Analysis of Three-Way Data and Other Topics in Clustering and Classification

ANALYSIS OF THREE-WAY DATA AND OTHER TOPICS IN CLUSTERING AND CLASSIFICATION

BY

MICHAEL P.B. GALLAUGHER, M.Sc.

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 OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2020) McMaster University

(Mathematics & Statistics) Hamilton, Ontario, Canada

TITLE:	Analysis of Three-Way Data and Other Topics in Clustering and Classification
AUTHOR:	Michael P.B. Gallaugher M.Sc., (Statistics) McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Paul D. McNicholas
NUMBER OF PAGES:	xx, 184

To my grandfather, Albert Jones

Abstract

Clustering and classification is the process of finding underlying group structure in heterogenous data. With the rise of the "big data" phenomenon, more complex data structures have made it so traditional clustering methods are oftentimes not advisable or feasible. This thesis presents methodology for analyzing three different examples of these more complex data types. The first is three-way (matrix variate) data, or data that come in the form of matrices. A large emphasis is placed on clustering skewed three-way data, and high dimensional three-way data. The second is click-stream data, which considers a user's internet search patterns. Finally, co-clustering methodology is discussed for very high-dimensional two-way (multivariate) data. Parameter estimation for all these methods is based on the expectation maximization (EM) algorithm. Both simulated and real data are used for illustration.

Acknowledgements

First and foremost, I would like to express my deepest appreciation to my supervisor Dr. Paul McNicholas. His support and dedication were instrumental in the completion of this thesis. I would also like to thank him for all the opportunities for both professional and personal development he provided through both conferences and international collaborations. I will forever be in his debt.

I would also like to show my sincere gratitude to Dr. Roman Viveros, and Dr. Jeffrey Racine who, along with Dr. McNicholas, were on my supervisory committee. I would also like to thank Dr. David Madigan who served as my external examiner, and Dr. Jim Reilly who served as the chair for my defence.

I would like to acknowledge the funds provided through the Vanier Canada Graduate Scholarship from the Natural Sciences and Engineering Research Council of Canada (NSERC), the Milos Novotny Fellowship from McMaster University, and the Department of Mathematics and Statistics.

Over the course of my PhD, I had the pleasure of working with international collaborators. I would specifically like to thank Dr. Christophe Biernacki (Inria, France), Dr. Volodymyr Melnykov (University of Alabama), and Dr. Antonio Punzo (University of Catania) who hosted me at their respective institutions.

I would like to express my appreciation to Sheree Cox (retired), Julie Fogarty, Diana Holmes, and all other office staff in the Department of Mathematics and Statistics for their help with administrative matters.

I would like to thank everyone in the McNicholas research group, who I had the privilege of working with during my time as a PhD student.

Finally, I would like to thank my family, especially my parents, grandparents and my sister for all of their love and support throughout the course of my education.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the past, data could often be analyzed using straightforward, off-the-shelf, statistical methods. Nowadays, however, with more complex data structures available today, due to the "big data" phenomenon, traditional methods are often not advisable and in many cases do not work.

This is particularly true in the area of clustering which is the process of revealing underlying (hidden) group structure in data, and is fundamental to computational statistics and machine learning. One may perform a cluster analysis on gene expression data to reveal previously unknown subtypes of a medical condition, a species of maize, etc. There are many methods for clustering presented in the literature, but they generally fall into two classes, distance based and model based clustering. Distance based methods group objects together based on their distance from each other, according to some distance measure, so that objects in the same cluster are "closer" to each other, and objects in different clusters are farther away from each other. There are many drawbacks to such methods, two of which are mentioned here. The first is that these methods are quite inflexible when it comes to cluster structure. The second

is that it may be difficult, or in some cases impossible, to define a distance metric to use for a particular type of data. The second general method, which is popular and very present in the literature, is model-based clustering. Such methods rely on the finite mixture model (see Chapter 2.1) which assumes that each observation comes from one of a number (G) of probability distributions. This reduces the problem of clustering to finding to which one of these G distributions each observation belongs. The effectiveness of model-based clustering comes from its flexibility. Specifically, the probability distribution can be chosen to allow high-dimensional data, a variety of data types, a data stream, or even a combination of two or all three of these. This thesis presents model-based clustering techniques for analyzing some of these complex, and not so complex, data types encountered today, including three-way data, with an emphasis on skewed three-way data, high- dimensional data, and clickstream data. A detailed outline is now given in Section 1.1.

1.1 Outline

1.1.1 Chapter 2

Chapter 2 will present a detailed background on model-based clustering, clustering skewed two-way (multivariate data), and clustering high-dimensional data. In addition, model selection, convergence, and performance criteria are discussed.

1.1.2 Chapters 3 to 6

Chapters 3 to 6 present a detailed development of methodology for clustering three-way data. Three-way data come in the form of matrices instead of traditional vectors,

and until relatively recently, there was a relative paucity of methods available for analyzing three-way data. Specifically, analyses were restricted to either vectorizing the data, or assuming matrix variate normality. The assumption of normality when analyzing three-way data, as in the two-way case, can be problematic in the presence of outliers or skewness. Chapter 3 develops a total of four skewed matrix variate distributions based on their multivariate counterparts. This work is based on two publications, Gallaugher and McNicholas (2017), and Gallaugher and McNicholas (2019c). Chapter 4, based on Gallaugher and McNicholas (2018b), then utilizes these four distributions in the mixture model context for model-based clustering and classification. To our knowledge, this is the first use of skewed matrix variate variate distributions in the mixture model paradigm.

Just like in the multivariate case, dimensionality can become a problem when analyzing three-way data due to the increase in the number of scale parameters. In Chapter 5, the mixture of factor analyzers model is extended to the matrix variate case. This model is called a mixture of matrix variate bilinear factor analyzers (MMVBFA) model. Chapter 6 then presents the skewed version of the mixtures of matrix variate bilinear factor analyzers using the four skewed distributions developed in Chapter 3. Chapters 5 and 6 are based on the publications Gallaugher and McNicholas (2018c) and Gallaugher and McNicholas (2019b), respectively.

1.1.3 Chapter 7

At this point in the thesis, the topics become more diverse. In Chapter 7, methodology for clustering and classification of clickstream data is presented. Many methods for modelling clickstream data are presented in the literature; however, very few of these

are in the area of model-based clustering. One such method is a mixture of discrete first order Markov chains. This type of methodology is useful for clickstream data from a website with multiple categories such as `amazon.com`, or a news website with categories such as weather, sports, breaking news, and so forth. The main drawback of this method is that it is unable take into account the amount of time spent on each website. This would be important, for example, if the website user accidentally entered the wrong category, and then immediately exit the category. Using a discrete time Markov chain would detect the entry to that category and may subsequently provide unhelpful product suggestions. The methodology presented herein allows the modelling of the amount of time spent in each category. This chapter is based on Gallaugher and McNicholas (2018a), available on arXiv.

1.1.4 Chapter 8

Chapter 8 presents a small extension of the mixtures of bilinear factor analyzers model by restricting the scale and factor loading matrices, creating a family of 64 models. This is essentially a matrix variate version of the parsimonious Gaussian mixture models presented by McNicholas and Murphy (2008), and allows for further parameter reduction and model flexibility. The material on which this chapter is based can be found in Gallaugher and McNicholas (2020).

1.1.5 Chapter 9

In a clustering scenario, imposing an assumption of multivariate normality can be problematic when the data is skewed. Because of this, many methods have been proposed in the literature for clustering skewed data; however, they generally fall

into one of two classes. The first is to consider a mixture model with skewed densities, and the second is to utilize a transformation to approximate normality alongside model-based clustering. Although the two methodologies are compared in their respective publications, there is no indication as to which situation one method might be preferable to another. In Chapter 9, measures of cluster overlap, skewness, and kurtosis are considered on benchmark data to extensively compare these two methods of clustering skewed data.

1.1.6 Chapter 10

The last topic presented in this thesis in Chapter 10 is in the area of co-clustering for high-dimensional two-way data. Co-clustering performs clustering on both the rows (observations) and columns (variables) of a data matrix. The result is the co-clustering of the data matrix into blocks with the observations in each block being independent and identically distributed. This is very useful for very high-dimensional datasets as the number of free parameters is independent of the dimensionality of the data, and because it can be performed when the number of variables exceeds the number of observations. However, the model is not very flexible, and increasing the flexibility requires an increase in the number of row-clusters and column-clusters which is generally not advisable. Herein, a parameter-wise co-clustering model is presented that allows for two different partitions in columns according to means and variances using the Gaussian distribution. This effectively improves the flexibility of the co-clustering model while still maintaining a high degree of parsimony. The material for this chapter is based on Gallaugher *et al.* (2018), available on arXiv.

Finally, potential areas of future work are discussed in Chapter 11.

Chapter 2

Background

2.1 Finite Mixture Models and Model-Based Clustering

Clustering and classification look at finding and analyzing underlying group structures in data. One common method used for clustering is model-based, and generally makes use of a G-component finite mixture model. A random variable $\mathbf{X}$ from a finite mixture model has density

$$f(\mathbf{x} \mid \boldsymbol{\vartheta}) = \sum_{g=1}^{G} \pi_g f_g(\mathbf{x} \mid \boldsymbol{\theta}_g),$$

where $\boldsymbol{\vartheta} = (\pi_1, \pi_2, \ldots, \pi_G, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_G)$, $f_g(\cdot)$ is the gth component density, and $\pi_g > 0$ is the gth mixing proportion such that $\sum_{g=1}^{G} \pi_g = 1$. McNicholas (2016a) traces the association between clustering and mixture models back to Tiedeman (1955), and the earliest use of a finite mixture model for clustering can be found in Wolfe (1965),

who uses a Gaussian mixture model. Other early work in this area can be found in Baum *et al.* (1970) and Scott and Symons (1971), and a recent review of model-based clustering is given by McNicholas (2016b).

Although the Gaussian mixture model is well-established for clustering, largely due to its mathematical tractability, quite some work has been done in the area of non-Gaussian mixtures. For example, some work has been done using symmetric component densities that parameterize concentration (tail weight), e.g., the t distribution (Peel and McLachlan, 2000; Andrews and McNicholas, 2011; Andrews *et al.*, 2011; Andrews and McNicholas, 2012; Lin *et al.*, 2014) and the power exponential distribution (Dang *et al.*, 2015). There are also several examples of mixtures of skewed distributions such as the NIG distribution (Karlis and Santourian, 2009; Subedi and McNicholas, 2014), the skew-t distribution (Lin, 2010; Vrbik and McNicholas, 2012, 2014; Lee and McLachlan, 2014; Murray *et al.*, 2014a,b), the shifted asymmetric Laplace distribution (Morris and McNicholas, 2013; Franczak *et al.*, 2014), the variance-gamma distribution (McNicholas *et al.*, 2017), and the generalized hyperbolic distribution (Browne and McNicholas, 2015).

Recently, there has been an interest in the mixtures of matrix variate distributions, e.g., Viroli (2011) and Anderlucci and Viroli (2015) consider multivariate longitudinal data with the matrix variate normal distribution and Doğru *et al.* (2016) consider a finite mixture of matrix variate t distributions.

2.2 Clustering and Classification for High Dimensional Data

Although the Gaussian mixture model is widely used, problems arise when the data dimensionality p increases. The main contribution to the number of free parameters is through the component covariance matrices $\boldsymbol{\Sigma}_g$. Therefore, as a starting point, many methods try to impose parsimonious constraints on $\boldsymbol{\Sigma}_g$. A detailed background is presented by Bouveyron and Brunet-Saumard (2014) and McNicholas (2016b).

In the multivariate case, the mixture of factor analyzers model is widely used. If $\mathbf{X}_i$ represents a p-dimensional random vector, with $\mathbf{x}_i$ as its realization, the factor analysis model for $\mathbf{X}_1, \ldots, \mathbf{X}_n$ is

$$\mathbf{X}_i = \boldsymbol{\mu} + \boldsymbol{\Lambda}\mathbf{U}_i + \boldsymbol{\varepsilon}_i,$$

where $\boldsymbol{\mu}$ is a location vector, $\boldsymbol{\Lambda}$ is a $p \times q$ matrix of factor loadings with $q < p$, $\mathbf{U}_i \sim \mathcal{N}_q(\mathbf{0}, \mathbf{I})$ denotes the latent factors, $\boldsymbol{\varepsilon}_i \sim \mathcal{N}_q(\mathbf{0}, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi} = \text{diag}(\psi_1, \psi_2, \ldots, \psi_p)$, and $\mathbf{U}_i$ and $\boldsymbol{\varepsilon}_i$ are each independently distributed and independent of one another. Under this model, the marginal distribution of $\mathbf{X}_i$ is $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{\Psi})$. Probabilistic principal component analysis (PPCA) arises as a special case with the isotropic constraint $\boldsymbol{\Psi} = \psi\mathbf{I}_p$ (Tipping and Bishop, 1999b).

Ghahramani and Hinton (1997) develop the mixture of factor analyzers model, which is a Gaussian mixture model with covariance structure $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}$. A small extension was presented by McLachlan and Peel (2000a), who utilize the more general structure $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g$. Tipping and Bishop (1999a) introduce the closely-related mixture of PPCAs with $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \psi_g\mathbf{I}$. McNicholas and Murphy

(2008) constructed a family of eight parsimonious Gaussian models by considering combinations of the constraints $\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}$, $\boldsymbol{\Psi}_g = \boldsymbol{\Psi}$ and $\boldsymbol{\Psi}_g = \psi_g \mathbf{I}$. McNicholas and Murphy (2010) and Bhattacharya and McNicholas (2014) extend the work McNicholas and Murphy (2008).

We note that under the most constrained model of McNicholas and Murphy (2008) there are a total of

$$\#\text{Params}_{\text{MFA}} = (G-1) + Gp + pq - q(q-1)/2 + 1 \tag{2.1}$$

free parameters. It is clear that although the number of free parameters associated with these models is linear in p, it is still nevertheless dependent on the dimension. Consequently, these models are still not suitable for very high dimensional data. Moreover, these methods may not be viable when $n > p$, which is common in applications such as gene expression data, word processing data, single nucleotide polymorphism data, etc.

There has also been work on extending the mixture of factor analyzers to other distributions, such as the skew-t distribution (Murray *et al.*, 2014b, 2017), the generalized hyperbolic distribution (Tortora *et al.*, 2016), the skew-normal distribution (Lin *et al.*, 2016), the variance-gamma distribution (McNicholas *et al.*, 2017), and others (e.g., Murray *et al.*, 2017).

Alternatively, Bouveyron *et al.* (2007) use the spectral decomposition of $\boldsymbol{\Sigma}_g$

$$\boldsymbol{\Sigma}_g = \mathbf{D}_g \boldsymbol{\Delta}_g \mathbf{D}_g',$$

where $\mathbf{D}_g$ is the orthogonal matrix of eigenvectors and $\mathbf{\Delta}_g$ is a diagonal matrix of corresponding eigenvalues for which they impose the structure $\mathbf{\Delta}_g = \text{diag}(a_{1g}, \ldots, a_{q_g g}, b_g, \ldots, b_g)$, where a_{kg} are the q_g largest eigenvalues and b_g is average of the remaining $p - q_g$ eigenvalues. This also greatly reduces the number of free parameters, i.e.,

$$\#\text{Params}_{\text{Bouveyron}} = (G-1) + Gp + \sum_{g=1}^{G} q_g[p - (q_g+1)/2] + \sum_{g=1}^{G} q_g + 2G. \qquad (2.2)$$

Again, however, the number of free parameters is dependent on the dimensionality of the data.

Finally, there are also variable selection procedures such as ℓ_1 penalization methods which take advantage of sparsity to perform variable selection and parameter estimation simultaneously. The first such proposed method is presented by Pan and Shen (2007) who consider equal, diagonal covariance matrices between groups and apply an ℓ_1 penalty to the mean vectors. A lasso method is then used for parameter estimation. This is extended by Zhou *et al.* (2009), who consider unconstrained covariance matrices and apply an ℓ_1 penalty for both the mean and covariance parameters. Although these methods are useful for dealing with the dimensionality problem, the ℓ_1 penalty shrinks the parameters, thus introducing bias, as discussed by Meynet and Maugis-Rabusseau (2012). Moreover, the Bayesian information criterion (BIC; Schwarz, 1978) may not be suitable for high-dimensional data. A detailed review of each of these methods is given by Biernacki and Maugis (2017).

2.3 Model Selection, Convergence, and Performance Criteria

2.3.1 Model Selection

In a general clustering scenario, the number of components (groups) G are not known *a priori*. It is, therefore, necessary to select an adequate number of components. There are two methods that are quite common in the literature. The first is the Bayesian information criterion (BIC; Schwarz, 1978), which is defined as

$$\text{BIC} = 2\ell(\hat{\boldsymbol{\vartheta}}) - p\log N, \tag{2.3}$$

where $\ell(\hat{\boldsymbol{\vartheta}})$ is the maximized log-likelihood, N is the number of observations, and p is the number of free parameters. Note that the BIC can sometimes be defined as the negative of (2.3).

Another criterion common in the literature is the integrated completed likelihood (ICL; Biernacki *et al.*, 2000). The ICL can be approximated as

$$\text{ICL} \approx \text{BIC} + 2\sum_{i=1}^{n_g}\sum_{g=1}^{G}\text{MAP}(\hat{z}_{ig})\log\hat{z}_{ig},$$

where

$$\text{MAP}(\hat{z}_{ig}) = \begin{cases} 1 & \text{if } \arg\max_{h=1,\ldots,G}\{\hat{z}_{ig}\} = g, \\ 0 & \text{otherwise.} \end{cases}$$

The ICL can be viewed as penalized version of the BIC, where the penalty is for uncertainty in the component membership. The the ICL is considered, in general the

results are very similar, and therefore for the analyses herein we only present results using the BIC.

2.3.2 Convergence Criterion

Parameter estimation for all methods presented herein utilize a form of the expectation maximization (EM; Dempster *et al.*, 1977) algorithm. Determining when the algorithm has converged is a difficult task. A simple convergence criterion is based on lack of progress in the log-likelihood, where the algorithm is terminated when $l^{(t+1)} - l^{(t)} < \epsilon$, where $\epsilon > 0$ is a small number. Oftentimes, however, the likelihood can plateau before increasing again, thus using lack of progress would terminate the algorithm prematurely (see McNicholas *et al.*, 2010, for examples). Another option, and one that is used for our analyses, is a criterion based on the Aitken acceleration (Aitken, 1926). The Aitken acceleration at iteration t is

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

where $l^{(t)}$ is the observed likelihood at iteration t. We then have an estimate, at iteration $t+1$, of the log-likelihood after many iterations:

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

(Böhning *et al.*, 1994; Lindsay, 1995). As suggested by McNicholas *et al.* (2010), the algorithm is terminated when $l_{\infty}^{(k+1)} - l^{(k)} \in (0, \epsilon)$. In general, we set the value of ϵ based on the magnitude of the log-likelihood. Specifically, for each AECM algorithm in our analyses, after five iterations we set ϵ to a value three orders of magnitude

lower than the log-likelihood. Unless stated otherwise, this criterion is used for all simulations and real data analyses.

2.3.3 Classification Performance

To assess classification performance, the adjusted Rand index (ARI; Hubert and Arabie, 1985) is used. The ARI is the Rand index (Rand, 1971) corrected for chance agreement. The ARI compares two different partitions—in our case, predicted and true classifications—and takes a value of 1 if there is perfect agreement. The expected value of the ARI under random classification is 0. A detailed discussion on the ARI is provided by Steinley (2004).

2.4 Inverse and Generalized Inverse Gaussian Distributions

The derivation of the matrix distributions and parameter estimation discussed in Chapters 3 and 4, will rely heavily on the generalized inverse Gaussian distribution, and to a lesser extent the inverse Gaussian distribution. A random variable Y follows an inverse Gaussian distribution if its probability density function is of the form

$$f(y|\delta,\gamma) = \frac{\delta}{\sqrt{2\pi}} \exp\{\delta\gamma\} y^{-\frac{3}{2}} \exp\left\{-\frac{1}{2}\left(\frac{\delta^2}{y} + \gamma^2 y\right)\right\},$$

for $\delta, \gamma > 0$. For notational purposes, we will denote this distribution by $\text{IG}(\delta, \gamma)$.

The generalized inverse Gaussian distribution has two different parameterizations, both of which will be useful. A random variable Y has a generalized inverse Gaussian

distribution parameterized by $a, b > 0$ and $\lambda \in \mathbb{R}$, denoted by $\text{GIG}(a, b, \lambda)$ if its probability density function can be written as

$$f(y|a, b, \lambda) = \frac{(a/b)^{\frac{\lambda}{2}} y^{\lambda-1}}{2K_\lambda(\sqrt{ab})} \exp\left\{-\frac{ay + b/y}{2}\right\}$$

where

$$K_\lambda(u) = \frac{1}{2}\int_0^\infty y^{\lambda-1} \exp\left\{-\frac{u}{2}\left(y + \frac{1}{y}\right)\right\} dy$$

is the modified Bessel function of the third kind with index λ. Some expectations of functions of a GIG random variable with this parameterization have a mathematically tractable form, e.g.,

$$\mathbb{E}(Y) = \sqrt{\frac{b}{a}} \frac{K_{\lambda+1}(\sqrt{ab})}{K_\lambda(\sqrt{ab})}, \tag{2.4}$$

$$\mathbb{E}\left(1/Y\right) = \sqrt{\frac{a}{b}} \frac{K_{\lambda+1}(\sqrt{ab})}{K_\lambda(\sqrt{ab})} - \frac{2\lambda}{b}, \tag{2.5}$$

$$\mathbb{E}(\log Y) = \log\left(\sqrt{\frac{b}{a}}\right) + \frac{1}{K_\lambda(\sqrt{ab})} \frac{\partial}{\partial\lambda} K_\lambda(\sqrt{ab}). \tag{2.6}$$

Although this parameterization of the GIG distribution will be useful for parameter estimation, for the purposes of deriving the density of the matrix variate generalized hyperbolic distribution, it is more useful to take the parameterization

$$g(y|\omega, \eta, \lambda) = \frac{(w/\eta)^{\lambda-1}}{2\eta K_\lambda(\omega)} \exp\left\{-\frac{\omega}{2}\left(\frac{w}{\eta} + \frac{\eta}{w}\right)\right\}, \tag{2.7}$$

where $\omega = \sqrt{ab}$ and $\eta = \sqrt{a/b}$ (Browne and McNicholas, 2015). For notational clarity, we will denote the parameterization given in (2.7) by $\text{I}(\omega, \eta, \lambda)$.

2.5 Variance-Mean Mixtures

A p-variate random vector $\mathbf{X}$ defined in terms of a variance-mean mixture, has a probability density function of the form

$$f(\mathbf{x}) = \int_0^\infty \phi_p(\mathbf{x}|\boldsymbol{\mu} + w\boldsymbol{\alpha}, w\boldsymbol{\Sigma})h(w|\boldsymbol{\theta})dw,$$

where the random variable $W > 0$ has density function $h(w|\boldsymbol{\theta})$, and $\phi_p(\cdot)$ represents the density function of the p-variate Gaussian distribution. This representation is equivalent to writing

$$\mathbf{X} = \boldsymbol{\mu} + W\boldsymbol{\alpha} + \sqrt{W}\mathbf{V}, \tag{2.8}$$

where $\boldsymbol{\mu}$ is a location parameter, $\boldsymbol{\alpha}$ is the skewness, $\mathbf{V} \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma}$ as the scale matrix, and W has density function $h(w|\boldsymbol{\theta})$. Note that W and $\mathbf{V}$ are independent. Many multivariate distributions can be obtained through a variance mean mixture by changing the distribution of W. For example, the multivariate skew-t distribution with ν degrees of freedom arises as a special case with $W \sim \text{IG}\left(\frac{\nu}{2}, \frac{\nu}{2}\right)$, where IG$(\cdot)$ denotes the inverse Gamma distribution with density function

$$f(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left\{-\frac{\beta}{x}\right\}.$$

The p-dimensional generalized hyperbolic distribution, $\text{GH}_p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}, \psi, \chi, \lambda)$, as given in McNeil *et al.* (2005), was shown to arise as a special case of (2.8) by taking $W \sim \text{GIG}(\psi, \chi, \lambda)$. However, there was a restriction that $|\boldsymbol{\Sigma}| = 1$. Simply relaxing this constraint results in an identifiability problem. In Browne and McNicholas (2015), this was discussed, and the authors proposed the reparameterization $\omega = \sqrt{\psi\chi}, \eta =$

$\sqrt{\chi/\psi}$. The representation of $\mathbf{X}$ is then as in (2.8), with $W \sim \text{I}(\omega, 1, \lambda)$.

