$	G
EEI	Equal	Equal	Axis-Aligned	$\lambda \boldsymbol{A}$	p
VEI	Variable	Equal	Axis-Aligned	$\lambda_g \boldsymbol{A}$	$p+G-1$
EVI	Equal	Variable	Axis-Aligned	$\lambda \boldsymbol{A}_g$	$pG-G+1$
VVI	Variable	Variable	Axis-Aligned	$\lambda_g \boldsymbol{A}_g$	pG
EEE	Equal	Equal	Equal	$\lambda \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^\top$	$p(p+1)/2$
EVE	Equal	Variable	Equal	$\lambda \boldsymbol{D}\boldsymbol{A}_g\boldsymbol{D}^\top$	$p(p+1)/2+(G-1)(p-1)$
VEE	Variable	Equal	Equal	$\lambda_g \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^\top$	$p(p+1)/2+(G-1)$
EEV	Equal	Equal	Variable	$\lambda \boldsymbol{D}_g\boldsymbol{A}\boldsymbol{D}_g^\top$	$Gp(p+1)/2-(G-1)p$
VVE	Variable	Variable	Equal	$\lambda_g \boldsymbol{D}\boldsymbol{A}_g\boldsymbol{D}^\top$	$p(p+1)/2+(G-1)p$
EVV	Equal	Variable	Variable	$\lambda \boldsymbol{D}_g\boldsymbol{A}_g\boldsymbol{D}_g^\top$	$Gp(p+1)/2-(G-1)$
VEV	Variable	Equal	Variable	$\lambda_g \boldsymbol{D}_g\boldsymbol{A}\boldsymbol{D}_g^\top$	$Gp(p+1)/2-(G-1)(p-1)$
VVV	Variable	Variable	Variable	$\lambda_g \boldsymbol{D}_g\boldsymbol{A}_g\boldsymbol{D}_g^\top$	$Gp(p+1)/2$

2.2 Model Estimation, Performance, and Convergence

Model estimation is based on the expectation-maximization algorithm (EM, Dempster *et al.*, 1977). The EM algorithm is an iterative approach for finding the maximum likelihood estimates when data is missing, and is a special case of the minorize-maximization variety (Lange, 2016; McLachlan and Krishnan, 2007). Let l^t denote a model's objective function at iteration t one aims to maximize using the EM algorithm. Model convergence for said EM algorithm is based on the Aitken acceleration criterion (Aitken, 1926) defined as

$$a^{(t)} = \frac{l^{(t+1)} - l^{(t)}}{l^{(t)} - l^{(t-1)}}, \tag{2.2}$$

Let

$$l_{\infty}^{(t+1)} = l^{(t)} + \frac{l^{(t+1)} - l^{(t)}}{1 - a^{(t)}}$$

be the observed estimate after many iterations at $t+1$ (Section 2.2.5 of McNicholas, 2016a). Termination of the algorithm occurs when $l_{\infty}^{(t+1)} - l^{(t)} \in (0, \varepsilon)$ for some pre-specified $\varepsilon > 0$. Model selection is based on the Bayesian information criterion (BIC; Schwarz *et al.*, 1978). Let ρ be the number of free parameters used. The BIC is then formulated as BIC $= 2l(\boldsymbol{\theta}) - \rho \log n$. To measure the performance on classification, the adjusted Rand index (ARI, Hubert and Arabie, 1985) is considered. The ARI is an adjusted for chance performance of the Rand index (Rand, 1971) for measuring classification.

2.3 Matrix Variate Normality

Two-way data can be regarded as the observation of N vectors, whereas three-way data can be considered the observation of N matrices. Common examples of three-way data include greyscale images and multivariate longitudinal data. Multivariate distributions have been successfully used in the analysis of two-way data, and matrix variate distributions are gaining popularity for the analysis of three-way data (e.g., Viroli, 2011; Anderlucci and Viroli, 2015; Gallaugher and McNicholas, 2018, 2020; Tomarchio *et al.*, 2021, 2022; Gallaugher *et al.*, 2022). Similar to the multivariate case, the most mathematically tractable matrix-variate distribution is the matrix-variate normal distribution. An $r \times c$ random matrix $\mathscr{X}$ comes from a matrix-variate

normal distribution if its density is of the form

$$\phi_{r\times c}(\boldsymbol{X} \mid \boldsymbol{M}, \boldsymbol{V}, \boldsymbol{U}) = \frac{1}{(2\pi)^{\frac{rc}{2}}|\boldsymbol{V}|^{\frac{r}{2}}|\boldsymbol{U}|^{\frac{c}{2}}} \exp\left\{-\frac{1}{2}\text{tr}\big(\boldsymbol{V}^{-1}(\boldsymbol{X}-\boldsymbol{M})^{\top}\boldsymbol{U}^{-1}(\boldsymbol{X}-\boldsymbol{M})\big)\right\}, \tag{2.3}$$

where $\boldsymbol{M}$ is the $r \times c$ mean matrix, $\boldsymbol{U}$ is the $r \times r$ row covariance matrix, and $\boldsymbol{V}$ is the $c \times c$ column covariance matrix. Note that the matrix-variate normal distribution is related to the multivariate normal distribution via the equivalence

$$\mathscr{X} \sim \mathcal{N}_{r\times c}(\boldsymbol{M}, \boldsymbol{V}, \boldsymbol{U}) \iff \text{vec}(\mathscr{X}) \sim \mathcal{N}_{rc}(\text{vec}(\boldsymbol{M}), \boldsymbol{V} \otimes \boldsymbol{U}) \tag{2.4}$$

(Gupta and Nagar, 1999), where $\otimes$ denotes the Kronecker product and $\text{vec}(\cdot)$ is the vectorization operator. Note that there is an identifiability issue with regard to the parameters $\boldsymbol{U}$ and $\boldsymbol{V}$, i.e., if k is a strictly positive constant, then

$$\frac{1}{k}\boldsymbol{V} \otimes k\boldsymbol{U} = \boldsymbol{V} \otimes \boldsymbol{U}$$

and so replacing $\boldsymbol{U}$ and $\boldsymbol{V}$ by $(1/k)\boldsymbol{U}$ and $k\boldsymbol{V}$ respectively, leaves (2.3) unchanged. Various solutions have been proposed to resolve this issue, including setting $\text{tr}(\boldsymbol{U}) = r$ or $\boldsymbol{U}_{11} = 1$ (see Anderlucci and Viroli, 2015; Gallaugher and McNicholas, 2018).

In summary, the matrix-variate normal distribution is defined through a vectorization of a multivariate normal. Note that the covariance matrices of row and column are non-unique as they are defined through a Kronecker product (Dutilleul, 1999). As a result, both distributions in (2.4) are parametrized by the product and not individual co-variance matrices (Gupta and Nagar, 1999). There are benefits for using a matrix-variate representation. The primary being that there is a considerable increase

in speed for estimating parameters within high dimensional settings. As a result, one should consider assessing whether matrix-variate normality is viable.

2.3.1 Assessing Multivariate Normality

The equivalence between the multivariate and matrix-variate normal distributions in (2.4) implies that if multivariate normality of the vectorized data does not hold, then the original matrices cannot be matrix-variate normal. Therefore, multivariate normality must first be established before considering matrix-variate normality. There are many approaches for testing multivariate normality, several of which can be found in the R package `MVN` (Korkmaz *et al.*, 2014). Most notably, tests proposed by Mardia (1970), Royston (1983), and Henze and Zirkler (1990) are found within the package. In terms of visual methods for multivariate normality, the multivariate generalization of the QQ plot (Easton and McCulloch, 1990) is perhaps the most popular. Although frequently used, this approach was further extended by the work of Holgersson (2006), through an introduction of what is referred to as a correlation plot. To conclude, comparisons of such tests can be found in Horswell and Looney (1992) and Alpu and Yuksek (2016), where Royston (1983) emerged as the most favoured.

2.3.2 Mahalanobis Squared Distance

The Mahalanobis distance is a well-established quantity in the literature (Mahalanobis, 1936). Hardin and Rocke (2005) illustrate its application in multivariate outlier detection and goodness-of-fit. Consider N p-dimensional vectors $\mathbf{y}_1, \ldots, \mathbf{y}_N$ such that each $\mathbf{y}_i$ is a realization of a multivariate random variable $\boldsymbol{\mathcal{Y}} \sim \mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The MSD for a given $\mathbf{y}_i$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ is

$$\mathcal{D}(\mathbf{y}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\mathbf{y}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}) . \tag{2.5}$$

It is well known (see Mardia *et al.*, 1979) that

$$\mathcal{D}(\mathbf{y}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \chi^2_p, \tag{2.6}$$

where χ^2_p is chi-square distributed with p degrees of freedom. Now, consider the estimates

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{y}_i \qquad \text{and} \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{y}_i - \hat{\boldsymbol{\mu}})^\top (\mathbf{y}_i - \hat{\boldsymbol{\mu}}).$$

Then, from Gnanadesikan and Kettenring (1972),

$$\frac{N}{(N-1)^2} \mathcal{D}(\mathbf{y}_i, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) \sim \text{Beta}\left(\frac{p}{2}, \frac{N-p-1}{2}\right). \tag{2.7}$$

If one considers the distribution for all MSDs within a given sample in (2.7), then a goodness-of-fit test naturally presents itself along with outlier detection and other statistical techniques for multivariate settings.

2.4 Hyperbolic Distributions and Transformations

A hyperbolic distribution can be characterized geometrically if the logarithm of its density function forms a hyperbola. Given the necessary parametrization, the respective density function may decrease much slower than its Gaussian counterpart in the limit. As a result, such distributions are useful for modelling behaviour with

extreme values or, asymmetric components within a model-based clustering context. An alternative to using hyperbolic distributions is that of transformations from normality. Given an appropriate system of transformation, one could capture the desired flexibility necessary to model asymmetric or skewed data. One caveat is that most transformation-based approaches always consider departure from normality as the starting point. However, this assumption can be relaxed as in Finak *et al.* (2009) where the Student-t distribution is considered. Both methods are often viable when modelling skewed data. Gallaugher *et al.* (2020) compares and contrasts both approaches within a model-based context for a variety of applications. Within this thesis, we consider both hyperbolic distributions and transformations for capturing extreme or asymmetric phenomena.

2.4.1 Normal Variance-Mean Distributions

The most distinguished of hyperbolic distributions is the normal variance-mean mixture family (NVMMs). NVMMs have grown in popularity for their robust ability to model skewed or asymmetric statistical problems. Such cases include, but are not limited to, finance (Luciano and Semeraro, 2010; Banihashemi, 2019), risk management (Kim and Kim, 2019), engineering (Snoussi and Idier, 2006), and hydrology (Ownuk *et al.*, 2021). Most notably characterized by their semi-heavy tailed property (Borak *et al.*, 2011), NVMMs are considered to be robust in cases where there are few spurious outliers when compared to their heavy-tailed counterparts; Weibull, Gumbel, etc. (Kotz and Nadarajah, 2015). Normal variance-mean mixtures are a set of distributions that arise from a combination of two random variables where one acts as a weighting function for the other. More formally, a p-dimensional random

variable $\mathcal{X}$ is said to be a normal variance-mean mixture if the density is formulated as

$$f(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\alpha},\boldsymbol{\theta}) = \int_0^\infty \phi_p(\boldsymbol{x};\boldsymbol{\mu}+y\boldsymbol{\alpha},y\boldsymbol{\Sigma})h(y;\boldsymbol{\theta})\mathrm{d}y, \quad \boldsymbol{x}\in\mathbb{R}^p. \tag{2.8}$$

Here, ϕ_p is the density of a p-dimensional multivariate Gaussian with mean $\boldsymbol{\mu}+y\boldsymbol{\alpha}$, covariance matrix $y\boldsymbol{\Sigma}$, and $h(y;\boldsymbol{\theta})$ as the density function of a univariate random variable defined on $y>0$ (Barndorff-Nielsen *et al.*, 1982). The function $h(y;\boldsymbol{\theta})$ is a weighting function whereby its form characterizes the resultant normal variance-mean mixture distribution. The tail behavior of an NVMM density is considered to be 'semi-heavy' satisfying a tempered stable distribution with a small truncation parameter (Rosiński, 2007; Borak *et al.*, 2011), i.e., a lighter tail than that of an extreme value law (Kotz and Nadarajah, 2015), but heavier than a Gaussian. For example, if $h(y;\boldsymbol{\theta})$ is the density of an exponentially distributed random variable with rate parameter 1, the integral $f(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\alpha},\boldsymbol{\theta})$ yields the density of a shifted asymmetric Laplace distribution (SAL, Kotz *et al.*, 2001). Furthermore, if $h(y;\boldsymbol{\theta})$ is taken to be a generalized inverse Gaussian (GIG, Barndorff-Nielsen and Halgreen, 1977) distribution, the resultant normal variance-mean mixture is a generalized hyperbolic distribution (GHD, McNeil *et al.*, 2015). The GIG plays a pivotal role for inference of normal variance-mean mixtures (Barndorff-Nielsen and Halgreen, 1977). The density of a GIG random variable Y is written as

$$d(y;a,b,\lambda) = \frac{(a/b)^{\frac{\lambda}{2}}y^{\lambda-1}}{2\mathcal{K}_\lambda(\sqrt{ab})}\exp\left\{-\frac{ay+b/y}{2}\right\}, \quad y>0, \tag{2.9}$$

where $a, b \in \mathbb{R}^+$, $\lambda \in \mathbb{R}$, and $\mathcal{K}_\lambda$ is the modified Bessel function of the third kind with index λ. This distribution has attractive properties where expectations follow closed forms and can be written as follows

$$\mathbb{E}[Y] = \sqrt{\frac{b}{a}} \frac{\mathcal{K}_{\lambda+1}(\sqrt{ab})}{\mathcal{K}_\lambda(\sqrt{ab})}, \tag{2.10}$$

$$\mathbb{E}[1/Y] = \sqrt{\frac{a}{b}} \frac{\mathcal{K}_{\lambda+1}(\sqrt{ab})}{\mathcal{K}_\lambda(\sqrt{ab})} - \frac{2\lambda}{b}, \tag{2.11}$$

$$\mathbb{E}[\log Y] = \frac{1}{2} \log\left(\frac{b}{a}\right) + \frac{1}{\mathcal{K}_\lambda(\sqrt{ab})} \frac{\partial}{\partial \lambda} \mathcal{K}_\lambda(\sqrt{ab}). \tag{2.12}$$

Due to other parametrizations of the GIG distribution, it is beneficial to denote $Y \sim \text{GIG}(a, b, \lambda)$ as the definition given in equation (2.9). However, if we allow $\omega = \sqrt{ab}$, and $\eta = \sqrt{\frac{b}{a}}$ then the density function becomes

$$h(y; \omega, \eta, \lambda) = \frac{(y/\eta)^{\lambda-1}}{2\eta \mathcal{K}_\lambda(\omega)} \exp\left\{-\frac{\omega}{2}\left(\frac{y}{\eta} + \frac{\eta}{y}\right)\right\}, \quad y > 0. \tag{2.13}$$

Here $\eta > 0$ is the scale parameter, $\omega > 0$ is the concentration parameter, and λ is an index parameter. To avoid confusion, we denote $Y \sim \text{I}(\omega, \eta, \lambda)$ to be a GIG distribution with the parametrization in (2.13).

It is often easier to interpret NNVMs through its stochastic representation. A p-dimensional random variable $\boldsymbol{\mathcal{X}}$ is said to be considered a normal variance-mean mixture if its stochastic representation satisfies

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mu} + Y\boldsymbol{\alpha} + \sqrt{Y}\boldsymbol{\mathcal{U}}, \tag{2.14}$$

with location vector $\boldsymbol{\mu}$, skewness vector $\boldsymbol{\alpha}$, and $\boldsymbol{\mathcal{U}} \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$. Y is considered a latent

variable, whereby its distribution characterizes the resultant member distribution of the NVMM family. Specifically, one can simulate realizations from NVMMs by first drawing values from the latent variables Y and $\boldsymbol{\mathcal{U}}$, then applying (2.14). In summary, NVMMs have an interpretative departure from normality as an infinite mixture of variance-mean weighted Gaussians.

2.4.2 S_{U} Johnson Distribution

The first notion of transformations to normality was put forth in Edgeworth (1898), concerning functions of a polynomial order. Though simple in concept, this seminal work spawned an abounding amount of research regarding methods of translation to Gaussian distributions. Most notably, the work of Box and Cox (1964) has become common practice amongst statisticians for such asymmetric scenarios (Sakia, 1992). Prominently referred to as the Box-Cox or power transform, the method allows data of extreme nature to be modelled with a mathematically tractable Gaussian distribution.

A similar, but lesser known alternative to such transformations considers the use of hyperbolic functions to capture the asymmetry. Johnson (1949b) introduces three systems of translation to account for a more realistic representation of distributions encountered in practice. Focus is placed on using functions which accurately represent the departure from normality referred to as skewness. Of the three systems denoted as S_{L}, S_{B}, and S_{U}; the latter is often-times most appropriate as it operates on the unbounded domain of $\mathbb{R}$. Despite being particularly effective in modelling extreme values (Burbidge *et al.*, 1988), there is a general paucity among literature regarding S_{U} transformations (Tsai, 2011). Nevertheless, several contributions extend S_{U} transformations methodologically for a variety of statistical problems (Jones and Pewsey,

2009; Stanfield *et al.*, 1996).

A p-dimensional random vector $\boldsymbol{\mathcal{X}}$ is said to emanate from a multivariate Gaussian with location $\boldsymbol{\mu}$, and covariance $\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^\top$, if its stochastic representation can be written as

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{\mathcal{Z}}.$$

Here, let $\boldsymbol{\mathcal{Z}}$ be a p-dimensional random vector distributed by a multivariate Gaussian $\mathcal{N}_p(\boldsymbol{0}, \boldsymbol{I})$. We now define a multivariate transformation function $\boldsymbol{\varphi}$ as in Nelson and Yamnitsky (1998) such that $\boldsymbol{\varphi}(\boldsymbol{x}) := (\varphi(x_1), \dots \varphi(x_p))$, where $\varphi(x_j) := \sinh(x_j)$ for some variate x_j. Now let $\boldsymbol{\omega} = (\omega_1, \dots, \omega_p)$ be the shift vector and let $\boldsymbol{\Lambda}$ be a diagonal matrix consisting of scale parameters for each dimension as $\boldsymbol{\Lambda} := \text{diag}(\delta_1, \dots, \delta_p), \delta_j > 0$. Finally, $\boldsymbol{\mathcal{Y}}$ is said to be distributed according to a multivariate S_U Johnson distribution if

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\omega} + \boldsymbol{\Lambda}\boldsymbol{\varphi}(\boldsymbol{\mathcal{X}}). \tag{2.15}$$

The density for $\boldsymbol{\mathcal{Y}}$ can be derived using the Transformation Theorem of random variables (Gupta and Kapoor, 2020). First, let us rearrange equation (2.15) and define a function $\boldsymbol{h}$ where

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\varphi}^{-1}(\boldsymbol{\Lambda}^{-1}(\boldsymbol{\mathcal{Y}} - \boldsymbol{\omega})) =: \boldsymbol{h}(\boldsymbol{\mathcal{Y}}; \boldsymbol{\vartheta}).$$

For some realization $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$, the function $\boldsymbol{h}(\boldsymbol{y}; \boldsymbol{\vartheta}) = (h(y_1; \boldsymbol{\vartheta}_1), \dots h(y_p; \boldsymbol{\vartheta}_p))$ can be thought of as the component-wise inverse hyperbolic sine transform with shift ω_j and scale δ_j. Specifically, given some variate y_j, and $\boldsymbol{\vartheta}_j := (\omega_j, \delta_j)$, the function is written as

$$h(y_j; \omega_j, \delta_j) = \text{arcsinh}\left(\frac{y_j - \omega_j}{\delta_j}\right). \tag{2.16}$$

Recognize that by definition, h is a well defined map from $\mathbb{R} \to \mathbb{R}$ with no singularities as $\delta_j > 0$. Let $\boldsymbol{J_h}(\boldsymbol{y};\boldsymbol{\vartheta})$ be the Jacobian with respect to the function $\boldsymbol{h}$; by the Transformation Theorem, the density of $\boldsymbol{\mathcal{Y}}$ is given as

$$\begin{aligned} f_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{y};\boldsymbol{\theta}) &= f_{\boldsymbol{\mathcal{X}}}(\boldsymbol{h}(\boldsymbol{y};\boldsymbol{\vartheta});\boldsymbol{\mu},\boldsymbol{\Sigma})|\boldsymbol{J_h}(\boldsymbol{y};\boldsymbol{\vartheta})| \\ &= \frac{\exp\left\{-\frac{1}{2}\left(\boldsymbol{h}(\boldsymbol{y};\boldsymbol{\vartheta})-\boldsymbol{\mu}\right)^{\top}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{h}(\boldsymbol{y};\boldsymbol{\vartheta})-\boldsymbol{\mu}\right)\right\}}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}\prod_{j=1}^{p}\left(\delta_j\sqrt{\left(\frac{y_j-\omega_j}{\delta_j}\right)^2+1}\right)}, \quad \boldsymbol{y}\in\mathbb{R}^p. \end{aligned} \tag{2.17}$$

The basis for this multivariate generalization can be traced back to Johnson (1949a) for modelling bean measurements of breadth and length. Stanfield *et al.* (1996) further extends the model to account for p-dimensional random variates. Most approaches derived from the S_{U} distribution retain the same functional form of (2.15), but differ in the parametrization of (2.16). The work of Burbidge *et al.* (1988) proposes a single scale parameter to transform extreme values; this is further extended by MacKinnon and Magee (1990) by offering parameters for both shift and scale. In principle, the S_{U} distribution can be thought of as a Gaussian random variate which is then transformed according to (2.15). Herein, we refer to the translated domain of $\boldsymbol{\mathcal{Y}}$ as hyperbolic space, and the domain of $\boldsymbol{\mathcal{X}}$ as Gaussian space. Therefore, $\boldsymbol{\omega}$ and $\boldsymbol{\Lambda}$ can be interpreted as the shift and scale in hyperbolic space. Once transformed, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ retain their usual Gaussian interpretations as discussed in Section 3 of Johnson (1949a).

CHAPTER 3

MIXTURE 2.1: MODEL-BASED CLUSTERING AND CLASSIFICATION, WITH OR WITHOUT MISSING DATA

This chapter focuses on the development of a software package for performing parsimonious clustering under missing data scenarios. Consider the following development timeline of software related to GPCMs.

3.1 Software Developments of GPCMs

A number of implementations of the GPCMs family in R (R Core Team, 2021) has been developed over the years. Among the most popular, the package `mclust` has

undergone its fifth fundamental change since its inception over 20 years ago (Scrucca *et al.*, 2016). Earlier versions of `mclust` (Fraley *et al.*, 2005) implemented 10 models of the 14-member GPCMs family. However, following the introduction of `mixture` version 1.0 (Browne and McNicholas, 2013), `mclust` version 5 included the remaining four. Notably, the same algorithms used for the two additional models in `mixture` version 1, were also used in `mclust` — specifically, the minorization-maximization algorithms of Browne and McNicholas (2014). In addition, despite its popularity, the ability to handle missing data had not been implemented within `mclust` for over 13 years until its fourth release (Fraley *et al.*, 2012).

In keeping with the GPCM tradition, consider a brand new upgrade: `mixture` version 2.1. The new version contains a C++ (Stroustrup, 2018) implementation of the GPCMs family with the ability to handle both skewed and missing data. This chapter focuses on the developments and specifics involved in creating `mixture` version 2.1. Based on an object oriented approach, an extension library `RcppArmadillo` is used to implement the family of 14 GPCMs (Eddelbuettel and Sanderson, 2014) which differs greatly from previous `mixture` versions. The object oriented approach is a traditional programming paradigm for the most natural and pragmatic C++ implementations (Zhou, 1996). This approach allows one to embed the imputation of missing data directly within the EM algorithm. Through the use of multiple inheritance, existing models are easily extended with minimal changes to the code base. This package features the following new additions. First, a brand new three-phase algorithm is developed for clustering missing data scenarios. Second, an extension of the original `mixture` package to include three skewed finite mixture models: skew-t, generalized hyperbolic, and variance gamma (Lee and McLachlan, 2013; Browne and

McNicholas, 2015; McNicholas *et al.*, 2017). Finally, the tEIGEN family for a finite mixture of student-t distributions is also implemented (Andrews *et al.*, 2018). All models mentioned above have 14 parsimonious settings to choose from allowing one to reduce dimensionality of the parameter space. In addition to parsimonious models, the new `mixture` package features an option to use the stochastic variant of the EM algorithm. To demonstrate robustness, both real and simulated datasets are used to compare the current version of `mclust` against the newly improved `mixture` package.

Installation of the package is as follows,

```
# Install package.
install.packages("mixture", dependencies=TRUE)

# Load mixture.
library(mixture)
```

3.2 Estimation

Estimation of the GPCM family is based on the EM algorithm. Suppose we observe $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ realizations of $\boldsymbol{\mathcal{X}}$, then the log-likelihood of a GPCM model is given as

$$l(\boldsymbol{\Theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \log \left(\sum_{g=1}^{G} \tau_g f(\boldsymbol{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right). \tag{3.1}$$

However, since the component membership of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ is unknown, it is difficult to maximize (3.1). In literature, when applying EM, (3.1) is usually referred to as the observed log-likelihood. A common approach is to introduce a latent random variable Z_{ig} which denotes the membership of observation $\boldsymbol{x}_i$. $Z_{ig} = 1$ when observation $\boldsymbol{x}_i$

belongs to component g, and $Z_{ig} = 0$ otherwise. As a result, denote the complete data log-likelihood as

$$l_c(\boldsymbol{\Theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \log \left\{ \tau_g f(\boldsymbol{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right\}. \tag{3.2}$$

In general, the EM algorithm can be broken down into two steps. The E-step calculates the expected value of missing data conditional on the observed, while the M-step maximizes (3.2) based on results from the E-step. The two steps alternate until convergence is established. Let $\boldsymbol{\Theta}^{(t)}$ be the parameters estimated from the current iteration t. The E-step is then given as

$$\hat{z}_{ig}^{(t)} = \mathbb{E}\left[Z_{ig} | \boldsymbol{x}_i, \boldsymbol{\Theta}^{(t)} \right] = \frac{\tau_g^{(t)} f(\boldsymbol{x}_i; \boldsymbol{\mu}_g^{(t)}, \boldsymbol{\Sigma}_g^{(t)})}{\sum_{k=1}^{G} \tau_k^{(t)} f(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)})}. \tag{3.3}$$

The resultant expectations $\hat{z}_{ig}^{(t)}$ of (3.3) are referred to as estimated component memberships (*a posteriori*), and can be interpreted as the probability that observation i belongs to component g given $\boldsymbol{x_i}$. The E-step is the same across all members of the GPCM family. One of the features implemented in `mixture` is deterministic annealing (3.2). Deterministic annealing (DA) is an optimization technique which avoids local minima of a given cost function (Zhou and Lange, 2010). The cost function in this case is (3.2). For a given $\nu \in \mathbb{R}(0, 1]$, DA is implemented within the E-step as

$$\hat{z}_{ig}^{(t)} = \mathbb{E}\left[Z_{ig} | \boldsymbol{x}_i, \boldsymbol{\Theta}^{(t)}, \nu \right] = \frac{\left[\tau_g^{(t)} f(\boldsymbol{x}_i; \boldsymbol{\mu}_g^{(t)}, \boldsymbol{\Sigma}_g^{(t)}) \right]^{\nu}}{\sum_{k=1}^{G} \left[\tau_k^{(t)} f(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}) \right]^{\nu}}. \tag{3.4}$$

DA is used to make the likelihood surface flatter, and can be viewed as a progression towards the full E-step (for details see Section 2.2.4 of McNicholas, 2016a).

The following code sample demonstrates the use of DA within the `gpcm` function:

```
### Load dataset
data(x2)

### use deterministic annealing for starting values
axDA = gpcm(x2, G=1:5, start=0,da=c(0.3,0.5,0.8,1.0))
summary(axDA)
```

The M-step varies across the 14 models within the GPCM family. However, the estimation of τ_g's and $\boldsymbol{\mu}_g$'s is common throughout. The maximization step on iteration $(t+1)$ for τ_g and $\boldsymbol{\mu}_g$ is given as

$$\hat{\tau}_g^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ig}^{(t)}}{n} \quad \text{and} \quad \hat{\boldsymbol{\mu}}_g^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ig}^{(t)} \boldsymbol{x}_i}{\sum_{i=1}^{n} \hat{z}_{ig}^{(t)}}.$$

For $\boldsymbol{\Sigma}_g$, the estimation is different for each model in the GPCM family. All models have the $\boldsymbol{\Sigma}_g$ estimations implemented according to Celeux and Govaert (1995) except EVE and VVE. The work by Browne and McNicholas (2014) develops a MM algorithm for estimating the covariance of models EVE and VVE. As a result, the MM algorithm for such models within `mixture` is implemented accordingly.

Model convergence, selection, and performance is performed in accordance with Section 2.2. Mixture model selection by BIC is fairly common as it is based on an approximation to Bayes factors (Kass and Raftery, 1995). For convenience, column 6 of Table 2.1 gives the number of free covariance parameters for each member of the GPCM family.

Within `mixture`, the following code sample demonstrates parsimonious model selection within the `gpcm` function:

```
# Load dataset and run models.
data(x2)
mm = gpcm(x2,G=3,mnames=c("VVV","EVE","VII")) # run three models.

# Display Results
summary(mm)
print(mm)
```

3.3 Imputation of Missing Data

The ability to handle missing data is an essential tool that most mixture model-based approaches lack. The imputation of missing data is a well researched topic; a complete review of methods can be found in literature such as Little and Rubin (2019) and Schafer (1997). The current approach for `mclust` is to use a general location model (GLOC, Schafer, 1997). However, the work by Di Zio *et al.* (2007) develops two approaches which imbed the imputation of missing data within the EM algorithm itself. Of the two investigated approaches, the conditional mean imputation (CMI) is found to be superior for preserving the sample mean. Using this existing literature, a similar approach is developed which outperforms `mclust` in both imputation and classification.