The p-dimensional variance-gamma distribution, $\text{VG}_p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}, \lambda, \psi)$, results as a limiting case of the generalized hyperbolic by taking $\lambda > 0$, and $\chi \to 0$. The precise details can be found in McNicholas *et al.* (2017); in essence, the variance-gamma distribution also arises as a special case of (2.8), with $W \sim \text{gamma}(\lambda, \psi/2)$, where $\text{gamma}(a, b)$ denotes the gamma distribution with density

$$f(w|a, b) = \frac{b^a}{\Gamma(a)} w^{a-1} \exp\{-bw\},$$

where $a, b > 0$. However, we again have an identifiability issue using this representation if we remove the constraint $|\boldsymbol{\Sigma}| = 1$. In McNicholas *et al.* (2017), the authors propose setting $\mathbb{E}(W) = 1$, resulting in the reparameterization $\gamma := \lambda = \psi/2$.

Finally, we have the p-dimensional normal inverse Gaussian (NIG) distribution, $\text{NIG}_p(\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}, \delta, \gamma)$. In Karlis and Santourian (2009), the authors derived the p-dimensional NIG distribution using a variance-mean mixture with $W \sim \text{IG}(\delta, \gamma)$. However, there was once again a restriction on the determinant of $\boldsymbol{\Sigma}$. To remove this restriction and maintain identifiability, Karlis and Santourian (2009) set $\delta = 1$, and set $\kappa = \gamma$.

2.6 Analysis of Three-Way Data

2.6.1 Examples of Three-Way Data

As already discussed, many methods exist for clustering multivariate data involving skewness or outliers, for clustering high dimensional data, and for dealing with data

of different types. However, there was, until recently, a relative paucity of methods for clustering three-way data. Consider that three-way data comes in the form of three-dimensional arrays, so that each observation is a matrix instead of a vector. With modern data, there are many emerging data types that naturally come as matrices. For example, a greyscale image comes in the form of pixel intensity matrix. In Figure 5.2, we show two different greyscale images. The first is taken from the MNIST dataset of handwritten digits, in this case a six, LeCun et al. (1998). The second is a face from the famous Olivetti faces dataset from the R package `RnavGraphImageData` (Waddell and Oldford, 2013). A second example of three-way data that is becoming very common in health studies, is multivariate longitudinal data. Longitudinal data consists of a measurement of interest that is collected over time and creating a vector for each individual. However, many times multiple variables are collected on an individual over time, and thus creating a vector for each column and resulting in a matrix observation on each individual.

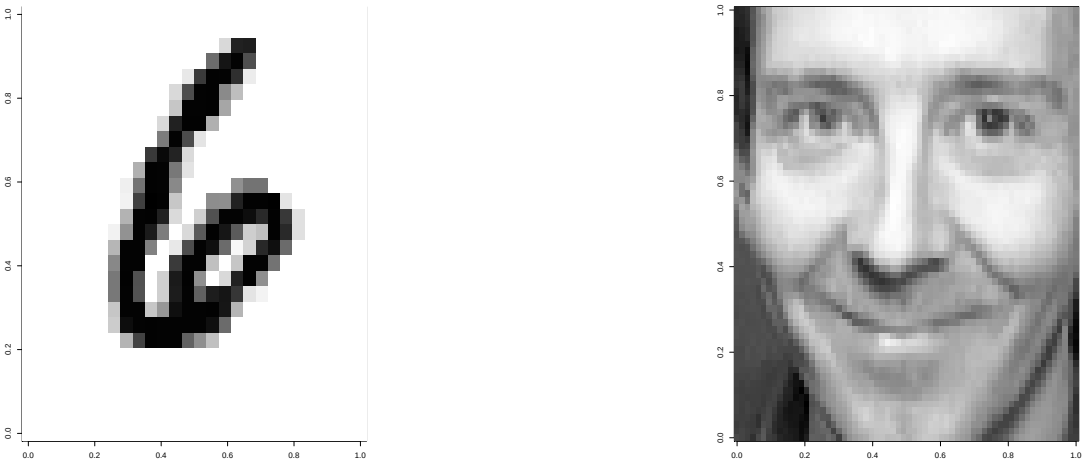


Figure 2.1: Two examples of greyscale images.

2.6.2 Matrix Variate Distributions

Three-way data such as multivariate longitudinal data or greyscale image data can be easily modelled using a matrix variate distribution. There are many examples of such distributions presented in the literature, the most notable being the matrix variate normal distribution. For notional clarity, $\mathbf{X}$ is used to denote a realization of a random matrix $\mathscr{X}$ unless stated otherwise. An $n \times p$ random matrix $\mathscr{X}$ follows an $n \times p$ matrix variate normal distribution with location parameter $\mathbf{M}$ and scale matrices $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ of dimensions $n \times n$ and $p \times p$, respectively, denoted by $\mathcal{N}_{n\times p}(\mathbf{M}, \mathbf{\Sigma}, \mathbf{\Psi})$ if the density of $\mathscr{X}$ can be written as

$$f(\mathbf{X} \mid \mathbf{M}, \mathbf{\Sigma}, \mathbf{\Psi}) = \frac{1}{(2\pi)^{\frac{np}{2}} |\mathbf{\Sigma}|^{\frac{p}{2}} |\mathbf{\Psi}|^{\frac{n}{2}}} \exp\left\{-\frac{1}{2} \operatorname{tr}\left(\mathbf{\Sigma}^{-1}(\mathbf{X}-\mathbf{M})\mathbf{\Psi}^{-1}(\mathbf{X}-\mathbf{M})'\right)\right\}. \tag{2.9}$$

A well-known, and useful, property of the matrix variate normal distribution (Harrar and Gupta, 2008) is

$$\mathscr{X} \sim \mathcal{N}_{n\times p}(\mathbf{M}, \mathbf{\Sigma}, \mathbf{\Psi}) \iff \operatorname{vec}(\mathscr{X}) \sim \mathcal{N}_{np}(\operatorname{vec}(\mathbf{M}), \mathbf{\Psi} \otimes \mathbf{\Sigma}), \tag{2.10}$$

where $\mathcal{N}_{np}(\cdot)$ is the multivariate normal density with dimension np, $\operatorname{vec}(\mathbf{M})$ is the vectorization of $\mathbf{M}$, and $\otimes$ is the Kronecker product.

The matrix variate normal has many elegant mathematical properties that have made it so popular, e.g., Viroli (2011) uses a mixture of matrix variate normal distributions for clustering. However, there are non-normal examples such as the Wishart distribution (Wishart, 1928) and the skew-normal distribution, e.g., Chen and Gupta (2005), Domínguez-Molina *et al.* (2007), and Harrar and Gupta (2008). More information on matrix variate distributions can be found in Gupta and Nagar (1999).

2.6.3 Note on Identifiability

It is important to note that the estimates for $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}$ for the matrix variate normal distribution are only unique up to a strictly positive constant. Therefore, to eliminate the identifiability issue, a constraint needs to be imposed on $\boldsymbol{\Sigma}$ or $\boldsymbol{\Psi}$. Anderlucci and Viroli (2015), suggest taking the trace of $\boldsymbol{\Psi}$ to be equal to p; however, it is much simpler to set the first diagonal element of $\boldsymbol{\Sigma}$ to be 1 and this is the constraint we use in the analyses herein.

Discussion of identifiability would not be complete without mention of the label switching problem in the case of clustering. This well-known problem is due to the invariance of the mixture model to relabelling of the components (Redner and Walker, 1984; Stephens, 2000). While the label switching problem is a real issue in the Bayesian paradigm (see Stephens, 2000; Celeux *et al.*, 2000, for some discussion), it is of no practical concern for the work carried out herein. However, it is a theoretical identifiability issue and we note that it be resolved by specifying some ordering on the model parameters, e.g., simply requiring that $\pi_1 > \pi_2 > \cdots > \pi_G$ often works and ordering on other parameters can be imposed as needed.

2.6.4 Benefits Over Vectorization

One alternative to matrix variate analysis for matrix variate data is to consider the vec- torization of the data and perform multivariate techniques. However, the benefits of using matrix variate methods are twofold. The first being specifically for the case of multivariate longitudinal data. Performing the analysis using a matrix variate model has the benefit of simultaneously considering the temporal covariances (via $\boldsymbol{\Sigma}$) as well as the covariances for the variables (via $\boldsymbol{\Psi}$). Performing multivariate analysis

on the vectorization of the data would not have this benefit without imposing some structure on the scale matrix. The second benefit is the reduction in the number of parameters. If the matrix variate data is $n \times p$, vectorization would result in np dimensional vectors, therefore resulting in $(n^2p^2 + np)/2$ free scale parameters when using multivariate analysis. However, when using a matrix variate model, there are two lower dimensional matrices that comprise the scale parameters with a total of $(n^2 + p^2 + n + p)/2$ free scale parameters. Thus, for $n = p$, there is a reduction from quartic to quadratic complexity in n and, for almost all values of n and p, there will be a (often substantial) reduction in the number of free scale parameters.

Chapter 3

Four Skewed Matrix Variate Distributions

3.1 Matrix Variate Skew-t Distribution

3.1.1 Derivation

Analogous to the multivariate case, an $n \times p$ random matrix $\mathscr{X}$ has a variance-mean mixture representation if

$$\mathscr{X} = \mathbf{M} + W\mathbf{A} + \sqrt{W}\mathscr{V}, \tag{3.1}$$

where $\mathbf{M}$ and $\mathbf{A}$ are $n \times p$ location and skewness matrices, respectively, $\mathscr{V} \sim \mathcal{N}_{n\times p}(\mathbf{0}, \mathbf{\Sigma}, \mathbf{\Psi})$, and $W \in \mathbb{R}^+$ is a positive random variable. We will say that an $n \times p$ random matrix $\mathscr{X}$ has a matrix variate skew-t distribution, $\text{MVST}_{n\times p}(\mathbf{M}, \mathbf{A}, \mathbf{\Sigma}, \mathbf{\Psi}, \nu)$, if

$W \sim \text{IG}\left(\frac{\nu}{2}, \frac{\nu}{2}\right)$. Analogous to its multivariate counterpart, $\mathbf{M}$ is a location matrix, $\mathbf{A}$ is a skewness matrix, $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}$ are scale matrices, and ν is the degrees of freedom.

It then follows that

$$\mathscr{X}|w \sim \mathcal{N}_{n\times p}\left(\mathbf{M} + w\mathbf{A}, w\boldsymbol{\Sigma}, \boldsymbol{\Psi}\right)$$

and thus the joint density of $\mathscr{X}$ and W is

$$\begin{aligned} f(\mathbf{X}, w \mid \boldsymbol{\vartheta}) &= f(\mathbf{X} \mid w) f(w) \\ &= \frac{\frac{\nu}{2}^{\frac{\nu}{2}}}{(2\pi)^{\frac{np}{2}}|\boldsymbol{\Sigma}|^{\frac{p}{2}}|\boldsymbol{\Psi}|^{\frac{n}{2}}\Gamma(\frac{\nu}{2})} w^{-\frac{\nu+np}{2}-1} \\ &\quad \times \exp\left\{-\frac{1}{2w}\left(\text{tr}\left(\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{M}-w\mathbf{A})\boldsymbol{\Psi}^{-1}(\mathbf{X}-\mathbf{M}-w\mathbf{A})'\right)+\nu\right)\right\}, \end{aligned} \tag{3.2}$$

where $\boldsymbol{\vartheta} = (\mathbf{M}, \mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}, \nu)$. We note that the exponential term in (3.2) can be written as

$$\exp\left\{\text{tr}(\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{M})\boldsymbol{\Psi}^{-1}\mathbf{A}')\right\} \times \exp\left\{-\frac{1}{2}\left[\frac{\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi})+\nu}{w} + w\rho(\mathbf{A};\boldsymbol{\Sigma},\boldsymbol{\Psi})\right]\right\},$$

where

$$\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi}) = \text{tr}(\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{M})\boldsymbol{\Psi}^{-1}(\mathbf{X}-\mathbf{M})') \quad \text{and} \quad \rho(\mathbf{A};\boldsymbol{\Sigma},\boldsymbol{\Psi}) = \text{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{A}\boldsymbol{\Psi}^{-1}\mathbf{A}').$$

Therefore, the marginal density of $\mathscr{X}$ is

$$\begin{aligned} f(\mathbf{X}) &= \int_0^\infty f(\mathbf{X}, w) dw \\ &= \frac{\frac{\nu}{2}^{\frac{\nu}{2}}}{(2\pi)^{\frac{np}{2}} |\boldsymbol{\Sigma}|^{\frac{p}{2}} |\boldsymbol{\Psi}|^{\frac{n}{2}} \Gamma(\frac{\nu}{2})} \exp\left\{ \operatorname{tr}(\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{M})\boldsymbol{\Psi}^{-1}\mathbf{A}')\right\} \\ &\quad \times \int_0^\infty w^{-\frac{\nu+np}{2}-1} \exp\left\{ -\frac{1}{2}\left[\frac{\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi})+\nu}{w} + w\rho(\mathbf{A},\boldsymbol{\Sigma},\boldsymbol{\Psi})\right]\right\} dw. \end{aligned}$$

Making the change of variables given by

$$y = \frac{\sqrt{\rho(\mathbf{A},\boldsymbol{\Sigma},\boldsymbol{\Psi})}}{\sqrt{\delta(X;\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi})+\nu}} w$$

we can write

$$\begin{aligned} f_{\text{MVST}}(\mathbf{X} \mid \boldsymbol{\vartheta}) = &\frac{2\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}} \exp\left\{ \operatorname{tr}(\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{M})\boldsymbol{\Psi}^{-1}\mathbf{A}')\right\}}{(2\pi)^{\frac{np}{2}} |\boldsymbol{\Sigma}|^{\frac{p}{2}} |\boldsymbol{\Psi}|^{\frac{n}{2}} \Gamma(\frac{\nu}{2})} \left(\frac{\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi})+\nu}{\rho(\mathbf{A},\boldsymbol{\Sigma},\boldsymbol{\Psi})}\right)^{-\frac{\nu+np}{4}} \\ &\times K_{-\frac{\nu+np}{2}}\left(\sqrt{[\rho(\mathbf{A},\boldsymbol{\Sigma},\boldsymbol{\Psi})]\,[\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi})+\nu]}\right). \end{aligned}$$

The density of $\mathscr{X}$, as derived here, be considered a matrix variate extension of the multivariate skew-t density used by Murray *et al.* (2014b,a). For the purposes of parameter estimation, note that the conditional density of W is

$$\begin{aligned} f(w \mid \mathbf{X}) &= \frac{f(\mathbf{X} \mid w) f(w)}{f(\mathbf{X})} \\ &= \frac{[\rho(\mathbf{A},\boldsymbol{\Sigma},\boldsymbol{\Psi})/(\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi})+\nu)]^{\frac{\lambda}{2}} w^{\lambda-1}}{2K_\lambda(\sqrt{\rho(\mathbf{A},\boldsymbol{\Sigma},\boldsymbol{\Psi})[\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi})+\nu]})} \\ &\quad \times \exp\left\{ -\frac{\rho(\mathbf{A},\boldsymbol{\Sigma},\boldsymbol{\Psi})w + [\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi})+\nu]/w}{2}\right\}. \end{aligned}$$

Therefore, $W \mid \mathbf{X} \sim \text{GIG}\left(\rho(\mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}), \delta(\mathbf{X}; \mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}) + \nu, \lambda\right)$, where $\lambda = -(\nu + np)/2$.

Finally, we note that

$$\mathscr{X} \sim \text{MVST}_{n\times p}(\mathbf{M}, \mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}, \nu) \iff \text{vec}(\mathscr{X}) \sim \text{MST}_{np}(\text{vec}(\mathbf{M}), \text{vec}(\mathbf{A}), \boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}, \nu), \tag{3.3}$$

where $\text{MST}_{np}(\cdot)$ denotes the multivariate skew-t distribution with location parameter $\text{vec}(\mathbf{M})$, skewness parameter $\text{vec}(\mathbf{A})$, scale matrix $\boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}$, and ν degrees of freedom. This can be easily seen from the representation given in (3.1) and the property of the matrix normal distribution given in (2.10). Note that the normal variance-mean mixture representation (3.1) as well as the relationship with the multivariate skew-t distribution (3.3) present two convenient methods to generate random matrices from the matrix variate skew t distribution.

Note that parameter estimation for the matrix variate skew-t is performed via an expectation conditional maximization (ECM) algorithm. To reduce space, the details are not presented here, as it is equivalent to the algorithm for a single component mixture of skew-t distributions presented in detail in Chapter 4.

3.1.2 Simulations

We conducted two simulations to look at the estimation of the parameters. In both simulations we took 50 different datasets of size 100, from a 3×4 matrix skew t

distribution. Also, in both simulations, we took

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.1 \\ 0.5 & 1 & 0.5 \\ 0.1 & 0.5 & 1 \end{pmatrix} \quad \boldsymbol{\Psi} = \begin{pmatrix} 1 & -0.5 & 0.5 & 0.1 \\ -0.5 & 1 & -0.5 & 0.6 \\ 0.5 & -0.5 & 1 & -0.4 \\ 0.1 & 0.6 & -0.4 & 1 \end{pmatrix}$$

and $\nu = 4$. In simulation 3.1, we took the location and skewness matrix to be $\mathbf{M}_1$, $\mathbf{A}_1$ respectively and $\mathbf{M}_2$, $\mathbf{A}_2$ in simulation 3.2, where

$$\mathbf{M}_1 = \begin{pmatrix} 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix} \qquad \mathbf{A}_1 = \begin{pmatrix} 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 \end{pmatrix}$$

$$\mathbf{M}_2 = \begin{pmatrix} 1 & -6 & -1 & -1 \\ -3 & 5 & -4 & 1 \\ 1 & -4 & -1 & 5 \end{pmatrix} \quad \mathbf{A}_2 = \begin{pmatrix} 1 & -1 & 0.5 & 0 \\ 0.5 & -0.5 & 0.5 & 0.5 \\ 0 & 0 & 0.5 & 0 \end{pmatrix}$$

In Figures 3.1 and 3.2, we show line plots of the marginals for each column (labelled V1, V2, V3, V4) of a typical dataset from simulation 3.1 and 3.2 respectively. The dashed red lines denote the mean.

In Figure 3.1, the skewness in columns 1, 2, and 4, for simulation 3.1, is very prominent when visually compared to column 3 which has zero skewness. The skewness is also apparent in the lineplots for simulation 3.2, however, because the values of the skewness are generally less than those for simulation 3.1, it is not as prominent.

In Table 3.1, we show a table with the component wise means of the parameters

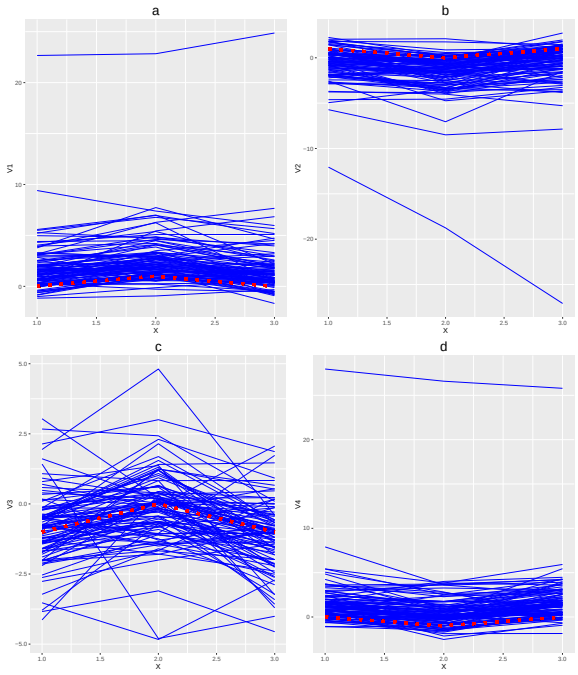


Figure 3.1: Typical marginals of the matrix variate skew-t distribution for Simulation 3.1 for (a) V1, (b) V2, (c) V3 and (d) V4. The red dashed lines denote the mean.

as well as the component wise standard deviations. We see that the estimates of the mean matrix and skewness matrix are very close to the true value for both simulations. Moreover, we see that the estimates of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}$ correspond approximately to the their true values as well, and thus so would the Kronecker product, which we don't show here to save space.

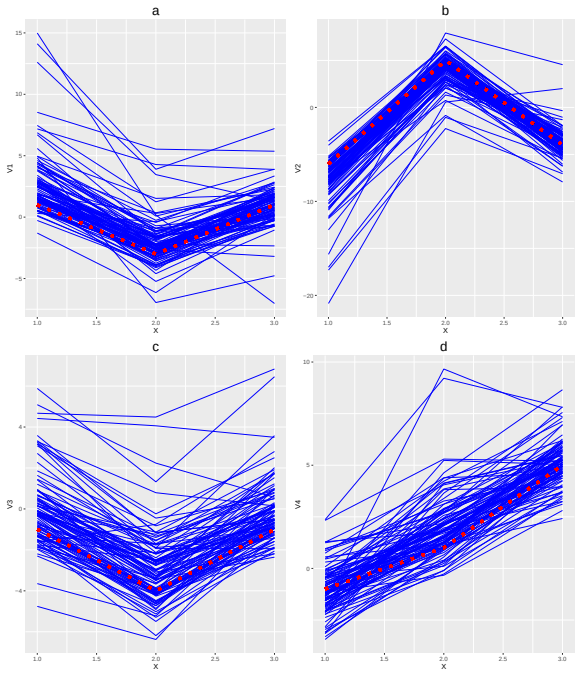


Figure 3.2: Typical marginals of the matrix variate skew-t distribution for Simulation 3.2 for (a) V1, (b) V2, (c) V3 and (d) V4. The red dashed lines denote the mean.

3.2 Three More Skewed Matrix Variate Distributions

3.2.1 Matrix Variate Generalized Hyperbolic Distribution

In a similar manner to the matrix variate skew-t distribution derive the density for a matrix variate generalized inverse Gaussian distribution. In this case, to avoid the indentifiability issue discussed in Browne and McNicholas (2015), we take $W \sim \text{I}(\omega, 1, \lambda)$, where ω is a concentration parameter and λ is the index parameter. The

Table 3.1: Component wise averages and standard deviations for the estimated parameters for simulations 3.1 and 3.2 for the matrix variate skew-t distribution.

Sim	$\mathbf{M}$(sd)	$\mathbf{A}$(sd)	$\mathbf{\Sigma}$(sd)	$\mathbf{\Psi}$(sd)	$\nu(sd)$
1	$\begin{bmatrix} -0.04 & 1.04 & -1.01 & -0.02 \\ 1.01 & 0.03 & 0.03 & -1.01 \\ 0.01 & 1.04 & -0.97 & -0.01 \end{bmatrix}$ $\left(\begin{bmatrix} 0.045 & 0.031 & 0.030 & 0.031 \\ 0.033 & 0.047 & 0.025 & 0.023 \\ 0.034 & 0.042 & 0.019 & 0.021 \end{bmatrix}\right)$	$\begin{bmatrix} 1.07 & -1.06 & 0.03 & 1.04 \\ 1.01 & -1.04 & -0.01 & 1.03 \\ 1.02 & -1.03 & -0.01 & 1.04 \end{bmatrix}$ $\left(\begin{bmatrix} 0.039 & 0.030 & 0.014 & 0.032 \\ 0.031 & 0.037 & 0.013 & 0.028 \\ 0.033 & 0.040 & 0.008 & 0.029 \end{bmatrix}\right)$	$\begin{bmatrix} 1.00 & 0.50 & 0.10 \\ 0.50 & 1.02 & 0.52 \\ 0.10 & 0.52 & 1.02 \end{bmatrix}$ $\left(\begin{bmatrix} 0.000 & 0.043 & 0.056 \\ 0.043 & 0.096 & 0.073 \\ 0.056 & 0.073 & 0.119 \end{bmatrix}\right)$	$\begin{bmatrix} 0.98 & -0.48 & 0.48 & 0.11 \\ -0.48 & 0.96 & -0.48 & 0.58 \\ 0.48 & -0.48 & 0.97 & -0.39 \\ 0.11 & 0.58 & -0.39 & 0.99 \end{bmatrix}$ $\left(\begin{bmatrix} 0.16 & 0.10 & 0.09 & 0.06 \\ 0.10 & 0.14 & 0.09 & 0.09 \\ 0.09 & 0.09 & 0.12 & 0.07 \\ 0.06 & 0.09 & 0.07 & 0.13 \end{bmatrix}\right)$	4.22 (0.63)
2	$\begin{bmatrix} 0.99 & -6.01 & -0.99 & -1.02 \\ -2.98 & 4.98 & -3.97 & 0.96 \\ 1.00 & -3.99 & -0.98 & 4.99 \end{bmatrix}$ $\left(\begin{bmatrix} 0.029 & 0.033 & 0.028 & 0.023 \\ 0.048 & 0.032 & 0.041 & 0.025 \\ 0.031 & 0.038 & 0.036 & 0.022 \end{bmatrix}\right)$	$\begin{bmatrix} 1.03 & -1.02 & 0.51 & 0.01 \\ 0.50 & -0.51 & 0.49 & 0.52 \\ 0.01 & -0.02 & 0.50 & 0.00 \end{bmatrix}$ $\left(\begin{bmatrix} 0.027 & 0.033 & 0.016 & 0.010 \\ 0.022 & 0.018 & 0.020 & 0.018 \\ 0.015 & 0.016 & 0.017 & 0.013 \end{bmatrix}\right)$	$\begin{bmatrix} 1.00 & 0.48 & 0.08 \\ 0.48 & 0.99 & 0.49 \\ 0.08 & 0.49 & 1.01 \end{bmatrix}$ $\left(\begin{bmatrix} 0.000 & 0.049 & 0.040 \\ 0.049 & 0.094 & 0.070 \\ 0.040 & 0.070 & 0.119 \end{bmatrix}\right)$	$\begin{bmatrix} 1.01 & -0.50 & 0.50 & 0.10 \\ -0.50 & 0.99 & -0.49 & 0.58 \\ 0.50 & -0.49 & 0.98 & -0.38 \\ 0.10 & 0.58 & -0.38 & 0.97 \end{bmatrix}$ $\left(\begin{bmatrix} 0.137 & 0.086 & 0.083 & 0.063 \\ 0.086 & 0.153 & 0.084 & 0.111 \\ 0.083 & 0.084 & 0.121 & 0.069 \\ 0.063 & 0.111 & 0.069 & 0.142 \end{bmatrix}\right)$	4.22 (0.92)

density of $\mathbf{X}$ in this case is

$$f_{\text{MVGH}}(\mathbf{X}|\boldsymbol{\vartheta}) = \frac{\exp\left\{\operatorname{tr}(\mathbf{\Sigma}^{-1}(\mathbf{X}-\mathbf{M})\mathbf{\Psi}^{-1}\mathbf{A}')\right\}}{(2\pi)^{\frac{np}{2}}|\mathbf{\Sigma}|^{\frac{p}{2}}|\mathbf{\Psi}|^{\frac{n}{2}}K_{\lambda}(\omega)}\left(\frac{\delta(\mathbf{X};\mathbf{M},\mathbf{\Sigma},\mathbf{\Psi})+\omega}{\rho(\mathbf{A},\mathbf{\Sigma},\mathbf{\Psi})+\omega}\right)^{\frac{(\lambda-\frac{np}{2})}{2}} \times K_{(\lambda-np/2)}\left(\sqrt{[\rho(\mathbf{A},\mathbf{\Sigma},\mathbf{\Psi})+\omega]\,[\delta(\mathbf{X};\mathbf{M},\mathbf{\Sigma},\mathbf{\Psi})+\omega]}\right),$$

where $\omega > 0$ is a concentration parameter, and $\lambda \in \mathbb{R}$ is an index parameter.

We note that the density of $\mathscr{X}$, as derived here, is similar to that in Browne and McNicholas (2015), and we denote this distribution by $\text{MVGH}_{n\times p}(\mathbf{M},\mathbf{A},\mathbf{\Sigma},\mathbf{\Psi},\lambda,\omega)$. For the purposes of parameter estimation, note that the conditional density of W is

$$\begin{aligned} f(w|\mathbf{X}) &= \frac{f(\mathbf{X}|w)f(w)}{f(\mathbf{X})} \\ &= \left(\frac{\rho(\mathbf{A},\mathbf{\Sigma},\mathbf{\Psi})+\omega}{\delta(\mathbf{X};\mathbf{M},\mathbf{\Sigma},\mathbf{\Psi})+\omega}\right)^{\frac{(\lambda-np/2)}{2}} \frac{w^{\lambda-np/2-1}}{2K_{(\lambda-np/2)}\sqrt{[\rho(\mathbf{A},\mathbf{\Sigma},\mathbf{\Psi})+\omega]\,[\delta(\mathbf{X};\mathbf{M},\mathbf{\Sigma},\mathbf{\Psi})+\omega]}} \\ &\qquad \times \exp\left\{-\frac{(\rho(\mathbf{A},\mathbf{\Sigma},\mathbf{\Psi})+\omega)w + [\delta(\mathbf{X};\mathbf{M},\mathbf{\Sigma},\mathbf{\Psi})+\omega]/w}{2}\right\}. \end{aligned}$$

Therefore, $W|\mathbf{X} \sim \text{GIG}\left(\rho(\mathbf{A},\mathbf{\Sigma},\mathbf{\Psi})+\omega, \delta(\mathbf{X};\mathbf{M},\mathbf{\Sigma},\mathbf{\Psi})+\omega, \lambda-np/2\right)$.

Note that a multiple scaled matrix variate generalized hyperbolic distribution was derived by Thabane and Safiul Haq (2004). While the distribution they derive is sometimes referred to as a matrix variate generalized hyperbolic distribution, the model of Thabane and Safiul Haq (2004) is in fact multiple scaled — a fact that may be confirmed by observing that they use a matrix variate distribution for the mixing variable $\mathbf{W}$. Not only does this mean that the distribution presented by Thabane and Safiul Haq (2004) is different to the matrix variate generalized hyperbolic distribution presented herein, but it also means that neither one of these distributions is a special case of the other. Some useful details about the multiple scaled generalized hyperbolic distribution are given by McNicholas (2016a, Chp. 7).

3.2.2 Matrix Variate Variance-Gamma Distribution

We now derive the density of a matrix variate variance-gamma distribution in much the same way as the generalized hyperbolic case. However, we now take $W \sim \text{gamma}(\gamma, \gamma)$, resulting in the joint distribution

$$\begin{aligned} f(\mathbf{X}, w|\boldsymbol{\vartheta}) =& \frac{\gamma^{\gamma}}{(2\pi)^{\frac{np}{2}}|\boldsymbol{\Sigma}|^{\frac{p}{2}}|\boldsymbol{\Psi}|^{\frac{n}{2}}\Gamma(\gamma)} w^{\gamma-\frac{np}{2}-1} \\ & \times \exp\left\{-\frac{1}{2w}\operatorname{tr}\left(\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{M}-w\mathbf{A})\boldsymbol{\Psi}^{-1}(\mathbf{X}-\mathbf{M}-w\mathbf{A})'\right)-\gamma w\right\}. \end{aligned}$$

Following the same procedure as before, the density of $\mathscr{X}$ is then

$$\begin{aligned} f_{\text{MVVG}}(\mathbf{X}|\boldsymbol{\vartheta}) =& \frac{2\gamma^{\gamma}\exp\left\{\operatorname{tr}(\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{M})\boldsymbol{\Psi}^{-1}\mathbf{A}')\right\}}{(2\pi)^{\frac{np}{2}}|\boldsymbol{\Sigma}|^{\frac{p}{2}}|\boldsymbol{\Psi}|^{\frac{n}{2}}\Gamma(\gamma)}\left(\frac{\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi})}{\rho(\mathbf{A},\boldsymbol{\Sigma},\boldsymbol{\Psi})+2\gamma}\right)^{\frac{(\gamma-np/2)}{2}} \\ & \times K_{\left(\gamma-\frac{np}{2}\right)}\left(\sqrt{[\rho(\mathbf{A},\boldsymbol{\Sigma},\boldsymbol{\Psi})+2\gamma]\,[\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma},\boldsymbol{\Psi})]}\right), \end{aligned}$$

where $\gamma > 0$. We will denote this distribution by $\text{MVVG}_{n\times p}(\mathbf{M}, \mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}, \gamma)$. Note that $W|\mathbf{X} \sim \text{GIG}\left(\rho(\mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}) + 2\gamma, \delta(\mathbf{X}; \mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}), \gamma - np/2\right)$.

3.2.3 Matrix Variate NIG Distribution

Finally, we consider a matrix variate NIG distribution. Derived in much the same way as the previous distributions, we take $W \sim \text{IG}(1, \kappa)$. The joint density of $\mathscr{X}$ and W is

$$f(\mathbf{X}, w|\boldsymbol{\vartheta}) = \frac{1}{(2\pi)^{\frac{np}{2}+1}|\boldsymbol{\Sigma}|^{\frac{p}{2}}|\boldsymbol{\Psi}|^{\frac{n}{2}}} w^{-\left(\frac{3+np}{2}\right)} \times \exp\left\{-\frac{1}{2w}\left(\text{tr}\left(\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{M}-w\mathbf{A})\boldsymbol{\Psi}^{-1}(\mathbf{X}-\mathbf{M}-w\mathbf{A})'\right)+1\right) - \frac{w\kappa^2}{2} + \kappa\right\},$$

and the density of $\mathscr{X}$ is then

$$f_{\text{MVNIG}}(\mathbf{X}|\boldsymbol{\vartheta}) = \frac{2\exp\left\{\text{tr}(\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\mathbf{M})\boldsymbol{\Psi}^{-1}\mathbf{A}') + \kappa\right\}}{(2\pi)^{\frac{np}{2}+1}|\boldsymbol{\Sigma}|^{\frac{p}{2}}|\boldsymbol{\Psi}|^{\frac{n}{2}}} \left(\frac{\delta(\mathbf{X}; \mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}) + 1}{\rho(\mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}) + \kappa^2}\right)^{-(1+np)/4} \times K_{-(1+np)/2}\left(\sqrt{\left[\rho(\mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}) + \kappa^2\right]\left[\delta(\mathbf{X}; \mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}) + 1\right]}\right),$$

where $\kappa > 0$. We denote this distribution by $\text{MVNIG}_{n\times p}(\mathbf{M}, \mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}, \kappa)$, and note that $W|\mathbf{X} \sim \text{GIG}\left(\rho(\mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}) + \kappa^2, \delta(\mathbf{X}; \mathbf{M}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}) + 1, -(1 + np)/2\right)$.

3.2.4 Simulations

We now consider a simple example for each of the three different distributions. Common elements between the distributions are as follows. We took 50 datasets each with

100 observations. For each distribution, we took

$$\mathbf{M} = \begin{pmatrix} -5 & 0 & 0 & 1 \\ -2 & 1 & 3 & 0 \\ 0 & 0 & 6 & 1 \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0.5 & -1 & 0 & -0.5 \\ 0 & -1 & 0 & 0 \end{pmatrix}.$$

and the scale matrices, $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}$ were

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.1 \\ 0.5 & 1 & 0.5 \\ 0.1 & 0.5 & 1 \end{pmatrix} \quad \boldsymbol{\Psi} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0.5 & 0.5 \\ 0 & 0.5 & 1 & 0.1 \\ 0 & 0.5 & 0.1 & 1 \end{pmatrix}.$$

We took the additional parameters to be $\lambda_2 = -2$, $\omega = 2$ for the generalized hyperbolic, $\gamma_2 = 4$ for the variance-gamma and $\kappa_2 = 2$ for the NIG. In Figure 3.3, we show the marginal distributions of the columns for each distribution of a typical dataset. We label the columns V1, V2, V3, and V4. The marginal location is shown by the red dashed line.

We now look at the parameter estimates. We show the component-wise means and standard deviations (in brackets), of the parameter estimates in Table 3.2. We see that for all three distributions, we get good average estimates in general. However, one obvious result that is unexpected is the estimate for λ for the matrix variate generalized hyperbolic distribution. The estimate is very different from the true value, and there is a very large amount of variation. We also notice a deflation in absolute value for the estimates of the skewness as well as a fair amount of variation. One possible explanation is that the generalized hyperbolic distribution is over-parameterized, and

thus the deflation in the estimates for the skewness could be compensation for the increased value of λ.

Table 3.2: Component wise averages and standard deviations for the estimated parameters for each of the three distributions.

Generalized Hyperbolic				
$\mathbf{M}$ (sd)	$\mathbf{A}$ (sd)	$\mathbf{\Sigma}$ (sd)	$\mathbf{\Psi}$ (sd)	λ (sd) ω (sd)
$\begin{bmatrix} -4.97 & 0.05 & -0.03 & 1.02 \\ -1.89 & 1.01 & 3.00 & 0.05 \\ 0.10 & -0.01 & 5.98 & 0.97 \end{bmatrix}$ $\left(\begin{bmatrix} 0.212 & 0.281 & 0.282 & 0.247 \\ 0.199 & 0.266 & 0.245 & 0.259 \\ 0.251 & 0.160 & 0.239 & 0.218 \end{bmatrix}\right)$	$\begin{bmatrix} 0.57 & -0.69 & 0.02 & 0.64 \\ 0.23 & -0.68 & -0.02 & -0.34 \\ -0.02 & -0.64 & 0.04 & 0.02 \end{bmatrix}$ $\left(\begin{bmatrix} 0.526 & 0.820 & 0.272 & 0.660 \\ 0.276 & 0.779 & 0.255 & 0.398 \\ 0.338 & 0.665 & 0.173 & 0.242 \end{bmatrix}\right)$	$\begin{bmatrix} 1.00 & 0.50 & 0.10 \\ 0.50 & 0.99 & 0.50 \\ 0.10 & 0.50 & 1.00 \end{bmatrix}$ $\left(\begin{bmatrix} 0.000 & 0.055 & 0.061 \\ 0.055 & 0.117 & 0.079 \\ 0.061 & 0.079 & 0.112 \end{bmatrix}\right)$	$\begin{bmatrix} 0.63 & 0.00 & 0.01 & 0.00 \\ 0.00 & 0.64 & 0.33 & 0.32 \\ 0.01 & 0.33 & 0.63 & 0.07 \\ 0.00 & 0.32 & 0.07 & 0.64 \end{bmatrix}$ $\left(\begin{bmatrix} 0.606 & 0.057 & 0.068 & 0.045 \\ 0.057 & 0.581 & 0.299 & 0.297 \\ 0.068 & 0.299 & 0.596 & 0.068 \\ 0.045 & 0.297 & 0.068 & 0.607 \end{bmatrix}\right)$	1.63 (2.42) 4.08 (1.33)
Variance Gamma				
$\mathbf{M}$ (sd)	$\mathbf{A}$ (sd)	$\mathbf{\Sigma}$ (sd)	$\mathbf{\Psi}$ (sd)	γ (sd)
$\begin{bmatrix} -4.98 & 0.01 & 0.04 & 0.96 \\ -1.98 & 1.00 & 3.02 & 0.02 \\ 0.02 & 0.05 & 6.07 & 1.03 \end{bmatrix}$ $\left(\begin{bmatrix} 0.280 & 0.229 & 0.254 & 0.260 \\ 0.233 & 0.240 & 0.206 & 0.216 \\ 0.238 & 0.242 & 0.206 & 0.195 \end{bmatrix}\right)$	$\begin{bmatrix} 0.98 & -0.99 & -0.00 & 1.04 \\ 0.49 & -0.98 & 0.01 & -0.52 \\ 0.00 & -1.05 & -0.06 & -0.04 \end{bmatrix}$ $\left(\begin{bmatrix} 0.307 & 0.269 & 0.256 & 0.282 \\ 0.248 & 0.256 & 0.222 & 0.247 \\ 0.260 & 0.245 & 0.232 & 0.225 \end{bmatrix}\right)$	$\begin{bmatrix} 1.00 & 0.51 & 0.10 \\ 0.51 & 1.01 & 0.51 \\ 0.10 & 0.51 & 1.02 \end{bmatrix}$ $\left(\begin{bmatrix} 0.000 & 0.048 & 0.063 \\ 0.048 & 0.095 & 0.081 \\ 0.063 & 0.081 & 0.129 \end{bmatrix}\right)$	$\begin{bmatrix} 0.99 & -0.01 & -0.01 & 0.00 \\ -0.01 & 0.98 & 0.47 & 0.51 \\ -0.01 & 0.47 & 0.98 & 0.09 \\ 0.00 & 0.51 & 0.09 & 1.00 \end{bmatrix}$ $\left(\begin{bmatrix} 0.121 & 0.064 & 0.053 & 0.060 \\ 0.064 & 0.103 & 0.074 & 0.072 \\ 0.053 & 0.074 & 0.121 & 0.059 \\ 0.060 & 0.072 & 0.059 & 0.126 \end{bmatrix}\right)$	4.20 (1.04)
Normal Inverse Gaussian				
$\mathbf{M}$ (sd)	$\mathbf{A}$ (sd)	$\mathbf{\Sigma}$ (sd)	$\mathbf{\Psi}$ (sd)	κ (sd)
$\begin{bmatrix} -5.02 & 0.04 & 0.01 & 1.03 \\ -1.99 & 1.04 & 2.99 & 0.05 \\ 0.02 & 0.01 & 5.98 & 1.01 \end{bmatrix}$ $\left(\begin{bmatrix} 0.143 & 0.134 & 0.133 & 0.137 \\ 0.137 & 0.123 & 0.140 & 0.117 \\ 0.148 & 0.120 & 0.128 & 0.114 \end{bmatrix}\right)$	$\begin{bmatrix} 1.16 & -1.18 & 0.01 & 1.02 \\ 0.55 & -1.19 & 0.04 & -0.64 \\ 0.01 & -1.11 & 0.04 & 0.02 \end{bmatrix}$ $\left(\begin{bmatrix} 0.506 & 0.446 & 0.306 & 0.418 \\ 0.390 & 0.462 & 0.323 & 0.357 \\ 0.298 & 0.433 & 0.271 & 0.249 \end{bmatrix}\right)$	$\begin{bmatrix} 1.00 & 0.49 & 0.11 \\ 0.49 & 1.01 & 0.51 \\ 0.11 & 0.51 & 1.00 \end{bmatrix}$ $\left(\begin{bmatrix} 0.000 & 0.045 & 0.053 \\ 0.045 & 0.107 & 0.077 \\ 0.053 & 0.077 & 0.119 \end{bmatrix}\right)$	$\begin{bmatrix} 1.02 & 0.01 & 0.01 & 0.02 \\ 0.01 & 1.06 & 0.54 & 0.53 \\ 0.01 & 0.54 & 1.06 & 0.11 \\ 0.02 & 0.53 & 0.11 & 1.07 \end{bmatrix}$ $\left(\begin{bmatrix} 0.250 & 0.065 & 0.064 & 0.072 \\ 0.065 & 0.285 & 0.175 & 0.139 \\ 0.064 & 0.175 & 0.281 & 0.072 \\ 0.072 & 0.139 & 0.072 & 0.245 \end{bmatrix}\right)$	2.12 (0.50)

3.3 Some Properties

Notice that just like matrix variate normal and matrix variate skew-t distributions, these three matrix variate skewed distributions are related to their multivariate counterparts.

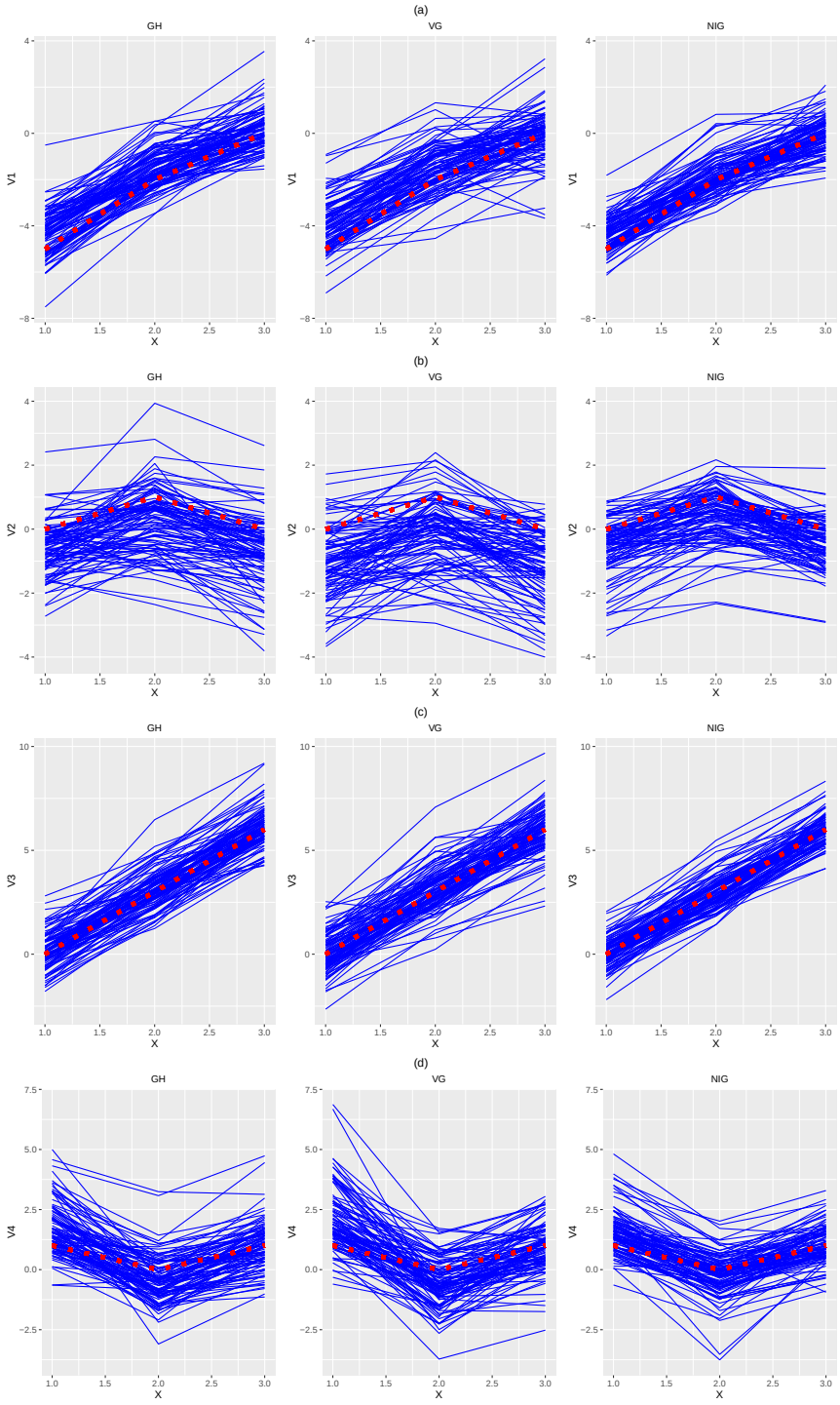


Figure 3.3: Marginal distributions for the matrix variate GH, VG and NIG distributions for (a) V1, (b) V2, (c) V3 and (d) V4. The marginal location (mode) is given by a red dashed line.

Specifically,

$$\begin{aligned}
\mathscr{X} \sim \text{MVGH}_{n\times p}(\mathbf{M}, \mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}, \omega, \lambda) &\iff \text{vec}(\mathscr{X}) \sim \text{GH}_{np}(\text{vec}(\mathbf{M}), \text{vec}(\mathbf{A}), \boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}, \omega, \lambda),\\
\mathscr{X} \sim \text{MVVG}_{n\times p}(\mathbf{M}, \mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}, \gamma) &\iff \text{vec}(\mathscr{X}) \sim \text{VG}_{np}(\text{vec}(\mathbf{M}), \text{vec}(\mathbf{A}), \boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}, \gamma),\\
\mathscr{X} \sim \text{MVNIG}_{n\times p}(\mathbf{M}, \mathbf{A}, \boldsymbol{\Sigma}, \boldsymbol{\Psi}, \kappa) &\iff \text{vec}(\mathscr{X}) \sim \text{NIG}_{np}(\text{vec}(\mathbf{M}), \text{vec}(\mathbf{A}), \boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}, \kappa).
\end{aligned}$$

These properties can be easily seen by using the representation of $\mathscr{X}$ given in (3.1) as well as the property of the matrix variate normal distribution given in (2.10).

We can also easily derive the moment generating functions for each of these three distributions. Using the representation for a random matrix $\mathscr{X}$ given in (3.1) and the moment generating function for the matrix variate normal distribution given in Dutilleul (1999), we have that the moment generating function in the general case of a matrix normal variance-mean mixture is

$$\begin{aligned}
M_{\mathscr{X}}(\mathbf{T}) &= \mathbb{E}[\exp\{\,\text{tr}(\mathbf{T}'\mathscr{X})\}] = \mathbb{E}[\mathbb{E}[\exp\{\,\text{tr}(\mathbf{T}'\mathscr{X})\} \mid W]]\\
&= \exp\{\,\text{tr}(\mathbf{T}'\mathbf{M})\}\mathbb{E}[\exp\{W\,\text{tr}(\mathbf{T}'\mathbf{A} + \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}'\boldsymbol{\Psi})\}]\\
&= \exp\{\,\text{tr}(\mathbf{T}'\mathbf{M})\}M_W(\,\text{tr}(\mathbf{T}'\mathbf{A} + \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}'\boldsymbol{\Psi})),
\end{aligned}$$

where $M_W(\cdot)$ is the moment generating function of W. Therefore, in the case of the generalized hyperbolic distribution, we have that the moment generating function is

$$\exp\{\,\text{tr}(\mathbf{T}'\mathbf{M})\}\left[1 - 2\frac{\text{tr}(\mathbf{T}'\mathbf{A} + \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}'\boldsymbol{\Psi})}{\omega}\right]^{-\frac{\lambda}{2}} \frac{K_\lambda\left(\sqrt{\omega(\omega - 2\,\text{tr}(\mathbf{T}'\mathbf{A} + \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}'\boldsymbol{\Psi}))}\right)}{K_\lambda(\omega)}.$$

For the variance gamma distribution, the moment generating function is

$$M_{\mathscr{X}}^{\text{MVVG}}(\mathbf{T}) = \exp\{\operatorname{tr}(\mathbf{T}'\mathbf{M})\}\left(1 - \frac{\operatorname{tr}(\mathbf{T}'\mathbf{A} + \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}'\boldsymbol{\Psi})}{\gamma}\right)^{-\gamma},$$

for $\operatorname{tr}(\mathbf{T}'\mathbf{A} + \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}'\boldsymbol{\Psi}) < \gamma$, and, in the case of the NIG distribution, the moment generating function is

$$M_{\mathscr{X}}^{\text{MVNIG}}(\mathbf{T}) = \exp\{\operatorname{tr}(\mathbf{T}'\mathbf{M})\}\exp\left\{\kappa\left(1 - \sqrt{1 - \frac{2\operatorname{tr}(\mathbf{T}'\mathbf{A} + \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}'\boldsymbol{\Psi})}{\kappa^2}}\right)\right\}.$$

Note that the moment generating function for the matrix variate skew-t distribution does not exist.