3.3.1 Conditional Mean Imputation

For a particular observation vector $\boldsymbol{x}_i$, there exists the potential to have up to $p-1$ missing entries. Let $\boldsymbol{m}_i$ universally correspond to a collection of indices to which entries are missing for vector $\boldsymbol{x}_i$. Furthermore, let $\boldsymbol{d}_i$ correspond to the collection of indices which are non-missing for vector $\boldsymbol{x}_i$. For example, if $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \text{NA}, x_{i4}, \ldots, x_{ip})$, then $\boldsymbol{m}_i = \{3\}$ and $\boldsymbol{d}_i = \{1, 2, 4, \ldots, p\}$. Essentially, $\boldsymbol{m}_i$ and $\boldsymbol{d}_i$ keep track of which entries are missing and non-missing for a particular observation vector. With this notation established, for a particular observation i, let $\boldsymbol{x}_{i,\boldsymbol{m}_i}$, and $\boldsymbol{x}_{i,\boldsymbol{d}_i}$ be vectors of missing values and non-missing values respectively. The imputation step of the CMI method is given as

$$\hat{\boldsymbol{x}}_{i,\boldsymbol{m}_i} = \mathbb{E}\left[\boldsymbol{x}_{i,\boldsymbol{m}_i} | \boldsymbol{z}_i, \boldsymbol{\Theta}, \boldsymbol{x}_{i,\boldsymbol{d}_i}\right] = \sum_{g=1}^{G} z_{ig} \mathbb{E}\left[\boldsymbol{x}_{i,\boldsymbol{m}_i} | \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \boldsymbol{x}_{i,\boldsymbol{d}_i}\right], \tag{3.5}$$

$$\mathbb{E}\left[\boldsymbol{x}_{i,\boldsymbol{m}_i} | \boldsymbol{\theta}_g, \boldsymbol{x}_{i,\boldsymbol{d}_i}\right] = \boldsymbol{\mu}_{g,\boldsymbol{m}_i} + \boldsymbol{\Sigma}_{g,\boldsymbol{m}_i,\boldsymbol{d}_i} \boldsymbol{\Sigma}_{g,\boldsymbol{d}_i,\boldsymbol{d}_i}^{-1} (\boldsymbol{x}_{i,\boldsymbol{d}_i} - \boldsymbol{\mu}_{g,\boldsymbol{d}_i}). \tag{3.6}$$

Here, $\boldsymbol{z}_i$ is the vector of *a posterori* for observation i. The vectors $\boldsymbol{\mu}_{g,\boldsymbol{m}_i}$ and $\boldsymbol{\mu}_{g,\boldsymbol{d}_i}$ are the mean vector entries of $\boldsymbol{\mu}_g$, associated with the missing and non-missing data entries respectively. $\boldsymbol{\Sigma}_{g,\boldsymbol{m}_i,\boldsymbol{d}_i}$ are the co-variance entries of $\boldsymbol{\Sigma}_g$, associated with the missing and non-missing entries. Note, $\boldsymbol{\Sigma}_{g,\boldsymbol{m}_i,\boldsymbol{d}_i}$ is a rectangular matrix if the number of missing entries exceed the number of non-missing entries, or vice versa. Finally, $\boldsymbol{\Sigma}_{g,\boldsymbol{d}_i,\boldsymbol{d}_i}$ is the square co-variance matrix of the non missing data taken from $\boldsymbol{\Sigma}_g$. The conditional mean imputation is fairly common and has been derived in Browne *et al.* (2022) and Keef *et al.* (2009).

Three Phase Imputation EM Algorithm

The imputation of missing data under presence of mixtures is a circular referencing problem. We see that (3.5) requires both the parameters $\boldsymbol{\Theta}$ and the *a posteriori* $\boldsymbol{z}_i$. However, to compute the E-step and M-step, you need to have already calculated $\hat{\boldsymbol{x}}_{i,\boldsymbol{m}}$, which in turn, requires you to impute missing data. Therein lies a circular reference between imputation and estimation resulting in a unique problem. As rectification, consider the introduction of a three phase imputation EM algorithm (3PEM) for missing data. The 3PEM is split into three separate procedures aiming to optimize (3.2) under the presence of missing data. We begin with the burn-in phase.

1. Given an initialization of the *a posteriori*, temporarily remove all observations with missing data entries, and their respective $\boldsymbol{z}_i$. As a consequence, this algorithm will not work in situations where every single observation has missing values.

2. Perform the EM algorithm b times with the non-missing observations to acquire parameters $\boldsymbol{\Theta}_\star$.

This phase is aimed at acquiring sufficient initialization parameters for the imputation of missing values. Next, we move onto the imputation phase. Place the missing data observations, and their estimates for $\boldsymbol{z}_i$ (see 3.3), back into the original dataset.

1. Impute the missing data $\hat{\boldsymbol{x}}_{i,\boldsymbol{m}}$ according to (3.5), with parameters $\boldsymbol{\Theta}_\star$.

2. Perform an E-step on the missing data, to acquire new $\boldsymbol{z}_i$.

3. Perform an M-step to gain new parameters $\boldsymbol{\Theta}_o$, which are estimated with the newly imputed data.

4. Impute the missing data one more time with $\boldsymbol{\Theta}_o$ as a preparation for the final phase.

Concluding with the final EM phase, use the initializations $\boldsymbol{\Theta}_o$ and begin an EM algorithm according to the following scheme.

1. Perform the E-step on the entire observed data.
2. Impute missing data according to (3.5).
3. Perform a M-step with the newly imputed data.
4. Calculate (3.1), and check convergence using (2.2).
5. Repeat steps $1 - 4$ if convergence has not been reached or if the maximum number of iterations have not been exhausted.

Each phase of the 3PEM algorithm is designed to handle a specific problem. The first phase handles the problem of estimating good initialization parameters for imputation. If one poorly imputes missing observations into bad starting values, there is a risk for the algorithm to become trapped in a local minima. The second phase handles the issue of performing a proper imputation with good initialization parameters. Since the burn-in phase results in proper estimates for parameters, the imputation is more accurate during the imputation phase. The last phase is designed to optimize (3.1) in the presence of missing data. For a convenient summary of the 3PEM algorithm, see Algorithm 1.

Algorithm 1: Three Phase EM Algorithm

Data: Temporarily remove all observations with missing data entries and their $\boldsymbol{z}_i$.

begin

Burn-In Phase

while *iterations* $< b$ **do**

E-step

M-step

Result: Acquire parameters $\boldsymbol{\Theta}_{no}$ estimated from the non-missing observations.

Data: Place the observations with missing data entires and their respective $\boldsymbol{z}_i$ (3.3) back.

begin

Imputation Phase

Impute the missing data $\hat{\boldsymbol{x}}_{i,\boldsymbol{m}}$ according to (3.5)

E-step

M-step

Impute $\hat{\boldsymbol{x}}_{i,\boldsymbol{m}}$

Result: Imputed $\hat{\boldsymbol{x}}_{i,\boldsymbol{m}}$ and acquired initialization parameters $\boldsymbol{\Theta}_o$.

begin

EM Phase

while *iterations* $<$ *max iterations* **do**

E-step

Impute $\hat{\boldsymbol{x}}_{i,\boldsymbol{m}}$

M-step

if *Converged* **then**

break

Result: Optimized (3.2).

3.3.2 Imputation Example

Any model within `mixture` can handle missing data easily through their respective function calls. Missing data values can be passed into the call by simply placing a place-holder `NA` within the dataset of choice. The following code example goes through a simulated missing data scenario and plots the results in Figure 3.1.

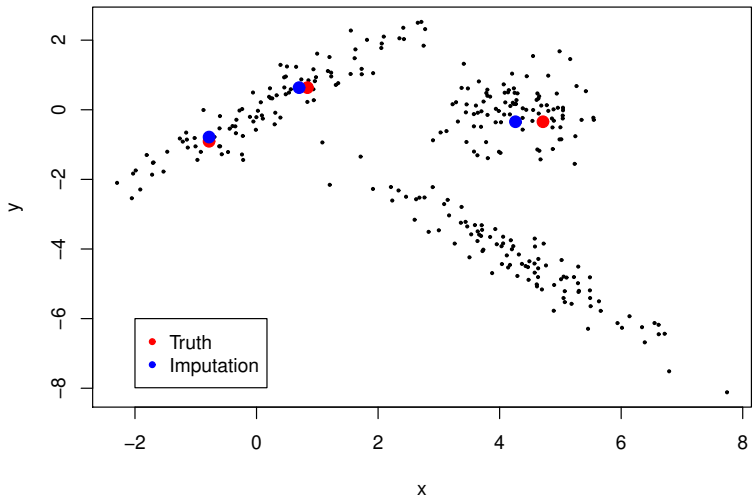


Figure 3.1: Imputation example plot for the `x2` dataset. Here, the true values (red) are simulated to have missing values and imputed by the algorithm (blue). The imputation results in a very close approximation to the actual value.

```
# Load x2 dataset and set seed.
data(x2)
set.seed(1)

# Randomely place NA's within the x2 dataset.
x2miss <- x2

# sample rows and columns
nobs_miss <- sample(x=1:dim(x2)[1], size=3)

# sample columns.
c_miss <- sample(x=1:2, size=3, replace=TRUE)
```

```
# place NA's
for(i in 1:3){
  x2miss[nobs_miss[i], c_miss[i]] <- NA
}

# plot targets in red.
plot(x2, pch=20, cex=0.5, xlab="x", ylab="y")
points(x2[nobs_miss,] , pch=20, cex = 2.0, col="red")

# run gpcm fitting
m = gpcm(x2miss, G = 1:3, start = 0)

# attempt e-step and plot imputations in green.
e_m_result <- e_step(x2miss, m$best_model)
points(e_m_result$X[nobs_miss,], pch=20, cex=2.0, col="blue")
legend(-2,-6, legend=c("Truth", "Imputation"),
            col=c("red", "blue"),
            pch=20)
```

3.4 Extensions to Skewed Distributions

There is a significant amount of literature on clustering skewed distributions. McNicholas (2016a) covers an in-depth overview on high-dimensional skewed clustering with both parsimonious and factor based approaches. Consider the following, let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ be n independent, p-dimensional random vectors formulated as

$$\boldsymbol{X}_i = \boldsymbol{\mu}_g + \boldsymbol{\alpha}_g Y_{ig} + \sqrt{Y_{ig}}\boldsymbol{U}_g, \quad | \quad Z_{ig} = 1. \tag{3.7}$$

Here, $\boldsymbol{\mu}_g$ is considered to be location vector emanating from component g, while $\boldsymbol{\alpha}_g$ is the skewness vector. The vector $\boldsymbol{U}_g$ is distributed as a multivariate normal with a zero mean vector, and a covariate matrix $\boldsymbol{\Sigma}_g$. Y_i is considered a latent variable, whereby the selection of its distribution constistutes the resultant member distribution of the normal variance mean mixture (NVMM) family . The density of $\boldsymbol{X}_i|Z_{ig}=1$ can be formulated as

$$f(\boldsymbol{x}_i;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g,\boldsymbol{\alpha}_g,\boldsymbol{\theta}_g)=\int_0^\infty \phi_p(\boldsymbol{x}_i;\boldsymbol{\mu}_g+y_i\boldsymbol{\alpha}_g,y_i\boldsymbol{\Sigma}_g)h(y_i;\boldsymbol{\theta}_g)\mathrm{d}y_i,\quad \boldsymbol{x}_i\in\mathbb{R}^p. \tag{3.8}$$

Here, ϕ_p is the density of a p-dimensional multivariate Gaussian with mean $\boldsymbol{\mu}_g+y_i\boldsymbol{\alpha}_g$, covariance matrix $y_i\boldsymbol{\Sigma}_g$, and $h(y_i;\boldsymbol{\theta}_g)$ as the density function of a univariate random variable defined on $y_i>0$ (Barndorff-Nielsen *et al.*, 1982). The function $h(y_i;\boldsymbol{\theta}_g)$ is a weighting function whereby its form characterizes the resultant normal variance-mean mixture distribution. Taking Y_{ig} to be distributed according to a generalized inverse Gaussian (GIG; Barndorff-Nielsen and Halgreen, 1977), $\boldsymbol{X}_i$ results in the member of the NVMM family with the most parameters, the generalized hyperbolic distribution (Barndorff-Nielsen *et al.*, 1982; Browne and McNicholas, 2015). Naturally, the density (3.8) then follows a closed form with $\boldsymbol{\theta}_g:=(\omega_g,\lambda_g)$ written as

$$f(\boldsymbol{x}_i;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g,\boldsymbol{\alpha}_g,\boldsymbol{\theta}_g)=\frac{\mathcal{K}_{\lambda_g-p/2}\left(\sqrt{(\omega_g+\boldsymbol{\alpha}_g^\top\boldsymbol{\Sigma}_g^{-1}\boldsymbol{\alpha}_g)(\omega_g+\delta(\boldsymbol{x}_i;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g))}\right)}{\mathcal{K}_{\lambda_g}(\omega_g)(2\pi)^{p/2}\mid\boldsymbol{\Sigma}_g\mid^{1/2}\exp\left\{(\boldsymbol{\mu}_g-\boldsymbol{x}_i)^\top\boldsymbol{\Sigma}_g^{-1}\boldsymbol{\alpha}_g\right\}}\left(\frac{\omega_g+\delta(\boldsymbol{x}_i;\boldsymbol{\alpha}_g,\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)}{\omega_g+\boldsymbol{\alpha}_g^\top\boldsymbol{\Sigma}_g^{-1}\boldsymbol{\alpha}_g}\right)^{\frac{\lambda_g-p/2}{2}}.$$

Here $\mathcal{K}_\lambda$ is the modified Bessel function of the third kind, λ_g is the index, ω_g is the concentration parameter, and $\delta(\boldsymbol{x}_i;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g):=(\boldsymbol{x}_i-\boldsymbol{\mu}_g)^\top\boldsymbol{\Sigma}_g^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_g)$. For specifics regarding mixtures of generalized hyperbolic distributions see Browne and McNicholas (2015). Different selections for the distribution of Y_{ig} results in a different member of

the NVMM family. For example, if $Y_{ig} \sim \text{Exp}(1)$, an exponential distribution with rate 1, then (3.8) results in a shifted asymmetric Laplace (Franczak *et al.*, 2014). Similarly, estimation of parameters is based on the EM algorithm with an added E-step, and M-step. Regarding the M-step, the estimation of parameters $\boldsymbol{\mu}_g$, and $\boldsymbol{\alpha}_g$ are the same under the NVMM family and is provided as follows:

$$\hat{\boldsymbol{\mu}}_g = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i \hat{z}_{ig}(\bar{a}_g b_{ig} - 1)}{\sum_{i=1}^{n} \hat{z}_{ig}(\bar{a}_g b_{ig} - 1)}, \qquad \hat{\boldsymbol{\alpha}}_g = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i \hat{z}_{ig}(\bar{b}_g - b_{ig})}{\sum_{i=1}^{n} \hat{z}_{ig}(\bar{a}_{ig} b_{ig} - 1)}, \tag{3.9}$$

where $n_g = \sum_{i=1}^{n} \hat{z}_{ig}$, $\bar{a}_g = (1/n_g) \sum_{i=1}^{n} \hat{z}_{ig} a_{ig}$, and $\bar{b}_g = (1/n_g) \sum_{i=1}^{n} \hat{z}_{ig} b_{ig}$. a_{ig} and b_{ig} are taken to be the expected values of $\mathbb{E}[Y_{ig}|X_i = \boldsymbol{x}_i]$ and $\mathbb{E}[1/Y_{ig}|X_i = \boldsymbol{x}_i]$ respectively. These expectations are quite involved, and differ for each skewed distribution (For specifics see Franczak *et al.*, 2014; McNicholas *et al.*, 2017; Wei *et al.*, 2019; O'Hagan *et al.*, 2016). However, across all possible parsimonious settings as in Table 2.1, they remain the same. It follows that the estimate for $\boldsymbol{\Sigma}_g$ is given as

$$\hat{\boldsymbol{\Sigma}}_g = \frac{1}{n_g} \sum_{i=1}^{n} \hat{z}_{ig} b_{ig} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_g)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_g)^\top - \hat{\boldsymbol{\alpha}}_g (\bar{\boldsymbol{x}}_i - \hat{\boldsymbol{\mu}}_g)^\top - (\bar{\boldsymbol{x}}_i - \hat{\boldsymbol{\mu}}_g)\hat{\boldsymbol{\alpha}}_g^\top + \bar{a}_g \hat{\boldsymbol{\alpha}}_g \hat{\boldsymbol{\alpha}}_g^\top. \tag{3.10}$$

This estimate is under the VVV constraint as outlined in Table 2.1. For all other models, follow the same procedures as outlined in Celeux and Govaert (1995) and Browne and McNicholas (2014) with the changes to corresponding matrix in (3.10). For the purposes of imputing skewed data, we modify (3.5) within the 3PEM algorithm to

that of Wei *et al.* (2019) as

$$\mathbb{E}\left[\boldsymbol{x}_{i,\boldsymbol{m}_i}|\boldsymbol{x}_{i,\boldsymbol{d}_i},\boldsymbol{\mu}_g,\boldsymbol{\alpha}_g,\boldsymbol{\Sigma}_g,\boldsymbol{\theta}_g\right]=\boldsymbol{\mu}_{g,\boldsymbol{m}_i}+a_{ig}\boldsymbol{\alpha}_{g,\boldsymbol{m}_i}+\boldsymbol{\Sigma}_{g,\boldsymbol{m}_i,\boldsymbol{d}_i}\boldsymbol{\Sigma}^{-1}_{g,\boldsymbol{d}_i,\boldsymbol{d}_i}(\boldsymbol{x}_{i,\boldsymbol{d}_i}-\boldsymbol{\mu}_{g,\boldsymbol{d}_i}-a_{ig}\boldsymbol{\alpha}_{g,\boldsymbol{d}_i}), \tag{3.11}$$

where $a_{ig}=\mathbb{E}\left[Y_{ig}|\boldsymbol{x}_i,\boldsymbol{\mu}_g,\boldsymbol{\alpha}_g,\boldsymbol{\Sigma}_g,\boldsymbol{\theta}_g\right]$. This imputation derives from Bayes theorem where $\boldsymbol{X}_i|(Y_i=y)\sim\mathcal{N}(\boldsymbol{\mu}+y\boldsymbol{\alpha},y\boldsymbol{\Sigma})$. Within the 3PEM algorithm, for each iteration, imputation is performed with the current estimates of $\boldsymbol{\mu}_g,\boldsymbol{\alpha}_g$, and $\boldsymbol{\Sigma}_g$. Finally, regarding estimation of latent parameters $\boldsymbol{\theta}_g$, each skewed distribution beckons their own estimation procedure. In some cases, such as skew-t, the estimation of the respective $\boldsymbol{\theta}_g$ parameters are of a closed form. However, for all other distributions, we must solve a non-linear equation that differs across every single skewed model. We employ the use of the `boost` C++ library for solving non-linear equations (Schäling, 2011). In addition, for the efficient computing of modified Bessel functions we employ the use of the `GSL` C++ library (Galassi *et al.*, 2019). As an example, the update for $\boldsymbol{\theta}_g:=\eta_g$ of a variance gamma distribution (McNicholas *et al.*, 2017) involves solving the following non-linear equation:

$$\varphi(\eta_g)-\log(\eta_g)-\bar{c}_g+\bar{a}_g-1=0.$$

Here $\bar{c}_g=\frac{1}{n_g}\sum_{i=1}^{n}z_{ig}\mathbb{E}\left[\log(Y_{ig})|\boldsymbol{X}_i=\boldsymbol{x}_i\right]$, and $\varphi(.)$ is the digamma function. To solve this non-linear equation using the `boost` library, we employ the use of Halley's method (Proinov and Ivanov, 2015). Halley's method is superior compared the standard Newton-Raphson solvers as it has a rate of convergence to the cubic order (Alefeld, 1981).

The `mixture` package contains three skewed approaches each with a family of 14 parsimonious models. The generalized hyperbolic parsimonious models (GHPCMs) provide the largest degree of parametrization for skewed data. The variance-gamma parsimonious models (VGPCMs) often provide the best BIC. Finally, the set of skew-t parsimonious models (STPCMs) are often the most numerically stable of models. In summary, across all models, we provide functionality for imputation. The `mixture` package contains three functions for fitting skewed models: `vgpcm`, `stpcm`, and `ghpcm`. The following code samples runs each of the skewed models on the `sx2` dataset:

```
# Load dataset.
data(sx2)

# Generalized Hyperbolic
ghMM = ghpcm(sx2, G=2, start=0)
summary(ghMM)

# Variance Gamma
vgMM = vgpcm(sx2, G=2, start=0)
summary(vgMM)

# Skew-t
stMM = stpcm(sx2, G=2, start=0)
summary(stMM)
```

3.5 Extensions to the Student's t Distribution

The Student's t distribution (Student, 1908) is particularly interesting when used in clustering scenarios. The density for some cluster g of the Student's t distribution is

formulated as

$$f_t(\boldsymbol{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g,\nu_g) = \frac{\Gamma(\frac{\nu_g+p}{2})|\boldsymbol{\Sigma}_g|^{-1/2}}{(\pi\nu_g)^{\frac{p}{2}}\Gamma\left(\frac{\nu_g}{2}\right)\left(1+\frac{(\boldsymbol{x}-\boldsymbol{\mu}_g)^\top\boldsymbol{\Sigma}_g^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_g)}{\nu_g}\right)^{\frac{\nu_g+p}{2}}}.$$

The parameters for scale ($\boldsymbol{\Sigma}_g$), and location ($\boldsymbol{\mu}_g$) follow the same paradigm as the previous distributions. For example, $\boldsymbol{\Sigma}_g$ can be decomposed into the 14 parsimonious models as in (2.1). However, the degree of freedom (ν_g) can be selected to either be constrained, or allowed to vary across clusters. The combination of both the 14 parsimonious settings, and the setting for the degree of freedom, comprises what is known as the tEIGEN family of parsimonious models (Andrews *et al.*, 2018). Suprisingly, despite having different covariances, the conditional mean imputation of missing data follows the same exact procedure as in (3.5). For specifics see Liu (1995). The tEIGEN family contains 28 models, and is provided through the `tpcm` function within the `mixture` package. The following code example demonstrates such functionality:

```
# Load sample dataset
data("x2")

# Unconstrained degrees of freedom.
mm = tpcm(x2,G=3)

# Constrained degrees of freedom.
mmC = tpcm(x2,G=3,constrained=TRUE)
```

3.6 The Stochastic EM Algorithm

For every iteration of the EM algorithm, it is guaranteed that (3.1) increase monotonically. However, attaining a global maximum is not guaranteed in practice as the optimization surface may be unfavorable; you may end in a local maxima. As an alternative, the stochastic EM algorithm (SEM), first formulated by Celeux (1985) extends the original in the following aspect. The SEM does not get stuck (Nielsen, 2000), provides greater information about the data (Diebolt and Ip, 1996), and in some cases, outperforms the original EM (Celeux *et al.*, 1996). The SEM algorithm differs from the original by one key step. Consider the *a posterori* for some observation i, at iteration t, specified in (3.3) as $\hat{\boldsymbol{z}}_i^{(t)} = (\hat{z}_{i1}^{(t)}, \ldots, \hat{z}_{iG}^{(t)})$. Next, consider $\hat{\boldsymbol{z}}_i^{(t)}$ to be the parameters of multinomial distribution, and sample

$$\star\boldsymbol{z}_i^{(t)} \sim \mathcal{M}(1, \hat{\boldsymbol{z}}_i^{(t)}). \tag{3.12}$$

Since $\star\boldsymbol{z}_i^{(t)}$ is random at every iteration, it allows for the opportunity to place membership of observation i in varying clusters. By consequence, this allows for the opportunity to reach a different likelihood surface even when initializations are the same. For example, consider the crabs dataset from the `MASS` package in R (Ripley *et al.*, 2013). This multivariate dataset consists of morphological measurements on Leptograpsus crabs. The `gpcm` function from the `mixture` package is ran on the `crabs` dataset with the exact same initializations, but differing optimization algorithms.

The following code sample produces desirable results.

```
# Load packages.
library(mixture)
library(MASS)
```

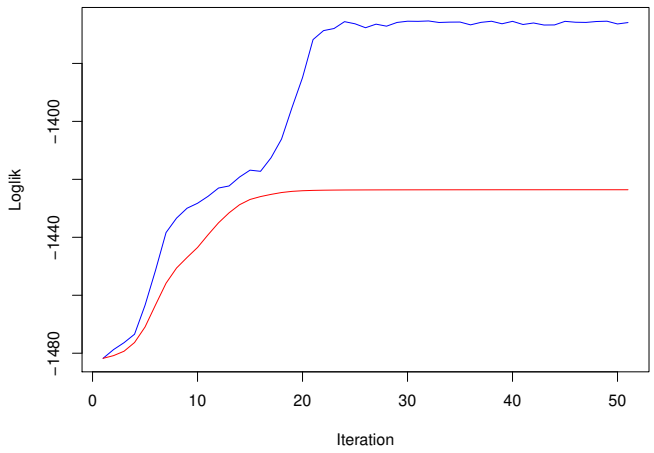


Figure 3.2: Incomplete data log-likelihood for `crabs` dataset. The SEM (blue) and EM (red) optimization algorithms are ran for 50 iterations for the `crabs` dataset.

```
# Setup.
set.seed(4)
XX <- as.matrix(crabs[,-c(1,2,3)]) # grab crabs measurements
zStart <- z_ig_random_soft(200,2) # same initializations

# Run EM.
mEM = gpcm(XX,"VVV",G=2,start= zStart,nmax = 50)

# Run SEM.
mSEM = gpcm(XX,"VVV",G=2,start=zStart, stochastic=TRUE, nmax=50)

# Plot EM likelihood.
plot(mSEM$model_objs[[1]]$logliks,
     type = "l", col = 'blue',
```

```
     xlab = "Iteration",
     ylab = "Loglik")

# Plot SEM likelihood.
lines(mEM$model_objs[[1]]$logliks,
     type = "l", col = 'red',
     xlab = "Iteration",
     ylab = "Loglik")
```

3.7 Simulation Study

In this section a simulation study is devised to measure performance on classification and imputation of missing values. The simulation study is performed on several datasets under various missing data settings. The introduction of datasets is as follows. The `x2` dataset, found within `mixture`, is a simulated two-dimensional multivariate EVE Gaussian with three groups. The `sx2` dataset is a skewed variant of `x2` with only two groups. Finally, the `banknote` dataset, found within `mclust`, is a collection of six measurements made on 100 genuine and 100 counterfeit old-Swiss bank notes.

The simulation study is split into several subsections pertaining to each dataset. Regarding missing data experiments, consider a proportion of missing values γ. Observations are randomly selected within a particular dataset. Once an observation is selected, one of the entries is replaced with missing values. For example, consider a dataset of $n = 100$ observations, and let $\gamma = 0.05$. According to these settings, 5 of the observations will have at least one missing entry. Within R, this is done by assigning `NA` to the entry along the observation of choice. It is worth noting that `mixture` will

only work where there is at least one non-missing entry along a particular observation vector. `x2` is simulated two-dimensional mixture of multivariate Gaussians. There are three groups generated consisting of 100 observations each totalling $n = 300$. The following code sample produces the right hand side of Figure 3.3:

The GPCM family is fitted on `x2` for $G = 1, \ldots, 6$ using both `mixture` and `mclust`. Both implementations of the GPCM family arrive at the exact same results consisting of the EVE model with $G = 3$ and a BIC of -1981. Consider the model comparison plots shown in Figure 3.3. For the purposes of stylization, it is beneficial to consider alternatives to line graphs when representing model fits. Since there are 14 models, the line graphs tend to become clustered and chaotic when estimating multiple G. Consider a level plot from the package `lattice` that presents model fit as a color scheme. The best models are presented as "white hot" and descend in performance to a darker brown. The level plot is a cleaner way to present model performance overall, and, is able to differentiate model fits with ease. Using `x2`, 1000 different datasets are randomly generated with missing values based on several settings of γ. As a measure of performance, the mean squared error (MSE) is used for comparing both packages (Wang and Bovik, 2009). In addition, $b = 5$ is held constant for the burn-in phase, across all settings of γ. The box-violin plots in Figure 3.5 shows that `mixture` outperforms `mclust` in the imputation of missing data across all settings of γ (Hintze and Nelson, 1998). In Figure 3.4a, the performance has very similar results indicating no difference between `mclust` and `mixture`. In most cases, the mean MSE of the new approach is much lower than the mean MSE of `mclust`. However, there are some extreme cases where the new method does not do so well, and, the `mclust` package outperforms. Figure 3.7 shows classification performance across several missing data

settings. The 3PEM method results in better classification performance according to ARI overall. There is one exception.