It is also important to note that these four skewed matrix variate distributions have the same identifiability issue as the matrix variate normal distribution.

3.4 Summary

In this chapter, a total of four skewed matrix variate distributions were derived from a matrix variate normal variance-mean mixture model. These four distributions were the matrix variate skew-t, generalized hyperbolic, variance-gamma and NIG distributions, respectively. When looking at the estimates in the simulations, we obtained fairly good results. One exception was the average estimates of λ and the skewness matrix $\mathbf{A}$ for the matrix variate generalized hyperbolic distribution. However, this could be due to over-parameterization.

Chapter 4

Finite Mixtures of Skewed Matrix Variate Distributions

4.1 Methodology

4.1.1 Likelihoods

In the mixture model context, $\mathscr{X}$ is assumed to come from a population with G subgroups each distributed according to the same one of the four skewed matrix variate distributions discussed previously. Now suppose N $n \times p$ matrices $\mathbf{X}_1, \ldots, \mathbf{X}_N$ are observed, then the observed-data likelihood is

$$L(\boldsymbol{\vartheta}) = \prod_{i=1}^{N} \sum_{g=1}^{G} \pi_g f(\mathbf{X}_i \mid \mathbf{M}_g, \mathbf{A}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g, \boldsymbol{\theta}_g),$$

where $\boldsymbol{\theta}_g$ are the parameters associated with the distribution of W_{ig}. For the purposes of parameter estimation, we proceed as if the observed data is incomplete. In

particular, we introduce the missing group membership indicators z_{ig}, where

$$z_{ig} = \begin{cases} 1 & \text{if } \mathbf{X}_i \text{ is in group } g, \\ 0 & \text{otherwise.} \end{cases}$$

In addition to the missing z_{ig}, we also have the latent variables $W_{ig} \in \mathbb{R}^+$ and we denote their densities by $h(w_{ig} \mid \boldsymbol{\theta}_g)$.

The complete-data log likelihood, in its general form for any of the distributions already discussed, is then

$$\ell_c(\boldsymbol{\vartheta}) = \mathcal{L}_1 + (\mathcal{L}_2 + C_2) + (\mathcal{L}_3 + C_3), \tag{4.1}$$

where C_2 and C_3 are constant with respect to the parameters, $\mathcal{L}_1 = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\pi_g$, $\mathcal{L}_2 = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}h(w_{ig} \mid \boldsymbol{\theta}_g) - C_2$, and

$$\begin{aligned}\mathcal{L}_3 = \frac{1}{2}\sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\big[&\operatorname{tr}\left(\boldsymbol{\Sigma}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)\boldsymbol{\Psi}_g^{-1}\mathbf{A}_g'\right) + \operatorname{tr}\left(\boldsymbol{\Sigma}_g^{-1}\mathbf{A}_g\boldsymbol{\Psi}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)'\right) \\ &-\frac{1}{w_{ig}}\operatorname{tr}(\boldsymbol{\Sigma}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)\boldsymbol{\Psi}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)') - w_{ig}\operatorname{tr}(\boldsymbol{\Sigma}_g^{-1}\mathbf{A}_g\boldsymbol{\Psi}_g^{-1}\mathbf{A}_g') \\ &-p\log(|\boldsymbol{\Sigma}_g|) - n\log(|\boldsymbol{\Psi}_g|)\big].\end{aligned}$$

4.1.2 Parameter Estimation

Parameter estimation is performed by using an expectation-conditional maximization (ECM) algorithm (Meng and Rubin, 1993).

1) Initialization: Initialize the parameters $\mathbf{M}_g, \mathbf{A}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g$ and other parameters related to the distribution. Set $t = 0$.

2) E Step: Update $\hat{z}_{ig}, a_{ig}, b_{ig}, c_{ig}$, where

$$\hat{z}_{ig}^{(t+1)} = \frac{\pi_g f(\mathbf{X}_i \mid \hat{\boldsymbol{\vartheta}}_g^{(t)})}{\sum_{h=1}^{G} \pi_h f(\mathbf{X}_i \mid \hat{\boldsymbol{\vartheta}}_h^{(t)})}, \qquad a_{ig}^{(t+1)} = \mathbb{E}(W_{ig}|\mathbf{X}_i, z_{ig} = 1, \hat{\boldsymbol{\vartheta}}_g^{(t)}),$$
$$b_{ig}^{(t+1)} = \mathbb{E}\left(\frac{1}{W_{ig}} \,|\, \mathbf{X}_i, z_{ig} = 1, \hat{\boldsymbol{\vartheta}}_g^{(t)}\right), \qquad c_{ig}^{(t+1)} = \mathbb{E}(\log(W_{ig})|\mathbf{X}_i, z_{ig} = 1, \hat{\boldsymbol{\vartheta}}_g^{(t)}).$$

Note that the specific updates will depend on the distribution. However, in each case, the conditional distribution of W_{ig} given the observed data and group memberships is a generalized inverse Gaussian distribution. Specifically,

$$W_{ig}^{\text{ST}} \mid \mathbf{X}_i, z_{ig} = 1 \sim \text{GIG}\left(\rho(\mathbf{A}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g), \delta(\mathbf{X}; \mathbf{M}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g) + \nu_g, -(\nu_g + np)/2\right),$$
$$W_{ig}^{\text{GH}} \mid \mathbf{X}_i, z_{ig} = 1 \sim \text{GIG}\left(\rho(\mathbf{A}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g) + \omega_g, \delta(\mathbf{X}; \mathbf{M}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g) + \omega_g, \lambda_g - np/2\right),$$
$$W_{ig}^{\text{VG}} \mid \mathbf{X}_i, z_{ig} = 1 \sim \text{GIG}\left(\rho(\mathbf{A}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g) + 2\gamma_g, \delta(\mathbf{X}; \mathbf{M}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g), \gamma_g - np/2\right),$$
$$W_{ig}^{\text{NIG}} \mid \mathbf{X}_i, z_{ig} = 1 \sim \text{GIG}\left(\rho(\mathbf{A}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g) + \kappa_g^2, \delta(\mathbf{X}; \mathbf{M}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g) + 1, -(1 + np)/2\right).$$

Therefore, the exact updates are obtained by using the expectations given in (2.4)–(2.6) for appropriate values of λ, a, and b.

3) First CM Step: Update the parameters $\pi_g, \mathbf{M}_g, \mathbf{A}_g$.

$$\hat{\pi}_g^{(t+1)} = \frac{N_g}{N}, \qquad \hat{\mathbf{M}}_g^{(t+1)} = \frac{\sum_{i=1}^{N} \hat{z}_{ig}^{(t+1)} \mathbf{X}_i \left(\overline{a}_g^{(t+1)} b_{ig}^{(t+1)} - 1\right)}{\sum_{i=1}^{N} \hat{z}_{ig}^{(t+1)} \overline{a}_g^{(t+1)} b_{ig}^{(t+1)} - N_g},$$
$$\hat{\mathbf{A}}^{(t+1)} = \frac{\sum_{i=1}^{N} \hat{z}_{ig}^{(t+1)} \mathbf{X}_i \left(\overline{b}_g^{(t+1)} - b_{ig}^{(t+1)}\right)}{\sum_{i=1}^{N} \hat{z}_{ig}^{(t+1)} \overline{a}_g^{(t+1)} b_{ig}^{(t+1)} - N_g},$$

where

$$N_g = \sum_{i=1}^{N} \hat{z}_{ig}^{(t+1)}, \qquad \overline{a}_g^{(t+1)} = \frac{\sum_{i=1}^{N} \hat{z}_{ig}^{(t+1)} a_{ig}^{(t+1)}}{N_g}, \qquad \overline{b}_g^{(t+1)} = \frac{\sum_{i=1}^{N} \hat{z}_{ig}^{(t+1)} b_{ig}^{(t+1)}}{N_g}.$$

4) Second CM Step: Update $\boldsymbol{\Sigma}_g$

$$\begin{aligned}\hat{\boldsymbol{\Sigma}}_g^{(t+1)} = \frac{1}{N_g p} \Bigg[\sum_{i=1}^{N} \hat{z}_{ig}^{(t+1)} \Bigg(& b_{ig}^{(t+1)} \left(\mathbf{X}_i - \hat{\mathbf{M}}_g^{(t+1)} \right) \hat{\boldsymbol{\Psi}}_g^{(t)\,-1} \left(\mathbf{X}_i - \hat{\mathbf{M}}_g^{(t+1)} \right)' \\ & - \hat{\mathbf{A}}_g^{(t+1)} \hat{\boldsymbol{\Psi}}_g^{(t)\,-1} \left(\mathbf{X}_i - \hat{\mathbf{M}}_g^{(t+1)} \right)' - \left(\mathbf{X}_i - \hat{\mathbf{M}}_g^{(t+1)} \right) \hat{\boldsymbol{\Psi}}_g^{(t)\,-1} \hat{\mathbf{A}}_g^{(t+1)\prime} \\ & + a_{ig}^{(t+1)} \hat{\mathbf{A}}_g^{(t+1)} \hat{\boldsymbol{\Psi}}_g^{(t)\,-1} \hat{\mathbf{A}}_g^{(t+1)\prime} \Bigg) \Bigg].\end{aligned} \tag{4.2}$$

5) Third CM Step: Update $\boldsymbol{\Psi}_g$

$$\begin{aligned}\hat{\boldsymbol{\Psi}}_g^{(t+1)} = \frac{1}{N_g n} \Bigg[\sum_{i=1}^{N} \hat{z}_{ig}^{(t+1)} \Bigg(& b_{ig}^{(t+1)} \left(\mathbf{X}_i - \hat{\mathbf{M}}_g^{(t+1)} \right)' \hat{\boldsymbol{\Sigma}}_g^{(t+1)\,-1} \left(\mathbf{X}_i - \hat{\mathbf{M}}_g^{(t+1)} \right) \\ & - \hat{\mathbf{A}}_g^{(t+1)\prime} \hat{\boldsymbol{\Sigma}}_g^{(t+1)\,-1} \left(\mathbf{X}_i - \hat{\mathbf{M}}_g^{(t+1)} \right) - \left(\mathbf{X}_i - \hat{\mathbf{M}}_g^{(t+1)} \right)' \hat{\boldsymbol{\Sigma}}_g^{(t+1)\,-1} \hat{\mathbf{A}}_g^{(t+1)} \\ & + a_{ig}^{(t+1)} \hat{\mathbf{A}}_g^{(t+1)\prime} \hat{\boldsymbol{\Sigma}}_g^{(t+1)\,-1} \hat{\mathbf{A}}_g^{(t+1)} \Bigg) \Bigg].\end{aligned} \tag{4.3}$$

6) Other CM Steps: The additional parameters introduced by the distribution of W_{ig} are now updated. These updates will vary according the distribution and the particulars for the MVST, MVGH, MVVG and MVNIG distributions are given below.

7) Check Convergence: If not converged, set $t = t + 1$ and return to step 2.

Matrix Variate Skew-t Distribution

In the case of the matrix variate skew-t distribution, the degrees of freedom ν_g need to be updated. This update cannot be obtained in closed form, and thus needs to be

performed numerically. We have

$$\mathcal{L}_2^{\text{MVST}} = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\left[\frac{\nu_g}{2}\log\left(\frac{\nu_g}{2}\right) - \log\left(\Gamma\left(\frac{\nu_g}{2}\right)\right) - \frac{\nu_g}{2}\left(\log(w_{ig}) + \frac{1}{w_{ig}}\right)\right].$$

Therefore, the update $\nu_g^{(t+1)}$ is obtained by solving (4.4) for ν_g, i.e.,

$$\log\left(\frac{\nu_g}{2}\right) + 1 - \varphi\left(\frac{\nu_g}{2}\right) - \frac{1}{N_g}\sum_{i=1}^{N}\hat{z}_{ig}^{(t+1)}(b_{ig}^{(t+1)} + c_{ig}^{(t+1)}) = 0, \tag{4.4}$$

where $\varphi(\cdot)$ denotes the digamma function.

Matrix Variate Generalized Hyperbolic Distribution

In the case of the matrix variate generalized hyperbolic distribution, updates for λ_g and ω_g are needed. In this case,

$$\mathcal{L}_1^{\text{MVGH}} = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\left[\log(K_{\lambda_g}(\omega_g)) - \lambda_g \log w_{ig} - \frac{1}{2}\omega_g\left(w_{ig} + \frac{1}{w_{ig}}\right)\right]. \tag{4.5}$$

The updates for λ_g and ω_g cannot be obtained in closed form. However, Browne and McNicholas (2015) discuss numerical methods for these updates and, because the portion of the likelihood function that include these parameters is the same as in the multivariate case, the updates described in Browne and McNicholas (2015) can be used directly here.

The updates for λ_g and ω_g rely on the log-convexity of $K_s(t)$ in both s and t (Baricz, 2010) and maximizing (4.5) via conditional maximization. The resulting

updates are

$$\hat{\lambda}_g^{(t+1)} = \bar{c}_g^{(t+1)} \hat{\lambda}_g^{(t)} \left[\frac{\partial}{\partial s} \log(K_s(\hat{\omega}_g^{(t)})) \Big|_{s=\hat{\lambda}_g^{(t)}} \right]^{-1}, \tag{4.6}$$

$$\hat{\omega}_g^{(t+1)} = \hat{\omega}_g^{(t)} - \left[\frac{\partial}{\partial s} q(\hat{\lambda}_g^{(t+1)}, s) \Big|_{s=\hat{\omega}_g^{(t)}} \right] \left[\frac{\partial^2}{\partial s^2} q(\hat{\lambda}_g^{(t+1)}, s) \Big|_{s=\hat{\omega}_g^{(t)}} \right]^{-1}, \tag{4.7}$$

where the derivative in (4.6) is calculated numerically,

$$q(\lambda_g, \omega_g) = \sum_{i=1}^{N} z_{ig} \left[\log(K_{\lambda_g}(\omega_g)) - \lambda_g \log w_{ig} - \frac{1}{2}\omega_g \left(w_{ig} + \frac{1}{w_{ig}} \right) \right]$$

and $\bar{c}_g^{(t+1)} = (1/N_g) \sum_{i=1}^{N} \hat{z}_{ig}^{(t+1)} c_{ig}^{(t+1)}$. The partials in (4.7) are described in Browne and McNicholas (2015) and can be written as

$$\frac{\partial}{\partial \omega_g} q(\lambda_g, \omega_g) = \frac{1}{2}[R_{\lambda_g}(\omega_g) + R_{-\lambda_g}(\omega_g) - (\bar{a}_g^{(t+1)} + \bar{b}_g^{(t+1)})],$$

and

$$\frac{\partial^2}{\partial \omega_g^2} q(\lambda_g, \omega_g) = \frac{1}{2} \left[R_{\lambda_g}(\omega_g)^2 - \frac{1 + 2\lambda_g}{\omega_g} R_{\lambda_g}(\omega_g) - 1 + R_{-\lambda_g}(\omega_g)^2 - \frac{1 - 2\lambda_g}{\omega_g} R_{-\lambda_g}(\omega_g) - 1 \right],$$

where $R_{\lambda_g}(\omega_g) = K_{\lambda_g+1}(\omega_g)/K_{\lambda_g}(\omega_g)$.

Matrix Variate Variance-Gamma Distribution

In the case of the matrix variate variance-gamma,

$$\mathcal{L}_1^{\text{MVVG}} = \sum_{i=1}^{N} \sum_{g=1}^{G} z_{ig} \left[\gamma_g \log \gamma_g - \log \Gamma(\gamma_g) + \gamma_g (\log w_{ig} - w_{ig}) \right].$$

The update for γ_g, as in the generalized hyperbolic case, cannot be obtained in closed form. Instead, the update $\gamma_g^{(t+1)}$ is obtained by solving (4.8) for γ_g, where

$$\log\gamma_g + 1 - \varphi(\gamma_g) + \bar{c}_g^{(t+1)} - \bar{a}_g^{(t+1)} = 0. \tag{4.8}$$

Matrix Variate NIG Distribution

In this case, κ_g needs to be updated. Note that

$$\mathcal{L}_2^{\text{MVNIG}} = \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\kappa_g - \frac{\kappa_g^2}{2} z_{ig} w_{ig}$$

and, therefore, the closed form updates for κ_g are

$$\kappa_g^{(t+1)} = \frac{1}{\overline{a}_g^{(t+1)}}.$$

4.1.3 Semi-Supervised Classification

In addition to clustering (unsupervised classification), the matrix variate mixture models introduced here can also be applied for semi-supervised classification. Suppose that N matrices are observed but that we know the labels for K of the N matrices; specifically, suppose that K of the N matrices come from one of G classes. Without loss of generality, order these matrices so it is the first K that have known labels: $\mathbf{X}_1, \ldots, \mathbf{X}_K, \mathbf{X}_{K+1}, \ldots \mathbf{X}_N$. Now, we know the values of z_{ig} for $i = 1, \ldots, K$ and the observed-data likelihood is

$$L(\boldsymbol{\vartheta}) = \prod_{i=1}^{K}\prod_{g=1}^{G}\left[\pi_g f(\mathbf{X}_i \mid \mathbf{M}_g, \mathbf{A}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g, \boldsymbol{\theta}_g)\right]^{z_{ig}} \prod_{j=K+1}^{N}\sum_{h=1}^{H}\pi_g f(\mathbf{X}_j \mid \mathbf{M}_h, \mathbf{A}_h, \boldsymbol{\Sigma}_h, \boldsymbol{\Psi}_h, \boldsymbol{\theta}_h),$$

where $\boldsymbol{\theta}_g$ are the parameters associated with the distribution of W_{ig}. In general, $H \geq G$; however, for the analyses herein, we make the common assumption that $H = G$. Parameter estimation, identifiability, etc., follow in an analogous fashion to the clustering case already described herein. Further details on semi-supervised classification in the mixture model setting are given in McLachlan and Peel (2000b) and McNicholas (2016a).

4.2 Illustrations

4.2.1 Overview

Two simulations are performed, where the first simulation has two groups and the second has three. The chosen parameters have no intrinsic meaning; however, they can be viewed as representations of multivariate longitudinal data and the parameters introduced by the distribution of W_{ig} are meant to illustrate the flexibility in concentration. Simulation 4.1 considers 3×4 data, Simulation 4.2 illustrates 4×3 data. In the first simulation, $\boldsymbol{\Sigma}_g$ and $\boldsymbol{\Psi}_g$ are set to

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & 0.5 & 0.1 \\ 0.5 & 1 & 0.5 \\ 0.1 & 0.5 & 1 \end{pmatrix}, \quad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & 0.1 & 0.1 \\ 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{pmatrix},$$

and

$$\boldsymbol{\Psi}_1 = \begin{pmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 0 & 1 & 0 \\ 0.5 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{\Psi}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0.5 & 0.5 \\ 0 & 0.5 & 1 & 0.2 \\ 0 & 0.5 & 0.2 & 1 \end{pmatrix}.$$

For notational purposes, let $\tilde{\boldsymbol{\Sigma}}_g$ and $\tilde{\boldsymbol{\Psi}}_g$ be the scale matrices used in Simulation 4.2. We set $\tilde{\boldsymbol{\Sigma}}_1 = \boldsymbol{\Psi}_1$, $\tilde{\boldsymbol{\Sigma}}_2 = \tilde{\boldsymbol{\Sigma}}_3 = \boldsymbol{\Psi}_2$ and $\tilde{\boldsymbol{\Psi}}_1 = \tilde{\boldsymbol{\Psi}}_3 = \boldsymbol{\Sigma}_1$ and $\tilde{\boldsymbol{\Psi}}_2 = \boldsymbol{\Sigma}_2$. For each distribution, the models are fitted for $G \in \{1, 2, 3, 4\}$ and the BIC is used to choose the number of groups.

4.2.2 Simulation 4.1

In Simulation 4.1, for all four distributions, we take the location and skewness matrices to be

$$\mathbf{M}_1 = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ -1 & 0 & 2 & -1 \end{pmatrix}, \quad \mathbf{M}_2 = \begin{pmatrix} 3 & 4 & 2 & 4 \\ 4 & 3 & 3 & 3 \\ 3 & 4 & 2 & 4 \end{pmatrix},$$

$$\mathbf{A}_1 = \begin{pmatrix} 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 \end{pmatrix}, \qquad \mathbf{A}_2 = \begin{pmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 0.5 & -1 \\ 1 & 1 & 0 & -1 \end{pmatrix}.$$

For the additional parameters, we took $\nu_1 = 4, \nu_2 = 20$ for the skew-t distribution, $\lambda_1 = \lambda_2 = 2$ and $\omega_1 = 4$, $\omega_2 = 2$ for the generalized hyperbolic distribution, $\gamma_1 = 7$, $\gamma_2 = 14$ for the variance-gamma distribution, and $\kappa_1 = 1/2, \kappa_2 = 2$ for the NIG distribution. Figure 4.1 shows a typical dataset for each distribution. For visualization, we

look at the marginal columns which we label V1, V2, V3 and V4. We see that for all of the columns, except column 4, there is a clear separation between the two groups. We also note that for the skew-t distribution, there was a severe outlier in group 2 (due to the small degrees of freedom) that we do not show for better visualization. The orange dotted line is the marginal location parameter for the first group, and the yellow dotted line is the marginal location for the second group.

Table 4.1 displays the number of groups (components) chosen and the average ARI values with the associated standard deviations. The ICL results were identical, and thus are not shown here. We see that the correct number of groups is chosen, with perfect classification, for all 30 of the datasets when using the MVST, MVVG, and MVNIG mixtures. However, this is not the case with MVGH mixture, which underperforms when compared to the other three. However, the eight datasets for which the incorrect number of components is chosen correspond to datasets for which the two-component MVGH solution did not converge and, in a real application, alternative starting values would be pursued until convergence is achieved for the $G = 2$ component case.

Table 4.1: The number of groups chosen by the BIC and the average ARI values, with standard deviations in parentheses, for Simulation 4,1. Note that the MVGH mixture did not converge for eight of the 30 runs with $G = 2$.

	$G = 1$	$G = 2$	$G = 3$	$G = 4$	$\overline{\text{ARI}}$ (std. dev.)
MVST	0	30	0	0	1.00 (0.00)
MVGH	4	22	1	3	0.85 (0.34)
MVVG	0	30	0	0	1.00 (0.00)
MVNIG	0	30	0	0	1.00 (0.00)

In Table 4.2, we show the average amount of time per dataset to run the algorithm for $G = 1, 2, 3, 4$. We note that these simulations were performed in parallel.