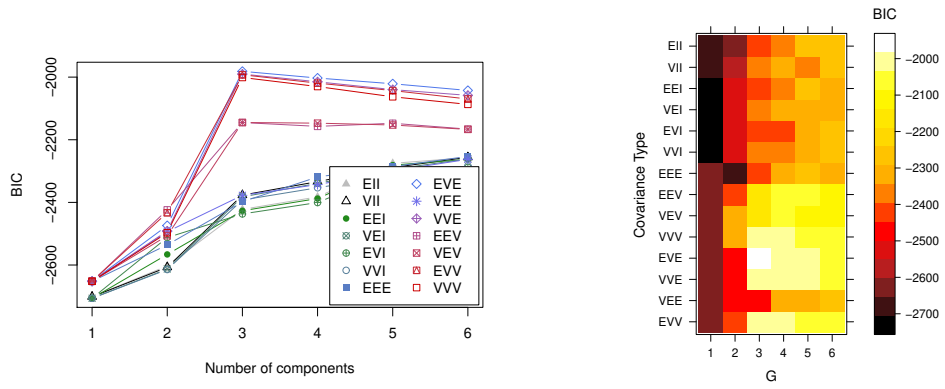
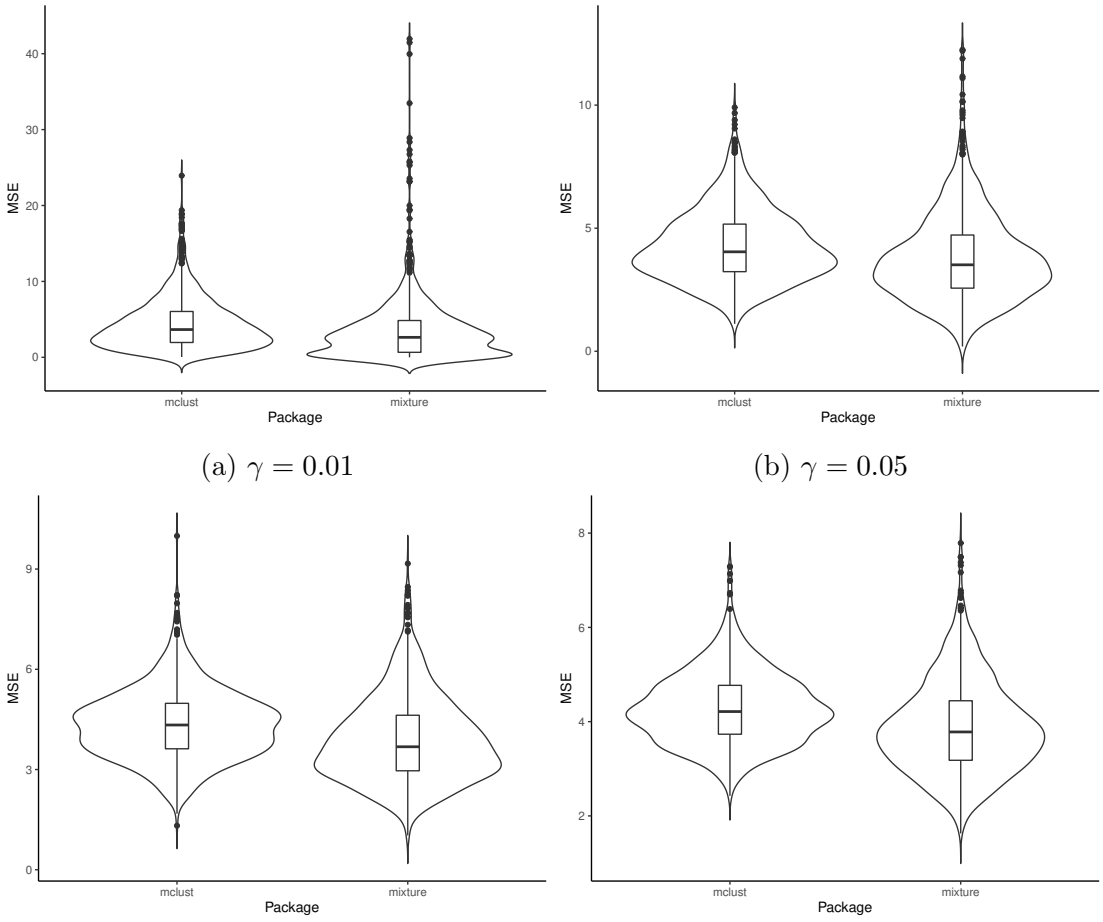


Figure 3.3: Model comparison plots between the mclust (left) vs mixture (right) packages, on the `x2` dataset.

The `banknote` dataset is a six-dimensional multivariate dataset. Within this simulation study, missing values are randomly generated along a randomly selected observation vector. There will be at least one, up to no more than 5 missing values along a particular vector. Note, the `banknote` dataset has a larger number of dimensions compared to `x2`. There are two groups each consisting of 100 observations. Missing values are simulated 1000 times where the 3PEM algorithm is performed with similar settings as in the `x2` study. Again, MSE is used as a measure of imputation performance, while ARI is used for classification. Figure 3.6 shows imputation performance for several settings of γ. The 3PEM approach is reported to have better performance in imputation when compared to `mclust` across all γ settings. As for classification, Figure 3.8 displays classification performance across several settings of γ. Again, the 3PEM approach has a better classification performance based on ARI across all settings. In summary, both ARI and MSE results indicate the approach is

superior when compared to `mclust` within a higher dimensional setting.

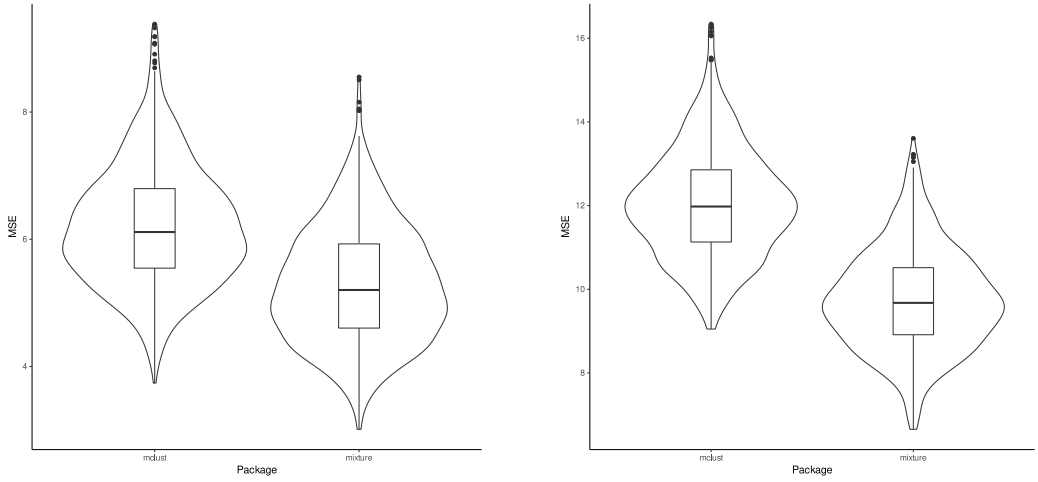
The `sx2` synthetic dataset is found within the `mixture` package. The `sx2` dataset contains two groups which are generated from a bivariate variance-gamma distribution with a VVV covariance structure. Using `sx2`, 1000 different datasets are randomly generated with missing values based on several settings of γ. The box-violin plots in Figure 3.5 shows that `mixture` outperforms `mclust` in the imputation of missing data across all settings of γ. In all cases, the mean MSE of the 3PEM approach is much lower than the mean MSE of `mclust`.



(a) $\gamma = 0.01$ (b) $\gamma = 0.05$

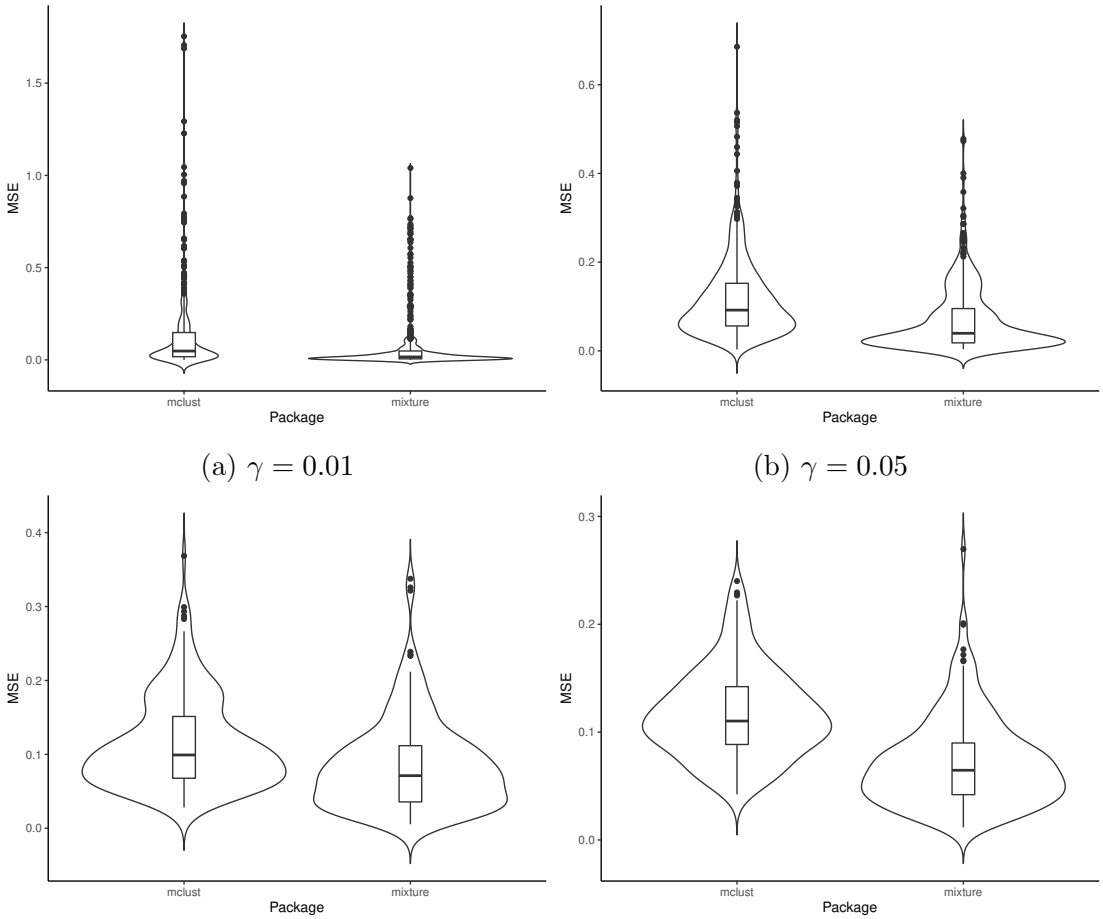
(c) $\gamma = 0.10$ (d) $\gamma = 0.20$

Figure 3.4: Simulation study results on imputation for `x2` data under a variety of missing data settings.



(a) $\gamma = 0.05$ (b) $\gamma = 0.10$

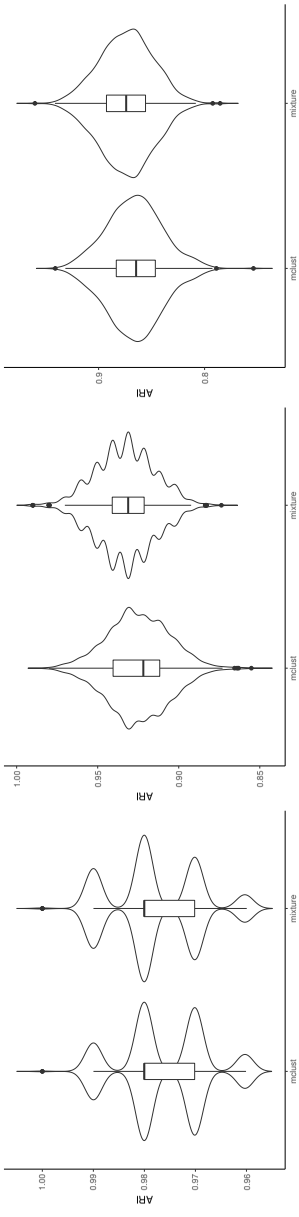
Figure 3.5: Simulation study results on imputation for `sx2`.



(a) $\gamma = 0.01$ (b) $\gamma = 0.05$

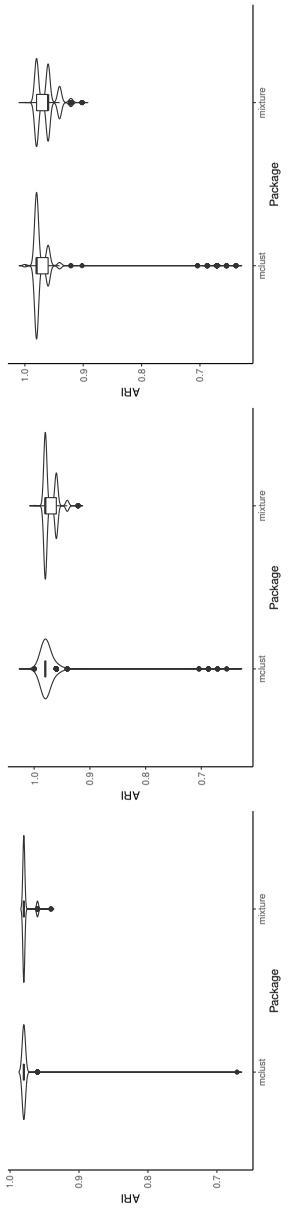
(c) $\gamma = 0.10$ (d) $\gamma = 0.20$

Figure 3.6: Simulation study results on imputation for `banknote` data under a variety of missing data settings.



(a) $\gamma = 0.01$ (b) $\gamma = 0.05$ (c) $\gamma = 0.10$

Figure 3.7: Simulation study results on classification for `x2` data under a variety of missing data settings.



(a) $\gamma = 0.01$ (b) $\gamma = 0.05$ (c) $\gamma = 0.10$

Figure 3.8: Simulation study results on classification for `banknote` data under a variety of missing data settings.

3.8 Summary

The `mixture` package contains a wide variety of models and features that greatly benefit those who wish to conduct clustering and classification. The ability to handle missing data across any model is particularly significant as most real-world datasets often contain missing entries. In most cases, these observations are filtered out but now with `mixture`, users have the ability to conduct analysis on such datasets without concern.

When comparing to existing packages, it is apparent from the simulation studies wherein `mixture` package outperforms `mclust` for missing data. The 3PEM algorithm provides superior results against the GLOC approach in both imputation and classification, whether skewed or symmetric. With both real and synthetic datasets, across multiple settings, the 3PEM algorithm is efficient in both clustering, and imputation performance. Classification performance is maintained, even in the presence of missing data. Model fit was performed through the `mixture` package on 80 2.20GHz Intel Xeon Silver CPUs.

In conclusion, there is a clear benefit for using `mixture` in the presence of missing data. In addition, `mixture` was written with an object oriented paradigm which easily allows for future additional add-ons. One future area in particular would be the imputation under censored data.

CHAPTER 4

EFFICIENT OPTIMIZATION OF NORMAL VARIANCE-MEAN MIXTURES

The optimization of NVMM models often comes at a cost both in stability and speed due to several issues. Most literature on normal variance-mean mixtures makes use of the EM algorithm. However, the calculation of expected values in the E-Step requires the computation of Bessel functions. The computational cost or overhead of such calculations are significant due to the use of auxiliary functions, and, backwards recurrence relations. Furthermore, the calculations are subject to inaccuracies as Bessel functions are misbehaved and prone to singularities for small inputs. Consider an alternative strategy for parameter estimation of normal variance-mean mixtures.

This strategy forgoes the evaluation of Bessel functions and instead, relies on posterior sampling. Based on the stochastic EM algorithm (Celeux, 1985), the new method improves performance and stability particularly in clustering applications. The stochastic EM (SEM) algorithm for variance-mean models shows a significant improvement in speed due to the use of random sampling over evaluating Bessel functions. Furthermore, it is shown that the SEM algorithm has comparable performance to the standard EM when optimizing log-likelihoods. As a clarifying measure when discussing "model performance" herein, it is in reference to the log-likelihood and other related measures of classification performance such as ARI. Conversely, when discussing efficiency or the "speed" of the algorithm, it is in reference to the computational overhead.

4.1 On the Computation of $\mathcal{K}_\lambda$

It is clear that the density (2.9) and expectations (2.10)–(2.12) of a GIG distribution are of closed form. However, there is consistent overhead for computing the modified Bessel function of the third kind ($\mathcal{K}_\lambda$). The evaluation of such functions are heavily involved, with several schemes to consider (Gil *et al.*, 2002; Temme, 1975; Amos, 1974). Of the most popular approaches, Temme's algorithm takes advantage of recurrence relations (Temme, 1975; Campbell, 1980). This algorithm has been implemented in the popular GSL library (Galassi *et al.*, 2019) and, is also contained in the `Rmath` header library for the R programming language (R Core Team, 2021). Despite its popularity, the algorithm is extremely involved and bears a heavy computational load. In fact, all methods for computing $\mathcal{K}_\lambda$ come at a heavy overhead cost. Moreover, the function itself is subject to singularities and/or loss in precision for

some small inputs. For some input ω, it is known that the relative error of Temme's algorithm decreases with increasing values of ω (Campbell, 1980). As a result, the new strategy for optimizing NVMMs aims to completely avoid the computation of $\mathcal{K}_\lambda$ except when absolutely necessary. Instead of computing heavy cost functions, consider another set algorithms and sampling procedures.

4.2 Current Optimization Methods

There have been several approaches for optimizing (4.1) across the family of NVMMs. A natural candidate for non-linear optimization is that of a quasi-Newton scheme (Ownuk *et al.*, 2021). However, such methods are limited due to evaluating derivatives, which contrive the computation of Bessel functions. Snoussi and Idier (2006) approaches the problem by a stochastic scheme via Markov chain Monte Carlo (MCMC). Although not explicitly stated, their approach is reminiscent of an SEM algorithm (Celeux, 1985). Across most literature there is a clear consensus of selecting an EM based approach for optimizing (4.1). Birge and Chavez-Bedoya (2021) use the EM for portfolio optimization, Protassov (2004) for modelling exchange rates, and Browne and McNicholas (2015) for clustering and classification.

As such, consider first the EM algorithm for the NVMM optimization scheme. Let a p-dimensional random variable $\boldsymbol{\mathcal{X}}$ be characterized by a normal variance-mean mixture as in (2.14). Given a sample of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, the log-likelihood function is formulated as

$$l(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \log\left(f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\alpha}, \boldsymbol{\theta})\right). \tag{4.1}$$

Here, $\boldsymbol{\theta}$ accounts for the parameters associated with the latent variable characterizing the specific member of the NVMM family. The EM iteratively maximizes the log-likelihood function when data is incomplete, i.e., when there is missing and/or latent data. In the context of NVMMs the missing data consists of the latent variables Y_i. Without loss of generality to other NVMMs, consider the most parameterized distribution; the generalized hyperbolic distribution (GHD). With several parameters to model location, scale, asymmetry, and concentration (Barndorff-Nielsen and Halgreen, 1977), the GHD has been applied in many statistical problems e.g. Protassov (2004). The stochastic representation of (2.14) is now redefined with $Y \sim \mathrm{I}(\omega, 1, \lambda), \omega > 0, \lambda \in \mathbb{R}$ and the resultant density of $\boldsymbol{\mathcal{X}}$ is of closed form with $\boldsymbol{\theta} := (\omega, \lambda)$, as

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \frac{\mathcal{K}_{\lambda - p/2}\big(\sqrt{(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha})(\omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}))}\big)}{\mathcal{K}_\lambda(\omega)(2\pi)^{p/2} \mid \boldsymbol{\Sigma} \mid^{1/2} \exp\Big\{(\boldsymbol{\mu} - \boldsymbol{x})^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}\Big\}} \left(\frac{\omega + \delta(\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}}\right)^{\frac{\lambda - p/2}{2}}, \tag{4.2}$$

and $\delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) := (\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$. The density itself is subject to complications. For example, if the denominators $\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}$ or $\mathcal{K}_\lambda(\omega)$ become extremely small, we have singularities. Conversely, if arguments for $\mathcal{K}_\lambda$ are extremely small, we have loss in precision. Since the density is the central component of the objective function, computing $\mathcal{K}_\lambda$ is not completely avoidable. Great care must be taken into account when computing such densities, and, is a constant issue for any distribution of the NNVM family. Campbell (1980) does provide a remedy where one computes $e^\omega \mathcal{K}_\lambda(\omega)$, and then normalizes by dividing with the leading term. From experience, this normalization has shown to bring stability in calculations; however, it still bears a large computational overhead. Galassi *et al.* (2019) provides both the standard

(`gsl_sf_bessel_Kn`), and, exponentially scaled (`gsl_sf_bessel_Kn_scaled`) versions for computing $\mathcal{K}_\lambda$.

The procedure for the EM algorithm alternates between imputing missing data, and maximizing parameters. Suppose that the latent variable is known ($Y = y$), then the conditional distribution of $\boldsymbol{\mathcal{X}}|(Y = y)$ is $\mathcal{N}(\boldsymbol{\mu} + y\boldsymbol{\alpha}, y\boldsymbol{\Sigma})$, where $\mathcal{N}$ is an appropriate multivariate Gaussian. Furthermore, it follows from Bayes theorem and Browne and McNicholas (2015) that

$$Y|\boldsymbol{\mathcal{X}} = \boldsymbol{x} \sim \text{GIG}(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}, \omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \lambda - p/2).$$

For every member of the NNVM family, the conditional distribution is of the same form, i.e., a GIG distribution, and therefore the proposed SEM algorithm would hold in generality for members of the NNVM family. In the interest of clarity, the algebra of derivations for each step of the EM have been previously derived in work such as Protassov (2004) and Browne and McNicholas (2015).

Define $\hat{\boldsymbol{\Phi}}^{(t)} := (\hat{\lambda}^{(t)}, \hat{\omega}^{(t)}, \hat{\boldsymbol{\mu}}^{(t)}, \hat{\boldsymbol{\Sigma}}^{(t)}, \hat{\boldsymbol{\alpha}}^{(t)})$ to be parameter estimates at iteration t. Due to closed forms of both posterior distributions, and equations (2.10 – 2.12), the expectation step (E-step) for some arbitrary parameter set $\boldsymbol{\Phi}$ can be written as:

$$\mathfrak{a}_i(\boldsymbol{\Phi}) := \mathbb{E}\left[Y_i|\boldsymbol{\mathcal{X}} = \boldsymbol{x}_i\right] = \sqrt{\frac{\omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}}} \frac{\mathcal{K}_{\lambda+1}(\sqrt{(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha})(\omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}))})}{\mathcal{K}_\lambda(\sqrt{(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha})(\omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}))})},$$

$$\mathfrak{b}_i(\boldsymbol{\Phi}) := \mathbb{E}\left[Y_i^{-1}|\boldsymbol{\mathcal{X}} = \boldsymbol{x}_i\right] = \sqrt{\frac{\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}}{\omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})}} \frac{\mathcal{K}_{\lambda+1}(\sqrt{(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha})(\omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}))})}{\mathcal{K}_\lambda(\sqrt{(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha})(\omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}))})} - \frac{2\lambda}{\omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})},$$

$$\mathfrak{c}_i(\boldsymbol{\Phi}) := \mathbb{E}\left[\log(Y_i)|\boldsymbol{\mathcal{X}} = \boldsymbol{x}_i\right] = \frac{1}{2}\log\left(\frac{\omega + \delta(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})}{\omega + \boldsymbol{\alpha}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}}\right)$$
$$+ \frac{1}{\mathcal{K}_\lambda(\sqrt{(\omega + \boldsymbol{\alpha}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha})(\omega + \delta(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}))})}\frac{\partial}{\partial\lambda}\mathcal{K}_\lambda(\sqrt{(\omega + \boldsymbol{\alpha}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha})(\omega + \delta(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}))}).$$

Initialization of the EM algorithm is of critical importance because the EM algorithm converges to a local maxima, and is highly dependent on initial starts. Consider Gaussian initializations for $\hat{\boldsymbol{\mu}}^{(0)}$and $\hat{\boldsymbol{\Sigma}}^{(0)}$; i.e. taking the sample mean, and sample covariance as initializers. Furthermore, set the initialization for $\hat{\boldsymbol{\alpha}}^{(0)} = \mathbf{0}$, $\hat{\lambda}^{(0)} = 1.0$, and $\hat{\omega}^{(t)} = 1.0$. This initialization is somewhat appropriate for the skewness and concentration, as the likelihood surface is quite flat with respect to these parameters (Protassov, 2004). Now, for some iteration t, simplify the following notation and consider $\mathfrak{a}_i^{(t)} := \mathfrak{a}_i(\boldsymbol{\Phi}^{(t-1)})$, $\mathfrak{b}_i^{(t)} := \mathfrak{b}_i(\boldsymbol{\Phi}^{(t-1)})$, $\mathfrak{c}_i^{(t)} := \mathfrak{c}_i(\boldsymbol{\Phi}^{(t-1)})$, $\bar{\mathfrak{a}} = (1/n)\sum_{i=1}^n \mathfrak{a}_i$, $\bar{\mathfrak{b}} = (1/n)\sum_{i=1}^n \mathfrak{b}_i$, and $\bar{\mathfrak{c}} = (1/n)\sum_{i=1}^n \mathfrak{c}_i$; the maximization step (M-step) is formulated as

$$\hat{\boldsymbol{\mu}}^{(t)} = \frac{\sum_{i=1}^n \boldsymbol{x}_i(\bar{\mathfrak{a}}^{(t)}\mathfrak{b}_i^{(t)} - 1)}{\sum_{i=1}^n(\bar{\mathfrak{a}}^{(t)}\mathfrak{b}_i^{(t)} - 1)}, \qquad \hat{\boldsymbol{\alpha}}^{(t)} = \frac{\sum_{i=1}^n \boldsymbol{x}_i(\bar{\mathfrak{b}}^{(t)} - \mathfrak{b}_i^{(t)})}{\sum_{i=1}^n(\bar{\mathfrak{a}}^{(t)}\mathfrak{b}_i^{(t)} - 1)},$$

$$\hat{\lambda}^{(t)} = \bar{\mathfrak{c}}\hat{\lambda}^{(t-1)}\left[\frac{\partial}{\partial\lambda}\log\mathcal{K}_\lambda(\hat{\omega}^{(t-1)})\Big|_{\lambda=\hat{\lambda}^{(t-1)}}\right],$$
$$\hat{\omega}^{(t)} = \hat{\omega}^{(t-1)} - \left[\frac{\partial}{\partial\omega}q(\omega,\hat{\lambda}^{(t)})\Big|_{\omega=\hat{\omega}^{(t-1)}}\right]\left[\frac{\partial^2}{\partial\partial\omega^2}q(\omega,\hat{\lambda}^{(t)})\Big|_{\omega=\hat{\omega}^{(t-1)}}\right]^{-1},$$

$$\hat{\boldsymbol{\Sigma}}^{(t)} = \frac{1}{n}\sum_{i=1}^n \mathfrak{b}_i^{(t)}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}^{(t)})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}^{(t)})^\top - \hat{\boldsymbol{\alpha}}^{(t)}(\bar{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}^{(t)})^\top - \hat{\boldsymbol{\alpha}}^{(t)\top}(\bar{\boldsymbol{x}} - \hat{\boldsymbol{\mu}}^{(t)}) + \bar{\mathfrak{a}}^{(t)}\hat{\boldsymbol{\alpha}}^{(t)}\hat{\boldsymbol{\alpha}}^{(t)\top},$$

where $\bar{\boldsymbol{x}} = (1/n)\sum_{i=1}^n \boldsymbol{x}_i$, and $q(\omega,\lambda) = -\log\mathcal{K}_\lambda(\omega) + (\lambda-1)\bar{c} - \frac{\omega}{2}(\bar{\mathfrak{a}} + \bar{\mathfrak{b}})$. Updates

for λ and ω are determined by conditional maximization because q is log-convex with respect to inputs (Browne and McNicholas, 2015; Baricz, 2010). Convergence is established via Aitken's acceleration as defined in (2.2) . For convenience, the above procedure has been summarized in Algorithm 2.

Algorithm 2: Standard EM algorithm for GHD

Input: Data $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$, and initialized $\hat{\boldsymbol{\Phi}}^{(0)} := (\hat{\lambda}^{(0)}, \hat{\omega}^{(0)}, \hat{\boldsymbol{\mu}}^{(0)}, \hat{\boldsymbol{\Sigma}}^{(0)}, \hat{\boldsymbol{\alpha}}^{(0)})$.
begin
  while *iterations* $<$ *max iterations* **do**
    E-step: Compute $\mathfrak{a}_i^{(t)}, \mathfrak{b}_i^{(t)}, \mathfrak{c}_i^{(t)}$ across all observations i.
    M-step: Compute $\hat{\boldsymbol{\Phi}}^{(t)} = (\hat{\lambda}^{(t)}, \hat{\omega}^{(t)}, \hat{\boldsymbol{\mu}}^{(t)}, \hat{\boldsymbol{\Sigma}}^{(t)}, \hat{\boldsymbol{\alpha}}^{(t)})$
    Check convergence using (2.2) **if** *Converged* **then**
      break
Output: Optimized set of parameters $\hat{\boldsymbol{\Phi}}$.