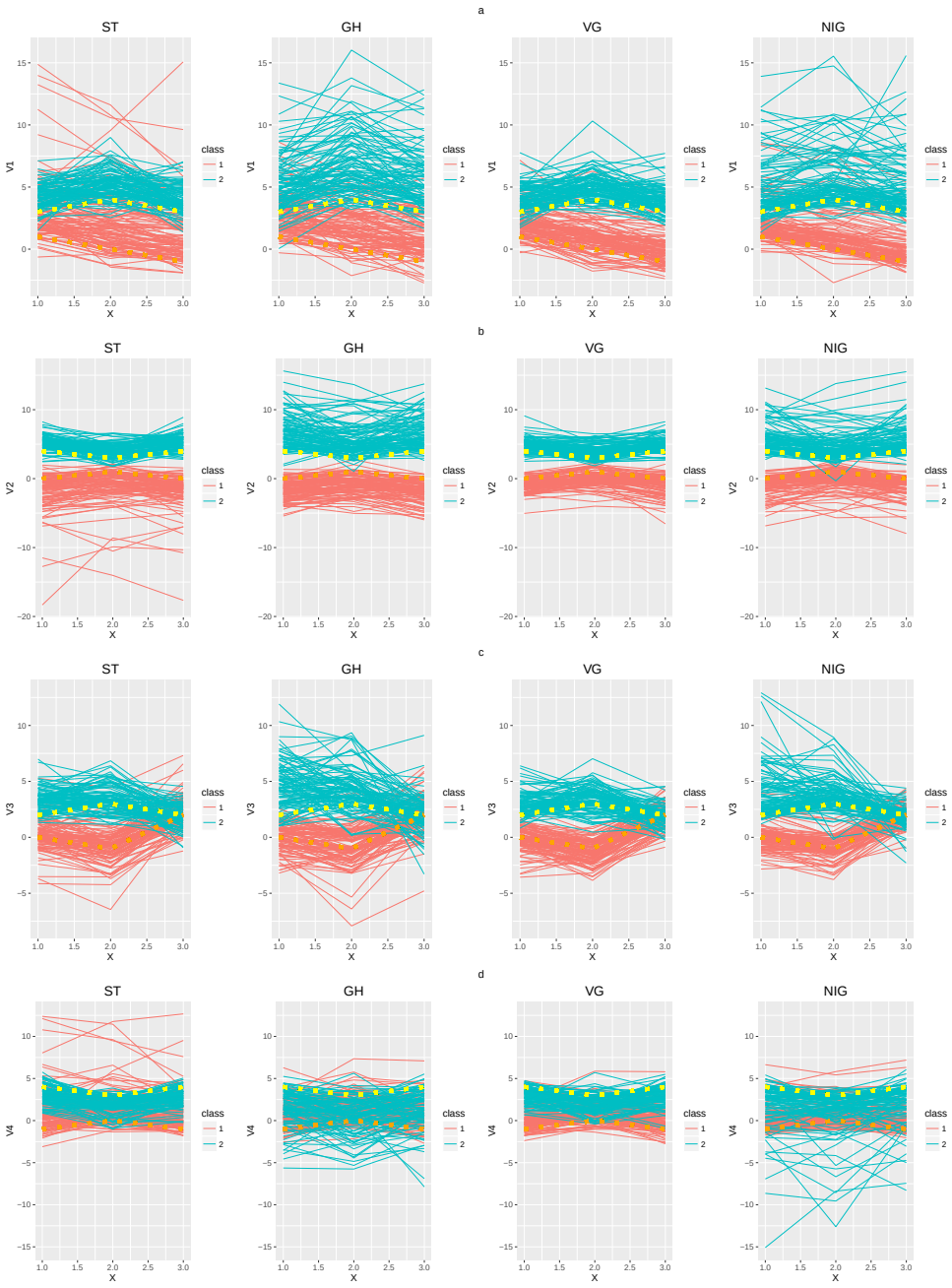


Figure 4.1: Marginal data for the columns for each of the four distributions for Simulation 4.1. The dotted lines represent the marginal location parameters with the orange as the marginal location for group 1 and the yellow for group 2.

Table 4.2: Average runtimes for Simulation 4.1.

Distribution	Average Time (s)
MVST	237.33
MVGH	625.90
MVVG	82.77
MVNIG	349.47

4.2.3 Simulation 4.2

In Simulation 4.2, a three group mixture was considered with 200 observations per group and the following location and skewness parameters.

$$\mathbf{M}_1 = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & -1 \end{pmatrix}, \quad \mathbf{M}_2 = \begin{pmatrix} -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad \mathbf{M}_3 = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix},$$

$$\mathbf{A}_1 = \begin{pmatrix} 1 & -1 & -1 \\ 1 & -0.5 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}, \quad \mathbf{A}_2 = \mathbf{A}_3 = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 0.5 & 0.5 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

The other parameters we set to $\nu_1 = 4$, $\nu_2 = 8$, $\nu_3 = 20$ for the MVST, $\lambda_1 = 4$, $\lambda_2 = 0$, $\lambda_3 = -2$ and $\omega_1 = 4$, $\omega_2 = \omega_3 = 2$ for the MVGH, $\gamma_1 = 7$, $\gamma_2 = 9$, $\gamma_3 = 14$ for the MVVG and $\kappa_1 = 1/2$, $\kappa_2 = 1$, $\kappa_3 = 2$ for the MVNIG.

Again, the marginal distributions of a typical dataset is shown in Figure 4.2. The dotted lines again represent the marginal locations, with orange for the first

group, yellow for the second, and purple for the third. In Table 4.3, the number of groups chosen by the BIC as well as the average ARI values, and associated standard deviations, are presented. Once again, the MVST, MVVG and MVNIG mixtures outperform the MVGH mixture; once again, this is due to convergence issues. The issue with convergence for the MVGH mixture with both simulations is possibly due to the update for, or impact of, the index parameters $\lambda_1, \ldots, \lambda_G$.

Table 4.3: The number of groups chosen by the BIC and the average ARI values, with standard deviations in parentheses, for Simulation 4.2. Note that the MVGH mixture did not converge for 22 of the 30 runs with $G = 2$.

	$G = 1$	$G = 2$	$G = 3$	$G = 4$	$\overline{\text{ARI}}$ (std. dev.)
MVST	0	0	30	0	0.97 (0.010)
MVGH	10	8	8	4	0.52 (0.41)
MVVG	0	0	30	0	0.98 (0.0077)
MVNIG	0	0	30	0	0.99 (0.0056)

Table 4.4 shows the average runtime per dataset for Simulation 4.2. Notice that for the MVGH, MVVG and MVNIG mixtures, each dataset took longer on average, with the MVGH mixture having the longest runtime as well as the largest increase. This is to be expected because there is an increase in the number of groups and observations; however, for the MVVG and MVNIG mixtures, the time differences between Simulations 4.1 and 4.2 is less notable. In fact, the MVST mixture actually took less time on average; however, this is because a few datasets for Simulation 4.1 ran to the maximum number of iterations for the $G = 4$ group mixture thus increasing the runtime.

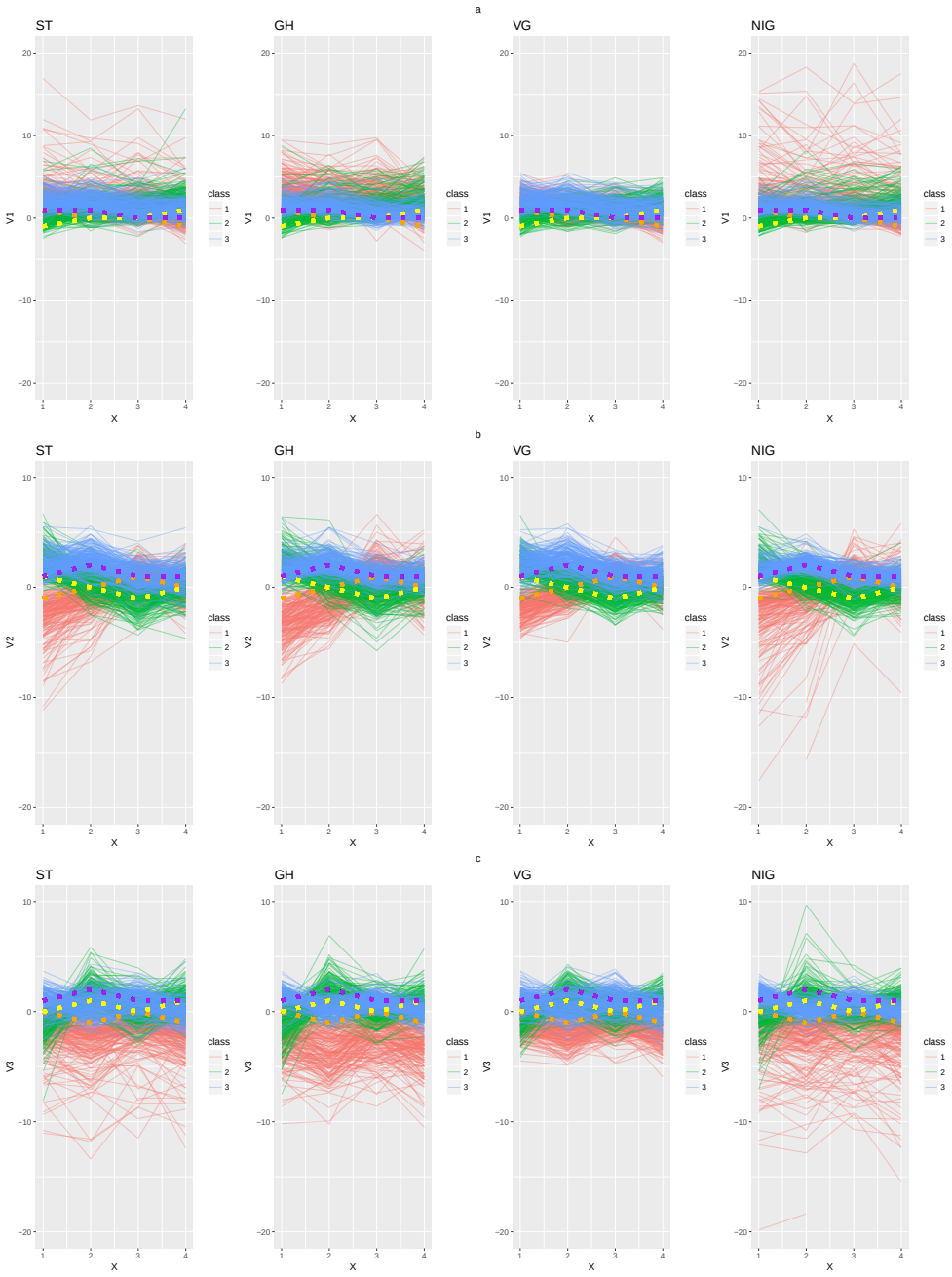


Figure 4.2: Marginal data for the columns for each of the four distributions for Simulation 4.2. The dotted lines represent the marginal location parameters with the orange as the marginal location for group 1, yellow for group 2, and purple for group 3.

Table 4.4: Average runtimes for Simulation 4.2.

Distribution	Average Time (s)
MVST	233.67
MVGH	2542.50
MVVG	171.90
MVNIG	581.63

4.3 Image Recognition Example

We now apply the matrix variate mixture models introduced herein to image recognition with the MNIST handwriting dataset (LeCun *et al.*, 1998). The original dataset consists of 60,000 training images of handwritten digits 0 to 9, which can be represented as 28×28 pixel matrices with greyscale intensities ranging from 0 to 255. However, because two unstructured 28×28 dimensional covariance matrices would need to be estimated, model fitting would be infeasible. We stress that this alone is an indication that dimension reduction techniques will need to be developed in the future. However, the main goal of this application is to demonstrate the discussed methods outside of the theoretical confines of the simulations. Therefore, we resized the original image to a 10×10 pixel matrix using the *resize* function in the `EBImage` package (Pau *et al.*, 2010) for the R software (R Core Team, 2019). However, there are problems with sparsity. Specifically, the outside columns and rows all contain values of 0 because they are outside of the main writing space. Accordingly, there is no variation in these outer columns and rows, therefore resulting in exactly singular $\boldsymbol{\Sigma}_g$ and $\boldsymbol{\Psi}_g$ updates. To solve this problem, we replace a value of 0 with a value between 0 and 2 with increments of 0.1 and added 50 to the non-zero values to make sure the noise did not interfere with the true signal.

Each of the matrix variate mixtures introduced herein is applied within the semi-supervised classification paradigm (Section 3.6). A total of 500 observations from digit 1 and 500 from digit 7 are sampled from the training set, and then 100 of each of these digits is considered unlabelled, i.e., 80% of the data are labelled. We performed the analysis on 30 different such sets. In Table 4.5, we show aggregate classification tables for the points considered unlabelled for each of the matrix variate mixtures. In Table 4.6, we show the average ARI values and the average misclassification rates for the unlabelled points. Note, that for some of the datasets, not all four mixtures converged; therefore, the total number of observations in the tables need not be the same for all four distributions. Looking at the classification tables, it is clear that all of these matrix variate mixtures overall misclassify digit 1 as digit 7 more often than digit 7 as digit 1. From both the ARI and MCR results, the MVVG mixture slightly outperforms the other three mixture. It is interesting to note that the MVGH mixture did not experience the same convergence issues as seen with the simulations.

Table 4.5: Cross-tabulations of true (1,7) versus predicted (P1, P7) classifications for the points considered unlabelled in the MNIST data, for each of the matrix variate mixtures introduced herein, aggregated over all runs (for which convergence was attained).

	MVST		MVGH		MVVG		MVNIG	
	P1	P7	P1	P7	P1	P7	P1	P7
1	2797	203	2813	187	2859	141	2798	202
7	127	2873	125	2875	122	2878	127	2873

Table 4.6: Average ARI values and misclassification rates (MCR), with associated standard deviations in parentheses, for each matrix variate mixture approach for the points considered unlabelled for the MNIST data, aggregated over all runs (for which convergence was attained).

	$\overline{\text{ARI}}$ (std. dev.)	$\overline{\text{MCR}}$(std. dev.)
MVST	0.79 (0.051)	0.055 (0.014)
MVGH	0.80 (0.056)	0.052 (0.016)
MVVG	0.83 (0.043)	0.044 (0.012)
MVNIG	0.79 (0.051)	0.055 (0.014)

4.4 Summary

Four matrix variate mixture distributions, with component densities that parameterize skew- ness, have been used for model-based clustering — and its semi-supervised analogue — of three-way data. Specifically, we considered MVST, MVGH, MVVG, and MVNIG mixtures, respectively, and an ECM algorithm was used for parameter estimation in each case. Simulated and real data were used for illustration. In the first simulation, there was good separation between the two groups and, in the second, we increased the number of groups, decreased the separation between the groups, and obtained similar results to the first. In both simulations, the MVGH mixture often underperformed when compared to the other three mixtures due to convergence issues. This could be resolved, for example, by restrict- ing the index parameter λ; however, doing this would essentially eliminate the additional flexibility enjoyed by the MVGH mixture. In the real data application, the MVVG mixture outperformed the other three mixtures in terms of both average ARI and average misclassification rate, and the MVVG mixture consistently ran faster than the other three mixtures.

Chapter 5

Mixtures of Matrix Variate Bilinear Factor Analyzers

5.1 Previous Work

Xie *et al.* (2008) and Yu *et al.* (2008) consider a matrix variate extension of PPCA in a linear fashion. For N independent $n \times p$ random matrices $\mathscr{X}_1, \ldots, \mathscr{X}_N$, the model assumes

$$\mathscr{X}_i = \mathbf{M} + \mathbf{\Lambda}\mathscr{U}_i\mathbf{\Delta}' + \mathscr{E}_i, \tag{5.1}$$

where $\mathbf{M}$ is an $n \times p$ location matrix, $\mathbf{\Lambda}$ is an $n \times q$ matrix of column factor loadings, $\mathbf{\Delta}$ is a $p \times r$ matrix of row factor loadings, $\mathscr{U}_i \sim \mathcal{N}_{q\times r}(\mathbf{0}, \mathbf{I}_q, \mathbf{I}_r)$, and $\mathscr{E}_i \sim \mathcal{N}_{n\times p}(\mathbf{0}, \sigma\mathbf{I}_n, \sigma\mathbf{I}_p)$, with $\sigma \in \mathbb{R}^+$. Note that the $\mathscr{U}_i$ and the $\mathscr{E}_i$ are each independently distributed and are independent of one another. The main disadvantage of this model is that, in general, $\mathscr{X}_i$ does not follow a matrix variate normal distribution.

Zhao *et al.* (2012) present bilinear probabilistic principal component analysis (BPPCA) which extends (5.1) by adding two projected error terms. The resulting model assumes

$$\mathscr{X}_i = \mathbf{M} + \mathbf{\Lambda}\mathscr{U}_i\mathbf{\Delta}' + \mathbf{\Lambda}\mathscr{E}_i^B + \mathscr{E}_i^A\mathbf{\Delta}' + \mathscr{E}_i, \tag{5.2}$$

where $\mathscr{E}_i^B \sim \mathcal{N}_{q\times p}(\mathbf{0}, \mathbf{I}_q, \sigma_B\mathbf{I}_p)$, $\mathscr{E}_i^A \sim \mathcal{N}_{n\times r}(\mathbf{0}, \sigma_A\mathbf{I}_n, \mathbf{I}_r)$, $\mathscr{E}_i \sim \mathcal{N}_{n\times p}(0, \sigma_A\mathbf{I}_n, \sigma_B\mathbf{I}_p)$, with $\sigma_A \in \mathbb{R}^+$ and $\sigma_B \in \mathbb{R}^+$, and the other terms are as defined for (5.1). In this model, each of the $\mathscr{U}_i$, $\mathscr{E}_i^B$, $\mathscr{E}_i^A$ and $\mathscr{E}_i$ are independently distributed and all are independent of each other.

It is important to note that the term "column factors" refers to reduction in the dimension of the columns, which is equivalent to the number of rows, and not a reduction in the number of columns. Likewise, the term "row factors" refers to the reduction in the dimension of the rows (number of columns). As discussed by Zhao *et al.* (2012) the interpretation of the terms $\mathscr{E}^B$ and $\mathscr{E}^A$ are the row and column noise respectively, whereas the final term $\mathscr{E}$ is the common noise. It can be shown using property (2.10) that under this model $\mathscr{X} \sim \mathcal{N}_{n\times p}(\mathbf{M}, \mathbf{\Lambda\Lambda}' + \sigma_A\mathbf{I}_n, \mathbf{\Delta\Delta}' + \sigma_B\mathbf{I}_p)$. Note that the covariance structure for the two covariance matrices of the matrix variate normal are analogous to the covariance structure for the (multivariate) factor analysis model.

5.2 Methodology

5.2.1 MMVBFA Model

An MMVBFA model is derived here by extending (5.2). Specifically, we remove the isotropic constraint and assume

$$\mathscr{X}_i = \mathbf{M}_g + \mathbf{\Lambda}_g \mathscr{U}_{ig} \mathbf{\Delta}_g' + \mathbf{\Lambda}_g \mathscr{E}_{ig}^B + \mathscr{E}_{ig}^A \mathbf{\Delta}_g' + \mathscr{E}_{ig} \tag{5.3}$$

with probability π_g, for $g = 1, 2, \ldots, G$, where $\mathbf{M}_g$ is an $n \times p$ location matrix, $\mathbf{\Lambda}_g$ is an $n \times q$ column factor loading matrix, with $q < n$, $\mathbf{\Delta}_g$ is a $p \times r$ row factor loading matrix, with $r < p$, and

$$\begin{aligned}
\mathscr{U}_{ig} &\sim \mathcal{N}_{q\times r}(\mathbf{0}, \mathbf{I}_q, \mathbf{I}_r),\\
\mathscr{E}_{ig}^B &\sim \mathcal{N}_{q\times p}(\mathbf{0}, \mathbf{I}_q, \mathbf{\Psi}_g),\\
\mathscr{E}_{ig}^A &\sim \mathcal{N}_{n\times r}(\mathbf{0}, \mathbf{\Sigma}_g, \mathbf{I}_r),\\
\mathscr{E}_{ig} &\sim \mathcal{N}_{n\times p}(\mathbf{0}, \mathbf{\Sigma}_g, \mathbf{\Psi}_g)
\end{aligned}$$

are independently distributed and independent of eachother, $\mathbf{\Sigma}_g = \text{diag}\{\sigma_{1g}, \sigma_{2g}, \ldots, \sigma_{ng}\}$, with $\sigma_{jg} \in \mathbb{R}^+$, $j \in \{1, \ldots, n\}$, and $\mathbf{\Psi}_g = \text{diag}\{\psi_{1g}, \psi_{2g}, \ldots, \psi_{pg}\}$, with $\psi_{kg} \in \mathbb{R}^+$, $k \in \{1, \ldots, p\}$.

Let $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})'$, with z_{ig} as defined as in Chapter 4. Using the vectorization of $\mathscr{X}_i$, and property (2.10), it can be shown that

$$\mathscr{X}_i \mid z_{ig} = 1 \sim \mathcal{N}_{n\times p}(\mathbf{M}_g, \mathbf{\Sigma}_g + \mathbf{\Lambda}_g \mathbf{\Lambda}_g', \mathbf{\Psi}_g + \mathbf{\Delta}_g \mathbf{\Delta}_g').$$

Therefore, the density of $\mathscr{X}_i$ can be written

$$f(\mathbf{X}_i|\boldsymbol{\vartheta}) = \sum_{g=1}^{G} \pi_g \varphi_{n\times p}(\mathbf{X}_i|\mathbf{M}_g, \boldsymbol{\Sigma}_g + \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g', \boldsymbol{\Psi}_g + \boldsymbol{\Delta}_g\boldsymbol{\Delta}_g'),$$

where $\varphi_{n\times p}(\cdot)$ denotes the $n \times p$ matrix variate normal density. Following a similar procedure to that described by Zhao *et al.* (2012), by introducing latent variables $\mathscr{Y}_{ig}^B$ and $\mathscr{V}_{ig}^B$, (5.3) can be written

$$\begin{aligned}
\mathscr{X}_i &= \mathbf{M}_g + \boldsymbol{\Lambda}_g\mathscr{Y}_{ig}^B + \mathscr{V}_{ig}^B,\\
\mathscr{Y}_{ig}^B &= \mathscr{U}_{ig}\boldsymbol{\Delta}_g' + \mathscr{E}_{ig}^B,\\
\mathscr{V}_{ig}^B &= \mathscr{E}_{ig}^A\boldsymbol{\Delta}_g' + \mathscr{E}_{ig}.
\end{aligned}$$

The two-stage interpretation of this formulation of the model is the same as that given by Zhao *et al.* (2012) where this can viewed as first projecting $\mathscr{X}_i$ in the column direction onto the latent matrix $\mathscr{Y}_{ig}^B$, and then $\mathscr{Y}_{ig}^B$ and $\mathscr{V}_{ig}^B$ are further projected in the row direction. Likewise, introducing $\mathscr{Y}_{ig}^A$ and $\mathscr{V}_{ig}^A$, (5.3) can be written

$$\begin{aligned}
\mathscr{X}_i &= \mathbf{M}_g + \mathscr{Y}_{ig}^A\boldsymbol{\Delta}_g' + \mathscr{V}_{ig}^A,\\
\mathscr{Y}_{ig}^A &= \boldsymbol{\Lambda}_g\mathscr{U}_{ig} + \mathscr{E}_{ig}^A,\\
\mathscr{V}_{ig}^A &= \boldsymbol{\Lambda}_g\mathscr{E}_{ig}^B + \mathscr{E}_{ig}.
\end{aligned}$$

The interpretation is the same as before only we project in the row direction first followed by the column direction. It can be shown that

$$\mathscr{Y}_{ig}^B|\mathbf{X}_i, z_{ig} = 1 \sim \mathcal{N}_{q\times p}(\mathbf{W}_g^{A^{-1}}\boldsymbol{\Lambda}_g'\boldsymbol{\Sigma}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g), \mathbf{W}_g^{A^{-1}}, \boldsymbol{\Psi}_g^*),$$

and

$$\mathscr{Y}_{ig}^{A}|\mathbf{X}_i, z_{ig}=1 \sim \mathcal{N}_{n\times r}((\mathbf{X}_i-\mathbf{M}_g)\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Delta}_g\mathbf{W}_g^{B^{-1}}, \boldsymbol{\Sigma}_g^*, \mathbf{W}_g^{B^{-1}}),$$

where $\mathbf{W}_g^A = \mathbf{I}_q + \boldsymbol{\Lambda}_g'\boldsymbol{\Sigma}_g^{-1}\boldsymbol{\Lambda}_g$, $\mathbf{W}_g^B = \mathbf{I}_r + \boldsymbol{\Delta}_g'\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Delta}_g$, $\boldsymbol{\Sigma}_g^* = \boldsymbol{\Sigma}_g + \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g'$, and $\boldsymbol{\Psi}_g^* = \boldsymbol{\Psi}_g + \boldsymbol{\Delta}_g\boldsymbol{\Delta}_g'$.

5.2.2 Parameter Estimation

Suppose we observe N observations $\mathbf{X}_1, \ldots, \mathbf{X}_N$ then the log-likelihood is given by

$$\mathcal{L}(\boldsymbol{\vartheta}) = \sum_{i=1}^{N}\log\sum_{g=1}^{G}\pi_g\varphi_{n\times p}(\mathbf{X}_i|\boldsymbol{\Sigma}_g+\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g', \boldsymbol{\Psi}_g+\boldsymbol{\Delta}_g\boldsymbol{\Delta}_g'). \tag{5.4}$$

To maximize (5.4), the observed data is viewed as incomplete and an AECM is then to maximize (5.4). There are three different sources of missingingness: the component memberships $\mathbf{z}_1, \ldots, \mathbf{z}_n$ as well as the latent variables $\mathscr{Y}_{ig}^B$ and $\mathscr{Y}_{ig}^A$. A three-stage AECM algorithm is now described for parameter estimation.

AECM Stage 1: In the first stage, the complete-data is taken to be the observed matrices $\mathbf{X}_1, \ldots, \mathbf{X}_N$ and the component memberships $\mathbf{z}_1, \ldots, \mathbf{z}_N$, and the update for $\mathbf{M}_g$ is calculated. The complete-data log-likelihood in the first stage is then

$$\ell^{(1)} = C + \sum_{g=1}^{G}\sum_{i=1}^{N} z_{ig}\left\{\log\pi_g - \frac{1}{2}\operatorname{tr}[\boldsymbol{\Sigma}_g^{*-1}(\mathbf{X}_i-\mathbf{M}_g)\boldsymbol{\Psi}^{*-1}_g(\mathbf{X}_i-\mathbf{M}_g)']\right\},$$

where C is a constant independent of $\mathbf{M}_g$, $\boldsymbol{\Sigma}_g^*$ and $\boldsymbol{\Psi}_g^*$. In the E-Step, the updates

for the component memberships z_{ig} are given by

$$\hat{z}_{ig} = \frac{\pi_g \varphi_{n\times p}(\mathbf{X}_i \mid \hat{\mathbf{M}}_g, \hat{\boldsymbol{\Sigma}}_g^*, \hat{\boldsymbol{\Psi}}_g^*)}{\sum_{h=1}^{G} \pi_g \varphi_{n\times p}(\mathbf{X}_i \mid \hat{\mathbf{M}}_h, \hat{\boldsymbol{\Sigma}}_h^*, \hat{\boldsymbol{\Psi}}_h^*)},$$

where $\varphi_{n\times p}(\cdot)$ denotes the $n \times p$ matrix variate normal density. In the CM-step, the update for $\mathbf{M}_g$ is calculated using

$$\hat{\mathbf{M}}_g = \frac{1}{N_g} \sum_{i=1}^{N} \hat{z}_{ig} \mathbf{X}_i,$$

where $N_g = \sum_{i=1}^{N} \hat{z}_{ig}$.