4.3 The SEM Algorithm for NVMMs

It has been demonstrated across literature that the EM algorithm works relatively well. However, there is consistent computational overhead for both the E-step and M-step. To alleviate such overhead, consider an alternative scheme. Instead of calculating expectations for $Y|\boldsymbol{\mathcal{X}} = \boldsymbol{x}$, we sample from the posterior distribution of $\text{GIG}(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}, \omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \lambda - p/2)$. In order for this to be effective, the sampling algorithm must be more efficient than directly calculating the expectations. Consider the work of Hörmann and Leydold (2014) which outlines three different sampling algorithms for the GIG distribution. The Ratio-of-Uniform's algorithm had been independently proposed by both Dagpunar (2007), and Lehner (1989). There

are two variants of this algorithm, one with mode-shift, and one without. Each variant supersedes one another in performance depending on the domain of parameter inputs. For convenience, Algorithms 4 and 5 are outlined in detail at the end of this chapter. However, according to Hörmann and Leydold (2014), the algorithm drops in performance for $\lambda < 1$, $\omega < 0.5$; as it does not have a uniformly bounded rejection constant. To remedy this issue, Hörmann and Leydold (2014) introduce a non-concave procedure that has superior performance overall (Algorithm 3). Although the algorithms look quite involved, in practice, they consist of a composition of strictly low-overhead functions. Programmatically, one can further reduce the cost of all algorithms by a few algebraic tricks as in the case of:

$$\mathcal{V}^2 \leq h^\star(Y) \Leftrightarrow \log \mathcal{V} > \frac{1}{2}(\lambda - 1)\log Y - \frac{\omega}{4}\left(Y + \frac{1}{Y}\right) - c,$$

where c is some normalizing constant, and h^* is the quasi-density. Such algebraic forms are of particular importance as they avoid the direct calculation of densities, and consequently, avoid calculating $\mathcal{K}_\lambda$. Furthermore, the sampling algorithms are defined on the second parameterization of the GIG distribution, namely $\mathrm{I}(\omega^\star, \eta^\star, \lambda - p/2)$. The appropriate re-parameterizations must be calculated from the original form of $\mathrm{GIG}(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}, \omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \lambda - p/2)$ as follows:

$$\begin{aligned}\omega^\star &= \sqrt{(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha})\,(\omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}))},\\ \eta^\star &= \sqrt{(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha})^{-1}\,(\omega + \delta(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}))}\,.\end{aligned}$$

All algorithms have been implemented in both the Julia and C++ programming languages. With the algorithms of choice established, consider an alternative scheme for the E-step. Instead of taking expectations, sample directly from the posterior

distribution as follows:

$$\begin{aligned}
\tilde{\mathfrak{a}}_i(\boldsymbol{\Phi}) &\sim \text{GIG}(\omega + \boldsymbol{\alpha}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\alpha}, \omega + \delta(\boldsymbol{x}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \lambda - p/2), \\
\tilde{\mathfrak{b}}_i(\boldsymbol{\Phi}) &= \frac{1}{\tilde{\mathfrak{a}}_i(\boldsymbol{\Phi})}, \\
\tilde{\mathfrak{c}}_i(\boldsymbol{\Phi}) &= \log\left(\tilde{\mathfrak{a}}_i(\boldsymbol{\Phi})\right).
\end{aligned}$$

The primary reduction in cost is that one only samples once from the distribution, and then applies the appropriate transforms. By circumventing the calculation of $\mathcal{K}_\lambda$ for the E-step, one can gain an immediate boost in performance. The new optimization algorithm for a GHD now falls under the SEM algorithm category. Previous literature shows that the SEM has comparable performance to the EM with an added benefit. Due to its stochastic nature, the SEM explores the likelihood surface more efficiently by avoiding local maxima (Celeux, 1985). The termination criteria of the SEM algorithm is taken to be the same as (2.2). To summarize, although this methodology has been outlined for the GHD, it is completely generalizable to any distribution of the NVMM family. Finally, since the E-step has been simplified, any previous or existing work on NVMMs can benefit greatly without much change in methodology or code.

4.4 Simulation Study

The purpose of this simulation study is to measure the performance of an E-step calculation. To achieve this, consider a series of simulation studies across different implementations and languages. Julia and C++ are selected due to their effective track record as high performance computing languages (Gibson, 2017). Within C++, there are several implementations of $\mathcal{K}_\lambda$. This is the key focus of the simulation study

as $\mathcal{K}_\lambda$ is the main bottleneck for the E-step calculation. Differences in implementation have an effect on performance for calculating $\mathcal{K}_\lambda$. As of `C++17`, there are several Bessel functions found in the `<cmath>` header library. In addition, GSL also offers similar implementations. The main objective is to compare GSL, and `C++17`, against the newly devised posterior sampling procedure.

The standard E-step is constant for any domain of parameter inputs, while the sampling procedure is not. Since the posterior sampler is based on three different sampling algorithms, one must formulate three different parameter domains for the study. Each parameter set is selected such that a specific sampling algorithm is benchmarked against the standard E-step. Each calculation is run 10000 times where the completion time of each experiment is recorded. In C++, the standard `<chrono>` library is used to measure time. Conversely, for Julia the package `BenchmarkTools.jl` is used to capture the same metrics. To gain a representative understanding of performance, the minimum, median, mean, and maximum time measurements are reported. The best in class metrics are bolded for convenience within each table. Regarding compiler optimizations, the highest possible level of optimization is set for the compiler. Great care was taken such that the compiler did not pre-compute values before actual run-time in order to avoid misrepresentations in performance.

Beginning with Table 4.1, Algorithm 5 is benchmarked against the standard E-step. The C++ mode-shift procedure (Algorithm 5) outperforms or matches the standard E-step performance in the best-case (minimum) and worst-case (maximum) scenarios. However, the median and mean calculations for the mode-shift benchmarks algorithm shows a result of an approximate 20% gain in performance overall. The Julia benchmarks show similar results with the exception of the worst-case metric

favouring Julia's GSL implementation.

Table 4.1: Reported times across competing latent step methods with parameters $\lambda = 1$, $\omega^\star = \rho^\star = 8/7$. Algorithm 4 is used for the SE step of the SEM algorithm. Time is measured in nanoseconds across 10000 runs.

Method	Min	Median	Mean	Max
C++ 17 (EM)	978	2375	2364	26613
C++ GSL (EM)	908	2375	2973	1852339
C++ Shift (SEM)	**908**	**1885**	**1735**	**17112**
Julia S.F. (EM)	1885	2864	4743	17727307
Julia GSL (EM)	1397	2375	2519	**55314**
Julia Shift (SEM)	**838**	**1316**	**1397**	79074

Table 4.2 reports the results of the no-mode-shift procedure (Algorithm 4). For C++, the sampling procedure meets or exceeds the other implementations in all categories. Based on the mean and median, one expects a 50% increase in performance overall. Again, Julia's GSL procedure attains the lead in worst case scenario. Overall, the sampling procedure shows great improvement over the standard E-step for the selected parameter set. Finally considering Table 4.3, the standard E-step is benchmarked against the non-concave sampler (Algorithm 3). Again, a similar pattern emerges where the sampler outperforms other implementations save for Julia's worst case scenario performance. Overall one should expect a 60% increase in performance during realistic scenarios. In conclusion, the results elucidate the superior performance of sampling procedures in comparison to the standard E step across multiple domains.

Table 4.2: Reported times across competing latent step methods with parameters $\lambda = 2.1$, $\omega^\star = 15/7$, and $\rho^\star = 8/7$. Algorithm 3 is used for the SE step in the SEM algorithm. Time is measured in nanoseconds across 10000 runs.

Method	Min	Median	Mean	Max
C++ 17 (EM)	908	2794	2626	73753
C++ GSL (EM)	1397	3282	3112	68864
C++ No Shift (SEM)	**908**	**1886**	**1832**	**14248**
Julia S.F. (EM)	2304	5210	3352	17774526
Julia GSL (EM)	1397	2793	2671	**54825**
Julia No Shift (SEM)	**838**	**1397**	**1385**	888521

Table 4.3: Reported times across competing latent step methods with parameters $\lambda = 1/2$, $\omega^\star = 1/7$, and $\rho^\star = 8/7$. Algorithm 2 is used for the SE step in the SEM algorithm. Time is measured in nanoseconds across 10000 runs.

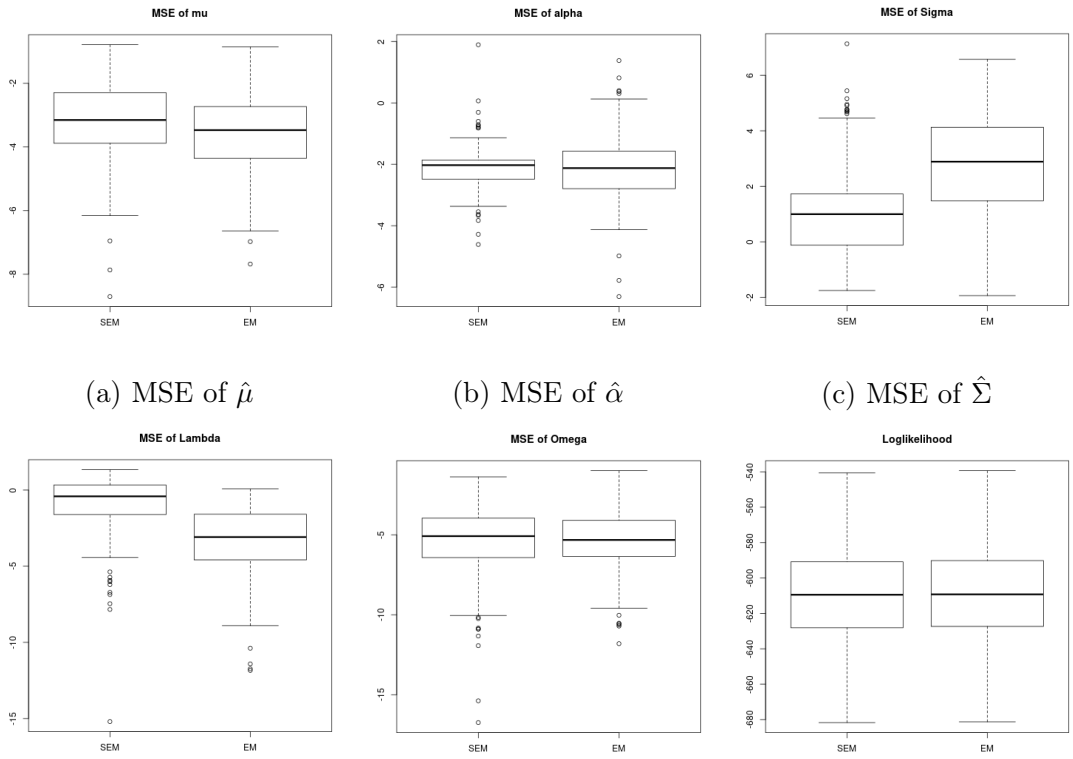
Method	Min	Median	Mean	Max
C++ 17 (EM)	908	2304	2316	322180
C++ GSL (EM)	907	2375	2888	70192
C++ Non-concave (SEM)	**907**	**1476**	**1698**	**19626**
Julia S.F. (EM)	1396	2445	4396	17283121
Julia GSL (EM)	1396	2375	2535	**56290**
Julia Non-concave (SEM)	**838**	**1337**	**1397**	782826

Consider another simulation study designed to assess performance with regards to estimating parameters. A single component generalized hyperbolic distribution is generated from the following parameter set:

$$\boldsymbol{\mu} = \begin{bmatrix} 0.0527 \\ 0.2227 \\ 0.5068 \end{bmatrix}, \quad \boldsymbol{\alpha} = \begin{bmatrix} -0.0656 \\ -0.2772 \\ -0.6306 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} 2.9461 & -0.3332 & -0.7606 \\ -0.3332 & 4.9783 & 1.3439 \\ -0.7606 & 1.3439 & 2.1733 \end{bmatrix},$$

$$\lambda = -0.6227, \quad \omega = 0.2034.$$

Across 1000 simulations, a dataset of $n = 300$ observations were drawn randomely from the GHD distribution with parameters defined above. Per each simulated dataset, both the EM and SEM were run for 200 iterations.



(a) MSE of $\hat{\mu}$ (b) MSE of $\hat{\alpha}$ (c) MSE of $\hat{\Sigma}$

(d) MSE of $\hat{\lambda}$ (e) MSE of $\hat{\omega}$ (f) Log-likelihood

Figure 4.1: MSE results of simulation study on a log-scale for parameters and likelihood.

Figure 4.1 shows the results from the simulation study. Beginning with the MSE for $\boldsymbol{\mu}$ in Figure 4.1a, we see comparable performance with respect to both algorithms. A one-sided t-test on log transformed MSE results in a p-value < 0.001, indicating that EM outperforms the SEM when estimating $\boldsymbol{\mu}$. With regards to estimating $\boldsymbol{\alpha}$, Figure 4.1b similarly shows comparable performance. A series of one-sided and two-sided t-tests show no conclusive statistical evidence that the performance for estimating

$\boldsymbol{\alpha}$ differs. However, we can see that the variance for SEM is much smaller than the EM. Figure 4.1c shows the MSE performance when estimating $\boldsymbol{\Sigma}$. Clearly the SEM outperforms the EM for estimating $\boldsymbol{\Sigma}$. A one-sided t-test with a p-value < 0.001 shows evidence favouring the SEM. Figure 4.1d shows the MSE results for estimating λ on a log-scale. Visually we see that EM outperforms the SEM when estimating λ. A one-sided t-test also indicates that the EM outperforms the SEM with a p-value < 0.001. With regards to ω, the Figure 4.1e shows very similar MSE performance. A series of one-sided and two-sided t-test fails to reject the null hypothesis, indicating no conclusive evidence that one method outperforms the other. Finally, Figure 4.1f shows the log-likelihood results of both algorithms. Visually we see that the two methods achieve very similar log-likelihoods indicating no difference in performance between the two methods. Furthermore, a series of one-sided and two-sided t-tests fail to reject the null hypothesis, showing no evidence that the methods differ in terms of log-likelihood. In conclusion, we see some benefit to using the SEM when estimating covariances. However, one clear noticible difference between both methods is that the SEM is signficantly faster than the EM as discussed in the previous study.

4.5 Application

One of the more difficult problems in unsupervised classification is that of skewed or asymmetric clusters. Much work has been done to alleviate such issues when performing clustering and classification. Fitting skewed distributions in a heterogeneous context brings upon a heavy computational load as one needs to fit G parameter sets. In the context of model-based clustering (McNicholas, 2016a), each set of parameters represents a single cluster component. The set of parameters characterizes

the shape, volume, direction and overall behaviour of each cluster within the data. For skewed or asymmetric clustering scenarios, the family of NVMMs have been particularly effective in modelling such behaviour. Consider the work of Browne and McNicholas (2015) where each cluster emanates from a generalized hyperbolic distribution. Methodologically, the standard E-step for a mixture is extremely similar to the single component case. Naturally, the posterior sampling method is a candidate for fitting such models. To replace the original E-step, the sampling method must show to be superior in several aspects. The first must be in the interest of efficiency. The sampling method must fit the same model with a significant reduction in run-time. Next, the sampling method must maximize the objective function (log-likelihood) to a degree equal to or greater than that of the standard E-step. Finally, in the context of clustering, the SEM procedure must either meet or exceed the standard EM with respect to classification accuracy.

The package `MixGHD` in R contains a large collection of models derived from the generalized hyperbolic distribution (Tortora *et al.*, 2021). There are many extensions to the multivariate GHD model within this package. Without loss of generality, and, for the purposes of this application, consider a modified `MGHD` function. The sampling procedure is implemented for `MGHD` via a C++ extension (Eddelbuettel *et al.*, 2011). The `crabs` dataset is selected consisting of 5 morphological measurements on *leptograpsus* crabs (Campbell and Mahon, 1974; Venables and Ripley, 2002). The dataset contains 200 observations consisting of two species (orange and blue), and sex. Therefore, the dataset consists of 50 crabs each per possible combination. For the purposes of this study, classification performance is assessed based on sex. Both algorithms are ran for 1000 iterations with `kmeans` initialization. The SEM took 5.24

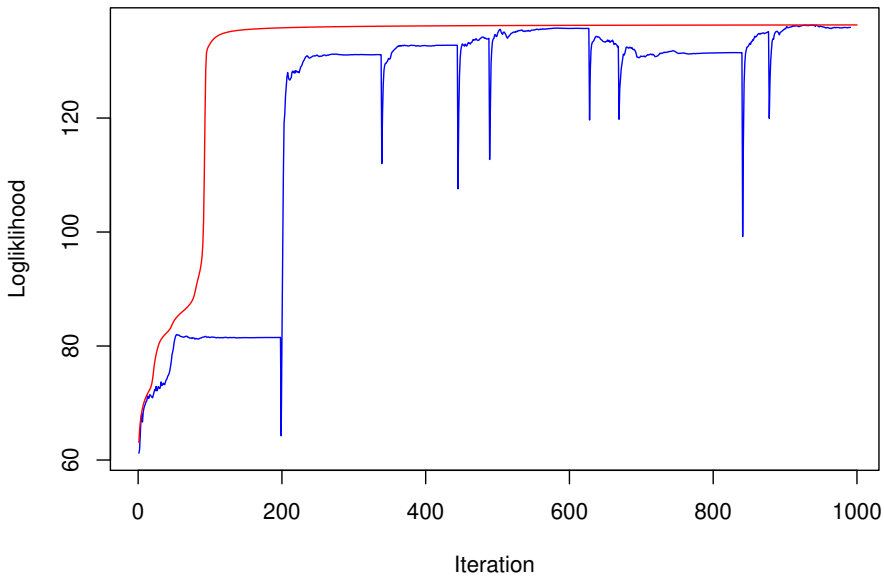


Figure 4.2: Log-likelihood over iterations for a `MGHD` fit on the `crabs` dataset. The posterior sampling method (blue) shows similar performance near the final iterations when compared to the standard EM (red).

seconds to finish while the standard EM took 8.80 seconds. Classification accuracy is reported as 93.50% for the SEM, and 92.50% for the standard EM. Figure 4.2 illustrates the log-likelihood by iteration for both algorithms. Surprisingly, the SEM has inferior performance for the first few iterations, but eventually matches the standard EM near the end. A final log-likelihood of 136.34 for the SEM and 137.64 for the standard EM shows the standard EM outperforming. However, in this case, the model with the slightly worse log-likelihood ended up with a higher accuracy. One caveat is that this does not hold in generality as demonstrated with the following example.

The package `mixture` in R contains several implementations of clustering models including that of the generalized hyperbolic variety (Pocuca *et al.*, 2021). With some

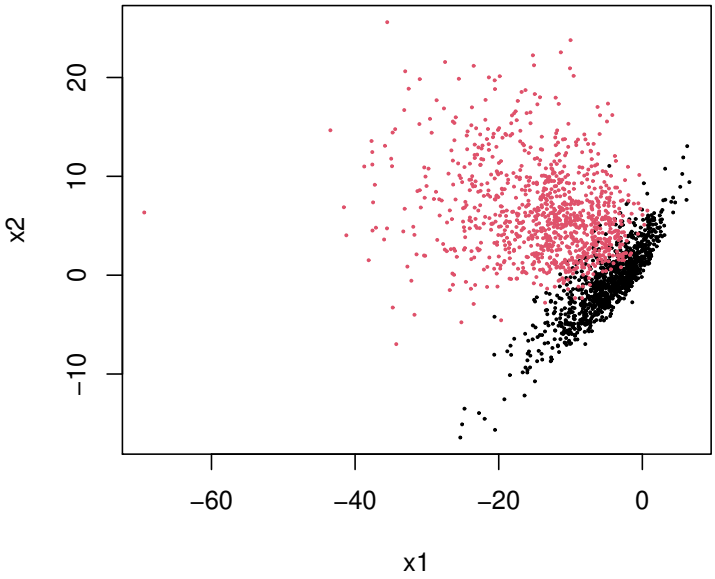


Figure 4.3: Scatter plot of `sx3` dataset coloured by memberships

very small modifications to the package, a clustering procedure is run on the dataset `sx3`. This data is of a two component bivariate generated from a variance-gamma distribution and is illustrated in Figure 4.3. Note, the variance-gamma distribution is a subclass of the generalized hyperbolic, and therefore, is appropriate in the context of this application. The two components are extremely close together which poses some challenge for a clustering scheme. Both the EM and SEM algorithms were fit on the dataset with random initializations across 1000 runs. The time to complete 1000 iterations of each algorithm, the log-likelihood, and the classification accuracy for each run is recorded. The average results are reported in Table 4.4. There is a noticeable difference in the average time, log-likelihood, and accuracy. The stochastic method shows an average speed-up of 23%. The average accuracy has also increased by 0.84%.

Table 4.4: Benchmarking the standard E-step against the stochastic variant over 1000 runs. The average and standard deviation of time (seconds), log-likelihood, and classification accuracy (%) are reported.

Method	Time	Log-likelihood	Accuracy
Standard	5.57 (0.11)	-12224.18 (0.665)	93.40 (0.1)
Stochastic	4.34 (0.94)	-12221.76 (2.681)	94.25 (1.3)

Overall, the results show that indeed the stochastic method outperforms the standard E-step through the new posterior sampling algorithm. Both in time, and in classification performance, the sampling method is a viable replacement for the standard E-step in a clustering context.

4.6 Discussion

The results of this work are not limited to multivariate clustering problems. Several uses of NVMMs extend to matrix variate and higher order distributions (Gallaugher and McNicholas, 2019, 2020; Gallaugher *et al.*, 2021a). Here, computational overhead plays a much larger role for such high dimensional datasets. Another potential application of the posterior sampling method would be for regression models that use an NVMM-like methodology. Gallaugher *et al.* (2021b) develops a skewed regression model with cluster-weighted components. The E-step for this model is extremely similar and, therefore, can also make use of the posterior sampling method. Another set of NVMM models that could benefit from this methodology include, inter alia, the normal inverse Gaussian (Barndorff-Nielsen, 1997; O'Hagan *et al.*, 2016), the variance gamma (Madan *et al.*, 1998), the skew-t (Gupta, 2003), and the shifted asymmetric Laplace. (Kotz *et al.*, 2001). Furthermore, more complex models such as the mixtures of GHD (Tortora *et al.*, 2021) could also benefit from the posterior

sampling method. From experience, a combination of both SEM and EM approaches would be optimal. The recommended strategy would be as follows. First, run the more performant SEM algorithm for several iterations, then, fine-tune the model fit using the standard EM for the last few iterations. In conclusion, this work applies several different sampling methods in the context of model estimation. The SEM is an efficient tool for optimizing normal variance-mean mixtures with a great potential for wide-spread adoption within NVMM methodologies.

Algorithm 3: Rejection method for non-$T_{-1/2}$-concave part

Input: Parameters λ, ω with $0 \leq \lambda < 1$ and $0 < \omega \leq \frac{2}{3}\sqrt{1-\lambda}$.
Output: A GIG distributed random variable Y.
begin

1: $m \leftarrow \omega/((1-\lambda) + \sqrt{(1-\lambda)^2 + \omega^2})$
2: $y_0 \leftarrow \omega/((1-\lambda)), \quad y_\star \leftarrow \max(y_0, 2/\omega)$
3: $k_1 \leftarrow g(m), \quad A_1 \leftarrow k y_0$
4: **if** $y_o < 2/\omega$ **then**
  $k_2 \leftarrow e^{-\omega}, \quad A_2 \leftarrow k_2((2/\omega)^\lambda - y_0^\lambda)/\lambda$
  if $\lambda = 0$ **then**
    $A_2 \leftarrow k_2 \log(2/\omega^2)$
5: **else**
  $k_2 \leftarrow 0, \quad A_2 \leftarrow 0$
6: $k_3 \leftarrow y_\star^{\lambda-1}, \quad A_3 \leftarrow 2k_3 \exp\{-y_\star\omega/2\}/\omega$
7: $A \leftarrow A_1 + A_2 + A_3$
8: **repeat**
  Generate $\mathcal{U} \sim \text{Unif}(0,1),\ \mathcal{V} \sim \text{Unif}(0, A)$
  if $V \leq A_1$ **then**
    $Y \leftarrow y_0 V/A_1,\ h \leftarrow k_1$
  else if $V \leq A_1 + A_2$ **then**
    $V \leftarrow V - A_1$
    $Y \leftarrow (y_0^\lambda + V\lambda/k_2)^{1/\lambda},\ h \leftarrow k_2 Y^{\lambda-1}$
    if $\lambda = 0$ **then**
      $Y \leftarrow \omega \exp(V \exp(\omega))$
  else
    $V \leftarrow V - (A_1 + A_2)$
    $Y \leftarrow -2/\omega \log(\exp(-y_\star\omega/2) - V\omega/(2k_3)),\ h \leftarrow k_3 \exp(-Y\omega/2)$
until $\mathcal{U}h \leq g(Y)$;
return Y

Algorithm 4: Ratio-of-Uniforms without mode shift

Input: Parameters λ, ω with $0 \leq \lambda < 1$ and $\min\{\frac{1}{2}, \frac{2}{3}\sqrt{1-\lambda}\} \leq \omega \leq 1$.
Output: A GIG distributed random variable Y.
begin
1: $m \leftarrow \omega/((1-\lambda) + \sqrt{(1-\lambda)^2 + \omega^2})$
2: $y^+ \leftarrow ((1+\lambda) + \sqrt{(1+\lambda)^2 + \omega^2})/\omega$
3: $v^+ \leftarrow \sqrt{h^\star(m)}$
4: $u^+ \leftarrow y^+\sqrt{h^\star(y^+)}$
5. **repeat**
  Generate $\mathcal{U} \sim \text{Unif}(0, u^+)$, and $\mathcal{V} \sim \text{Unif}(0, v^+)$
  $Y \leftarrow \mathcal{U}/\mathcal{V}$
until $\mathcal{V}^2 \leq h^\star(Y)$;
return Y

Algorithm 5: Ratio-of-Uniforms with mode shift

Input: Parameters λ, ω with $\lambda > 1$, and $\omega > 1$.
Output: A GIG distributed random variable Y.
begin
1: $m \leftarrow \omega/((1-\lambda) + \sqrt{(1-\lambda)^2 + \omega^2})$
2: $\varphi_1 \leftarrow -\frac{2(\lambda+1)}{\omega} - m$, $\varphi_2 \leftarrow \frac{2(\lambda-1)}{\omega}m - 1$,
3: $\mathfrak{p} \leftarrow \varphi_2 - \frac{\varphi_1^2}{3}$, $\mathfrak{q} \leftarrow \frac{2\varphi_1^3}{27} - \frac{\varphi_1\varphi_2}{3} + m$
4: $\vartheta \leftarrow \arccos(-\frac{\mathfrak{q}}{2}\sqrt{-\frac{27}{\mathfrak{p}^3}})$
5. $x^- \leftarrow \sqrt{-\frac{4}{3}\mathfrak{p}}\cos(\frac{\vartheta}{3} + \frac{4}{3}\pi) - \frac{\varphi_1}{3}$
6. $x^+ \leftarrow \sqrt{-\frac{4}{3}\mathfrak{p}}\cos(\frac{\vartheta}{3}) - \frac{\varphi_1}{3}$
7. $v^+ \leftarrow \sqrt{g(m)}$
8. $u^- \leftarrow (x^- - m)\sqrt{g(x^-)}$, $u^+ \leftarrow (x^+ - m)\sqrt{g(x^+)}$
9. **repeat**
  Generate $\mathcal{U} \sim \text{Unif}(u^-, u^+)$, and $\mathcal{V} \sim \text{Unif}(0, v^+)$
  $Y \leftarrow \mathcal{U}/\mathcal{V} + m$
until $\mathcal{V}^2 \leq g(Y)$;
return Y

CHAPTER 5

THE MISSING AND THE ASYMMETRIC: CLUSTERING WITH FINITE MIXTURES OF S_U JOHNSON DISTRIBUTIONS

A growing area of interest with regards to non-Gaussian data is that of clustering and classification. In statistical practice, it is common to have a population which consists of multiple sub-populations. The heterogeneity of such data poses particular challenges when proposing an appropriate model. An added challenge occurs when skewness is present within sub-populations and a Gaussian assumption becomes inadequate. The area of mixture model-based clustering constitutes a powerful framework

to tackle such issues. Consider the work of Zhu and Melnykov (2018), which introduces Manly transformations in the context of a finite mixture model. Methods of transformation within finite mixtures provide the ability to handle a multitude of patterns within data; be it symmetric or skewed. Alternatives to mixtures of transformations includes the work of Browne and McNicholas (2015) which models component-wise asymmetry with the generalized hyperbolic distribution (Barndorff-Nielsen and Halgreen, 1977). With parameters for location, variance, skewness, and concentration; the approach shows comparable performance against mixtures of transformations (Gallaugher *et al.*, 2020). In the context of mixture modelling, applications of the Johnson system of transformation can be found in work such as Dun and Kong (2022), where a mixture of Johnson's S_{B} distributions is developed to segment grey-scale liver images. In this chapter, the unbounded transformation system S_{U} is used to introduce finite mixtures of multivariate S_{U} Johnson distributions.

5.1 Finite Mixtures of S_{U} Johnson Distributions

Consider the notion of a finite mixture of S_{U} Johnson distributions. Let $\boldsymbol{\mathcal{X}}$ be a p-dimensional random variate which emanates from a G component finite mixture model. For group $g \in \{1, \ldots, G\}$, let $\boldsymbol{\mu}_g$, and $\boldsymbol{A}_g$ be the canonical shift and scale of a multivariate Gaussian distribution ($\mathcal{N}_p$) such that $\boldsymbol{\Sigma}_g = \boldsymbol{A}_g \boldsymbol{A}_g^\top$. Now, consider the hyperbolic scale matrix $\boldsymbol{\Lambda}_g := \mathrm{diag}(\delta_{1g}, \ldots, \delta_{pg})$, $\delta_{jg} > 0$ and shift vector $\boldsymbol{\omega}_g$ as previously defined in Section 2.4.2. Next, allow some latent variable Z to be distributed according to a multinomial distribution ($\mathcal{M}$) with parameters $\pi_1, \ldots, \pi_G$. Finally, a p-dimensional random variate $\boldsymbol{\mathcal{Y}}$ is said to emanate from a finite mixture of S_{U} Johnson distributions if its stochastic representation can be written as

$$
\begin{aligned}
Z &\sim \mathcal{M}(\pi_1, \ldots, \pi_G), \\
\boldsymbol{\mathcal{X}} &\sim \mathcal{N}_p(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \quad | \, Z = g, \\
\boldsymbol{\mathcal{Y}} &= \boldsymbol{\omega}_g + \boldsymbol{\Lambda}_g \boldsymbol{\varphi}(\boldsymbol{\mathcal{X}}) \quad | \, Z = g.
\end{aligned}
$$

The corresponding density for some realization $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$ is derived from equation (2.17) as

$$
\begin{aligned}
f_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{y}; \boldsymbol{\Theta}) &= \sum_{g=1}^{G} \pi_g f_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{y}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\omega}_g, \boldsymbol{\Lambda}_g), \\
f_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{y}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\omega}_g, \boldsymbol{\Lambda}_g) &= \frac{\exp\left\{-\frac{1}{2}\left(\boldsymbol{h}(\boldsymbol{y}; \boldsymbol{\vartheta}_g) - \boldsymbol{\mu}_g\right)^\top \boldsymbol{\Sigma}_g^{-1}\left(\boldsymbol{h}(\boldsymbol{y}; \boldsymbol{\vartheta}_g) - \boldsymbol{\mu}_g\right)\right\}}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}_g|^{\frac{1}{2}} \prod_{j=1}^{p} \left(\delta_{jg} \sqrt{\left(\frac{y_j - \omega_{jg}}{\delta_{jg}}\right)^2 + 1}\right)} \quad \boldsymbol{y} \in \mathbb{R}^p.
\end{aligned}
\tag{5.1}
$$

This form generally arises from transformations within mixtures as a canonical multivariate Gaussian density multiplied by a Jacobian term. See sections 2.3 of Zhu and Melnykov (2018) and 3.1 of Gutierrez *et al.* (1995) for similar derivations as (5.1).