AECM Stage 2: In the second stage, the complete-data is taken to be the observed $\mathbf{X}_1, \ldots, \mathbf{X}_N$, the component memberships $\mathbf{z}_1, \ldots, \mathbf{z}_N$ and the latent factors $\mathscr{Y}_{\mathbf{i}}^B = (\mathscr{Y}_{i1}^B, \mathscr{Y}_{i2}^B, \ldots, \mathscr{Y}_{iG}^B)$. The complete-data log-likelihood is then

$$\begin{aligned}\ell^{(2)} = C - \frac{N_g p}{2} \log|\boldsymbol{\Sigma}_g| - \frac{1}{2} \sum_{g=1}^{G} \sum_{i=1}^{N} z_{ig} \mathrm{tr}\big[&\boldsymbol{\Sigma}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)\boldsymbol{\Psi}_g^{*-1}(\mathbf{X}_i - \mathbf{M}_g)' \\ &- \boldsymbol{\Sigma}_g^{-1}\boldsymbol{\Lambda}_g \mathscr{Y}_{ig}^B \boldsymbol{\Psi}_g^{*-1}(\mathbf{X}_i - \mathbf{M}_g)' - \boldsymbol{\Sigma}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)\boldsymbol{\Psi}_g^{*-1}\mathscr{Y}_{ig}^{B\prime}\boldsymbol{\Lambda}_g' \\ &+ \boldsymbol{\Sigma}_g^{-1}\boldsymbol{\Lambda}_g \mathscr{Y}_{ig}^B \boldsymbol{\Psi}_g^{*-1}\mathscr{Y}_{ig}^{B\prime}\boldsymbol{\Lambda}_g'\big].\end{aligned}$$

In the E-Step, the following expectations are calculated:

$$a_{ig}^B := \mathbb{E}[\mathscr{Y}_{ig}^B \mid \hat{\boldsymbol{\vartheta}}, \mathbf{X}_i, z_{ig} = 1] = \hat{\mathbf{W}}_g^{A-1} \hat{\boldsymbol{\Lambda}}_g' \hat{\boldsymbol{\Sigma}}_g^{-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g),$$

and

$$b_{ig}^B := \mathbb{E}[\mathscr{Y}_{ig}^B \boldsymbol{\Psi}_g^{*-1} \mathscr{Y}_{ig}^{B\prime} \mid \hat{\boldsymbol{\vartheta}}, \mathbf{X}_i, z_{ig} = 1] = p\hat{\mathbf{W}}_g^{A-1} + a_{ig}^B \hat{\boldsymbol{\Psi}}_g^{*-1} a_{ig}^{B\prime}.$$

As usual, these expectations are calculated using the current estimates of the parameters. In the CM-step $\boldsymbol{\Lambda}_g$ and $\boldsymbol{\Sigma}_g$ are updated via

$$\hat{\boldsymbol{\Lambda}}_g = \sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*-1} a_{ig}^{B\prime} \left(\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}^{B} \right)^{-1},$$
$$\hat{\boldsymbol{\Sigma}}_g = \frac{1}{N_g p} \operatorname{diag}\{\mathbf{S}_g^B\},$$

where

$$\mathbf{S}_g^B = \sum_{i=1}^{N} \hat{z}_{ig} \big[(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g)' - \hat{\boldsymbol{\Lambda}}_g a_{ig}^B \hat{\boldsymbol{\Psi}}_g^{*-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g)' \big].$$

AECM Stage 3: In the last stage of the AECM algorithm, the complete data is taken to be the observed $\mathbf{X}_1, \ldots, \mathbf{X}_N$, the component memberships $\mathbf{z}_1, \ldots, \mathbf{z}_N$ and the latent factors $\mathscr{Y}_{\mathbf{i}}^A = (\mathscr{Y}_{i1}^A, \mathscr{Y}_{i2}^A, \ldots, \mathscr{Y}_{iG}^A)$. In this step, the complete-data log-likelihood is

$$\begin{aligned}\ell^{(3)} = C - \frac{N_g n}{2} \log|\boldsymbol{\Psi}_g| - \frac{1}{2} \sum_{g=1}^{G} \sum_{i=1}^{N} z_{ig} \operatorname{tr} \big[& \boldsymbol{\Psi}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)'\boldsymbol{\Sigma}_g^{*-1}(\mathbf{X}_i - \mathbf{M}_g) \\ & - \boldsymbol{\Psi}_g^{-1}\boldsymbol{\Delta}_g \mathscr{Y}_{ig}^{A\prime}\boldsymbol{\Sigma}_g^{*-1}(\mathbf{X}_i - \mathbf{M}_g) - \boldsymbol{\Psi}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)'\boldsymbol{\Sigma}_g^{*-1}\mathscr{Y}_{ig}^{A}\boldsymbol{\Delta}_g' \\ & + \boldsymbol{\Psi}_g^{-1}\boldsymbol{\Delta}_g \mathscr{Y}_{ig}^{A\prime}\boldsymbol{\Sigma}_g^{*-1}\mathscr{Y}_{ig}^{A}\boldsymbol{\Delta}_g' \big].\end{aligned}$$

In the E-Step, expectations similar to those in the second step are calculated.

$$a_{ig}^A := \mathbb{E}[\mathscr{Y}_{ig}^A \mid \hat{\boldsymbol{\vartheta}}_g \mathbf{X}_i, z_{ig} = 1] = (\mathbf{X}_i - \hat{\mathbf{M}}_g)\boldsymbol{\Psi}_g^{-1}\hat{\boldsymbol{\Delta}}_g \hat{\mathbf{W}}_g^{B-1},$$

and

$$b_{ig}^A := \mathbb{E}[\mathscr{Y}_{ig}^{A\prime}\boldsymbol{\Sigma}_g^{*-1}\mathscr{Y}_{ig}^A \mid \hat{\boldsymbol{\vartheta}}_g, \mathbf{X}_i, z_{ig} = 1] = n\hat{\mathbf{W}}_g^{B-1} + a_{ig}^{A\prime}\hat{\boldsymbol{\Sigma}}_g^{*-1} a_{ig}^A.$$

In the CM-step we update $\boldsymbol{\Delta}_g$ and $\boldsymbol{\Psi}_g$ given by

$$\hat{\boldsymbol{\Delta}}_g = \sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*-1} a_{ig}^A \left(\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}^A \right)^{-1},$$
$$\hat{\boldsymbol{\Psi}}_g = \frac{1}{N_g n} \operatorname{diag}\{\mathbf{S}_g^A\},$$

where

$$\mathbf{S}_g^A = \sum_{i=1}^{N} \hat{z}_{ig} \big[(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g) - \hat{\boldsymbol{\Delta}}_g a_{ig}^{A\prime} \hat{\boldsymbol{\Sigma}}_g^{*-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g) \big].$$

To initialize the AECM algorithm, we employ an alternating emEM strategy (Biernacki *et al.*, 2003). This consists of running the AECM algorithm for a small number of iterations for different random starting values of the parameters and then use the parameters that maximize the likelihood to continue with the AECM algorithm until convergence.

5.2.3 Reduction in Number of Free Covariance Parameters

Because the covariance structure of both covariance matrices in the MVVBFA model is equivalent to the covariance structure in the multivariate MFA model many of the results on the number of free covariance parameters may be used here. Specifically there are $nq + n - q(q-1)/2$ free covariance parameters in $\boldsymbol{\Sigma}_g^*$ and $pr + p - r(r-1)/2$ free covariance parameters in $\boldsymbol{\Psi}_g^*$ (Lawley and Maxwell, 1962). Therefore, reduction in the number of free covariance parameters for the row covariance matrix is

$$\frac{1}{2}n(n+1) - nq - n + \frac{1}{2}q(q-1) = \frac{1}{2}\big[(n-q)^2 - (n+q)\big],$$

which is positive for $(n-q)^2 > n+q$. Likewise for the column covariance matrix the reduction in the number of parameters is

$$\frac{1}{2}p(p+1) - pr - p + \frac{1}{2}r(r-1) = \frac{1}{2}\big[(p-r)^2 - (p+r)\big],$$

which is positive for $(p-r)^2 > p+r$.

In applications herein, the model is fit for a range of row factors and column factors. If the number of row or column factors chosen by the BIC is the maximum in that range, the relevant number of factors will be increased so long as the aforementioned conditions are met.

5.3 Data Analyses

5.3.1 Simulations

Simulation 5.1

In the first simulation, $G = 2$ groups are considered with 10×7 matrices. The mixing proportions are taken to be $\pi_1 = \pi_2 = 0.5$, and we set $N \in \{200, 400, 800\}$. Observations are simulated from (5.3) with $q = 2$ column factors and $r = 3$ row factors. For each value of N, 50 datasets are simulated. For each dataset, for each N, the correct number of groups, column and row factors are selected. In addition, perfect classification is achieved (ARI = 1). In Table 5.1, we show the average value of $\|\mathbf{M}_g - \hat{\mathbf{M}}_g\|_1$, for $g = 1, 2$ and for each value of N, over the 50 datasets. Note that

if $\mathbf{W}$ is an $n \times p$ matrix then

$$\|\mathbf{W}\|_1 = \max_{1 \leq j \leq p} \sum_{i=1}^{n} |w_{ij}|.$$

As expected, the estimates of $\mathbf{M}_g$ get closer to the true values as the sample size N in increased. Moreover, the variability of $\|\mathbf{M}_g - \hat{\mathbf{M}}_g\|_1$ decreases as the sample size increases.

Table 5.1: Average $\|\mathbf{M}_g - \hat{\mathbf{M}}_g\|_1$ values over 50 datasets, for $g = 1, 2$ and $N = 200, 400, 800$, in Simulation 5.1, with standard deviations in parentheses.

	N		
g	200	400	800
1	13.97(3.61)	9.66(2.65)	6.48(1.69)
2	12.08(3.25)	7.45(1.79)	5.69(1.32)

Simulation 5.2

The second simulation considers $G = 3$ groups with 28×17 matrices. The mixing proportions are $\pi_1 = \pi_3 = 0.4$ and $\pi_2 = 0.2$, and $N \in \{250, 500, 1000\}$. Again, 50 datasets are simulated for each N with $q = 2$ column factors and $r = 3$ row factors. As in Simulation 5.1, the correct number of groups, column and row factors are chosen and perfect classification is achieved. In Table 5.2, we again show the average 1-norms for the differences between the true and estimated location parameters.

5.3.2 MNIST Digit Recognition

We consider the 28×28 MNIST digit dataset (LeCun *et al.*, 1998), which contains over 60,000 greyscale images of handwritten Arabic digits 0 to 9. The images are

Table 5.2: Average $\|\mathbf{M}_g - \hat{\mathbf{M}}_g\|_1$ values over 50 datasets, for $g = 1, 2, 3$ and $N = 250, 500, 1000$, in Simulation 5.2, with standard deviations in parentheses.

	N		
g	250	500	1000
1	36.28(7.95)	26.36(5.12)	19.37(4.62)
2	55.23(11.75)	40.42(9.64)	29.30(6.26)
3	39.10(8.89)	27.09(6.37)	19.99(4.45)

represented by 28×28 pixel matrices with greyscale intensities ranging from 0 to 255. Because of the lack of variability in the outer rows and columns, some random noise is added while adding 50 to each of the non-zero elements to avoid confusing the noise with a true signal. We are interested in comparing digit 1 to digit 7, as was considered in Gallaugher and McNicholas (2018b). Similar to Gallaugher and McNicholas (2018b), we consider semi-supervised classification with 25%, 50% and 75% supervision. In each case, 25 datasets are considered, each consisting of 200 observations from each digit, and we fit the model for 10 to 20 column and row factors.

In Table 5.3, we show an aggregated classification table between the true and predicted classifications at each level of supervision for the points considered unlabelled. As expected, slightly better classification performance is obtained when the level of supervision is increased. Moreover, there is a more substantial difference when going from 25% supervision to 50% supervision than from 50% to 75%.

Table 5.4 shows the average ARI and MCR over the 25 datasets, with the respective standard deviations, for each level of supervision. We note that we obtain better results than Gallaugher and McNicholas (2018b) even with a lower level of supervision; however, the results in Gallaugher and McNicholas (2018b) were based on resized images due to dimensionality constraints whereas this analysis was performed

Table 5.3: Cross-tabulations of true (1,7) versus predicted (P1, P7) classifications for the observations considered unlabelled in the MNIST data at each level of supervision, aggregated over all runs.

	25% Supervision		50% Supervision		75% Supervision	
	P1	P7	P1	P7	P1	P7
1	3550	173	2449	53	1232	26
7	200	3577	51	2447	18	1221

on the original images.

Table 5.4: Average ARI values and MCR, with associated standard deviations in parentheses, for each level of supervision for the points considered unlabelled for the MNIST data, aggregated over all runs.

	$\overline{\text{ARI}}$ (std. dev.)	$\overline{\text{MCR}}$(std. dev.)
25%	0.82(0.15)	0.050(0.046)
50%	0.92 (0.056)	0.021 (0.015)
75%	0.93 (0.056)	0.018 (0.015)

In Table 5.5 the frequency of the number of factors chosen for each level of supervision over the 25 datasets is shown. For the majority of the datasets, the number of row and column factors lie between 13 and 15.

Finally, in Figure 5.1, heatmaps are displayed for the average estimates of the location matrices over the 25 runs for each level of supervision for both digits. We see a slight increase in quality when going from 25% to 50% supervision for digit 7 with the centre of the digit being a little smoother with 50% supervision. There is no noticeable difference when going from 50% to 75% supervision. This similarity across the three levels of supervision illustrates the power of semi-supervised classification.

Table 5.5: Numbers of row and columns factors chosen for the MNIST dataset for 25%, 50% and 75% supervision.

	10	11	12	13	14	15	16	17	18	19	20
	25% Supervision										
Row Factors	0	0	0	2	7	6	4	3	2	1	0
Column Factors	0	0	2	6	7	6	3	1	0	0	0
	50% Supervision										
Row Factors	0	0	0	4	6	10	2	0	1	1	1
Column Factors	0	0	2	9	7	5	1	1	0	0	0
	75% Supervision										
Row Factors	0	0	0	1	9	9	3	3	0	0	0
Column Factors	0	0	0	9	11	4	0	0	0	0	1

5.3.3 Olivetti Faces Dataset

Finally, consider the Olivetti faces dataset from the R package `RnavGraphImageData` (Waddell and Oldford, 2013). The dataset consists of greyscale images of faces that were taken between 1992 and 1994 at AT&T laboratories in Cambridge. There were 40 individuals with 10 images of each individual for a total of 400 64×64 images. The images were taken with varied lighting, expressions (eyes open/closed, smile/frown etc.), and glasses or no glasses. We fit the model for 15 to 30 column and row factors, and for $G = 1, \ldots, 9$ components. The BIC chooses three components with 23 column factors and 26 row factors. The estimated mixing proportions are $\pi_1 = 0.22, \pi_2 = 0.49, \pi_3 = 0.29$. In Figure 5.2, we show a heatmap of the estimated location parameters for each component. The heatmap for component 3 arguably shows the clearest image and appears to display the glasses feature.

Upon looking at individual faces classified to component 3 (Figure 5.2), all the faces have glasses. Moreover, all faces with glasses are classified to component 3 with the exception of two which are classified to component 2. The faces with closed eyes are scattered throughout the three different components and are not classified

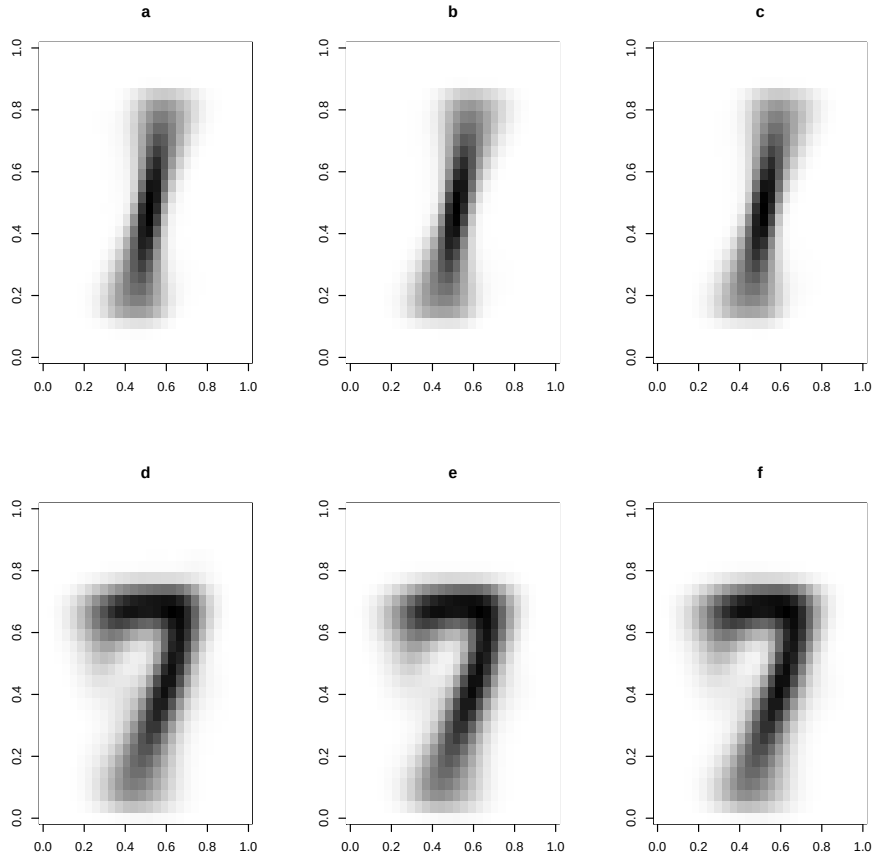


Figure 5.1: Heatmaps for the average estimated location matrices taken over the 25 runs for digit 1 at 25%, 50% and 75% supervision, respectively (a, b, c), and digit 7 at 25%, 50% and 75% supervision, respectively (d, e, f).

to any one component. Although it is a difficult to determine the main feature that differentiates component 1 from component 2, it is apparent that the eyebrows for the faces classified to component 1 tend to be more prominent and higher above the eyelid. Of course, a semi-supervised approach to these data could be used to detect specific classes, similar to the MNIST analysis (Section 5.3.2). However, the unsupervised analysis here has shown that the MMVBFA approach can be effective at detecting subgroups without training.

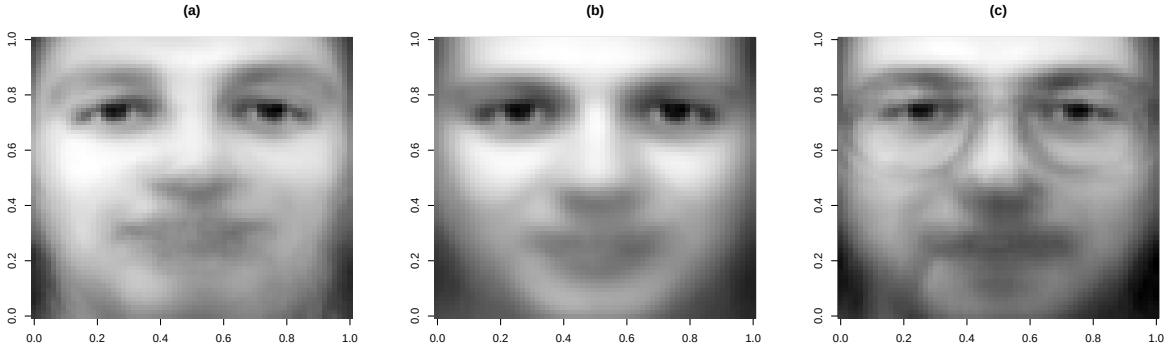


Figure 5.2: Estimated location matrices for (a) component 1, (b) component 2, and (c) component 3 for the faces dataset.

5.4 Summary

In this chapter, we developed a MMVBFA model for use in clustering and classification of matrix variate data. Two simulations as well as two real data examples were used for illustration. For each of the simulations, the correct number of components and column/row factors were chosen by the BIC for all of the datasets. Perfect classification performance was also obtained in the simulations. In the MNIST digit application, even with a lower level of supervision, we obtained better results than Gallaugher and McNicholas (2018b). However, this is probably due to the fact that the MMVBFA model could use the full 28×28 image. In the faces application, the BIC chooses three groups with the third group being defined by the presence of the glasses facial feature.

Chapter 6

Mixtures of Skewed Matrix Variate Bilinear Factor Analyzers

6.1 Model Specification

We now consider a mixture of skewed bilinear factor analyzers according to one of the four skewed distributions discussed previously. Each random matrix $\mathscr{X}_i$ from a random sample distributed according to one of the four distributions can be written

$$\mathscr{X}_i = \mathbf{M}_g + W_{ig}\mathbf{A}_g + \mathscr{V}_{ig}$$

with probability π_g for $g \in \{1, 2, \ldots, G\}$, $\pi_g > 0$, $\sum_{i=1}^{G} \pi_g = 1$, where $\mathbf{M}_g$ is the location of the gth component, $\mathbf{A}_g$ is the skewness, and W_{ig} is a random variable with density $h(w_{ig}|\boldsymbol{\theta}_g)$. The distribution of the random variable W_{ig} — and so the density $h(w_{ig}|\boldsymbol{\theta}_g)$ — will change depending on the distribution of $\mathscr{X}_i$, i.e., skew-t, generalized

hyperbolic, variance-gamma, or NIG. Assume also that $\mathscr{V}_{ig}$ can be written as

$$\mathscr{V}_{ig} = \boldsymbol{\Lambda}_g \mathscr{U}_{ig} \boldsymbol{\Delta}_g' + \boldsymbol{\Lambda}_g \mathscr{E}_{ig}^B + \mathscr{E}_{ig}^A \boldsymbol{\Delta}_g' + \mathscr{E}_{ig},$$

where $\boldsymbol{\Lambda}_g$ is a $n \times q$ matrix of column factor loadings, $\boldsymbol{\Delta}_g$ is a $p \times r$ matrix of row factor loadings, and

$$\begin{aligned} \mathscr{U}_{ig}|w_{ig} &\sim \mathcal{N}_{q\times r}(\mathbf{0}, w_{ig}\mathbf{I}_q, \mathbf{I}_p), & \mathscr{E}_{ig}^B|w_{ig} &\sim \mathcal{N}_{q\times p}(\mathbf{0}, w_{ig}\mathbf{I}_q, \boldsymbol{\Psi}_g), \\ \mathscr{E}_{ig}^A|w_{ig} &\sim \mathcal{N}_{n\times r}(\mathbf{0}, w_{ig}\boldsymbol{\Sigma}_g, \mathbf{I}_r), & \mathscr{E}_{ig}|w_{ig} &\sim \mathcal{N}_{n\times p}(\mathbf{0}, w_{ig}\boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g). \end{aligned}$$

Note that $\mathscr{U}_{ig}, \mathscr{E}_{ig}^B, \mathscr{E}_{ig}^A$, and $\mathscr{E}_{ig}$ are all independently distributed and independent of each other.

To facilitate clustering, introduce the indicator z_{ig}, where $z_{ig} = 1$ if observation i belongs to group g, and $z_{ig} = 0$ otherwise. Then, it can be shown that

$$\mathscr{X}_i \mid z_{ig} = 1 \sim \mathrm{D}_{n\times p}(\mathbf{M}_g, \mathbf{A}_g, \boldsymbol{\Sigma}_g + \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g', \boldsymbol{\Psi}_g + \boldsymbol{\Delta}_g\boldsymbol{\Delta}_g', \boldsymbol{\theta}_g),$$

where D is the distribution in question, and $\boldsymbol{\theta}_g$ is the set of parameters related to the distribution of W_{ig}.