5.2 Expectation Maximization Algorithm

As with previous works herein, maximum likelihood estimation (MLE) is the most common approach to estimate the parameters of a finite mixture model. The use of the EM is common for mixture models of this type, and, is consistently present across literature (Gutierrez *et al.*, 1995; McNicholas, 2016b). Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ be realizations

of a finite mixture of multivariate S_{U} Johnson distributions. The corresponding likelihood (L) and log-likelihood (l) is written as

$$L(\boldsymbol{\Theta}; \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = \prod_{i=1}^{n} \sum_{g=1}^{G} \pi_g f_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{y}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\omega}_g, \boldsymbol{\Lambda}_g),$$

$$l(\boldsymbol{\Theta}; \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = \sum_{i=1}^{n} \log \left(\sum_{g=1}^{G} \pi_g f_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{y}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\omega}_g, \boldsymbol{\Lambda}_g) \right). \tag{5.2}$$

The optimization of the log-likelihood (5.2) with respect to each of the parameters $\boldsymbol{\Theta}$ is difficult due to the non-linear nature of the function. The EM algorithm provides an iterative approach for estimating parameters. Beginning with the notion of missing data, let Z_{ig} be a latent variable indicating component memberships and z_{ig}, its realization. $Z_{ig} = 1$ if observation i belongs to component g, and $Z_{ig} = 0$ otherwise. The complete data likelihood (L_c) and log-likelihood (l_c) can then be derived as

$$L_c(\boldsymbol{\Theta}; \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n, \boldsymbol{z}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \left[\pi_g f_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{y}_i; \boldsymbol{\theta}) \right]^{z_{ig}}$$

$$= \prod_{i=1}^{n} \prod_{g=1}^{G} \left[\pi_g \frac{\exp\left\{ -\frac{1}{2} \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \boldsymbol{\mu}_g \right)^{\top} \boldsymbol{\Sigma}_g^{-1} \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \boldsymbol{\mu}_g \right) \right\}}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}_g|^{\frac{1}{2}} \prod_{j=1}^{p} \left(\delta_{jg} \sqrt{\left(\frac{y_{ij} - \omega_{jg}}{\delta_{jg}} \right)^2 + 1} \right)} \right]^{z_{ig}},$$

$$
\begin{aligned}
l_c(\boldsymbol{\Theta}; \boldsymbol{y}_1, \dots, \boldsymbol{y}_n, \boldsymbol{z}) = & \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\log(\pi_g) - \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\left(\boldsymbol{h}(\boldsymbol{y}_i;\boldsymbol{\vartheta}_g) - \boldsymbol{\mu}_g\right)^\top \boldsymbol{\Sigma}_g^{-1}\left(\boldsymbol{h}(\boldsymbol{y}_i;\boldsymbol{\vartheta}_g) - \boldsymbol{\mu}_g\right) \\
& - \frac{np}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\log(|\boldsymbol{\Sigma}_g|) - \sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{j=1}^{p} z_{ig}\log(\delta_{jg}) \\
& - \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{j=1}^{p} z_{ig}\log\left(\left(\frac{y_{ij}-\omega_{jg}}{\delta_{jg}}\right)^2 + 1\right).
\end{aligned} \tag{5.3}
$$

Given the true component memberships $\boldsymbol{z}$, optimize l_c with respect to $\boldsymbol{\Theta}$. This is what is known as the maximization or M-step. However, again, maximizing l_c is difficult due to the nature of non-linear transformations operating on $\boldsymbol{y}_i$. Here, the new approach differs from Zhu and Melnykov (2018) by reducing the number of parameters to optimize as follows. By profiling out nuisance parameters of (5.3), one reduces the number of parameters to optimize. The new objective function is derived in appendix A.1 referred to as the complete-profile log-likelihood. The new objective function is written as

$$
\begin{aligned}
l_{\text{cp}}(\boldsymbol{\vartheta}; \boldsymbol{y}_1, \dots, \boldsymbol{y}_n, \boldsymbol{z}) = & \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\log(\hat{\pi}_g) - \frac{np}{2}\log(2\pi) + \frac{np}{2}\sum_{g=1}^{G}\log(n_g) - \frac{np}{2} \\
& -\frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\text{logdet}\left(\sum_{i=1}^{n} z_{ig}\left(\boldsymbol{h}(\boldsymbol{y}_i;\boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right)\left(\boldsymbol{h}(\boldsymbol{y}_i;\boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right)^\top\right) \\
& - \sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{j=1}^{p} z_{ig}\log(\delta_{jg}) - \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{j=1}^{p} z_{ig}\log\left(\left(\frac{y_{ij}-\omega_{jg}}{\delta_{jg}}\right)^2 + 1\right).
\end{aligned} \tag{5.4}
$$

Notice that the function l_{cp} is free from parameters associated with the Gaussian space, and only concerns itself with the scale ($\boldsymbol{\omega}_g$), and shift ($\boldsymbol{\Lambda}_g$) of hyperbolic space

contained in $\boldsymbol{\vartheta}_g$, and $\hat{\boldsymbol{\mu}}_g$. Notice also that one can disregard the first row of (5.4) as terms are constant with respect to parameters being estimated. Optimization of (5.4) can be performed via any desired non-linear optimization scheme such as Nedler and Mead (1965) or Zhu *et al.* (1997). This methodology is implemented in Python 3.8.5 (Van Rossum and Drake Jr, 1995) where such algorithms are freely available within the `scipy` (Virtanen *et al.*, 2020) and `torch` (Paszke *et al.*, 2019) libraries. In practice, component memberships are often unknown as in the case of unsupervised learning (clustering) and therefore, we must estimate them. Given a set of parameter estimates $\hat{\boldsymbol{\Theta}}^{(t)}$ for some iteration t, compute the expectation of component memberships *a posteriori* such that

$$\mathbb{E}\left[Z_{ig}=1|\hat{\boldsymbol{\Theta}}^{(t)}\right]=\hat{z}_{ig}^{(t)}=\frac{\hat{\pi}_g^{(t)}\left(\frac{\exp\left\{-\frac{1}{2}\left(\boldsymbol{h}(\boldsymbol{y}_i;\hat{\boldsymbol{\vartheta}}_g^{(t)})-\hat{\boldsymbol{\mu}}_g^{(t)}\right)^\top\hat{\boldsymbol{\Sigma}}_g^{-1^{(t)}}\left(\boldsymbol{h}(\boldsymbol{y}_i;\hat{\boldsymbol{\vartheta}}_g^{(t)})-\hat{\boldsymbol{\mu}}_g^{(t)}\right)\right\}}{(2\pi)^{\frac{p}{2}}|\hat{\boldsymbol{\Sigma}}_g^{(t)}|^{\frac{1}{2}}\prod_{j=1}^{p}\left(\hat{\delta}_{jg}^{(t)}\sqrt{\left(\frac{y_{ij}-\hat{\omega}_{jg}^{(t)}}{\hat{\delta}_{jg}^{(t)}}\right)^2+1}\right)}\right)}{\sum_{k=1}^{G}\hat{\pi}_k^{(t)}\left(\frac{\exp\left\{-\frac{1}{2}\left(\boldsymbol{h}(\boldsymbol{y}_i;\hat{\boldsymbol{\vartheta}}_k^{(t)})-\hat{\boldsymbol{\mu}}_k^{(t)}\right)^\top\hat{\boldsymbol{\Sigma}}_k^{-1^{(t)}}\left(\boldsymbol{h}(\boldsymbol{y}_i;\hat{\boldsymbol{\vartheta}}_k^{(t)})-\hat{\boldsymbol{\mu}}_k^{(t)}\right)\right\}}{(2\pi)^{\frac{p}{2}}|\hat{\boldsymbol{\Sigma}}_k^{(t)}|^{\frac{1}{2}}\prod_{j=1}^{p}\left(\hat{\delta}_{jk}^{(t)}\sqrt{\left(\frac{y_{ij}-\hat{\omega}_{jk}^{(t)}}{\hat{\delta}_{jk}^{(t)}}\right)^2+1}\right)}\right)}. \tag{5.5}$$

Within the EM, (5.5) is what is referred to as the expectation or E-step. Once $\hat{z}_{ig}^{(t)}$ $\forall i,g$ have been calculated, substitute $\hat{z}_{ig}^{(t)}$'s into (5.4,A.1) acquiring $\hat{\boldsymbol{\Theta}}^{(t+1)}$ using a non-linear optimization scheme. The algorithm then iterates between E-step and M-step until convergence is reached. Initialization of the algorithm EM algorithm can be performed randomly or by k-means (k in this case meaning G, Lloyd, 1982). Convergence is assessed via Aiken's convergence criteria given in (2.2). The setting for tolerance is often nuanced and is generally set manually. As Karlis and Xekalaki (2003) elucidates, the EM is highly dependent on the choice of initial values. The EM

algorithm is prone to becoming trapped within the log-likelihood surface in areas away from the global maximum. To solve this issue, consider a mixed strategy (Karlis and Xekalaki, 2003). Starting from several different initializations, run a small number of iterations and record their log-likelihood progressions. From here, cull for the parameter set with the largest log-likelihood. Continuation from this parameter set with the EM algorithm until convergence is established using criterion (2.2). As demonstrated by Karlis and Xekalaki (2003), this approach for the EM avoids being stuck in a local maxima. Model performance, convergence and selection for the S_{U} Johnson is assessed in accordance with Section 2.2.

5.3 Imputation of Missing Data

In realistic scenarios, there is often some data that is missing. For some observation vector $\boldsymbol{y}_i$, a data point may not be recorded for entry y_{ij}. This results in a perforated dataset where some observations are not fully realized. In addition, the perforation (pattern of missing data) may or may not be deterministic (see Figure 1.1 of Little and Rubin, 2019). For the purpose of simplicity, assume the observed data has no relationship between the pattern of perforation, and, the observed values themselves. To elaborate, missing data entries are a random subset of the data, and there is no systematic phenomena going on that makes some data more likely to be missing than others. By this definition, data is considered to be missing completely at random (MCAR, Little and Rubin, 2019). To impute missing values, consider two different methods from the work of Di Zio *et al.* (2007). The first method is to impute missing values based on conditional mean imputation (CMI). The second method is to randomly sample values from a source distribution which is referred to as random draws

imputation (RDI).

Both methods for the multivariate S_{U} Johnson distribution are derived as follows. For a particular observation vector $\boldsymbol{y}_i$, there exists the potential to have up to $p-1$ missing entries. Let m correspond to the index of the missing entry for vector $\boldsymbol{y}_i$. Furthermore, let $\boldsymbol{d}_i$ correspond to the collection of indices which are non-missing for said vector $\boldsymbol{y}_i$. For example, if $\boldsymbol{y}_i = (y_{i1}, y_{i2}, \mathrm{NA}, y_{i4}, \ldots, y_{ip})$, then $m = 3$ and $\boldsymbol{d}_i = \{1, 2, 4, \ldots, p\}$. The vector $\boldsymbol{d}_i$ keeps track of which entries are non-missing for a particular observation vector. With this notation established, for a particular observation i, let $y_{i,m}$ denote the missing value, and $\boldsymbol{y}_{i,\boldsymbol{d}_i}$ denote the non-missing values respectively. The imputation step of the CMI method under the finite mixture model is given as

$$\mathbb{E}\left[y_{i,m}|\boldsymbol{y}_{i,\boldsymbol{d}_i}, \boldsymbol{z}_i, \boldsymbol{\Theta}\right] = \sum_{g=1}^{G} \boldsymbol{z}_{ig} \mathbb{E}\left[y_{i,m}|\boldsymbol{y}_{i,\boldsymbol{d}_i}, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\omega}_g, \boldsymbol{\Lambda}_g\right]. \tag{5.6}$$

For some component g, the conditional distribution of $y_{i,m}|\boldsymbol{y}_{i,\boldsymbol{d}_i}$ is a S_{U} Johnson distribution with parameters adjusted for conditioning on non-missing entries (see Appendix A.2 for full derivation). The expectation is of a closed form (see equation 37 of Johnson, 1949b, pg. 163) and written as

$$\mathbb{E}\left[y_{i,m}|\boldsymbol{y}_{i,\boldsymbol{d}_i}, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\omega}_g, \boldsymbol{\Lambda}_g\right] = \mathbb{E}\left[y_{i,m}|\boldsymbol{y}_{i,\boldsymbol{d}_i}, \tilde{\boldsymbol{\mu}}_g, \tilde{\sigma}_g^2, \omega_{mg}, \delta_{mg}\right] = \omega_{mg} - (\delta_{mg}) e^{\frac{\tilde{\sigma}_g^2}{2}} \sinh(-\tilde{\mu}).$$

The probability that observation i belongs to group g is calculated *a posteriori*. This poses issues in the presence of missing data as estimation of component memberships $\boldsymbol{z}_i$ require fully recorded observations. Furthermore, imputation of said missing data is calculated via (5.6) which again, poses issues since it relies on fully realized

non-missing observations. Wei *et al.* (2019) provides a solution where component memberships can be calculated by strictly via the non-missing entries $\boldsymbol{y}_{i,\boldsymbol{d}_i}$. The appropriate calculation for (5.5) is now adjusted with the corresponding parameters $\boldsymbol{\mu}_{\boldsymbol{d}_i,g}, \boldsymbol{\Sigma}_{\boldsymbol{d}_i\boldsymbol{d}_i,g}, \boldsymbol{\omega}_{\boldsymbol{d}_i,g}$, and $\boldsymbol{\Lambda}_{\boldsymbol{d}_i,\boldsymbol{d}_i,g}$. These parameters are associated with the non-missing entries $\boldsymbol{y}_{i,\boldsymbol{d}_i}$ and are used to attain $\hat{\boldsymbol{z}}_{ig}$ for (5.6). All other parameter estimates $\hat{\boldsymbol{\Theta}}$ can be attained by running the EM algorithm on a complete dataset with no missing entries. The imputation step of RDI for the finite mixtures of S_{U} model can be derived from the definition of an S_{U} distribution. Let $\hat{\boldsymbol{\Theta}}$ and $\hat{\boldsymbol{z}}_{ig}$ be estimates similarly attained as in CMI. Now generate a random draw as follows

$$\hat{z}_i \sim \mathcal{M}(\hat{\boldsymbol{z}}_{i1}, \ldots, \hat{\boldsymbol{z}}_{iG}), \tag{5.7}$$

$$\hat{x}_{ij} \sim \mathcal{N}(\tilde{\mu}_g, \tilde{\sigma}_g^2) \quad | \, \hat{z}_i = g, \tag{5.8}$$

$$\hat{y}_{ij} = \omega_{jg} + \delta_{jg} \sinh(\hat{x}_{ij}) \quad | \, \hat{z}_i = g. \tag{5.9}$$

The RDI can be broken down into three steps. (5.7) draws from a multinomial distribution the component membership for observation i based on the non-missing entries $\boldsymbol{y}_{i,\boldsymbol{d}_i}$. (5.8) draws from a univariate Gaussian distribution conditioned on non-missing entries. Finally (5.9) transforms from Gaussian space into the desired domain of the S_{U} Johnson to complete the imputation for the desired missing entry y_{ij}. Of the two methods, Di Zio *et al.* (2007) demonstrates that the RDI method has better performance when estimating Gaussian variance parameters $\boldsymbol{\Sigma}_g$. However, the CMI method has better performance for estimating $\boldsymbol{\mu}_g$. For the newly developed model, it is not clear whether one method will outperform the other for the S_{U} Johnson distribution. Specifically, it is uncertain whether CMI or RDI will be superior for

imputing missing values. To motivate the newly developed methodology, consider a series of investigations for geotechnical engineering data. These datum are at times, incomplete; as they are compiled from multiple sets of bivariate data sourced from different locations. When these bivariate datasets are consolidated, there are often missing measurement records within the general multivariate dataset.

5.4 Application

5.4.1 Shanghai Clay Dataset

The shanghai clay dataset denominated as `SH-CLAY/11/4051` consists of 4051 observations by 11 clay measurements, which are recorded across 50 borehole sites in Shanghai (Zhang *et al.*, 2020). In addition to the 11 measurements, the depth of the borehole per sample was also recorded. The presence of asymmetry and skewness within the dataset poses a unique challenge to model clay properties accordingly. As Zhang *et al.* (2020) elucidates, there is often insufficient site related data to estimate design parameters at the desired project location. As a result, the need for modelling such multivariate datasets contrives the development of appropriate statistical models. Zhou *et al.* (2022) investigates several candidate transformation models and establishes the S_{U} Johnson distribution as the superior model. However, these approaches do not consider an underlying heterogeneous structure. Upon a simple visual inspection of the dataset, it is evident that there are at least two sources of heterogeneity within `SH-CLAY/11/4051` (for details, see Figure 4 of Zhou *et al.*, 2022). Furthermore, given the model performance benchmarks related to normality, a clear presence of asymmetry is intrinsic to the data (see Table 5 of Zhou *et al.*, 2022).

All things considered, the work of Zhang *et al.* (2020) and Zhou *et al.* (2022) establishes a precedent for the use of S_{U} Johnson distributions within the context of this dataset. The dataset variates can be collected into two groups pertaining to the type of measurement outlined in Table 5.1. The Index group pertains to data measurements that are intrinsic soil properties, and, by definition are usually unit-less. These records are extracted from laboratory tests of borehole soil samples. The Mechanical variate group pertains to data attained from test-specific measurements such as the vane shear test (VST), and, the unconfined compression soil test (UCST). There is a significant amount of missing data within the dataset. Only 2 observations are fully complete which poses significant issues (see Table 1, column 2 of Zhang *et al.*, 2020).

Table 5.1: Description of the `SH-CLAY/11/4051` dataset and its attributes grouped by variate measurement type. Index variates are intrinsic soil properties and by definition have no units of measurement. Mechanical variates are records taken from the unconfined compression soil test (UCST), and vane shear test (VST).

Type	Attribute	Description
Index	Y_{LL}	Liquid limit
	Y_{PI}	Plasticity index
	Y_{LI}	Liquid index
	Y_{e}	Void ratio
Mechanical	Y_{K0}	At rest lateral pressure coefficient
	Y_{VE}	Vertical effective stress (kPa)
	Y_{SSUCST}	Shear strength of UCST (kPa)
	Y_{SUCST}	Sensitivity coefficient of UCST
	Y_{SSVST}	Shear strength of VST (kPa)
	Y_{SVST}	Sensitivity coefficient of VST
	Y_{Pen}	Penetration resistance (kPa)

Without loss of generality, consider only modelling a subset of the variates to demonstrate robustness. Strictly, only the Index group is modelled as there are

an adequate number of complete records to estimate $\boldsymbol{\Theta}$. Upon filtering out incomplete records, $n = 2066$ observations were extracted and organized as $\boldsymbol{\mathcal{Y}} = (Y_{LL}, Y_{PI}, Y_{LI}, Y_e)$. Table 5.2 reports the descriptive statistics of shanghai dataset. Note the difference in scale across variates. To prevent numerical issues where one variable may influence estimation over others, the dataset is standardized with the sample mean and deviation (Milligan and Cooper, 1988). Specifically, standardization prevents variables with larger scales from dominating how clusters are defined. There is considerable skewness for the marginal distribution of Y_{LI} according to Pearson's skewness statistic. Y_{LI} refers to the liquid index of soil which is a proxy for how much water content there is in a sample. Due to presence of skewness, Y_{LI} is used as a reference point for Figure 5.1 visualizing the structure of data.

Table 5.2: Descriptive statistics of `SH-CLAY/11/4051` dataset, skewness is calculate via Pearson's coefficient of skewness statistic.

Variate	Min	Median	Mean	Max	Skew	Std
Y_{LL}	26.300	41.250	40.350	58.700	-0.095	4.901
Y_{PI}	10.300	18.200	18.090	30.900	0.040	3.222
Y_{LI}	0.490	1.165	1.149	2.190	0.266	0.253
Y_e	0.770	1.209	1.223	1.863	0.117	0.181

Immediately one can see presence of heterogeneity across the bivariate plots of Figure 5.1. The orientations for (Y_{LI}, Y_{LL}) and (Y_{LI}, Y_{PI}) appear to be on the same axis which implies that those specific measurements are characterized by similar behaviour and orientation. Most of the data is quite concentrated in the (Y_{LI}, Y_e) plot which may result in greater cluster overlap during estimation. In addition, there are several data points that lie quite distant from the majority of samples such as in the lower right-hand corner of the (Y_{LI}, Y_{PI}) plot. The analogous data point can also be found in the lower right-hand corner of the (Y_{LI}, Y_{LL}) plot implying that again, the

pairs (Y_{LI}, Y_{LL}) and (Y_{LI}, Y_{PI}) exhibit similar behaviour and orientation.

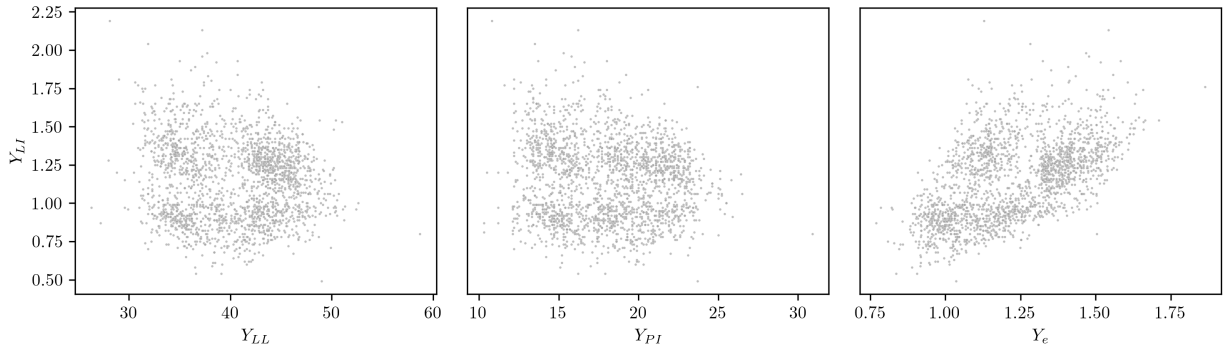


Figure 5.1: A series of bivariate scatter plots with Y_{LI} as a point of reference for Y_{LL}, Y_{PI}, Y_e. Shows data heterogeneity and asymmetric clusters across all domains for each Index variate.

To model heterogeneity and asymmetry of the data, consider the following. Assume that realizations $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$ emanate from a finite mixture model of up to G components. By fitting an appropriate finite mixture model, one can reclaim the component membership of each $\boldsymbol{y}$. Several types of initializations are considered for the S_{U} model. Initialization methods include k-means, random soft, and random hard. As a means of comparison to other methodologies, similar transformation models are also considered for the dataset. The work of Zhu and Melnykov (2018) through the `ManlyMix` package (Zhu and Melnykov, 2017) considers modelling skewness with Manly transforms. Another candidate to consider is a mixtures of t-distributions with power transforms through the `flowClust` package (Lo *et al.*, 2009). A total of 1000 runs with varying initializations are performed for each candidate model. Table 5.3 reports the top performing models via BIC. It is evident that a S_{U} Johnson mixture model outperforms all other candidates with a BIC of -8286.117 and $G = 6$. The second most performant model is a Manly mixture with a BIC of -8291.592 and

$G = 6$. Furthermore, all candidate models selected $G = 6$ components as their best fit; indicating some intrinsic consistency across different methodological approaches for this dataset. Similarly, comparable analyses can be found in work such as Gallaugher *et al.* (2020) where skewed approaches had similar performance in terms of classification, but differed in BIC.

Table 5.3: Model fits on complete data of Index measurements from the `SH-CLAY/11/4051` dataset. Grouped and organized by each candidate model, the top four fits out of 1000 runs are reported. The top performer is shown in bold.

Model	BIC	G
S_{U} Johnson	**−8286.117**	**6**
	−8313.977	5
	−8395.002	4
	−8421.113	7
`ManlyMix`	−8291.592	6
	−8387.434	7
	−8634.226	4
	−8668.603	5
`flowClust`	−8304.792	6
	−8318.946	7
	−8365.024	8
	−8453.637	5

The heterogeneous nature of the data yields a particularly interesting fit for an S_{U} Johnson mixture model. Figure 5.2 visualizes the component memberships assigned by the $G = 6$ S_{U} Johnson model. Here, we see that some components overlap more heavily than others. In addition, the orientation and location of components across variables within plots appear fairly consistent. Beginning with the four top-most components denoted in green, brown, purple and pink. The green/brown components are superimposed on one another with strong overlaps. The purple/pink component pair exhibits the exact same behaviour. However, the remaining components coloured

in light-blue/dark-green are fairly distinct from all others; not exhibiting such structure. Consider the purple/pink component pairing, both components display a similar shape visually within the data. The pink component is more concentrated than the purple, and does not contain any spurious points. In contrast, the purple counterpart contains several points that are quite distant from the general population. Additionally, the component shapes for purple/pink are quite similar visually, and, is especially evident when considering the (Y_{LI}, Y_e) plot. A similar pattern can be found for the brown/green pair. The green component seems to model the concentrated centre of the data, while the brown component accounts for the sparsity. Furthermore, all component shapes are visually asymmetric and do not exhibit ellipsoidal properties such as in the case of mixtures of multivariate Gaussian.

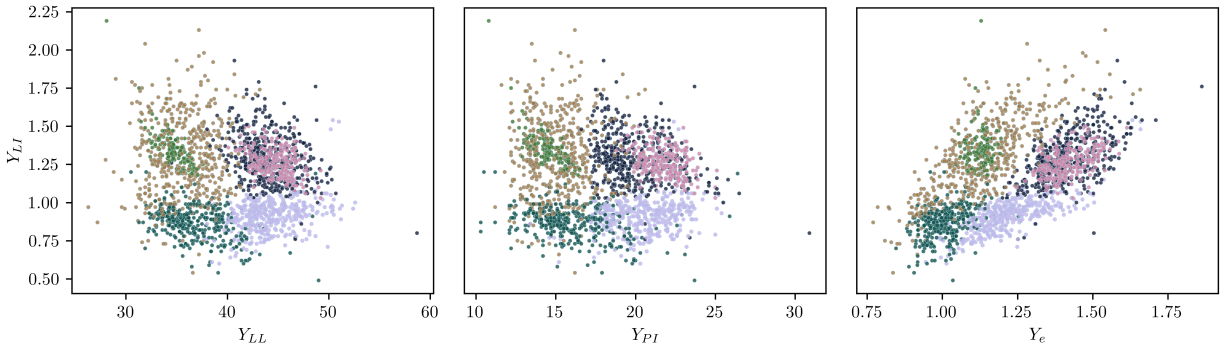


Figure 5.2: S_{U} Johnson model fit visualized on a series of bivariate scatter plots with Y_{LI} as a point of reference against Y_{LL}, Y_{PI}, Y_e. Memberships are denoted by color and assigned by hard classification.

The presence of heterogeneous soil properties within the dataset is not surprising. Shanghai local code for geotechnical site investigation defines different geological layers and sub-layers constituting Shanghai clay (see Figure 2 of Zhang *et al.*, 2020).

The nature of Shanghai soil data is consistent with the general theory of geotechnical poromechanics. Contemporary deterministic models such as Kebria *et al.* (2022) model freezing processes within a heterogeneous soil media consisting of silt, clay, sand, and gravel. All things considered, the analysis indicates viability of the S_{U} Johnson mixture model to accurately encapsulate soil phenomena. One caveat to consider, is that borehole samples were not taken uniformly across depth (Figure 3 of Zhang *et al.*, 2020). This sample bias may lead to underestimating the actual number of components and thus under-representing the heterogeneity of data. Nevertheless, the analysis of components demonstrates the flexibility of the S_{U} Johnson model to account for asymmetry, concentration, and spuriosity of data within a heterogeneous context.

5.4.2 Imputation of Missing Clay Data

There are a considerable amount of missing records within the `SH-CLAY/11/4051` dataset. Even when considering only Index measurements, there are 1985 observations with missing entries. Imputation methods are restricted to only measurements with at least one non-missing record. However, two observations are fully missing with regards to Index measurements. This implies that for this borehole extraction, no soil samples were sent to the lab to record Index measurements. However, other Mechanical measurements are available for these two observations. By the results of previous analyses, these two observations are not considered due to the focus on Index measurement types.

For the remaining $n = 1983$ samples, Index measurements are only partially available. For a given observation, there are records within the data where not all Index

measurements were recorded for a given observation i. Some obeservations do not contain a full set of Index measurements. There is a clear asymmetry in proportions of the type of missing data within the dataset. Table 5.4 shows a comprehensive overview of missing data patterns. Over 90% of missing data is strictly for the Y_{LL}, Y_{LI} variates. The variates Y_{LI}, Y_e are a far-second with 8.17% missing.

Table 5.4: Pattern of missing data for the Index variables of the `SH-CLAY/11/4051` dataset.

Missing Variate	Y_e	Y_{LI}, Y_e	Y_{LL}, Y_{LI}	Y_{LL}, Y_{LI}, Y_e	Y_{LL}, Y_{PI}, Y_{LI}
Proportion (%)	0.05	8.17	90.97	0.55	0.25
Count	1	162	1804	11	5

The imbalance within proportions of missing data patterns pose some difficulty in imputation since it violates the MCAR assumption. MCAR assumes that there is no underlying mechanism that makes one type of data more likely to be missing than others. In contrast, the dataset clearly indicates that liquid measurements are more likely to be missing. However, consider the following. According to Figure 5.1, there is a clear similarity in orientation/structure within the (Y_{LI}, Y_{LL}), and (Y_{LI}, Y_{PI}) pairs. The strength of imputation methods lie in leveraging already recorded entries, and, for most non-missing entries, Y_{PI} is indeed recorded. Despite the imbalanced nature of missing data patterns, it is assumed that this structure overcomes any limitations of the MCAR assumption for imputing liquid measurements.

Both methods of imputation are performed on the $n = 1983$ incomplete observations. Imputation is performed using the estimated parameters taken from the $G = 6$ S_{U} Johnson model within Section 5.4.1. Figure 5.3 displays the results of the conditional mean imputation. The imputed points coloured in blue are overlaid on the original dataset to visualize the similar structure of data. The CMI method is able

to capture the variation of points with small areas of concentration. This is evident in the (Y_{LL}, Y_{LI}) bivariate plot where imputed points are quite concentrated in the upper right-hand area. Despite the similarities between (Y_{LL}, Y_{LI}) and (Y_{PI}, Y_{LI}), the imputed points for (Y_{PI}, Y_{LI}) are not as concentrated. This concentration of imputed data for (Y_{LL}, Y_{LI}) is a direct result of the high proportion of missingness for said variate pair as previously reported in Table 5.4. Furthermore, there are some points in Figure 5.3 which are quite spurious in relation to the general structure of data. This type of dispersion is fairly consistent with the original dataset displayed in grey.

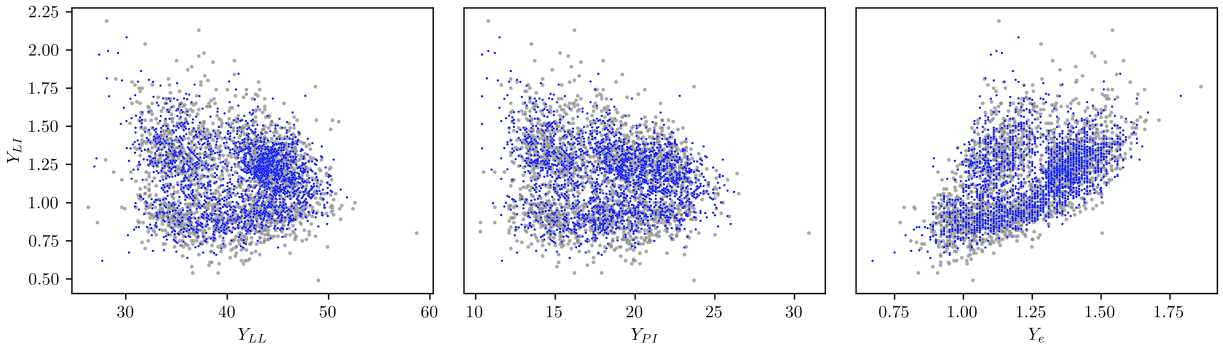


Figure 5.3: A series of bivariate scatter plots for original (grey) and CMI imputed points (blue) with Y_{LI} as a point of reference for Y_{LL}, Y_{PI}, Y_e. Imputed points (blue) capture the underlying structure of the original points (grey).

Overall, the CMI method visually captures the spirit of the data despite violation of the MCAR assumption. The RDI method is implemented in a similar fashion for the $n = 1983$ partially missing observations. Figure 5.4 visualizes the result where again, the imputed values (red) are overlaid on the original dataset (grey). From the visuals, the RDI method indeed captures the underlying structure of data but with a less dispersed imputation. Overall, the imputed points are less spurious than the CMI counterpart and impute closer to the overall structure. In addition, imputed

data appears less concentrated than the CMI counterpart.

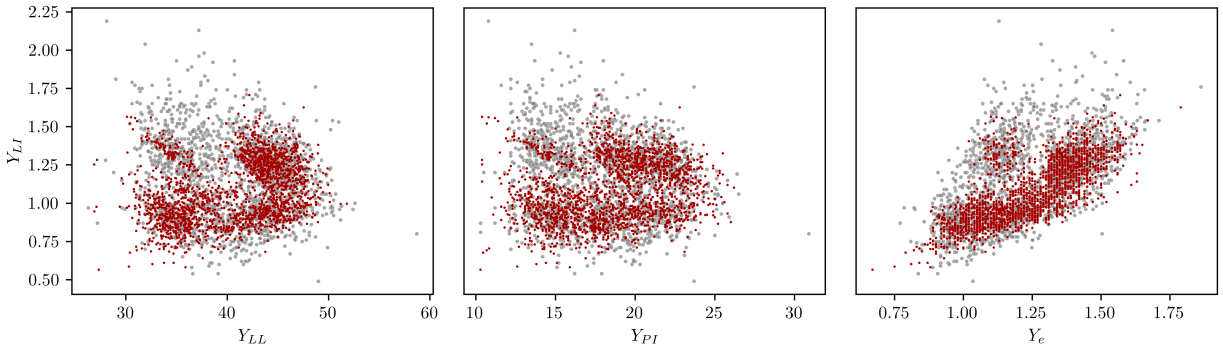


Figure 5.4: A series of bivariate scatter plots for original (grey) and RDI imputed points (red) with Y_{LI} as a point of reference for Y_{LL}, Y_{PI}, Y_e. Imputed points (red) capture the underlying structure of the original points (grey).

For both methods, the results appear to capture the essence of the original dataset. However, with no ground truth to compare against, it is difficult to discern whether one method is superior over another. Furthermore, the violation of the assumed MCAR perspective complicates a fair comparison as data are not randomly missing. As a result, a simulation study is developed to closely match existing data. Consider a MAR mechanism to both measure imputation performance, and, facilitate a fair comparison between methods.

5.5 Simulation Study

A simulation study is designed to assess imputation performance by creating a synthetic dataset. Naturally, the closest candidate for a ground-truth is the aforementioned $n = 2066$ fully complete records of Index measurements. To capture the same perforation mechanism present within the $n = 1983$ incomplete Index measurements, consider the following multinomial sample drawing process. First, randomly select

an observation from the complete Index dataset where each observation has an equal probability of being sampled. Next, sample from a multinomial distribution with probabilities equal to the row of proportions in Table 5.4. Based on category drawn, perforate the drawn observation based on the corresponding pattern of missing data. For example, if category 1 is drawn, perforate the void ratio Y_e for that observation leaving it missing. Perform the same process 1000 times and impute the newly perforated missing values. Finally, to measure performance, calculate the mean squared error (MSE) between the original observations, and, the newly imputed ones. The results for both imputation methods are aggregated and visualized in Figure 5.5 on a log scale.

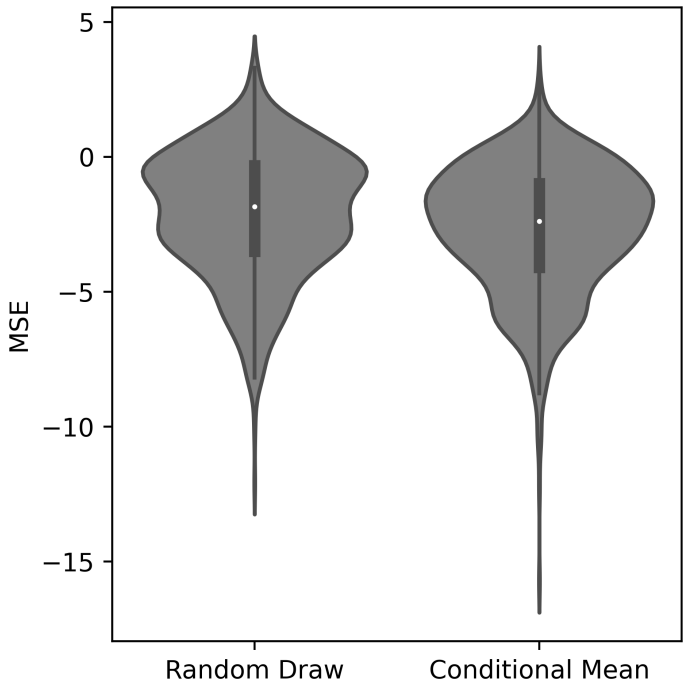


Figure 5.5: Violin plot of MSEs for CMI and RDI methods on a log scale. Results show that CMI has better performance overall even under worst-case scenarios (highest MSE).

Both the best-case (lowest MSE) and worst-case (highest MSE) show that CMI has better performance. For the general case, the mean and median MSEs are also lower for CMI than RDI. Overall, the results indicate that CMI is able to significantly outperform RDI. In addition, CMI intrinsically has lower computational overhead compared to stochastic methods such as RDI. From results herein, CMI is the clear choice for imputation overall.

5.6 Discussion

This chapter develops a multivariate finite mixture S_{U} Johnson model with the ability to handle missing data entries. Through an application in geotechnical soil analysis, the robustness of the S_{U} Johnson model demonstrates the ability to handle asymmetry and concentration within a heterogeneous context. The results from the application section show two pairs of components that overlap significantly. For each pair, one component captured the concentration while its counterpart captured dispersion. Consider the following perspective, in a contaminated model as in Punzo and McNicholas (2016), a cluster is defined through a mathematically tractable multivariate Gaussian distribution but with a superimposed inflated high-variance component. Through this perspective one may treat the pink/purple component pair to be of the same cluster. Naturally, this contrives an avenue for future work to consider a contaminated S_{U} Johnson model. Since the newly developed methodology is a component-wise transformation of a multivariate Gaussian model, one can naturally extend the contaminated Gaussian in similar fashion using the S_{U} system. Furthermore, the results of the simulation study show that of the two methods described in Di Zio *et al.* (2007), CMI is the clear choice. Another imputation method to consider

is a conditional median imputation. The median of an S_{U} Johnson distribution is easily calculated and has a closed form. In addition, median imputation would have even less computational overhead than the mean, constituting a natural extension of this imputation method. Furthermore, one could also model the missing data mechanism to relax the MCAR assumption to MAR. Nevertheless, visually both imputation methods capture the structure of data, and, can be used to impute missing values. In summary, the finite mixture S_{U} Johnson model proves itself to be highly robust for missing and asymmetric data.

CHAPTER 6

VISUAL ASSESSMENT OF MATRIX-VARIATE NORMALITY

In recent years, both the dimensionality and quantity of data have become increasingly large, leading to what is commonly known as the "big data phenomenon". In the case of longitudinal data, for example, it is becoming increasingly common to have repeated measurements on more than one characteristic for an individual, leading to a dataset where each observation can be recorded as a matrix. These multiple repeated measures over time, sometimes known as multivariate longitudinal data, are one example of three-way (matrix-variate) data.

Many approaches for analyzing two-way (multivariate) data are based on the multivariate normal distribution and much work has been done on assessing the normality of such data. Royston (1983) extend the univariate Shapiro-Wilk test for normality

(Shapiro and Wilk, 1965) to large samples of higher dimension, and Mudholkar *et al.* (1992) define a distribution for the Mahalanobis squared distance (MSD) under the assumption of multivariate normality. Moreover, visual methods such as the QQ plot (Easton and McCulloch, 1990) are quite common for assessing the assumption of multivariate normality.

Herein, these concepts are extended to three-way data by developing a visual approach for testing matrix-variate normality. Furthermore, previous work in this area is encapsulated within a framework for testing matrix-variate normality.

6.1 Distance-Distance Plot

A new *post hoc* method is proposed for visually assessing the matrix-variate structure of a dataset. Consider N $r \times c$ matrices $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$ such that each $\boldsymbol{X}_i$ is a realization of a matrix-variate random variable $\mathscr{X} \sim \mathcal{N}_{r\times c}(\boldsymbol{M}, \boldsymbol{V}, \boldsymbol{U})$. Recall the relationship between the matrix-variate normal and multivariate normal distributions in (2.4), and let $\boldsymbol{\mu} = \text{vec}(\boldsymbol{M})$ and $\boldsymbol{\Sigma} = \boldsymbol{V} \otimes \boldsymbol{U}$. Consider the estimates

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \text{vec}(\boldsymbol{X}_i) \qquad \text{and} \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N} \left\{\text{vec}(\boldsymbol{X}_i) - \hat{\boldsymbol{\mu}}\right\}\left\{\text{vec}(\boldsymbol{X}_i) - \hat{\boldsymbol{\mu}}\right\}^\top.$$

Now, calculate the MSD for each observation $\boldsymbol{X}_i$ in a given sample as follows:

$$\mathcal{D}(\boldsymbol{X}_i, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) = \left\{\text{vec}(\boldsymbol{X}_i) - \hat{\boldsymbol{\mu}}\right\}^\top \hat{\boldsymbol{\Sigma}}^{-1} \left\{\text{vec}(\boldsymbol{X}_i) - \hat{\boldsymbol{\mu}}\right\}. \tag{6.1}$$

We have that

$$\frac{N}{(N-1)^2}\mathcal{D}(\boldsymbol{X}_i, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) \sim \text{Beta}\left(\frac{rc}{2}, \frac{N-rc-1}{2}\right). \tag{6.2}$$

Moreover, the maximum likelihood estimates for $\boldsymbol{M}$, $\boldsymbol{U}$ and $\boldsymbol{V}$ are

$$\hat{\boldsymbol{M}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{X}_i, \tag{6.3}$$

$$\hat{\boldsymbol{U}} = \frac{1}{cN}\sum_{i=1}^{N} \big(\boldsymbol{X}_i - \hat{\boldsymbol{M}}\big)\hat{\boldsymbol{V}}^{-1}\big(\boldsymbol{X}_i - \hat{\boldsymbol{M}}\big)^\top, \tag{6.4}$$

$$\hat{\boldsymbol{V}} = \frac{1}{rN}\sum_{i=1}^{N} \big(\boldsymbol{X}_i - \hat{\boldsymbol{M}}\big)^\top\hat{\boldsymbol{U}}^{-1}\big(\boldsymbol{X}_i - \hat{\boldsymbol{M}}\big). \tag{6.5}$$

This estimation procedure alternates between estimating $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{V}}$ until convergence is established in the likelihood (see Dutilleul, 1999, for details). Now, let

$$\mathcal{D}_M(\boldsymbol{X}_i, \boldsymbol{M}, \boldsymbol{V}, \boldsymbol{U}) = \text{tr}\left\{\boldsymbol{U}^{-1}(\boldsymbol{X}_i - \boldsymbol{M})\boldsymbol{V}^{-1}(\boldsymbol{X}_i - \boldsymbol{M})^\top\right\}. \tag{6.6}$$

This quantity in (6.6) can be viewed as the matrix-variate version of the MSD. As a result, this terminology is adopted going forward when referencing this quantity. Given preceding notation, consider the following lemma.

Lemma 6.1 If a Kronecker product structure exists for $\boldsymbol{\Sigma}$, then

$$\mathcal{D}(\boldsymbol{X}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{D}_M(\boldsymbol{X}_i, \boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V}), \tag{6.7}$$

$$\mathcal{D}_M(\boldsymbol{X}_i, \hat{\boldsymbol{M}}, \hat{\boldsymbol{U}}, \hat{\boldsymbol{V}}) \xrightarrow{P} \mathcal{D}_M(\boldsymbol{X}_i, \boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V}), \tag{6.8}$$

$$\mathcal{D}(\boldsymbol{X}_i, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) \xrightarrow{P} \mathcal{D}(\boldsymbol{X}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{6.9}$$

where $\xrightarrow{P}$ denotes convergence in probability.

Proof. Result (6.7) is trivial and follows directly from (2.4)—for completeness, details are given in Appendix B.1. Now, a proof is given for (6.8) as follows. Note that

$$\hat{\boldsymbol{M}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{X}_i$$

is the MLE for the mean matrix. Because the matrix-variate normal distribution is part of the exponential family (Gupta and Nagar, 1999), all MLEs exist and are consistent (DasGupta, 2008). Therefore, $\hat{\boldsymbol{M}} \xrightarrow{P} \boldsymbol{M}$. As mentioned previously, the estimates of the scale matrices are unique only up to a strictly positive multiplicative constant; however, their Kronecker product $\boldsymbol{V} \otimes \boldsymbol{U} =: \boldsymbol{\Sigma}$ is unique. Therefore

$$\hat{\boldsymbol{V}} \otimes \hat{\boldsymbol{U}} \xrightarrow{P} \boldsymbol{V} \otimes \boldsymbol{U} = \boldsymbol{\Sigma}.$$

From these two results, and the continuous mapping theorem (stated as Theorem B.2), we have

$$\mathcal{D}_M(\boldsymbol{X}_i, \hat{\boldsymbol{M}}, \hat{\boldsymbol{U}}, \hat{\boldsymbol{V}}) \xrightarrow{P} \mathcal{D}_M(\boldsymbol{X}_i, \boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V}).$$

Proceeding to the proof of (6.9), note that the multivariate normal distribution is a member of the exponential family (Gupta and Nagar, 1999). Therefore, the unbiased estimates $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ converge in probability to the true parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively. From the continuous mapping theorem, it follows that

$$\{\text{vec}(\boldsymbol{X}_i) - \hat{\boldsymbol{\mu}}\}^\top \hat{\boldsymbol{\Sigma}}^{-1} \{\text{vec}(\boldsymbol{X}_i) - \hat{\boldsymbol{\mu}}\} \xrightarrow{P} \{\text{vec}(\boldsymbol{X}_i) - \boldsymbol{\mu}\}^\top \boldsymbol{\Sigma}^{-1} \{\text{vec}(\boldsymbol{X}_i) - \boldsymbol{\mu}\},$$

i.e.,

$$\mathcal{D}(\boldsymbol{X}_i, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) \xrightarrow{P} \mathcal{D}(\boldsymbol{X}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

□

From of the results in Lemma 6.1, it seems useful to visualize matrix-variate normality by comparing the estimated MSDs. Consider a plot of $\mathcal{D}$ versus $\mathcal{D}_M$, which is denoted henceforth as the distance-distance (DD) plot. The DD plot is a scatter plot of the Mahalanobis distances using the estimated parameters from the multivariate and matrix-variate normal distributions, respectively. As a clarifying visual measure, the MSDs are standardized by the same scaling factor for all $\mathcal{D}_M$ and $\mathcal{D}$ alike. By scaling all MSDs with the single calculated maximum distance of all $\mathcal{D}_M$ and $\mathcal{D}$, both scales now range from 0 to 1. Figure 6.1 illustrates the visual approach to determining the matrix-variate normal structure. On the left, we have the DD plot for a matrix-variate normal structure with a red line at $\mathcal{D} = \mathcal{D}_M$ for reference. For convenience, the parameters used to generate Figure 6.1 are listed in Appendix B.3.1. Note that the distances lie roughly along the line with little variability between the multivariate and matrix-variate MSDs. On the right side, however, we have that the MSDs exhibit more variability and do not lie along the

reference line—this is because the data were simulated from a strictly multivariate normal distribution, i.e., without a Kronecker product covariance structure. Interpretation of these plots are as follows. The closer the points fall to the red reference line, the greater the evidence that these data emanate from a matrix-variate normal distribution. The plot on the left shows greater evidence that a matrix-variate assumption is reasonable because both distances $\mathcal{D}_M$ and $\mathcal{D}$ must converge to each other asymptotically. However, the plot on the right demonstrates asymmetry along this line and suggests that a matrix-variate assumption is unreasonable.

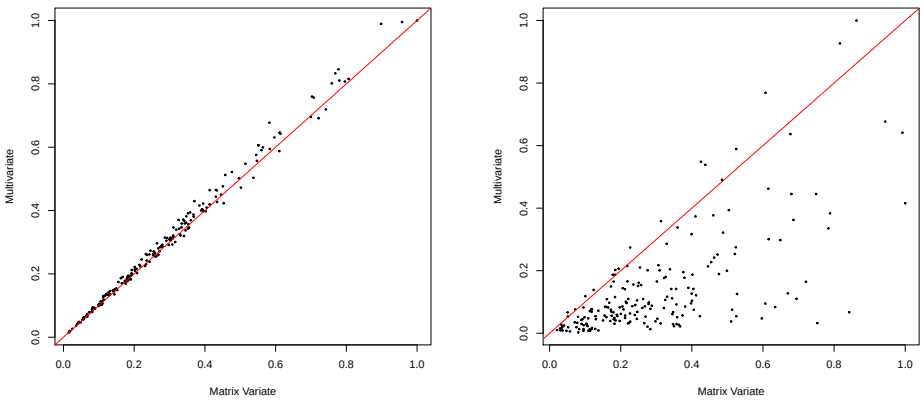


Figure 6.1: DD plots for simulated data ($N = 200, r = 2, c = 2$) with randomly chosen mean and variance parameters, indicating the presence (left) and absence (right) of a matrix-variate normal structure, i.e., of a Kronecker product covariance structure in the multivariate case.

6.1.1 Testing for Matrix-Variate Normality

In addition to visually assessing matrix-variate normality via the DD plots, it is also possible to test for matrix-variate normality by combining existing methods. This test requires two phases. In the first phase, multivariate normality must be established using the methods described in Section 2.3.1. Once multivariate normality has been established, matrix-variate normality can then be considered. Because of the relationship between the multivariate and matrix-variate normal distributions, the second phase of the testing procedure is equivalent to testing for a Kronecker-structured covariance matrix for the vectorized data. More succinctly, one desires to perform a hypothesis test with the following hypotheses:

$$H_0 : \text{Kronecker structure is present (i.e., } \boldsymbol{\Sigma} = \boldsymbol{V} \otimes \boldsymbol{U}).$$
$$H_a : \text{Kronecker structure is not present (i.e., } \boldsymbol{\Sigma} \neq \boldsymbol{V} \otimes \boldsymbol{U}).$$

Hypothesis tests for assessing Kronecker structure in a multivariate normal setting are well represented in literature. Srivastava *et al.* (2008) formulate some of these tests and perform simulation studies which show good performance overall. Lu and Zimmerman (2005) and Naik and Rao (2001) show similar results using likelihood ratio tests (LRTs) for Kronecker structure. The likelihood ratio test for Kronecker structure is outlined as follows. Let $\boldsymbol{X}_1, \ldots \boldsymbol{X}_n$ be iid normally distributed $r \times c$ matrices. If the Kronecker structure holds, then $\text{vec}(\boldsymbol{X}_i)$ is normally distributed under the covariance structure $\boldsymbol{\Sigma} = \boldsymbol{V} \otimes \boldsymbol{U}$. However, in the general case where the Kronecker structure may not hold, the covariance $\boldsymbol{\Sigma}$ is unconstrained for $\text{vec}(\boldsymbol{X}_i)$.

Therefore, the corresponding LRT statistic is given as

$$\hat{\vartheta} = 2[l\big(\boldsymbol{X}; \hat{\boldsymbol{M}}, \hat{\boldsymbol{U}}, \hat{\boldsymbol{V}}\big) - l\big(\mathrm{vec}(\boldsymbol{X}); \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\big)],$$

where l is the log-likelihood function. From well-known asymptotic results, for large N, $\hat{\vartheta} \sim \chi^2_{\mathrm{df}}$, where

$$\mathrm{df} = \frac{p(p+1)}{2} - \frac{r(r+1)}{2} - \frac{c(c+1)}{2}.$$

The null hypothesis is rejected at a level α if

$$\hat{\vartheta} > \chi^2_{\mathrm{df}[1-\alpha]},$$

where $\chi^2_{\mathrm{df}[1-\alpha]}$ is the appropriate critical value.

It should be noted that if the null hypothesis is rejected in either phase 1 or 2 of the testing procedure, then the assumption of matrix-variate normality is rejected. However, matrix-variate normality may still be a plausible assumption if multivariate normality holds and the DD plot indicates possible matrix-variate normality. For example, Figure 6.2 shows a DD plot for a case where the null hypothesis in the second phase of the testing procedure is rejected (at the 5% significance level), even though the data is in fact matrix-variate normal. From the DD plot one would conclude, and rightly so, that matrix-variate normality is indeed plausible. Therefore, in addition to the already developed tests, the proposed DD plots should also be taken into consideration when assessing matrix-variate normality as they may catch cases of Type I error in the aforementioned tests of hypotheses.

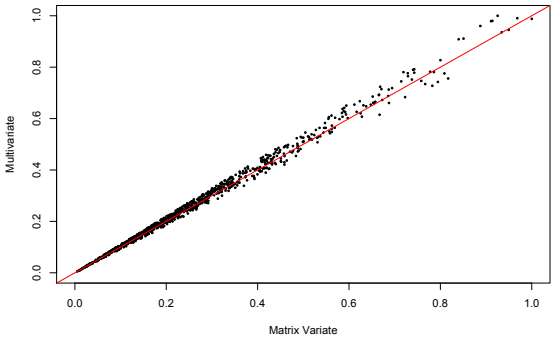


Figure 6.2: DD plot for a simulated matrix-variate normal dataset for which the second phase of the matrix-variate normal testing procedure determined there was no Kronecker structure ($\hat{p} = 0.01502$, $N = 1000$, $r = c = 2$). However, the DD plot shows that distances tend to follow the reference line contriving evidence that matrix-variate normality may hold.

6.2 Simulation Study

A simulation study is performed to display the efficacy of the proposed DD plots for assessing matrix-variate normality. Consider the two different studies as follows. For the first study, the sample size is held constant but the dimensions vary. This is to show the effect of dimension on DD plots, and particularly the interpretability of results. In the second study, the matrix dimensions are held constant but the sample size is varied. This is to show the effect of sample size on DD plots. The methodology was implemented in the Julia programming language (Bezanson *et al.*, 2017) and is available within the `MatrixVariate.jl` package (Počuča *et al.*, 2019). Let us first investigate the effect of dimensionality for assessing matrix-variate normality. The number of observations is set at $N = 1000$ while dimension takes the values $p \in \{4, 100\}$. Note that square matrices are used, i.e., $p = 4$ corresponds to 2×2 matrices and $p = 100$ corresponds to 10×10 matrices. Figure B.1 shows the

DD plots for each case, with each row having a different dimension p. Figures B.1a and B.1c show the DD plots for data that are matrix-variate normal, and Figures B.1b and B.1d correspond to data that are strictly multivariate normal and not matrix-variate normal. In Figures B.1a and B.1c, the matrix-variate and multivariate MSDs roughly coincide with one another and the variability about the reference line increases with dimensionality. In contrast, Figures B.1b and B.1d show highly variable and random MSDs. As the data for the plots on the right are strictly multivariate normal and not matrix-variate normal, the MSDs should not and do not coincide with one another. Finally, if the sample size is not sufficiently large to account for higher dimensions, the DD plot becomes uninterpretable due to the estimates not being reliable.

The second simulation varies the sample size while keeping the dimension constant. In these simulations, the dimension is set as $p = 100$ for the generated random vectors (10×10 matrices). Figure B.2 displays an array of DD plots for matrix normal and multivariate normal datasets when $p = 100$ and $N \in \{500, 2000\}$. Similar to the first investigation, Figures B.2a and B.2c represent datasets which are matrix-variate normal, and Figures B.2b and B.2d represent datasets which are multivariate normal but not matrix-variate normal. The plots demonstrate that the MSDs from matrix-variate normal data follow the reference line with some random variability, which drastically reduces as the sample size increases. When data are strictly multivariate normal and the matrix-variate structure is absent: the variability is large, the distances are skewed, and the MSDs diverge from the reference line. Note that Figure B.2a displays the effect of high dimension when the sample size is insufficient for estimating parameters. One can see that the

distribution of distances is slightly offset from the reference line, but less offset than in Figure B.2b. As the sample size is increased from 500 to 2000, this offset is corrected, and the distribution of distances indeed follows along the reference line. This indicates that the DD plot is highly consistent with asymptotic results, and, works for an increasing sample size where dimensionality is kept constant.

6.3 Application

The MNIST dataset is an image database of handwritten digits from United States Census Bureau employees and high school students. Each image is represented as a 28×28 matrix with entries corresponding to the grayscale intensity of each pixel (LeCun *et al.*, 1998). MNIST is considered as the quintessential baseline dataset for image classification problems. Benchmarks.AI (2022) records the top ranking methodologies and their respective classification performance. Most methods place significance on supervised learning methodologies, leaving a general dearth of unsupervised and semi-supervised statistically interpretable approaches. In model-based classification, work such as Gallaugher and McNicholas (2020) show that skewed models outperform their Gaussian counterparts in a semi-supervised approach to the MNIST image classification problem. As a result, there is a need to assess matrix-variate normality of the MNIST data to support evidence wherein a matrix-variate normal model is not appropriate. Consider the handwritten digits 1 and 7, where sample sizes are 6742 and 6265, respectively. The data $(r = 28, c = 28)$ were preprocessed in accordance with Section 5 of Gallaugher and McNicholas (2018). Figure 6.3 illustrates the DD plots for digits 1 and 7. Both DD plots indicate no evidence of matrix-variate normality. Furthermore, multivariate

normality fails to hold for both digits according to Korkmaz *et al.* (2014) with p-values less than 0.05. The results shown in this work show presence of skewness or other violations of matrix-variate normality, which is in agreement with previously attained performance of skewed matrix-variate distributions on the MNIST digits (see Table 5 of Gallaugher and McNicholas, 2020). In summary, for this dataset, the DD plot indicates that matrix-variate normality is not plausible.

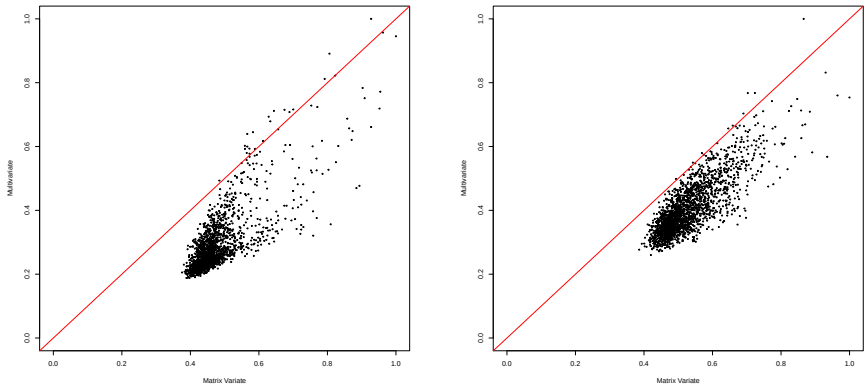


Figure 6.3: DD plots for MNIST digit 1 (left panel) and digit 7 (right panel) indicates lack of presence of a matrix-variate normal structure. In addition, tests of multivariate normality according to Korkmaz *et al.* (2014) indicate no presence of multivariate normality with p-values of $1.164E^{-4}$ and $3.243E^{-6}$, respectively.