As in the matrix variate normal case, this model has a two stage interpretation given by

$$\begin{aligned} \mathscr{X}_i &= \mathbf{M}_g + W_{ig}\mathbf{A} + \boldsymbol{\Lambda}_g \mathscr{Y}_{ig}^B + \mathscr{R}_{ig}^B, \\ \mathscr{Y}_{ig}^B &= \mathscr{U}_{ig}\boldsymbol{\Delta}_g' + \mathscr{E}_{ig}^B, \\ \mathscr{R}_{ig}^B &= \mathscr{E}_{ig}^A \boldsymbol{\Delta}_g' + \mathscr{E}_{ig}, \end{aligned}$$

and

$$\begin{aligned}\mathscr{X}_i &= \mathbf{M}_g + W_{ig}\mathbf{A} + \mathscr{Y}_{ig}^A\boldsymbol{\Delta}_g' + \mathscr{R}_{ig}^A,\\ \mathscr{Y}_{ig}^A &= \boldsymbol{\Lambda}_g\mathscr{U}_{ig} + \mathscr{E}_{ig}^A,\\ \mathscr{R}_{ig}^A &= \boldsymbol{\Lambda}_g\mathscr{E}_{ig}^B + \mathscr{E}_{ig},\end{aligned}$$

which will be useful for parameter estimation.

6.2 Parameter Estimation

Suppose we observe the N $n \times p$ matrices $\mathbf{X}_1, \ldots, \mathbf{X}_N$ distributed according to one of the four distributions. We assume that these data are incomplete and employ an alternating expectation conditional maximization (AECM) algorithm (Meng and van Dyk, 1997). This algorithm is now described after inititalization.

AECM 1st Stage The complete-data in the first stage consists of the observed data $\mathbf{X}_i$, the latent variables $\mathbf{W}_i = (W_{i1}, \ldots, W_{iG})'$, and the unknown group labels $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})'$ for $i = 1, 2, \ldots, N$. In this case, the complete-data log-likelihood is

$$\begin{aligned}\ell_{\text{C1}} = C + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\Bigg[&\log\pi_g + \log h(w_{ig}|\boldsymbol{\theta}_g)\\ &- \frac{1}{2}\operatorname{tr}\Bigg\{\frac{1}{W_{ig}}(\boldsymbol{\Sigma}_g^*)^{-1}(\mathbf{X}_i - \mathbf{M}_g)(\boldsymbol{\Psi}_g^*)^{-1}(\mathbf{X}_i - \mathbf{M}_g)'\\ &- (\boldsymbol{\Sigma}_g^*)^{-1}(\mathbf{X}_i - \mathbf{M}_g)(\boldsymbol{\Psi}_g^*)^{-1}\mathbf{A}_g' - (\boldsymbol{\Sigma}_g^*)^{-1}\mathbf{A}_g(\boldsymbol{\Psi}_g^*)^{-1}(\mathbf{X}_i - \mathbf{M}_g)'\\ &+ W_{ig}(\boldsymbol{\Sigma}_g^*)^{-1}\mathbf{A}_g(\boldsymbol{\Psi}_g^*)^{-1}\mathbf{A}_g'\Bigg\}\Bigg],\end{aligned}$$

where $\boldsymbol{\Sigma}_g^* = \boldsymbol{\Sigma}_g + \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g'$, $\boldsymbol{\Psi}_g^* = \boldsymbol{\Psi}_g + \boldsymbol{\Delta}_g\boldsymbol{\Delta}_g'$ and C is constant with respect to the parameters.

In the E-step, we calculate the following conditional expectations:

$$\hat{z}_{ig} = \frac{\pi_g f(\mathbf{X}_i \mid \hat{\boldsymbol{\vartheta}}_g)}{\sum_{h=1}^{G} \pi_h f(\mathbf{X}_i \mid \hat{\boldsymbol{\vartheta}}_h)}, \qquad a_{ig} = \mathbb{E}(W_{ig} \mid \mathbf{X}_i, z_{ig} = 1, \hat{\boldsymbol{\vartheta}}_g),$$
$$b_{ig} = \mathbb{E}\left(\frac{1}{W_{ig}} \;\middle|\; \mathbf{X}_i, z_{ig} = 1, \hat{\boldsymbol{\vartheta}}_g\right), \qquad c_{ig} = \mathbb{E}(\log W_{ig} \mid \mathbf{X}_i, z_{ig} = 1, \hat{\boldsymbol{\vartheta}}_g).$$

As usual, all expectations are conditional on current parameter estimates; however, to avoid cluttered notation, we do not use iteration-specific notation. Similar to the mixtures of skewed matrix variate distributions presented in Chapter 4, it can be shown

$$W_{ig}^{\text{ST}} \mid \mathbf{X}_i, z_{ig} = 1 \sim \text{GIG}\left(\rho(\mathbf{A}_g, \boldsymbol{\Sigma}_g^*, \boldsymbol{\Psi}_g^*), \delta(\mathbf{X}; \mathbf{M}_g, \boldsymbol{\Sigma}_g^*, \boldsymbol{\Psi}_g^*) + \nu_g, -(\nu_g + np)/2\right),$$
$$W_{ig}^{\text{GH}} \mid \mathbf{X}_i, z_{ig} = 1 \sim \text{GIG}\left(\rho(\mathbf{A}_g, \boldsymbol{\Sigma}_g^*, \boldsymbol{\Psi}_g^*) + \omega_g, \delta(\mathbf{X}; \mathbf{M}_g, \boldsymbol{\Sigma}_g^*, \boldsymbol{\Psi}_g^*) + \omega_g, \lambda_g - np/2\right),$$
$$W_{ig}^{\text{VG}} \mid \mathbf{X}_i, z_{ig} = 1 \sim \text{GIG}\left(\rho(\mathbf{A}_g, \boldsymbol{\Sigma}_g^*, \boldsymbol{\Psi}_g^*) + 2\gamma_g, \delta(\mathbf{X}; \mathbf{M}_g, \boldsymbol{\Sigma}_g^*, \boldsymbol{\Psi}_g^*), \gamma_g - np/2\right),$$
$$W_{ig}^{\text{NIG}} \mid \mathbf{X}_i, z_{ig} = 1 \sim \text{GIG}\left(\rho(\mathbf{A}_g, \boldsymbol{\Sigma}_g^*, \boldsymbol{\Psi}_g^*) + \kappa_g^2, \delta(\mathbf{X}; \mathbf{M}_g, \boldsymbol{\Sigma}_g^*, \boldsymbol{\Psi}_g^*) + 1, -(1 + np)/2\right).$$

Therefore, the exact updates are again obtained by using the expectations given in (2.4)–(2.6) for appropriate values of λ, a, and b.

In the M-step, we update $\hat{\pi}_g, \hat{\mathbf{M}}_g, \hat{\mathbf{A}}_g$, and $\hat{\boldsymbol{\theta}}_g$ for $g = 1, \ldots, G$. We have:

$$\hat{\pi}_g = \frac{N_g}{N}, \qquad \hat{\mathbf{M}}_g = \frac{\sum_{i=1}^{N} \hat{z}_{ig}\left(\overline{a}_g b_{ig} - 1\right)\mathbf{X}_i}{\sum_{i=1}^{N} \hat{z}_{ig}\overline{a}_g b_{ig} - N_g}, \qquad \hat{\mathbf{A}} = \frac{\sum_{i=1}^{N} \hat{z}_{ig}\left(\overline{b}_g - b_{ig}\right)\mathbf{X}_i}{\sum_{i=1}^{N} \hat{z}_{ig}\overline{a}_g b_{ig} - N_g},$$

where

$$N_g = \sum_{i=1}^{N} \hat{z}_{ig}, \qquad \overline{a}_g = \frac{1}{N_g}\sum_{i=1}^{N} \hat{z}_{ig} a_{ig}, \qquad \overline{b}_g = \frac{1}{N_g}\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}.$$

The update for $\boldsymbol{\theta}_g$ is dependent on the distribution and will be identical to one of those given in Gallaugher and McNicholas (2018b).

AECM Stage 2 In the second stage, the complete-data consists of the observed data $\mathbf{X}_i$, the latent variables $\mathbf{W}_i$, the unknown group labels $\mathbf{z}_i$, and the latent matrices $\mathscr{Y}_i^B = (\mathscr{Y}_{i1}^B, \ldots, \mathscr{Y}_{iG}^B)$ for $i = 1, \ldots, N$. The complete-data log-likelihood at this stage is

$$\begin{aligned}
\ell_{\text{C2}} = C + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\big[& \log \pi_g + \log h(W_{ig}|\nu_g) + \log \phi_{q\times p}(\mathscr{Y}_{ig}^B|\mathbf{0}, W_{ig}\mathbf{I}_q, \boldsymbol{\Psi}_g^*) \\
& + \log \phi_{n\times p}(\mathbf{X}_i|\mathbf{M}_g + W_{ig}\mathbf{A}_g + \boldsymbol{\Lambda}_g\mathscr{Y}_{ig}^B, W_{ig}\boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g^*)\big] \\
= C + \sum_{i=1}^{N}\sum_{g=1}^{G} -\frac{1}{2} z_{ig}\Bigg[& - p\log|\boldsymbol{\Sigma}_g| + \operatorname{tr}\Bigg\{ \frac{1}{W_{ig}}\boldsymbol{\Sigma}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)(\boldsymbol{\Psi}_g^*)^{-1}(\mathbf{X}_i - \mathbf{M}_g)' \\
& - \boldsymbol{\Sigma}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)(\boldsymbol{\Psi}_g^*)^{-1}\mathbf{A}_g' - \frac{1}{W_{ig}}\boldsymbol{\Sigma}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)(\boldsymbol{\Psi}_g^*)^{-1}\mathscr{Y}_{ig}^{B\prime}\boldsymbol{\Lambda}_g' \\
& - \boldsymbol{\Sigma}_g^{-1}\mathbf{A}_g(\boldsymbol{\Psi}_g^*)^{-1}(\mathbf{X}_i - \mathbf{M}_g)' + W_{ig}\boldsymbol{\Sigma}_g^{-1}\mathbf{A}_g(\boldsymbol{\Psi}_g^*)^{-1}\mathbf{A}_g' + \boldsymbol{\Sigma}_g^{-1}\mathbf{A}_g(\boldsymbol{\Psi}_g^*)^{-1}\mathscr{Y}_{ig}^{B\prime}\boldsymbol{\Lambda}_g' \\
& - \frac{1}{W_{ig}}\boldsymbol{\Sigma}_g^{-1}\boldsymbol{\Lambda}_g\mathscr{Y}_{ig}^B(\boldsymbol{\Psi}_g^*)^{-1}(\mathbf{X}_i - \mathbf{M}_g)' + \boldsymbol{\Sigma}_g^{-1}\boldsymbol{\Lambda}_g\mathscr{Y}_{ig}^B(\boldsymbol{\Psi}_g^*)^{-1}\mathbf{A}_g' \\
& + \frac{1}{W_{ig}}\boldsymbol{\Sigma}_g^{-1}\boldsymbol{\Lambda}_g\mathscr{Y}_{ig}^B(\boldsymbol{\Psi}_g^*)^{-1}\mathscr{Y}_{ig}^{B\prime}\boldsymbol{\Lambda}_g' \Bigg\}\Bigg].
\end{aligned}$$

In the E-step, it can be shown that

$$\begin{aligned}
&\mathscr{Y}_{ig}^B \mid \mathbf{X}_i, W_{ig}, z_{ig} = 1 \sim \\
&\quad \mathcal{N}_{q\times p}((\mathbf{I}_q + \boldsymbol{\Lambda}_g'\boldsymbol{\Sigma}_g^{-1}\boldsymbol{\Lambda}_g)^{-1}\boldsymbol{\Lambda}_g'\boldsymbol{\Sigma}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g - W_{ig}\mathbf{A}_g), W_{ig}(\mathbf{I}_q + \boldsymbol{\Lambda}_g'\boldsymbol{\Sigma}_g^{-1}\boldsymbol{\Lambda}_g)^{-1}, \boldsymbol{\Psi}_g^*)
\end{aligned}$$

and so we can calculate the expectations

$$\begin{aligned}
\mathbf{E}_{1ig}^{(2)} &:= \mathbb{E}[\mathscr{Y}_{ig}^{B}|\hat{\boldsymbol{\vartheta}}, \mathbf{X}_i, z_{ig}=1] = \mathbf{L}_g(\mathbf{X}_i - \hat{\mathbf{M}}_g - a_{ig}\hat{\mathbf{A}}_g),\\
\mathbf{E}_{2ig}^{(2)} &:= \mathbb{E}\left[\frac{1}{W_{ig}}\mathscr{Y}_{ig}^{B}\middle|\hat{\boldsymbol{\vartheta}}, \mathbf{X}_i, z_{ig}=1\right] = \mathbf{L}_g(b_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g) - \hat{\mathbf{A}}_g),\\
\mathbf{E}_{3ig}^{(2)} &:= \mathbb{E}\left[\frac{1}{W_{ig}}\mathscr{Y}_{ig}^{B}(\boldsymbol{\Psi}_g^*)^{-1}\mathscr{Y}_{ig}^{B\prime}\middle|\hat{\boldsymbol{\vartheta}}, \mathbf{X}_i, z_{ig}=1\right]\\
&= p(\mathbf{I}_q + \hat{\boldsymbol{\Lambda}}_g'\hat{\boldsymbol{\Sigma}}_g^{-1}\hat{\boldsymbol{\Lambda}}_g)^{-1} + b_{ig}\mathbf{L}_g(\mathbf{X}_i - \hat{\mathbf{M}}_g)(\boldsymbol{\Psi}_g^*)^{-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\mathbf{L}_g'\\
&\quad - \mathbf{L}_g((\mathbf{X}_i - \hat{\mathbf{M}}_g)(\hat{\boldsymbol{\Psi}}_g^*)^{-1}\hat{\mathbf{A}}_g' + \hat{\mathbf{A}}_g(\hat{\boldsymbol{\Psi}}_g^*)^{-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g)')\mathbf{L}_g' + a_{ig}\mathbf{L}_g\hat{\mathbf{A}}_g(\hat{\boldsymbol{\Psi}}_g^*)^{-1}\hat{\mathbf{A}}_g'\mathbf{L}_g',
\end{aligned}$$

where $\mathbf{L}_g = (\mathbf{I}_q + \hat{\boldsymbol{\Lambda}}_g'\hat{\boldsymbol{\Sigma}}_g^{-1}\hat{\boldsymbol{\Lambda}}_g)^{-1}\hat{\boldsymbol{\Lambda}}_g'\hat{\boldsymbol{\Sigma}}_g^{-1}$.

In the M-step, the updates for $\boldsymbol{\Lambda}_g$ and $\boldsymbol{\Sigma}_g$ are calculated. These updates are given by

$$\hat{\boldsymbol{\Lambda}}_g = \sum_{i=1}^{N}\hat{z}_{ig}\left[(\mathbf{X}_i - \hat{\mathbf{M}}_g)(\hat{\boldsymbol{\Psi}}_g^*)^{-1}\mathbf{E}_{2ig}^{(2)\prime} - \hat{\mathbf{A}}_g(\hat{\boldsymbol{\Psi}}_g^*)^{-1}\mathbf{E}_{1ig}^{(2)\prime}\right]\left(\sum_{i=1}^{N}z_{ig}\mathbf{E}_{3ig}^{(2)}\right)^{-1}$$

and $\hat{\boldsymbol{\Sigma}}_g = \operatorname{diag}(\mathbf{S}_g^L)$, respectively, where

$$\begin{aligned}
\mathbf{S}_g^L = \frac{1}{N_g p}\sum_{i=1}^{N}\hat{z}_{ig}\big[&b_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)(\hat{\boldsymbol{\Psi}}_g^*)^{-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g)' - (\hat{\mathbf{A}}_g + \hat{\boldsymbol{\Lambda}}_g\mathbf{E}_{2ig}^{(2)})(\hat{\boldsymbol{\Psi}}_g^*)^{-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\\
&- (\mathbf{X}_i - \hat{\mathbf{M}}_g)(\hat{\boldsymbol{\Psi}}_g^*)^{-1}\hat{\mathbf{A}}_g' + a_{ig}\hat{\mathbf{A}}_g(\hat{\boldsymbol{\Psi}}_g^*)^{-1}\hat{\mathbf{A}}_g + \hat{\boldsymbol{\Lambda}}_g\mathbf{E}_{1ig}^{(1)}(\hat{\boldsymbol{\Psi}}_g^*)^{-1}\hat{\mathbf{A}}_g'\\
&- (\mathbf{X}_i - \hat{\mathbf{M}}_g)(\hat{\boldsymbol{\Psi}}_g^*)^{-1}\mathbf{E}_{2ig}^{(2)\prime}\hat{\boldsymbol{\Lambda}}_g' + \hat{\mathbf{A}}_g(\hat{\boldsymbol{\Psi}}_g^*)^{-1}\mathbf{E}_{1ig}^{(2)\prime}\hat{\boldsymbol{\Lambda}}_g' + \hat{\boldsymbol{\Lambda}}_g\mathbf{E}_{3ig}^{(2)}\hat{\boldsymbol{\Lambda}}_g'\big].
\end{aligned}$$

AECM Stage 3 In the third stage, the complete-data consists of the observed data $\mathbf{X}_i$, the latent variables $\mathbf{W}_i$, the labels $\mathbf{z}_i$ and the latent matrices $\mathscr{Y}_i^A = (\mathscr{Y}_{i1}^A, \ldots, \mathscr{Y}_{iG}^A)$

for $i = 1, \ldots, N$. The complete-data log-likelihood at this stage is

$$\begin{aligned}
\ell_{\text{C3}} &= C + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig}\big[\log\pi_g + \log h(W_{ig}|\nu_g) + \log\phi_{q\times p}(\mathscr{Y}_{ig}^{A}|\mathbf{0}, W_{ig}\boldsymbol{\Sigma}_g^*, \mathbf{I}_p) \\
&\quad + \log\phi_{n\times p}(\mathbf{X}_i|\mathbf{M}_g + W_{ig}\mathbf{A}_g + \mathscr{Y}_{ig}^{A}\boldsymbol{\Delta}_g', W_{ig}\boldsymbol{\Sigma}_g^*, \boldsymbol{\Psi}_g)\big] \\
&= C + \sum_{i=1}^{N}\sum_{g=1}^{G} -\frac{1}{2} z_{ig}\Bigg[- n\log|\boldsymbol{\Psi}_g| + \operatorname{tr}\Bigg\{\frac{1}{W_{ig}}\boldsymbol{\Psi}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)'(\boldsymbol{\Sigma}_g^*)^{-1}(\mathbf{X}_i - \mathbf{M}_g) \\
&\quad - \boldsymbol{\Psi}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)'(\boldsymbol{\Sigma}_g^*)^{-1}\mathbf{A}_g - \frac{1}{W_{ig}}\boldsymbol{\Psi}_g^{-1}(\mathbf{X}_i - \mathbf{M}_g)'(\boldsymbol{\Sigma}_g^*)^{-1}\mathscr{Y}_{ig}^{A}\boldsymbol{\Delta}_g' \\
&\quad - \boldsymbol{\Psi}_g^{-1}\mathbf{A}_g'(\boldsymbol{\Sigma}_g^*)^{-1}(\mathbf{X}_i - \mathbf{M}_g) + W_{ig}\boldsymbol{\Psi}_g^{-1}\mathbf{A}_g'(\boldsymbol{\Sigma}_g^*)^{-1}\mathbf{A}_g \\
&\quad + \boldsymbol{\Psi}_g^{-1}\mathbf{A}_g'(\boldsymbol{\Sigma}_g^*)^{-1}\mathscr{Y}_{ig}^{A}\boldsymbol{\Delta}_g' - \frac{1}{W_{ig}}\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Delta}_g\mathscr{Y}_{ig}^{A\prime}(\boldsymbol{\Sigma}_g^*)^{-1}(\mathbf{X}_i - \mathbf{M}_g) \\
&\quad + \boldsymbol{\Psi}_g^{-1}\boldsymbol{\Delta}_g\mathscr{Y}_{ig}^{A\prime}(\boldsymbol{\Sigma}_g^*)^{-1}\mathbf{A}_g + \frac{1}{W_{ig}}\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Delta}_g\mathscr{Y}_{ig}^{A\prime}(\boldsymbol{\Sigma}_g^*)^{-1}\mathscr{Y}_{ig}^{A}\boldsymbol{\Delta}_g'\Bigg\}\Bigg].
\end{aligned}$$

In the E-step, it can be shown that

$$\begin{aligned}
&\mathscr{Y}_{ig}^{A}|\mathbf{X}_i, W_{ig}, z_{ig} = 1 \sim \\
&\quad \mathcal{N}_{n\times r}((\mathbf{X}_i - \mathbf{M}_g - W_{ig}\mathbf{A}_g)\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Delta}_g(\mathbf{I}_r + \boldsymbol{\Delta}_g'\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Delta}_g)^{-1}, W_{ig}\boldsymbol{\Sigma}_g^*, (\mathbf{I}_r + \boldsymbol{\Delta}_g'\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Delta}_g)^{-1})
\end{aligned}$$

and so we can calculate the expectations

$$\begin{aligned}
\mathbf{E}_{1ig}^{(3)} &:= \mathbb{E}[\mathscr{Y}_{ig}^{A}|\hat{\boldsymbol{\vartheta}}, \mathbf{X}_i, z_{ig} = 1] = (\mathbf{X}_i - \hat{\mathbf{M}}_g - a_{ig}\hat{\mathbf{A}}_g)\mathbf{D}_g, \\
\mathbf{E}_{2ig}^{(3)} &:= \mathbb{E}\left[\frac{1}{W_{ig}}\mathscr{Y}_{ig}^{A}|\hat{\boldsymbol{\vartheta}}, \mathbf{X}_i, z_{ig} = 1\right] = (b_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g) - \hat{\mathbf{A}}_g)\mathbf{D}_g, \\
\mathbf{E}_{3ig}^{(3)} &:= \mathbb{E}\left[\frac{1}{W_{ig}}\mathscr{Y}_{ig}^{A\prime}(\boldsymbol{\Sigma}_g^*)^{-1}\mathscr{Y}_{ig}^{A}|\hat{\boldsymbol{\vartheta}}, \mathbf{X}_i, z_{ig} = 1\right] \\
&= n(\mathbf{I}_r + \hat{\boldsymbol{\Delta}}_g'\hat{\boldsymbol{\Psi}}_g^{-1}\hat{\boldsymbol{\Delta}}_g)^{-1} + b_{ig}\mathbf{D}_g'(\mathbf{X}_i - \hat{\mathbf{M}}_g)'(\hat{\boldsymbol{\Sigma}}_g^*)^{-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\mathbf{D}_g \\
&\quad - \mathbf{D}_g'((\mathbf{X}_i - \hat{\mathbf{M}}_g)'(\hat{\boldsymbol{\Sigma}}_g^*)^{-1}\hat{\mathbf{A}}_g + \hat{\mathbf{A}}_g'(\hat{\boldsymbol{\Sigma}}_g^*)^{-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g))\mathbf{D}_g + a_{ig}\mathbf{D}_g'\hat{\mathbf{A}}_g'(\hat{\boldsymbol{\Sigma}}_g^*)^{-1}\hat{\mathbf{A}}_g\mathbf{D}_g,
\end{aligned}$$

where $\mathbf{D}_g = \hat{\mathbf{\Psi}}_g^{-1}\hat{\mathbf{\Delta}}_g(\mathbf{I}_r + \hat{\mathbf{\Delta}}_g'\hat{\mathbf{\Psi}}_g^{-1}\hat{\mathbf{\Delta}}_g)^{-1}$.

In the M-step, the updates for $\mathbf{\Delta}_g$ and $\mathbf{\Psi}_g$ are calculated. These updates are given by

$$\hat{\mathbf{\Delta}}_g = \sum_{i=1}^{N} \hat{z}_{ig}[(\mathbf{X}_i - \hat{\mathbf{M}}_g)'(\hat{\mathbf{\Sigma}}_g^*)^{-1}\mathbf{E}_{2ig}^{(3)} - \hat{\mathbf{A}}_g'(\hat{\mathbf{\Sigma}}_g^*)^{-1}\mathbf{E}_{1ig}^{(3)}](\sum_{i=1}^{N} z_{ig}\mathbf{E}_{3ig}^{(3)})^{-1}$$

and $\hat{\mathbf{\Psi}}_g = \text{diag}(\mathbf{S}_g^D)$, respectively, where

$$\begin{aligned}\mathbf{S}_g^D = \frac{1}{N_g p}\sum_{i=1}^{N} \hat{z}_{ig}[&b_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'(\hat{\mathbf{\Sigma}}_g^*)^{-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g) - (\hat{\mathbf{A}}_g' + \hat{\mathbf{\Delta}}_g\mathbf{E}_{2ig}^{(3)'})(\hat{\mathbf{\Sigma}}_g^*)^{-1}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\\ &- (\mathbf{X}_i - \hat{\mathbf{M}}_g)'(\hat{\mathbf{\Sigma}}_g^*)^{-1}\hat{\mathbf{A}}_g + a_{ig}\hat{\mathbf{A}}_g'(\hat{\mathbf{\Sigma}}_g^*)^{-1}\hat{\mathbf{A}}_g + \hat{\mathbf{\Delta}}_g\mathbf{E}_{1ig}^{(3)'}(\hat{\mathbf{\Sigma}}_g^*)^{-1}\hat{\mathbf{A}}_g\\ &- (\mathbf{X}_i - \hat{\mathbf{M}}_g)'(\hat{\mathbf{\Sigma}}_g^*)^{-1}\mathbf{E}_{2ig}^{(3)}\hat{\mathbf{\Delta}}_g' + \hat{\mathbf{A}}_g'(\hat{\mathbf{\Sigma}}_g^*)^{-1}\mathbf{E}_{1ig}^{(3)}\hat{\mathbf{\Delta}}_g' + \hat{\mathbf{\Delta}}_g\mathbf{E}_{3ig}^{(3)}\hat{\mathbf{\Delta}}_g'].\end{aligned}$$

In our simulations and data analyses, we used soft initializations by generating group memberships at random using a uniform distribution. From these initial soft group memberships $\hat{z}_{ig}$, we initialize the location matrices using

$$\hat{\mathbf{M}}_g = \frac{1}{N_g}\sum_{i=1}^{N} \hat{z}_{ig}\mathbf{X}_i,$$

where $N_g = \sum_{i=1}^{N} \hat{z}_{ig}$. Each skewness matrix is initialized as a matrix with all entries equal to 0.1—note that matrices with all entries equal to 0 cannot be used because the component densities would not be defined. The diagonal scale matrices, $\mathbf{\Sigma}_g$ and $\mathbf{\Psi}_g$ are initialized as follows

$$\hat{\mathbf{\Sigma}}_g = \frac{1}{pN_g}\,\text{diag}\left\{\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\right\},$$

and

$$\hat{\boldsymbol{\Psi}}_g = \frac{1}{nN_g} \operatorname{diag}\left\{\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'(\mathbf{X}_i - \hat{\mathbf{M}}_g)\right\}.$$

The factor loadings are initialized randomly from a uniform distribution on $[-1, 1]$.