6.4 Summary

A framework for assessing matrix-variate normality, both visually and using a statistical test, has been introduced. The new graphical technique for assessing matrix-variate normality, called the DD plot, is based on comparing Mahalanobis squared differences. The DD plot was shown to be effective for assessing

matrix-variate normality for various dimensions and sample sizes. In addition, a two-phase testing procedure was discussed for testing matrix-variate normality by combining existing tests. However, as illustrated by means of an example, the testing procedure should be used in conjunction with the proposed DD plot when assessing matrix-variate normality. The DD plots, along with the two-phase testing procedure, constitute a powerful combination for assessing matrix-variate normality. Future work will consider developing a method for calculating confidence intervals for the MSDs. A good candidate distribution to consider is a bivariate chi-square distribution (see Gunst and Webster, 1973). Another avenue for future work is to consider this approach in the context of skewed matrix-variate data, as real datasets are quite often asymmetric and/or skewed. Finally, one may extend this approach to the realm of tensor-variate data, i.e, d-way data for $d > 3$ as in the case of accelerometer data (Tait *et al.*, 2020; Gallaugher *et al.*, 2021a).

CHAPTER 7

FUTURE DIRECTIONS AND EXTENSIONS

This monograph composes four innovative approaches for dealing with highly complex data. Chapters 3-5 compose of methodologies aimed to handle both skewed and/or missing data within a heterogeneous context. The final chapter deviates; considering data of higher order.

Future developments aim for extending current methods into the area of matrix-variate distributions. The work of Gallaugher *et al.* (2021a) provides a starting point for delving into asymmetric matrix variate distributions. These models have shown great promise in modelling heterogeneity in images, and other higher-order data. The imputation methods proposed herein can be combined with higher-order models to impute missing data. In addition, the order of data is not

strictly limited to $d < 2$. Tait *et al.* (2020) and Gallaugher *et al.* (2021a) consider modelling both symmetric and skewed with a tensor variate extension. Such models can benefit from optimization techniques as defined in Chapter 4.

Another paradigm to consider is that of dimensionality reduction. Variable selection techniques are often limited in scope to symmetric distributions. The work of Neal (2022) elucidates a compelling framework for selecting variables when the distribution of data is both skewed and heterogeneous. Such methodologies consider first modelling data through an appropriate asymmetric finite mixture model, then, removing variables which are deemed ineffective by some criterion. A matrix-variate or higher-order extension can be derived in a similar fashion using the same principles as in Neal (2022).

The flexibility of semi-heavy tailed distributions are robust against outliers but are not immune. Another avenue of future work to consider is an extension to Clark and McNicholas (2019). The OCLUST algorithm uses subset log-likelihoods to trim outliers in Gaussian mixture models, and is available in an R package (Clark and McNicholas, 2022). Replacing a Gaussian assumption may yield better results and contrives some interest.

As a final note, it is sensible to acknowledge the limitations of methodologies within this monograph. The dimensionality of data poses several issues numerically when inverting matrices, calculating quadratic forms, and dealing with Bessel functions. Great care must be taken into account when dealing with such issues as they may influence performance of model estimation. The use of high-performance programming languages is absolutely essential to remedy such issues. As technical debt accumulates, the use of more complex models will dissipate unless there is

some remedial, counteracting system. Any work, any method, which does not consider the performance or efficiency of their program; will undoubtedly be severely limited in its adoption as datasets become increasingly complex in the future.

BIBLIOGRAPHY

Aitken, A. (1926). A series formula for the roots of algebraic and transcendental equations. *Proceedings of the Royal Society of Edinburgh*, **45**(1), 14–22.

Alefeld, G. (1981). On the convergence of halley's method. *The American Mathematical Monthly*, **88**(7), 530–536.

Alpu, Ö. and Yuksek, D. (2016). Comparison of some multivariate normality tests: A simulation study. *International Journal of Advanced and Applied Sciences*, **3**(12), 73–85.

Amos, D. E. (1974). Computation of modified Bessel functions and their ratios. *Mathematics of Computation*, **28**(125), 239–251.

Anderlucci, L. and Viroli, C. (2015). Covariance pattern mixture models for the analysis of multivariate heterogeneous longitudinal data. *Ann. Appl. Stat.*, **9**(2), 777–800.

Andrews, J. L., Wickins, J. R., Boers, N. M., and McNicholas, P. D. (2018). teigen: An R package for model-based clustering and classification via the multivariate t distribution. *Journal of Statistical Software*, **83**(7), 1–32.

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

Banihashemi, S. (2019). Portfolio management by normal mean-variance mixture distributions. In *2019 3rd International Conference on Data Science and Business Analytics (ICDSBA)*, pages 218–222. IEEE.

Baricz, Á. (2010). Turán type inequalities for some probability density functions. *Studia scientiarum mathematicarum Hungarica*, **47**(2), 175–189.

Barndorff-Nielsen, O. and Halgreen, C. (1977). Infinite divisibility of the hyperbolic and generalized inverse Gaussian distributions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, **38**(4), 309–311.

Barndorff-Nielsen, O., Kent, J., and Sørensen, M. (1982). Normal variance-mean mixtures and z distributions. *International Statistical Review/Revue Internationale de Statistique*, pages 145–159.

Barndorff-Nielsen, O. E. (1997). Processes of normal inverse Gaussian type. *Finance and stochastics*, **2**(1), 41–68.

Benchmarks.AI (2022). Mnist: Classify handwritten digits. Directory of AI Benchmarks, Sanford NC.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, **59**(1), 65–98.

Birge, J. R. and Chavez-Bedoya, L. (2021). Portfolio optimization under the generalized hyperbolic distribution: optimal allocation, performance and tail behavior. *Quantitative Finance*, **21**(2), 199–219.

Borak, S., Misiorek, A., and Weron, R. (2011). Models for heavy-tailed asset returns. In *Statistical tools for finance and insurance*, pages 21–55. Springer.

Box, G. E. and Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, **26**(2), 211–243.

Browne, R. P. and McNicholas, P. D. (2013). *mixture: Mixture Models for Clustering and Classification*. R package version 1.0.

Browne, R. P. and McNicholas, P. D. (2014). Estimating common principal components in high dimensions. *Advances in Data Analysis and Classification*, **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.

Browne, R. P., McNicholas, P. D., and Findlay, C. J. (2022). A partial em algorithm for model-based clustering with highly diverse missing data patterns. *Stat*, **11**(1), e437.

Burbidge, J. B., Magee, L., and Robb, A. L. (1988). Alternative transformations to handle extreme values of the dependent variable. *Journal of the American Statistical Association*, **83**(401), 123–127.

Campbell, J. (1980). On Temme's algorithm for the modified Bessel function of the third kind. *ACM Transactions on Mathematical Software (TOMS)*, **6**(4), 581–586.

Campbell, N. and Mahon, R. (1974). A multivariate study of variation in two species of rock crab of the genus leptograpsus. *Australian Journal of Zoology*, **22**(3), 417–425.

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

Celeux, G., Chauveau, D., and Diebolt, J. (1996). Stochastic versions of the em algorithm: an experimental study in the mixture case. *Journal of statistical computation and simulation*, **55**(4), 287–314.

Clark, K. M. and McNicholas, P. D. (2019). Using subset log-likelihoods to trim outliers in gaussian mixture models. *arXiv preprint arXiv:1907.01136*.

Clark, K. M. and McNicholas, P. D. (2022). *oclust: Gaussian Model-Based Clustering with Outliers*. R package version 0.2.0.

Dagpunar, J. S. (2007). *Simulation and Monte Carlo: With applications in finance and MCMC*. John Wiley & Sons.

DasGupta, A. (2008). *Asymptotic theory of statistics and probability*. Springer Science & Business Media.

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.

Dennis, S. B. (2009). Matrix Mathematics: Theory, Facts and Formulas. *Princeton University Press, Princeton, NJ*, **42**, 1139.

Di Zio, M., Guarnera, U., and Luzi, O. (2007). Imputation through finite Gaussian mixture models. *Computational Statistics & Data Analysis*, **51**(11), 5305–5316.

Diebolt, J. and Ip, E. H. (1996). Stochastic em: method and application. In *Markov chain Monte Carlo in practice*, pages 259–273. Springer.

Drucker, P. F. (2006). *Classic Drucker: essential wisdom of Peter Drucker from the pages of Harvard Business Review*. Harvard Business Press, Boston.

Dun, Y. and Kong, Y. (2022). Efficient Johnson-SB mixture model for segmentation of ct liver image. *Journal of Healthcare Engineering*, **2022**.

Dutilleul, P. (1999). The MLE algorithm for the matrix normal distribution. *Journal of Statistical Computation and Simulation*, **64**(2), 105–123.

Easton, G. S. and McCulloch, R. E. (1990). A multivariate generalization of quantile-quantile plots. *Journal of the American Statistical Association*, **85**(410), 376–386.

Eddelbuettel, D. and Sanderson, C. (2014). RcppArmadillo: Accelerating R with high-performance C++ linear algebra. *Computational Statistics & Data Analysis*, **71**, 1054–1063.

Eddelbuettel, D., François, R., Allaire, J., Ushey, K., Kou, Q., Russel, N., Chambers, J., and Bates, D. (2011). Rcpp: Seamless r and c++ integration. *Journal of statistical software*, **40**(8), 1–18.

Edgeworth, F. Y. (1898). On the representation of statistics by mathematical formulae (Part I.). *Journal of the Royal Statistical Society*, pages 670–700.

Finak, G., Bashashati, A., Brinkman, R., and Gottardo, R. (2009). Merging mixture components for cell population identification in flow cytometry. *Advances in bioinformatics*, **2009**.

Fraley, C., Raftery, A. E., and Wehrens, R. (2005). *mclust: Model-based Cluster Analysis*. R package version 2.1-11.

Fraley, C., Raftery, A. E., Murphy, T. B., and Scrucca, L. (2012). mclust version 4 for R: normal mixture modeling for model-based clustering, classification, and density estimation. Technical report, Technical report.

Franczak, B. C., Browne, R. P., and McNicholas, P. D. (2014). Mixtures of shifted asymmetriclaplace distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(6), 1149–1157.

Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Alken, P., Booth, M., and Rossi, F. (2019). *GNU scientific library*.

Gallaugher, M. P. and McNicholas, P. D. (2019). Three skewed matrix variate distributions. *Statistics & Probability Letters*, **145**, 103–109.

Gallaugher, M. P. and McNicholas, P. D. (2020). Mixtures of skewed matrix variate bilinear factor analyzers. *Advances in data analysis and classification*, **14**(2), 415–434.

Gallaugher, M. P., McNicholas, P. D., Melnykov, V., and Zhu, X. (2020). Skewed

distributions or transformations? modelling skewness for a cluster analysis. *arXiv preprint arXiv:2011.09152*.

Gallaugher, M. P., Tait, P. A., and McNicholas, P. D. (2021a). Four skewed tensor distributions. *arXiv preprint arXiv:2106.08984*.

Gallaugher, M. P., Tomarchio, S. D., McNicholas, P. D., and Punzo, A. (2021b). Multivariate cluster weighted models using skewed distributions. *Advances in Data Analysis and Classification*, (In Press: https://doi.org/10.1007/s11634-021-00480-5).

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

Gallaugher, M. P. B., Tomarchio, S. D., McNicholas, P. D., and Punzo, A. (2022). Model-based clustering via skewed matrix-variate cluster-weighted models. *Journal of Statistical Computation and Simulation*, **92**(13), 2645–2666.

Ghahramani, Z., Hinton, G. E., *et al.* (1996). The EM algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto.

Gibson, J. (2017). The Julia programming language: the future of scientific computing. In *APS Division of Fluid Dynamics Meeting Abstracts*, pages L39–011.

Gil, A., Segura, J., and Temme, N. M. (2002). Evaluation of the modified Bessel function of the third kind of imaginary orders. *Journal of Computational physics*, **175**(2), 398–411.

Gnanadesikan, R. and Kettenring, J. R. (1972). Robust estimates, residuals, and outlier detection with multiresponse data. *Biometrics*, **28**(1), 81–124.

Gunst, R. F. and Webster, J. T. (1973). Density functions of the bivariate chi-square distribution. *Journal of statistical computation and simulation*, **2**(3), 275–288.

Gupta, A. (2003). Multivariate skew t-distribution. *Statistics: A Journal of Theoretical and Applied Statistics*, **37**(4), 359–363.

Gupta, A. and Nagar, D. (1999). *Matrix Variate Distributions*. Monographs and Surveys in Pure and Applied Mathematics. Taylor & Francis.

Gupta, S. and Kapoor, V. (2020). *Fundamentals of mathematical statistics*. Sultan Chand & Sons.

Gutierrez, R. G., Carroll, R. J., Wang, N., Lee, G.-H., and Taylor, B. H. (1995). Analysis of tomato root initiation using a normal mixture distribution. *Biometrics*, pages 1461–1468.

Hardin, J. and Rocke, D. M. (2005). The distribution of robust distances. *Journal of Computational and Graphical Statistics*, **14**(4), 928–946.

Henze, N. and Zirkler, B. (1990). A class of invariant consistent tests for multivariate normality. *Communications in Statistics – Theory and Methods*, **19**(10), 3595–3617.

Hintze, J. L. and Nelson, R. D. (1998). Violin plots: a box plot-density trace synergism. *The American Statistician*, **52**(2), 181–184.

Holgersson, H. (2006). A graphical method for assessing multivariate normality. *Computational Statistics*, **21**(1), 141–149.

Hörmann, W. and Leydold, J. (2014). Generating generalized inverse Gaussian random variates. *Statistics and Computing*, **24**(4), 547–557.

Horswell, R. L. and Looney, S. W. (1992). A comparison of tests for multivariate normality that are based on measures of multivariate skewness and kurtosis. *Journal of Statistical Computation and Simulation*, **42**(1-2), 21–38.

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

Johnson, N. L. (1949a). Bivariate distributions based on simple translation systems. *Biometrika*, **36**(3/4), 297–304.

Johnson, N. L. (1949b). Systems of frequency curves generated by methods of translation. *Biometrika*, **36**(1/2), 149–176.

Jones, M. C. and Pewsey, A. (2009). Sinh-arcsinh distributions. *Biometrika*, **96**(4), 761–780.

Karlis, D. and Xekalaki, E. (2003). Choosing initial values for the EM algorithm for finite mixtures. *Computational Statistics & Data Analysis*, **41**(3-4), 577–590.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, **90**(430), 773–795.

Kebria, M. M., Na, S., and Yu, F. (2022). An algorithmic framework for computational estimation of soil freezing characteristic curves. *International*

Journal for Numerical and Analytical Methods in Geomechanics, **46**(8), 1544–1565.

Keef, C., Tawn, J., and Svensson, C. (2009). Spatial risk assessment for extreme river flows. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **58**(5), 601–618.

Kim, J. H. and Kim, S.-Y. (2019). Tail risk measures and risk allocation for the class of multivariate normal mean–variance mixture distributions. *Insurance: Mathematics and Economics*, **86**, 145–157.

Korkmaz, S., Goksuluk, D., and Zararsiz, G. (2014). MVN: An R package for assessing multivariate normality. *The R Journal*, **6**(2), 151–162.

Kotz, S. and Nadarajah, S. (2015). *Extreme value distributions: Theory and applications*. Imperial College Press.

Kotz, S., Kozubowski, T. J., and Podgórski, K. (2001). Asymmetric multivariate laplace distribution. In *The Laplace distribution and generalizations*, pages 239–272. Springer.

Lange, K. (2016). *MM optimization algorithms*. Society for Industrial and Applied Mathematics, Philadelphia.

LeCun, Y., Cortes, C., and Burges, C. J. (1998). The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*.

Lee, S. X. and McLachlan, G. J. (2013). On mixtures of skew normal and skew-t distributions. *Advances in Data Analysis and Classification*, **7**(3), 241–266.

Lehner, K. (1989). Erzeugung von zufallszahlen aus zwei exotischen verteilungen. *Diplomar-beit, Techn. Universitaet Graz.*

Little, R. J. and Rubin, D. B. (2019). *Statistical analysis with missing data*, volume 793. John Wiley & Sons, Hoboken, 2 edition.

Liu, C. (1995). Missing data imputation using the multivariate t distribution. *Journal of Multivariate Analysis*, **53**(1), 139–158.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE transactions on information theory*, **28**(2), 129–137.

Lo, K., Hahne, F., Brinkman, R., and Gottardo, R. (2009). Flowclust: a bioconductor package for automated gating of flow cytometry data. *BMC Bioinformatics*, **10**. R package version 4.2.1.

Lu, N. and Zimmerman, D. L. (2005). The likelihood ratio test for a separable covariance matrix. *Statistics and Probability Letters*, **73**(4), 449–457.

Luciano, E. and Semeraro, P. (2010). A generalized normal mean-variance mixture for return processes in finance. *International Journal of Theoretical and Applied Finance*, **13**(03), 415–440.

MacKinnon, J. G. and Magee, L. (1990). Transforming the dependent variable in regression models. *International Economic Review*, pages 315–339.

MacQueen, J. *et al.* (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Madan, D. B., Carr, P. P., and Chang, E. C. (1998). The variance gamma process and option pricing. *Review of Finance*, **2**(1), 79–105.

Mahalanobis, P. (1936). On the generalized distance in statistic divergence in relation to breeding system in some crop plants. *Indian Journal of Genetics*, **26**, 188–198.

Mann, H. B. and Wald, A. (1943). On stochastic limit and order relationships. *The Annals of Mathematical Statistics*, **14**(3), 217–226.

Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, **57**(3), 519–530.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Statistics*. Academic Press, London.

McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D., and Barton, D. (2012). Big data: the management revolution. *Harvard Business Review*, **90**(10), 60–68.

McLachlan, G. and Krishnan, T. (2007). *The EM algorithm and extensions*, volume 382. John Wiley & Sons, Hoboken, 2 edition.

McNeil, A. J., Frey, R., and Embrechts, P. (2015). *Quantitative risk management: concepts, techniques and tools-revised edition*. Princeton: Princeton University Press.

McNicholas, P. D. (2016a). *Mixture Model-Based Classification*. Chapman & Hall/CRC Press, Boca Raton.

McNicholas, P. D. (2016b). Model-based clustering. *Journal of Classification*, **33**(3), 331–373.

McNicholas, P. D. and Murphy, T. B. (2008). Parsimonious gaussian mixture models. *Statistics and Computing*, **18**(3), 285–296.

McNicholas, S. M., McNicholas, P. D., and Browne, R. P. (2017). A mixture of variance-gamma factor analyzers. In *Big and Complex Data Analysis*, pages 369–385. Springer.

Milligan, G. W. and Cooper, M. C. (1988). A study of standardization of variables in cluster analysis. *Journal of Classification*, **5**(2), 181–204.

Mudholkar, G. S., McDermott, M., and Srivastava, D. K. (1992). A test of p-variate normality. *Biometrika*, **79**(4), 850–854.

Murphy, K. and Murphy, T. B. (2020). Gaussian parsimonious clustering models with covariates and a noise component. *Advances in Data Analysis and Classification*, **14**(2), 293–325.

Murphy, S. A. and Van der Vaart, A. W. (2000). On profile likelihood. *Journal of the American Statistical Association*, **95**(450), 449–465.

Naik, D. N. and Rao, S. S. (2001). Analysis of multivariate repeated measures data with a Kronecker product structured covariance matrix. *Journal of Applied Statistics*, **28**(1), 91–105.

Neal, M. (2022). *Variable Selection for Skewed Clustering and Classification*. McMaster University, Hamilton.

Nedler, J. and Mead, R. (1965). A simplex method for function minimization, compt. *Computer Journal*, **7**, 308–313.

Nelson, B. L. and Yamnitsky, M. (1998). Input modeling tools for complex problems. In *1998 Winter Simulation Conference. Proceedings (Cat. No. 98CH36274)*, volume 1, pages 105–112. IEEE.

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

Ownuk, J., Baghishani, H., and Nezakati, A. (2021). Heavy or semi-heavy tail, that is the question. *Journal of Applied Statistics*, **48**(4), 646–668.

O'Hagan, A., Murphy, T. B., Gormley, I. C., McNicholas, P. D., and Karlis, D. (2016). Clustering with the multivariate normal inverse Gaussian distribution. *Computational Statistics & Data Analysis*, **93**, 18–30.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pocuca, N., Browne, R. P., and McNicholas, P. D. (2021). *mixture: Mixture Models for Clustering and Classification*. R package version 2.0.4.

Počuča, N., Gallaugher, M. P. B., and McNicholas, P. D. (2019). *MatrixVariate.jl:*

A complete statistical framework for analyzing matrix variate data. Julia package version 0.2.0.

Proinov, P. D. and Ivanov, S. I. (2015). On the convergence of halley's method for simultaneous computation of polynomial zeros. *Journal of Numerical Mathematics*, **23**(4), 379–394.

Protassov, R. S. (2004). EM-based maximum likelihood parameter estimation for multivariate generalized hyperbolic distributions with fixed λ. *Statistics and Computing*, **14**(1), 67–77.

Punzo, A. and McNicholas, P. D. (2016). Parsimonious mixtures of multivariate contaminated normal distributions. *Biometrical Journal*, **58**(6), 1506–1537.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

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

Ripley, B., Venables, B., Bates, D. M., Hornik, K., Gebhardt, A., Firth, D., and Ripley, M. B. (2013). Package 'mass'. *Cran r*, **538**, 113–120.

Rosiński, J. (2007). Tempering stable processes. *Stochastic processes and their applications*, **117**(6), 677–707.

Royston, J. P. (1983). Some techniques for assessing multivariate normality based on the Shapiro-Wilk W. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **32**(2), 121–133.

Sakia, R. M. (1992). The Box-Cox transformation technique: a review. *Journal of the Royal Statistical Society: Series D (The Statistician)*, **41**(2), 169–178.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. Chapman and Hall/CRC, Boca Raton, 1 edition.

Schäling, B. (2011). *The boost C++ libraries*. Boris Schäling, 2 edition.

Schwarz, G. *et al.* (1978). Estimating the dimension of a model. *The Annals of Statistics*, **6**(2), 461–464.

Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R journal*, **8**(1), 289.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., and Young, M. (2014). Machine learning: The high interest credit card of technical debt.(2014).

Shapiro, S. and Wilk, M. (1965). An analysis of variance test for normality. *Biometrika*, **52**(3), 591–611.

Snoussi, H. and Idier, J. (2006). Bayesian blind separation of generalized hyperbolic processes in noisy and underdeterminate mixtures. *IEEE Transactions on Signal Processing*, **54**(9), 3257–3269.

Spitzer, J. J. (1982). A primer on Box-Cox estimation. *The Review of Economics and Statistics*, pages 307–313.

Srivastava, M. S., Nahtman, T., and Von Rosen, D. (2008). Models with a Kronecker product covariance structure: estimation and testing. *Mathematical Methods of Statistics*, **17**(4), 357–370.

Stanfield, P. M., Wilson, J. R., Mirka, G. A., Glasscock, N. F., Psihogios, J. P., and Davis, J. R. (1996). Multivariate input modeling with johnson distributions. In *Proceedings Winter Simulation Conference*, pages 1457–1464. IEEE.

Stroustrup, B. (2018). *A Tour of C++*. Addison-Wesley Professional, Upper Saddle River, 2 edition.

Student (1908). The probable error of a mean. *Biometrika*, **6**(1), 1–25.

Tait, P. A., McNicholas, P. D., and Obeid, J. (2020). Clustering higher order data: An application to pediatric multi-variable longitudinal data. *arXiv preprint arXiv:1907.08566v3*.

Temme, N. M. (1975). On the numerical evaluation of the modified Bessel function of the third kind. *Journal of Computational Physics*, **19**(3), 324–337.

Tiedeman, D. V. (1955). On the study of types. *Symposium on pattern analysis*, pages 1–14.

Tomarchio, S. D., McNicholas, P. D., and Punzo, A. (2021). Matrix normal cluster-weighted models. *Journal of Classification*, **38**(3), 556—575.

Tomarchio, S. D., Gallaugher, M. P. B., Punzo, A., and McNicholas, P. D. (2022). Mixtures of contaminated matrix variate normal distributions. *Journal of Computational and Graphical Statistics*, **31**(2), 413–421.

Tortora, C., McNicholas, P. D., and Browne, R. P. (2016). A mixture of generalized hyperbolic factor analyzers. *Advances in Data Analysis and Classification*, **10**(4), 423–440.

Tortora, C., Browne, R. P., ElSherbiny, A., Franczak, B. C., and McNicholas, P. D. (2021). Model-based clustering, classification, and discriminant analysis using the generalized hyperbolic distribution: Mixghd R package. *Journal of Statistical Software*, **98**(1), 1–24.

Tsai, C. S.-Y. (2011). The real world is not normal. *Morningstar Alternative Investments Observer: Chicago, IL, USA*.

Van Rossum, G. and Drake Jr, F. L. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition.

Viroli, C. (2011). Finite mixtures of matrix normal distributions for classifying three-way data. *Statistics and Computing*, **21**(4), 511–522.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van

Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, **17**, 261–272.

Vrbik, I. and McNicholas, P. D. (2014). Parsimonious skew mixture models for model-based clustering and classification. *Computational Statistics & Data Analysis*, **71**, 196–210.

Wang, Z. and Bovik, A. C. (2009). Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, **26**(1), 98–117.

Wei, Y., Tang, Y., and McNicholas, P. D. (2019). Mixtures of generalized hyperbolic distributions and mixtures of skew-t distributions for model-based clustering with incomplete data. *Computational Statistics & Data Analysis*, **130**, 18–41.

Wolfe, J. (1965). A computer program for the maximum likelihood analysis of types. *Technical Bulletin*, pages 15–65.

Zhang, D., Zhou, Y., Phoon, K.-K., and Huang, H. (2020). Multivariate probability distribution of shanghai clay properties. *Engineering Geology*, **273**, 105675.

Zhou, E. (1996). Object-oriented programming, C++ and power system simulation. *IEEE Transactions on Power Systems*, **11**(1), 206–215.

Zhou, H. and Lange, K. L. (2010). On the bumpy road to the dominant mode. *Scandinavian Journal of Statistics*, **37**(4), 612–631.

Zhou, Y., Zhang, D., Huang, H., and Xue, Y. (2022). Effect of normal transformation methods on performance of multivariate normal distribution. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, **8**(1), 04021074.

Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J. (1997). Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, **23**(4), 550–560.

Zhu, X. and Melnykov, V. (2017). ManlyMix: An R package for Manly mixture modeling. *R Journal*, **9**(2), 176.

Zhu, X. and Melnykov, V. (2018). Manly transformation in finite mixture modeling. *Computational Statistics & Data Analysis*, **121**, 190–208.

APPENDIX A

A.1 Complete-Profile-Loglikelihood

There are a large number of parameters required to estimate in within the S_{U} Johnson mixture model. It is favourable to reduce the dimensionality of parameter optimizations through the derivation of a profile log-likelihood (Spitzer, 1982). The work of Murphy and Van der Vaart (2000) shows that profile log-likelihoods behave like ordinary likelihoods, in that they have a quadratic expansion, and therefore, can be used to fit MLE's. Spitzer (1982) provides a primer on Box-Cox parameter estimations using profile log-likelihoods. Due to the similarities between both S_{U} and power transformations, consider a similar approach as follows.

Let $\boldsymbol{\Phi} := (\boldsymbol{\Theta} \setminus \boldsymbol{\vartheta})$ designate all parameters that are not associated with the scale/shift of hyperbolic space. Let $n_g = \sum_{i=1}^{n} z_{ig}$. The MLE estimates $\hat{\boldsymbol{\Phi}}$ of (5.3)

are derived as follows

$$\hat{\pi}_g = \frac{\sum_{i=1}^{n} z_{ig}}{n_g}, \quad \hat{\boldsymbol{\mu}}_g = \frac{\sum_{i=1}^{n} z_{ig} \boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g)}{n_g},$$

$$\hat{\boldsymbol{\Sigma}}_g = \frac{\sum_{i=1}^{n} z_{ig} \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right) \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right)^\top}{n_g}. \tag{A.1}$$

Substituting maximum likelihood estimates of (A.1) into (5.3) yields the complete-profile log-likelihood for $\boldsymbol{\vartheta}$ as

$$\begin{aligned} l_c(\boldsymbol{\vartheta}; \boldsymbol{y}_1, \dots, \boldsymbol{y}_n, \boldsymbol{z}, \hat{\boldsymbol{\Phi}}) = & \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \log(\hat{\pi}_g) - \frac{1}{2} \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right)^\top \hat{\boldsymbol{\Sigma}}_g^{-1} \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right) \\ & - \frac{np}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \log(|\hat{\boldsymbol{\Sigma}}_g|) - \sum_{i=1}^{n} \sum_{g=1}^{G} \sum_{j=1}^{p} z_{ig} \log(\delta_{jg}) \\ & - \frac{1}{2} \sum_{i=1}^{n} \sum_{g=1}^{G} \sum_{j=1}^{p} z_{ig} \log\left(\left(\frac{y_{ij} - \omega_{jg}}{\delta_{jg}}\right)^2 + 1\right). \end{aligned}$$

The second row can be shown to be free of $\boldsymbol{\vartheta}$ as follows:

$$\begin{aligned} & -\frac{1}{2} \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right)^\top \hat{\boldsymbol{\Sigma}}_g^{-1} \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right) \\ = & -\frac{1}{2} \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right)^\top \left(\frac{\sum_{i=1}^{n} z_{ig} \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right) \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right)^\top}{n_g}\right)^{-1} \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right). \end{aligned}$$

Simplifying the expression further, let $\boldsymbol{b}_{ig} = \left(\boldsymbol{h}(\boldsymbol{y}_i; \boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right)$ allowing

$$
\begin{aligned}
&- \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\boldsymbol{b}_{ig}^{\top}\left(\frac{\sum_{i=1}^{n} z_{ig}\boldsymbol{b}_{ig}\boldsymbol{b}_{ig}^{\top}}{n_g}\right)^{-1}\boldsymbol{b}_{ig} \\
&= -\frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n} n_g z_{ig}\boldsymbol{b}_{ig}^{\top}\left(\sum_{i=1}^{n} z_{ig}\boldsymbol{b}_{ig}\boldsymbol{b}_{ig}^{\top}\right)^{-1}\boldsymbol{b}_{ig} \\
&= -\frac{1}{2}\sum_{g=1}^{G} n_g \sum_{i=1}^{n} z_{ig}\text{trace}\left(\boldsymbol{b}_{ig}^{\top}\left(\sum_{i=1}^{n} z_{ig}\boldsymbol{b}_{ig}\boldsymbol{b}_{ig}^{\top}\right)^{-1}\boldsymbol{b}_{ig}\right) \\
&= -\frac{1}{2}\sum_{g=1}^{G} n_g \sum_{i=1}^{n} z_{ig}\text{trace}\left(\boldsymbol{b}_{ig}\boldsymbol{b}_{ig}^{\top}\left(\sum_{i=1}^{n} z_{ig}\boldsymbol{b}_{ig}\boldsymbol{b}_{ig}^{\top}\right)^{-1}\right) \\
&= -\frac{1}{2}\sum_{g=1}^{G} n_g \text{trace}\left(\sum_{i=1}^{n} z_{ig}\boldsymbol{b}_{ig}\boldsymbol{b}_{ig}^{\top}\left(\sum_{i=1}^{n} z_{ig}\boldsymbol{b}_{ig}\boldsymbol{b}_{ig}^{\top}\right)^{-1}\right) = -\frac{1}{2}\sum_{g=1}^{G} n_g \text{trace}\left(\mathbf{I}\right) = -\frac{np}{2}.
\end{aligned}
$$

With this simplification, the complete-profile likelihood can be written as:

$$
\begin{aligned}
l_{\text{cp}}(\boldsymbol{\vartheta}; \boldsymbol{y}_1, \dots, \boldsymbol{y}_n) = &\sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\log(\hat{\pi}_g) - \frac{np}{2}\log(2\pi) + \frac{np}{2}\sum_{g=1}^{G}\log(n_g) - \frac{np}{2} \\
&- \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\text{logdet}\left(\sum_{i=1}^{n} z_{ig}\left(\boldsymbol{h}(\boldsymbol{y}_i;\boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right)\left(\boldsymbol{h}(\boldsymbol{y}_i;\boldsymbol{\vartheta}_g) - \hat{\boldsymbol{\mu}}_g\right)^{\top}\right) \\
&- \sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{j=1}^{p} z_{ig}\log(\delta_{jg}) - \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{j=1}^{p} z_{ig}\log\left(\left(\frac{y_{ij}-\omega_{jg}}{\delta_{jg}}\right)^2 + 1\right).
\end{aligned}
$$

Notice that the first row of terms are strictly constants with respect to $\boldsymbol{\vartheta}$ and are deemed unnecessary to compute during optimization procedure. By concentrating out the terms for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, the optimization procedure is reduced down to strictly $2pG$ number of parameters.

A.2 Derivation of a Conditional S_{U} Distribution

Let $\boldsymbol{\mathcal{Y}}$ emanate from a S_{U} distribution parametrized by $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\omega}$, and $\boldsymbol{\Lambda}$. For some realization $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$, consider the conditional distribution of $y_j|\boldsymbol{y_d}$ where $\boldsymbol{y_d}$ corresponds to entries of $\boldsymbol{y}$ excluding y_j. We now show that the distribution of $y_j|\boldsymbol{y_d}$ is a univariate S_{U} Johnson distribution parametrized by $\tilde{\mu}, \tilde{\sigma}, \omega_j, \delta_j$ as follows. The join density function of $\boldsymbol{y} = (y_j, \boldsymbol{y_d})$ is written as

$$f(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\omega}, \boldsymbol{\Lambda}) = \frac{\exp\left\{-\frac{1}{2}\left(\boldsymbol{h}(\boldsymbol{y}; \boldsymbol{\vartheta}) - \boldsymbol{\mu}\right)^{\top} \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{h}(\boldsymbol{y}; \boldsymbol{\vartheta}) - \boldsymbol{\mu}\right)\right\}}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}} \prod_{k=1}^{p} \left(\delta_k \sqrt{\left(\frac{y_k - \omega_k}{\delta_k}\right)^2 + 1}\right)}, \quad \boldsymbol{y} \in \mathbb{R}^p. \quad \text{(A.2)}$$

Within our derivation, and without loss of generality, we place index j as the first entry for convenience. Working with the denominator of (A.2), we factor into two separate forms using equation (2.8.13 of Dennis, 2009) as

$$\begin{aligned}
&(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}} \prod_{k=1}^{p} \left(\delta_k \sqrt{\left(\frac{y_k - \omega_k}{\delta_k}\right)^2 + 1}\right) |\boldsymbol{\Sigma}|^{\frac{1}{2}} \\
&= (2\pi)^{\frac{1}{2}} (2\pi)^{\frac{(p-1)}{2}} \left(\delta_j \sqrt{\left(\frac{y_j - \omega_j}{\delta_j}\right)^2 + 1}\right) \prod_{k=2}^{p} \left(\delta_k \sqrt{\left(\frac{y_k - \omega_k}{\delta_k}\right)^2 + 1}\right) |\boldsymbol{\Sigma}_{\boldsymbol{d},\boldsymbol{d}}|^{\frac{1}{2}} |\sigma_{jj}^2 - \boldsymbol{\Sigma}_{j,\boldsymbol{d}} \boldsymbol{\Sigma}_{\boldsymbol{d},\boldsymbol{d}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{d},j}|^{\frac{1}{2}} \\
&= (2\pi)^{\frac{1}{2}} |\sigma_{jj}^2 - \boldsymbol{\Sigma}_{j,\boldsymbol{d}} \boldsymbol{\Sigma}_{\boldsymbol{d},\boldsymbol{d}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{d},j}|^{\frac{1}{2}} \left(\delta_j \sqrt{\left(\frac{y_j - \omega_j}{\delta_j}\right)^2 + 1}\right) (2\pi)^{\frac{(p-1)}{2}} \prod_{k=2}^{p} \left(\delta_k \sqrt{\left(\frac{y_k - \omega_k}{\delta_k}\right)^2 + 1}\right) |\boldsymbol{\Sigma}_{\boldsymbol{d},\boldsymbol{d}}|^{\frac{1}{2}}.
\end{aligned} \quad \text{(A.3)}$$

Working with the numerator of (A.2), we can decompose the quadratic form of the exponent into two parts. Let $\boldsymbol{a} = \boldsymbol{h}(\boldsymbol{y}; \boldsymbol{\vartheta})$ for ease of notation. Since $\boldsymbol{h}$ is a component-wise function, we can treat each entry within as $\boldsymbol{a} = \left[a_j, \boldsymbol{a_d}\right]$. Now,

using equation (2.8.27 of Dennis, 2009), we decompose the quadratic form as

$$
\begin{aligned}
(\boldsymbol{a}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{a}-\boldsymbol{\mu}) &= \begin{bmatrix} a_j-\mu_j, \boldsymbol{a_d}-\boldsymbol{\mu_d} \end{bmatrix} \begin{bmatrix} \sigma^2_{jj}, & \boldsymbol{\Sigma_{jd}} \\ \boldsymbol{\Sigma_{dj}}, & \boldsymbol{\Sigma_{dd}} \end{bmatrix}^{-1} \begin{bmatrix} a_j-\mu_j \\ \boldsymbol{a_d}-\boldsymbol{\mu_d} \end{bmatrix} \\
&= (a_j-\mu_j)^\top(\sigma^2_{jj}-\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}}\boldsymbol{\Sigma_{dj}})^{-1}(a_j-\mu_j) \\
&\quad - 2(a_j-\mu_j)^\top(\sigma^2_{jj}-\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}})^{-1}\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}}(\boldsymbol{a_d}-\boldsymbol{\mu_d}) \\
&\quad + (\boldsymbol{a_d}-\boldsymbol{\mu_d})^\top(\boldsymbol{\Sigma^{-1}_{dd}}+\boldsymbol{\Sigma^{-1}_{dd}}\boldsymbol{\Sigma}_{dm}(\sigma^2_{jj}-\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}}\boldsymbol{\Sigma_{dj}})^{-1}\boldsymbol{\Sigma}_{md}\boldsymbol{\Sigma^{-1}_{dd}})(\boldsymbol{a_d}-\boldsymbol{\mu_d}) \\
&= (a_j-\mu_j)^\top(\sigma^2_{jj}-\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}}\boldsymbol{\Sigma_{dj}})^{-1}(a_j-\mu_j) \\
&\quad - 2(a_j-\mu_j)^\top(\sigma^2_{jj}-\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}})^{-1}\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}}(\boldsymbol{a_d}-\boldsymbol{\mu_d}) \\
&\quad + (\boldsymbol{a_d}-\boldsymbol{\mu_d})^\top(\boldsymbol{\Sigma^{-1}_{dd}}\boldsymbol{\Sigma_{dj}}(\sigma^2_{jj}-\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}}\boldsymbol{\Sigma_{dj}})^{-1}\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}})(\boldsymbol{a_d}-\boldsymbol{\mu_d}) \\
&\quad + (\boldsymbol{a_d}-\boldsymbol{\mu_d})^\top\boldsymbol{\Sigma^{-1}_{dd}}(\boldsymbol{a_d}-\boldsymbol{\mu_d}) \\
&= (a_j-\mu_j-\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}}(\boldsymbol{a_d}-\boldsymbol{\mu_d}))^\top(\sigma^2_{jj}-\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}}\boldsymbol{\Sigma_{dj}})^{-1} \\
&\quad (a_j-\mu_j-\boldsymbol{\Sigma_{jd}}\boldsymbol{\Sigma^{-1}_{dd}}(\boldsymbol{a_d}-\boldsymbol{\mu_d})) \\
&\quad + (\boldsymbol{a_d}-\boldsymbol{\mu_d})^\top\boldsymbol{\Sigma^{-1}_{dd}}(\boldsymbol{a_d}-\boldsymbol{\mu_d}). \qquad \text{(A.4)}
\end{aligned}
$$

Given the preceding results, we derive the density of the conditional distribution $y_j|\boldsymbol{y_d}$ as

$$
f(y_j|\boldsymbol{y_d};\tilde{\mu},\tilde{\sigma},\omega_j,\delta_j) = \frac{f(\boldsymbol{y};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\omega},\boldsymbol{\Lambda})}{f(\boldsymbol{y_d};\boldsymbol{\mu_d},\boldsymbol{\Sigma_d},\boldsymbol{\omega_d},\boldsymbol{\Lambda_d})}
$$

$$= \frac{\exp\left\{-\frac{1}{2}\left(\boldsymbol{a}-\boldsymbol{\mu}\right)^{\top}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{a}-\boldsymbol{\mu}\right)\right\}}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}\prod_{k=1}^{p}\left(\delta_k\sqrt{\left(\frac{y_k-\omega_k}{\delta_k}\right)^2+1}\right)} \frac{(2\pi)^{\frac{p-1}{2}}|\boldsymbol{\Sigma}_{\boldsymbol{dd}}|^{\frac{1}{2}}\prod_{l=2}^{p}\left(\delta_l\sqrt{\left(\frac{y_l-\omega_l}{\delta_l}\right)^2+1}\right)}{\exp\left\{-\frac{1}{2}(\boldsymbol{a_d}-\boldsymbol{\mu_d})^{\top}\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}(\boldsymbol{a_d}-\boldsymbol{\mu_d})\right\}}$$

$$= \frac{(2\pi)^{\frac{p-1}{2}}|\boldsymbol{\Sigma}_{\boldsymbol{dd}}|^{\frac{1}{2}}\prod_{l=2}^{p}\left(\delta_l\sqrt{\left(\frac{y_l-\omega_l}{\delta_l}\right)^2+1}\right)}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}\prod_{k=1}^{p}\left(\delta_k\sqrt{\left(\frac{y_k-\omega_k}{\delta_k}\right)^2+1}\right)} \frac{\exp\left\{-\frac{1}{2}\left(\boldsymbol{a}-\boldsymbol{\mu}\right)^{\top}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{a}-\boldsymbol{\mu}\right)\right\}}{\exp\left\{-\frac{1}{2}(\boldsymbol{a_d}-\boldsymbol{\mu_d})^{\top}\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}(\boldsymbol{a_d}-\boldsymbol{\mu_d})\right\}}. \tag{A.5}$$

Working with the left hand side of (A.5), and using result (A.3), the expression is reduced to

$$\frac{(2\pi)^{\frac{p-1}{2}}|\boldsymbol{\Sigma}_{\boldsymbol{dd}}|^{\frac{1}{2}}\prod_{l=2}^{p}\left(\delta_l\sqrt{\left(\frac{y_l-\omega_l}{\delta_l}\right)^2+1}\right)}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}\prod_{k=1}^{p}\left(\delta_k\sqrt{\left(\frac{y_k-\omega_k}{\delta_k}\right)^2+1}\right)}$$

$$= \frac{(2\pi)^{\frac{p-1}{2}}|\boldsymbol{\Sigma}_{\boldsymbol{dd}}|^{\frac{1}{2}}\prod_{l=2}^{p}\left(\delta_l\sqrt{\left(\frac{y_l-\omega_l}{\delta_l}\right)^2+1}\right)}{(2\pi)^{\frac{1}{2}}|\sigma_{jj}^2-\boldsymbol{\Sigma}_{j,\boldsymbol{d}}\boldsymbol{\Sigma}_{\boldsymbol{d},\boldsymbol{d}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{d},j}|^{\frac{1}{2}}\left(\delta_j\sqrt{\left(\frac{y_j-\omega_j}{\delta_j}\right)^2+1}\right)(2\pi)^{\frac{(p-1)}{2}}\prod_{k=2}^{p}\left(\delta_k\sqrt{\left(\frac{y_k-\omega_k}{\delta_k}\right)^2+1}\right)|\boldsymbol{\Sigma}_{\boldsymbol{d},\boldsymbol{d}}|^{\frac{1}{2}}}$$

$$= \frac{1}{(2\pi)^{\frac{1}{2}}|\sigma_{jj}^2-\boldsymbol{\Sigma}_{j,\boldsymbol{d}}\boldsymbol{\Sigma}_{\boldsymbol{d},\boldsymbol{d}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{d},j}|^{\frac{1}{2}}\left(\delta_j\sqrt{\left(\frac{y_j-\omega_j}{\delta_j}\right)^2+1}\right)}. \tag{A.6}$$

Working with the exponential terms at right hand side of (A.5), and, using result

(A.4), results in

$$
\begin{aligned}
&(\boldsymbol{a}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{a}-\boldsymbol{\mu}) - (\boldsymbol{a_d}-\boldsymbol{\mu_d})^\top\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}(\boldsymbol{a_d}-\boldsymbol{\mu_d}) \\
&= (a_j-\mu_j-\boldsymbol{\Sigma}_{j\boldsymbol{d}}\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}(\boldsymbol{a_d}-\boldsymbol{\mu_d}))^\top(\sigma_{jj}^2-\boldsymbol{\Sigma}_{j\boldsymbol{d}}\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{d}j})^{-1}(a_j-\mu_j-\boldsymbol{\Sigma}_{j\boldsymbol{d}}\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}(\boldsymbol{a_d}-\boldsymbol{\mu_d})) \\
&\quad + (\boldsymbol{a_d}-\boldsymbol{\mu_d})^\top\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}(\boldsymbol{a_d}-\boldsymbol{\mu_d}) - (\boldsymbol{a_d}-\boldsymbol{\mu_d})^\top\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}(\boldsymbol{a_d}-\boldsymbol{\mu_d}) \\
&= (a_j-\mu_j-\boldsymbol{\Sigma}_{j\boldsymbol{d}}\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}(\boldsymbol{a_d}-\boldsymbol{\mu_d}))^\top(\sigma_{jj}^2-\boldsymbol{\Sigma}_{j\boldsymbol{d}}\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{d}j})^{-1}(a_j-\mu_j-\boldsymbol{\Sigma}_{j\boldsymbol{d}}\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}(\boldsymbol{a_d}-\boldsymbol{\mu_d})).
\end{aligned}
\tag{A.7}
$$

Finally, combining results (A.6,A.7), concludes with $\tilde{\mu} = \mu_j + \boldsymbol{\Sigma}_{j\boldsymbol{d}}\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}(\boldsymbol{a_d}-\boldsymbol{\mu_d})$, $\tilde{\sigma}^2 = \sigma_{jj}^2 - \boldsymbol{\Sigma}_{j\boldsymbol{d}}\boldsymbol{\Sigma}_{\boldsymbol{dd}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{d}j}$, and

$$
\begin{aligned}
f(y_j|\boldsymbol{y_d};\tilde{\mu},\tilde{\sigma},\omega_j,\delta_j) &= \frac{\exp\left\{-\frac{1}{2}(a_j-\tilde{\mu})^\top(\tilde{\sigma}^2)^{-1}(a_j-\tilde{\mu})\right\}}{(2\pi)^{\frac{1}{2}}|\tilde{\sigma}^2|^{\frac{1}{2}}\left(\delta_j\sqrt{\left(\frac{y_j-\omega_j}{\delta_j}\right)^2+1}\right)} \\
&= \frac{\exp\left\{-\frac{1}{2}\left(\frac{h(y_j;\omega_j,\delta_j)-\tilde{\mu}}{\tilde{\sigma}}\right)^2\right\}}{(2\pi)^{\frac{1}{2}}\tilde{\sigma}\left(\delta_j\sqrt{\left(\frac{y_j-\omega_j}{\delta_j}\right)^2+1}\right)}.
\end{aligned}
\tag{A.8}
$$

The density derived above is a univariate S_{U} Johnson distribution with hyperbolic parameters ω_j, δ_j, and Gaussian parameters $\tilde{\mu}$, $\tilde{\sigma}$. Notice that the conditioning on $\boldsymbol{y_d}$ only adjusts the Gaussian parameters giving rise to the same joint distribution as $\boldsymbol{y}$.

APPENDIX B

B.1 Proof of Distance Equality

Here follows the proof for (12). Let $\boldsymbol{\mu} = \text{vec}(\boldsymbol{M})$ and $\boldsymbol{\Sigma} = \boldsymbol{V} \otimes \boldsymbol{U}$, then

$$\begin{aligned}
\mathcal{D}(\boldsymbol{X}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \big(\text{vec}(\boldsymbol{X}_i) - \boldsymbol{\mu}\big)^\top \boldsymbol{\Sigma}^{-1} \big(\text{vec}(\boldsymbol{X}_i) - \boldsymbol{\mu}\big) \\
&= \{\text{vec}(\boldsymbol{X}_i) - \text{vec}(\boldsymbol{M})\}^\top (\boldsymbol{V} \otimes \boldsymbol{U})^{-1} \{\text{vec}(\boldsymbol{X}_i) - \text{vec}(\boldsymbol{M})\} \\
&= \text{vec}\,(\boldsymbol{X}_i - \boldsymbol{M})^\top \left(\boldsymbol{V}^{-1} \otimes \boldsymbol{U}^{-1}\right) \text{vec}\,(\boldsymbol{X}_i - \boldsymbol{M}) \\
&= \text{vec}\,(\boldsymbol{X}_i - \boldsymbol{M})^\top \,\text{vec}\left\{\boldsymbol{U}^{-1}(\boldsymbol{X}_i - \boldsymbol{M})\boldsymbol{V}^{-1}\right\} \\
&= \text{tr}\left\{\boldsymbol{V}^{-1}(\boldsymbol{X}_i - \boldsymbol{M})^\top \boldsymbol{U}^{-1}(\boldsymbol{X}_i - \boldsymbol{M})\right\} \\
&= \text{tr}\left\{\boldsymbol{U}^{-1}(\boldsymbol{X}_i - \boldsymbol{M})\boldsymbol{V}^{-1}(\boldsymbol{X}_i - \boldsymbol{M})^\top\right\} \\
&= \mathcal{D}_M(\boldsymbol{X}_i, \boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V}).
\end{aligned}$$

Showing the reverse,

$$\mathcal{D}(\boldsymbol{X}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{D}_M(\boldsymbol{X}_i, \boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V}),$$

i.e.,

$$\begin{aligned}\big(\text{vec}(\boldsymbol{X}_i)-\boldsymbol{\mu}\big)^\top\boldsymbol{\Sigma}^{-1}\big(\text{vec}(\boldsymbol{X}_i)-\boldsymbol{\mu}\big) &= \text{tr}\left\{\boldsymbol{U}^{-1}(\boldsymbol{X}_i-\boldsymbol{M})\boldsymbol{V}^{-1}(\boldsymbol{X}_i-\boldsymbol{M})^\top\right\}\\ &= \text{tr}\left\{\boldsymbol{V}^{-1}(\boldsymbol{X}_i-\boldsymbol{M})^\top\boldsymbol{U}^{-1}(\boldsymbol{X}_i-\boldsymbol{M})\right\}\\ &= \text{vec}\left(\boldsymbol{X}_i-\boldsymbol{M}\right)^\top\text{vec}\left\{\boldsymbol{U}^{-1}(\boldsymbol{X}_i-\boldsymbol{M})\boldsymbol{V}^{-1}\right\}\\ &= \left\{\text{vec}(\boldsymbol{X}_i)-\text{vec}(\boldsymbol{M})\right\}^\top(\boldsymbol{V}\otimes\boldsymbol{U})^{-1}\left\{\text{vec}(\boldsymbol{X}_i)-\text{vec}(\boldsymbol{M})\right\}.\end{aligned}$$

This equality only holds if $\boldsymbol{\mu}=\text{vec}(\boldsymbol{M})$ and $\boldsymbol{\Sigma}=\boldsymbol{V}\otimes\boldsymbol{U}$. Therefore, under matrix-variate normality $\mathcal{D}(\boldsymbol{X}_i,\boldsymbol{\mu},\boldsymbol{\Sigma})=\mathcal{D}_M(\boldsymbol{X}_i,\boldsymbol{M},\boldsymbol{U},\boldsymbol{V})$, with $\boldsymbol{\mu}=\text{vec}(\boldsymbol{M})$ and $\boldsymbol{\Sigma}=\boldsymbol{V}\otimes\boldsymbol{U}$.

B.2 Continuous Mapping Theorem

The continuous mapping theorem was first proved in 1943 and is sometimes referred to as the Mann-Wald theorem (Mann and Wald, 1943).

Theorem B.1 Let $\{X_N\}$, $\{Y_N\}$, X, and Y be random elements on some metric space S. In addition, let g be a bivariate continuous map from one metric space S to another S'. Then,

$$X_N, Y_N \xrightarrow{P} X, Y \quad \Rightarrow \quad g(X_N, Y_N) \xrightarrow{P} g(X, Y).$$

B.3 Parameters used to Generate Figures

B.3.1 Parameters for Figure 6.1

Left-hand plot:

$$\boldsymbol{M} = \begin{bmatrix} 1.4426 & 3.5974 \\ 0.4797 & 1.6333 \end{bmatrix}, \boldsymbol{U} = \begin{bmatrix} 8.7093 & 0.4839 \\ 0.4839 & 0.0577 \end{bmatrix},$$

$$\boldsymbol{V} = \begin{bmatrix} 1.0887 & 0.0605 \\ 0.0605 & 0.0072 \end{bmatrix}.$$

The right-hand plot uses the same $\boldsymbol{M}$ as before but

$$\boldsymbol{\Sigma} = \begin{bmatrix} 6.0373 & 6.3725 & 10.1365 & 3.5837 \\ 6.3725 & 7.4186 & 10.0872 & 4.4695 \\ 10.1365 & 10.0872 & 18.8089 & 6.1764 \\ 3.5837 & 4.4695 & 6.1764 & 4.4742 \end{bmatrix}.$$

B.3.2 Parameters for Figure 6.2, B.1a

The parameters are the same as the left-hand plot of Figure 6.1 listed in Section B.3.1.

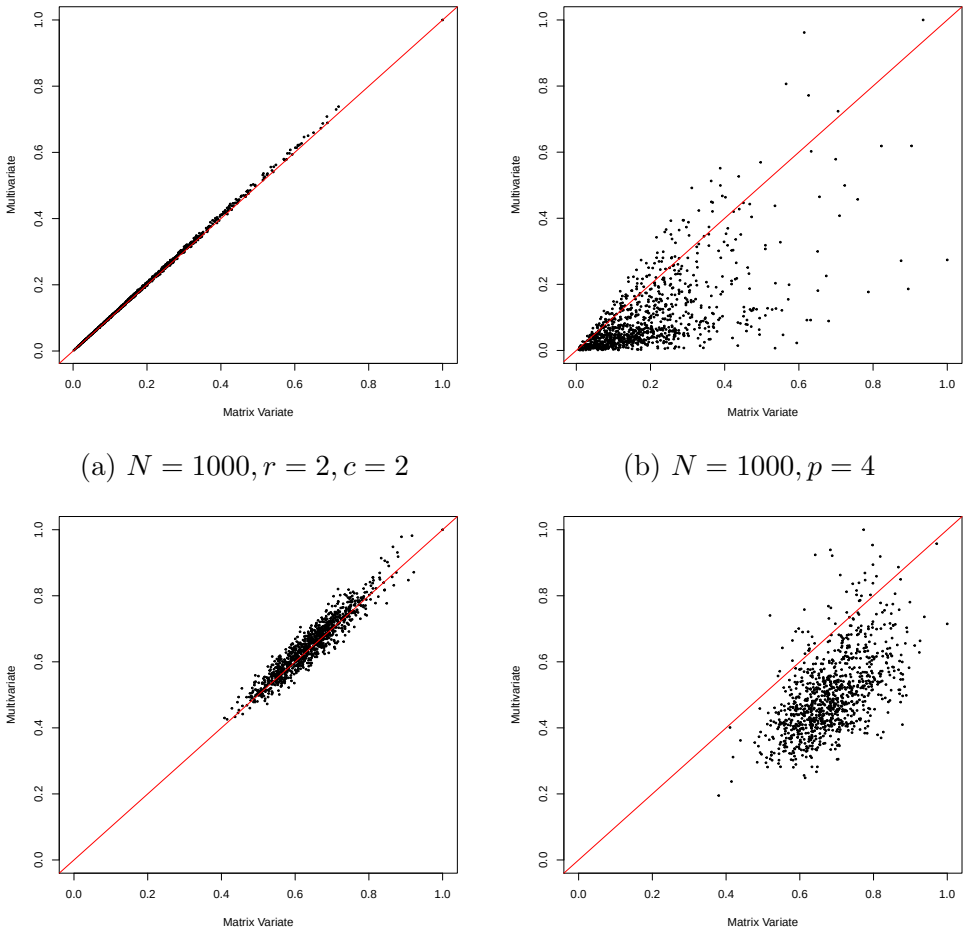
B.3.3 Parameters for Figure B.1b

The parameters are the same as the right-hand plot of Figure 6.1 listed in Section B.3.1.

B.3.4 Parameters for Remaining Figures

Parameters for all remaining figures are not listed because $p = 100$.

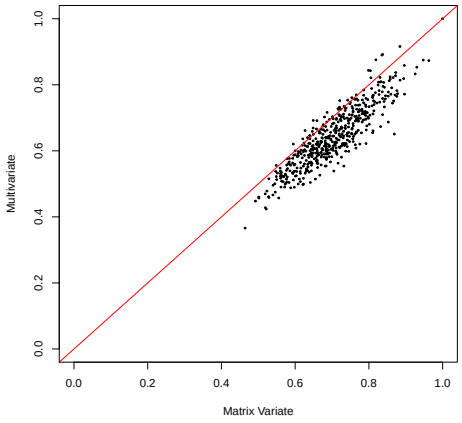
B.4 DD Plots for Simulation Study



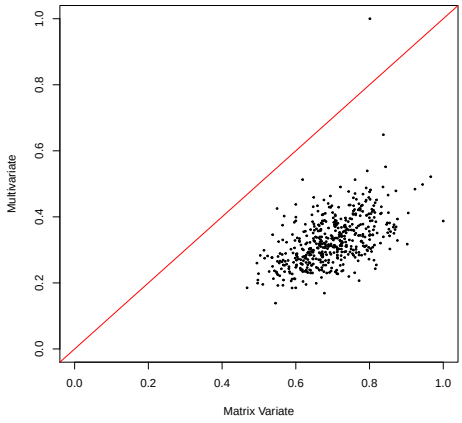
(a) $N = 1000, r = 2, c = 2$ (b) $N = 1000, p = 4$

(c) $N = 1000, r = 10, c = 10$ (d) $N = 1000, p = 100$

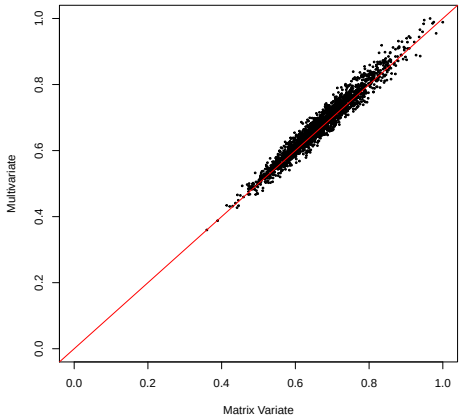
Figure B.1: DD plots for simulated data with $N = 1000$ and $p \in \{4, 100\}$ where matrix-variate normal structure is present (left hand figures) and absent (right hand figures).



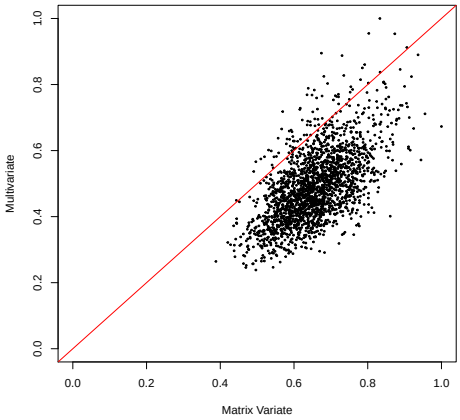
(a) $N = 500, p = 100$



(b) $N = 500, p = 100$



(c) $N = 2000, p = 100$



(d) $N = 2000, p = 100$

Figure B.2: DD plots for simulated data with $N \in \{500, 2000\}$ and $p = 100$, where a matrix-variate normal structure is present (left) or absent (right).