6.2.1 Computational Issues

One situation that needs to be addressed for all four of these distributions, but particularly the variance-gamma distribution, is the infinite likelihood problem. This occurs as a result of the update for $\hat{\mathbf{M}}_g$ becoming very close, and in some cases equal to, an observation $\mathbf{X}_i$ when the algorithm gets close to convergence. A similar situation occurs in the multivariate case for the mixture of SAL distributions described in Franczak *et al.* (2014) and we follow a similar procedure when faced with this issue. While iterating the algorithm, when the likelihood becomes numerically infinite, we set the estimate of $\hat{\mathbf{M}}_g$ to the previous estimate which we will call $\hat{\mathbf{M}}_g^*$. We then update $\hat{\mathbf{A}}_g$ according to

$$\hat{\mathbf{A}}_g^* = \frac{\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g^*)}{\sum_{i=1}^{N} \hat{z}_{ig} a_{ig}}.$$

The updates for all other parameters remain the same. As mentioned in Franczak *et al.* (2014), this solution is a little naive; however, it does generally work quite well. It is not surprising that this problem is particularly prevalent in the case of the variance-gamma distribution because the SAL distribution arises as a special case of the variance-gamma distribution.

Another computational concern is in the evaluation of the Bessel functions. In the computation of the GIG expected values and the component densities, it may be the case that the argument is far larger than the magnitude of the index—especially

in higher dimensional cases. Therefore, in these situations, the result is computationally equivalent to zero which causes issues with other computations. In such a situation, we calculate the exponentiated version of the Bessel function, i.e., we calculate $\exp(u)K_\lambda(u)$ and subsequent calculations can be easily adjusted.

6.3 Simulation Study

A simulation study was performed for each of the four models presented herein. For each of the four models, we consider $d \times d$ matrices with $d \in \{10, 30\}$ and, for each value of d, we consider datasets coming from a mixture with two components and $\pi_1 = \pi_2 = 0.5$. The datasets have sample sizes $N \in \{100, 200, 400\}$ and the following parameters are used for all four models for each combination of d and N. We take $\mathbf{M}_1 = \mathbf{0}$ and $\mathbf{M}_2 = \mathbf{M}_1 + \mathbf{C}$, where $\mathbf{C}$ is a matrix with all entries equal to c for $c \in \{1, 2, 4\}$. All other parameters are held constant. We take $\mathbf{\Sigma}_1 = 2\mathbf{I}_d$, $\mathbf{\Sigma}_2 = \mathbf{I}_d$, $\mathbf{\Psi}_1 = \mathbf{I}_d$, $\mathbf{\Psi}_2 = 2\mathbf{I}_d$, and $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{1}$, where $\mathbf{1}$ is a matrix of 1's. Three column factors and two row factors are used with their values being randomly drawn from a uniform distribution on $[-1, 1]$. See Table 6.1 for distribution-specific parameters.

Table 6.1: Distribution-specific parameters used for the simulations, where the acronyms all take the form MMVDFA and denote "mixture of matrix variate D factor analyzers" with D being either skew-t (ST), generalized hyperbolic (GH), variance-gamma (VG), or NIG.

	Component 1	Component 2
MMVSTFA	$\nu_1 = 4$	$\nu_2 = 20$
MMVGHFA	$\omega_1 = 4, \lambda_1 = -4$	$\omega_2 = 10, \lambda_2 = 4$
MMVVGFA	$\gamma_1 = 4$	$\gamma_2 = 10$
MMVNIGFA	$\kappa_1 = 2$	$\kappa_2 = 4$

We fit the MMVSTFA model to data that is simulated from the MMVSTFA

model using the parameters above together with the distribution-specific parameters in Table 6.1. We take an analogous approach with the MMVGHFA, MMVVGFA, and MMVNIGFA models. However, we fit the MMVBFA model to data that is simulated from the MMVVGFA model—this is done to facilitate an illustration that uses data simulated from a mixture of skewed matrix variate distributions. We fit all models for $G \in \{1, 2, 3, 4\}$ and $q, r \in \{1, 2, 3, 4, 5\}$. In Tables 6.2 and 6.3, we show the number of times that the BIC correctly chooses the number of groups, row factors, and column factors. In Table 6.4, the average ARI and corresponding standard deviation for each setting is shown. As one would expect, for each model introduced herein, the classification performance generally improves as N increases. However, this is not the case for the MMVBFA model. In the case $d = 10$, it is interesting to note that the number of correct choices made by the BIC for the row and column factors generally decreases as we increase the separation (Table 6.2). However, when d is increased to 30, there is no clear trend in this regard (Table 6.3). The classification performance for the four models introduced here in is excellent overall (Table 6.4). However, when fitting the MMVBFA model to data simulated from the MMVVGFA model, the BIC never chooses the correct number of groups for $N \in \{200, 400\}$. Furthermore, although not apparent from the tables, the model generally overfits the number of groups which, as in the multivariate case, is to be expected when using a Gaussian mixture model in the presence of skewness or outliers.

Table 6.2: Number of datasets for which the BIC correctly chose the number of groups, row factors, and column factors ($d = 10$).

		MMVSTFA			MMVGHFA			MMVVGFA			MMVNIGFA			MMVBFA		
c	N	G	q	r	G	q	r	G	q	r	G	q	r	G	q	r
	100	18	15	19	25	16	24	18	10	12	21	21	20	17	15	19
1	200	23	18	21	25	25	25	21	11	8	24	24	24	0	19	23
	400	25	21	21	25	25	25	22	17	15	24	24	25	0	19	14
	100	18	14	17	25	9	22	16	7	4	17	18	19	16	17	15
2	200	24	18	19	25	22	22	23	10	2	23	23	24	0	20	20
	400	25	23	23	25	25	25	19	20	19	25	24	25	0	21	14
	100	8	13	14	23	5	10	17	11	0	24	2	7	19	23	9
4	200	22	9	16	25	4	16	21	8	8	24	7	24	0	18	18
	400	25	12	21	25	22	12	17	10	19	21	0	14	0	17	19

Table 6.3: Number of datasets for which the BIC correctly chose the number of groups, row factors, and column factors ($d = 30$).

		MMVSTFA			MMVGHFA			MMVVGFA			MMVNIGFA			MMVBFA		
c	N	G	q	r	G	q	r	G	q	r	G	q	r	G	q	r
	100	24	11	12	25	15	18	25	12	12	25	20	21	15	6	2
1	200	25	17	18	25	22	23	25	21	20	25	23	25	0	4	3
	400	25	22	23	25	25	24	25	25	20	25	25	25	0	10	1
	100	24	15	17	25	17	18	25	13	11	25	22	23	14	5	0
2	200	25	22	19	25	19	22	25	20	22	25	23	25	0	5	3
	400	25	19	20	25	22	24	25	23	24	25	24	25	0	9	8
	100	24	17	17	25	12	14	25	17	14	25	23	16	18	2	2
4	200	25	18	20	25	18	23	25	21	22	25	21	21	0	3	8
	400	25	15	15	25	20	24	25	19	20	25	21	22	0	5	7

Table 6.4: Average ARI values over 25 runs for each setting with standard deviations in parentheses.

		MMVSTFA		MMVGHFA		MMVVGFA		MMVNIG		MMVBFA	
c	N	$d = 10$	$d = 30$	$d = 10$	$d = 30$	$d = 10$	$d = 30$	$d = 10$	$d = 30$	$d = 10$	$d = 30$
1	100	0.91(0.08)	0.96(0.01)	0.97(0.03)	0.97(0.02)	0.90(0.1)	0.97(0.02)	0.98(0.05)	1.00(0.0)	0.90(0.1)	0.91(0.1)
	200	0.98 (0.03)	0.99(0.009)	1.00(0.006)	1.00(0.007)	0.97(0.03)	0.99(0.01)	1.00(0.007)	1.00(0.0)	0.75(0.05)	0.76(0.01)
	400	1.00 (0.005)	1.00(0.004)	1.00(0.0)	1.00(0.0)	0.99(0.03)	1.00(0.0)	1.00(0.006)	1.00(0.0)	0.69(0.1)	0.54(0.07)
2	100	0.94(0.03)	0.96(0.02)	0.96(0.03)	0.97(0.03)	0.88(0.1)	0.98(0.02)	0.96(0.07)	1.00(0.0)	0.91(0.1)	0.90(0.1)
	200	0.98 (0.02)	0.99(0.009)	1.00(0.0)	1.00(0.0)	0.97(0.05)	1.00(0.007)	0.99(0.02)	1.00(0.0)	0.76(0.01)	0.76(0.02)
	400	1.00 (0.005)	1.00(0.003)	1.00(0.0)	1.00(0.0)	0.98(0.05)	1.00(0.0)	1.00(0.0)	1.00(0.0)	0.66(0.1)	0.53(0.07)
4	100	0.84(0.08)	0.97(0.03)	0.94(0.04)	0.97(0.02)	0.92(0.1)	1.00(0.01)	0.98(0.08)	1.00(0.0)	0.94(0.1)	0.93(0.1)
	200	0.98 (0.02)	1.00(0.004)	1.00(0.0)	1.00(0.0)	0.97(0.05)	1.00(0.0)	0.99(0.02)	1.00(0.0)	0.76(0.02)	0.76(0.01)
	400	1.00 (0.004)	1.00(0.002)	1.00(0.0)	1.00(0.0)	0.98(0.03)	1.00(0.0)	1.00(0.03)	1.00(0.0)	0.73(0.08)	0.54(0.07)

6.4 MNIST Digits

Gallaugher and McNicholas (2018b,c) consider the MNIST digits dataset; specifically, looking at digits 1 and 7 because they are similar in appearance. Herein, we consider the digits 1, 6, and 7. This dataset consists of 60,000 (training) images of Arabic numerals 0 to 9. We consider different levels of supervision and perform either clustering or semi-supervised classification. Specifically we look at 0% (clustering), 25%, and 50% supervision. For each level of supervision, 25 datasets consisting of 200 images each of digits 1, 6, and 7 are taken. As discussed in Gallaugher and McNicholas (2018b), because of the lack of variability in the outlying rows and columns of the data matrices, random noise is added to ensure non-singularity of the scale matrices. Each of the four models developed herein, as well as the MMVBFA model, are fitted for 1 to 17 row and column factors. In Table 6.5, the average ARI and misclassification rate (MCR) values are presented for each model and each level of supervision.

Table 6.5: Average ARI and MCR values for the MNIST dataset for each level of supervision, with respective standard deviations in parentheses for digits 1,6, and 7.

Supervision		MMVSTFA	MMVGHFA	MMVVGFA	MMVNIGFA	MMVBFA
0% (clustering)	ARI	0.58(0.09)	0.58(0.09)	0.62(0.1)	0.47(0.1)	0.36(0.09)
	MCR	0.17(0.04)	0.17(0.08)	0.15(0.04)	0.22(0.05)	0.28(0.09)
25%	ARI	0.72(0.1)	0.72(0.1)	0.75(0.1)	0.64(0.2)	0.51(0.16)
	MCR	0.10(0.04)	0.10(0.04)	0.094(0.04)	0.14(0.07)	0.20(0.07)
50%	ARI	0.83(0.07)	0.85(0.03)	0.83(0.07)	0.81(0.1)	0.72(0.06)
	MCR	0.059(0.03)	0.052(0.02)	0.061(0.03)	0.067(0.05)	0.10(0.06)

In the completely unsupervised case, three of the skewed models have a MCR of around 16%. However, at 25% supervision, this decreases to around 10% and, at 50% supervision, this falls again to around 5%. At all three levels of supervision, it is clear that all four skewed mixture models introduced herein outperform the MMVBFA

model. In fact, the performance of the MMVBFA model at 25% supervision is not as good as that of the MMVVGFA, MMVGHFA or MMVSTFA models in the completely unsupervised case (i.e., 0% supervision).

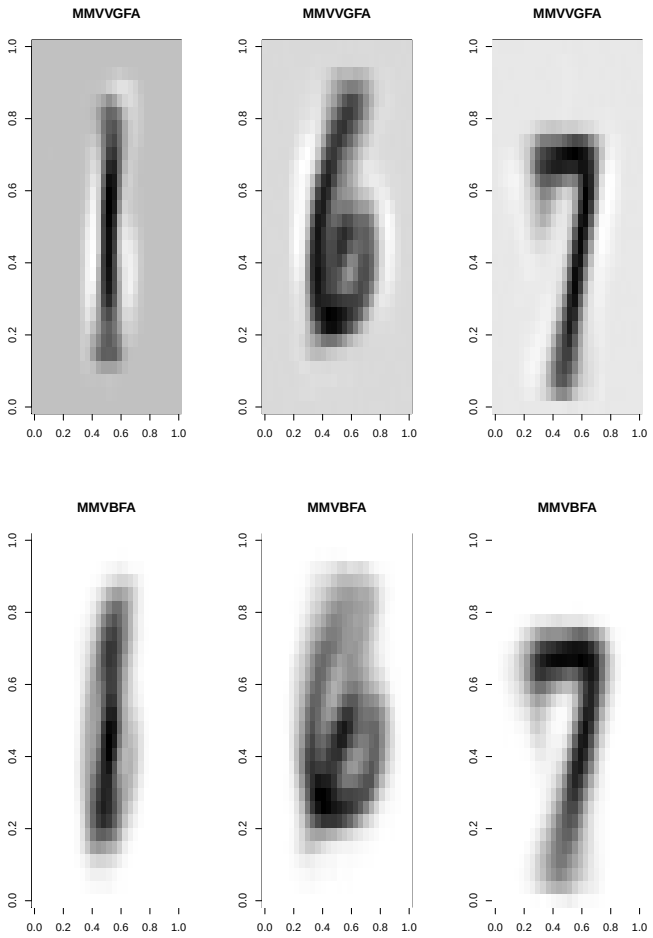


Figure 6.1: Heat maps of estimated location matrices for the MMVBFA and MMVVGFA models for each class in the unsupervised case.

It is of interest to compare heat maps of the estimated location matrices for the MMVBFA and MMVVGFA models for one of the datasets in the unsupervised case (Figure 6.1). It can be seen that the images are a lot clearer for the MMVVGFA model

compared to the MMVBFA model. This is particularly prominent when considering the the results for the digit 6, for which one can see a possible 1 or 7 in the background for the MMVBFA heat map. Moreover, for digit 1, one can see a faint 6 in the background when looking at the MMVBFA heat map.

6.5 Summary

The MMVBFA model has been extended to four skewed distributions; specifically, the matrix variate skew-t, generalized hyperbolic, variance-gamma, and NIG distributions. AECM algorithms were developed for parameter estimation, and the novel approaches were illustrated on real and simulated data. In the simulations, the models introduced herein generally exhibited very good performance under various scenarios. As expected, the MMVBFA model did not perform well when applied to data from the MMVVGFA model. In the real data example, all four of the skewed matrix variate models introduced herein performed better than the MMVBFA model. As one would expect, the difference in performance was most stark in the clustering case.

Software to implement the approaches introduced herein, written in the Julia language (Bezanson *et al.*, 2017; McNicholas and Tait, 2019), is available in the `MatrixVariate.jl` repository (Počuča *et al.*, 2019).

Chapter 7

Clustering and Semi-Supervised Classification for Clickstream Data via Mixture Models

7.1 Background

Clickstream data present an important means of investigating users' internet behaviour. Unsupervised classification, a.k.a. clustering or cluster analysis, or semi-supervised classification of such data can be very useful in many different areas of endeavour. Examples can be found in areas as diverse as online marketing and anti-terrorism. Early examples of clustering clickstream data can be found in the work of Banerjee and Ghosh (2000, 2001), which looked at concept based clustering, and longest common sequences respectively. Other examples clustering and classification of clickstreams can be found in Montgomery *et al.* (2004), Aggarwal *et al.* (2003) and Wei *et al.* (2012); notably, none of these approaches draw on mixture models.

The first use of mixture models for clustering for clickstreams can be found in Cadez *et al.* (2003), who considered a mixture of first-order Markov models. One problem, as mentioned in Cadez *et al.* (2003), is that the number of parameters can become very high when the number of website categories is very large. To alleviate this potential problem, Melnykov (2016c) looked at bi-clustering of the clickstreams and the states to effectively reduce the number of states. Although this was shown to be successful in simulations, in the real data analyses, only two states were grouped together.

Herein, a mixture of first-order continuous time Markov models is introduced for unsupervised and semi-supervised classification of clickstream data. Specifically, the type of clickstream data considered herein would come from a website with multiple categories, such as `amazon.com`, or a news website with categories such as weather, breaking news, sports, etc., with the clickstreams recording the movement of a user from one category to another, as well as the amount of time spent in each category. In practice, the incorporation of continuous time may be desirable in detecting the true underlying group structure for internet users. Consider the example of monitoring potentially inappropriate or criminal behaviour: our approach allows for the fact that an internet user might accidentally click on a link to, or be redirected to, inappropriate content and then immediately exit the site. When considering a discrete time Markov chain, the user would have been recorded as entering the inappropriate site, and could possibly be flagged as a problematic user. However, if the amount of time spent on the website can be taken into consideration, this may not be classified as suspicious activity. In the case of online shopping, a user could again click on the wrong category but immediately switch. Again, using a discrete time model would not be able to take

this into account, and could lead to incorrect product suggestions.

The basis of the proposed methodology rests on the work done in Albert (1960), who considered the estimation of the infinitesimal generator in a continuous time Markov model for a single component. Employing this in the mixture-model context using the EM algorithm is where the novelty lies.

7.2 Mixtures of First-Order Markov Models

Cadez *et al.* (2003) and Melnykov (2016c) consider a mixture of first-order Markov models to cluster clickstreams. Consider a website consisting of many different webpages that can be accessed from one of J categories. The clickstream of interest is given by the transitions from one category to another. Suppose N clickstreams are observed from a population with G types. Now, assume that N_1 of these clickstreams have unknown labels and denote these clickstreams by $\mathbf{x}_i^{(1)} = (x_{i1}^{(1)}, x_{i2}^{(1)}, \ldots, x_{iL_i}^{(1)})', i \in \{1, 2, \ldots, N_1\}$, and N_2 of these are labelled denoted similarly by $\mathbf{x}_i^{(2)} = (x_{i1}^{(2)}, x_{i2}^{(2)}, \ldots, x_{iL_i}^{(2)})', i \in \{1, 2, \ldots, N_2\}$, where $N_2 = N - N_1$ and L_i is the length of clickstream i. For notational purposes, note that $\mathbf{x}_i^{(1)}$ is an L_i-dimensional vector of the states for the unlabelled clickstream i, and that each element can take values in the state space, which corresponds to the number of categories. For example, if there are 7 categories on a website, each element of $\mathbf{x}_i^{(1)}$ can take values in $\{1, 2, \ldots, 7\}$. The same applies to the labelled observations $\mathbf{x}_i^{(2)}$.

The one-step transition matrix for group g is

$$\boldsymbol{\Lambda}_g = \begin{pmatrix} \lambda_{g11} & \lambda_{g12} & \cdots & \lambda_{g1J} \\ \lambda_{g21} & \lambda_{g22} & \cdots & \lambda_{g2J} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{gJ1} & \lambda_{gJ2} & \cdots & \lambda_{gJJ} \end{pmatrix}.$$

Now, define the initial probabilities $\alpha_{gx_{i1}} = P(X_{i1} = x_{i1}|\mathbf{x}_i \text{ is in group } g)$, for $i = 1, 2, \ldots, N$. For ease of notation, and recalling that $x_{i1} \in \{1, 2, \ldots, J\}$, denote an initial probability vector for each group g by $\boldsymbol{\alpha}_g = (\alpha_{g1}, \alpha_{g2}, \ldots, \alpha_{gJ})$. Finally, for ease of notation, denote the total number of transitions from state j to state k for unlabelled clickstream i by $n_{ijk}^{(1)}$ and likewise for labelled clickstream i by $n_{ijk}^{(2)}$.

The observed likelihood is given by

$$\begin{aligned} \mathcal{L}_{\text{obs}}(\boldsymbol{\vartheta}|\mathcal{D}_{\text{o}}) = & \underbrace{\prod_{i=1}^{N_1}\sum_{g=1}^{G}\left\{\pi_g\left[\prod_{j=1}^{J}\alpha_{gj}^{I(x_{i1}=j)}\right]\left[\prod_{j=1}^{J}\prod_{k=1}^{J}\lambda_{gjk}^{n_{ijk}}\right]\right\}}_{\text{Unlabelled Observations}} \\ & \times \underbrace{\prod_{i=1}^{N_2}\prod_{g=1}^{G}\left\{\pi_g\left[\prod_{j=1}^{J}\alpha_{gj}^{I(x_{i1}=j)}\right]\left[\prod_{j=1}^{J}\prod_{k=1}^{J}\lambda_{gjk}^{n_{ijk}}\right]\right\}^{z_{ig}^{(2)}}}_{\text{Labelled Observations}}, \end{aligned} \tag{7.1}$$

where $\mathcal{D}_{\text{o}}$ is the observed data and

$$z_{ig}^{(2)} = \begin{cases} 1 & \text{if labelled observation } i \text{ is in group } g, \\ 0 & \text{otherwise.} \end{cases}$$

The expectation-maximization (EM) algorithm (Dempster *et al.*, 1977) is used

for parameter estimation. The EM algorithm works with the complete-data log-likelihood, i.e., the log-likelihood of the observed data together with the missing data. On the E-step, the expected value of the complete-data log-likelihood is computed and, on the M-step, it is maximized conditional on the current parameter estimates. The E- and M-steps are iterated until some stopping criterion is satisfied. Defining the latent $z_{ig}^{(1)}$ to be the group indicators for the unlabelled observations, analogous to $z_{ig}^{(2)}$ in (7.2), the complete-data likelihood in this case can be written

$$\mathcal{L}_{\text{c}} = \prod_{m=1}^{2}\prod_{i=1}^{N_m}\prod_{g=1}^{G}\left\{\left[\prod_{j=1}^{J}\alpha_{gj}^{I(x_{i1}=j)}\right]\left[\prod_{j=1}^{J}\prod_{k=1}^{J}\lambda_{gjk}^{n_{ijk}}\right]\right\}^{z_{ig}^{(m)}}.$$

In this particular case, where the only latent variables in the EM are the $z_{ig}^{(1)}$ values, the EM algorithm can be outlined as follows.

Initialization: Initialize the parameters π_g, $\boldsymbol{\alpha}_g$, and $\boldsymbol{\Lambda}_g$ for all $g = 1, \ldots, G$.

E Step: Update each $\hat{z}_{ig}^{(1)}$ by calculating

$$\hat{z}_{ig}^{(1)} = \frac{\hat{\pi}_g\left[\prod_{j=1}^{J}\hat{\alpha}_{gj}^{I(x_{i1}^{(1)}=j)}\right]\left[\prod_{j=1}^{J}\prod_{k=1}^{J}\hat{\lambda}_{gjk}^{n_{ijk}^{(1)}}\right]}{\sum_{g=1}^{G}\hat{\pi}_g\left[\prod_{j=1}^{J}\hat{\alpha}_{gj}^{I(x_{i1}^{(1)}=j)}\right]\left[\prod_{j=1}^{J}\prod_{k=1}^{J}\hat{\lambda}_{gjk}^{n_{ijk}^{(1)}}\right]}.$$

M Step: Update the parameter estimates via:

$$\hat{\pi}_g = \frac{\sum_{m=1}^{2}\sum_{i=1}^{N_m}\hat{z}_{ig}^{(m)}}{N}, \tag{7.2a}$$

$$\hat{\alpha}_{gj} = \frac{\sum_{m=1}^{2}\sum_{i=1}^{N_m}\hat{z}_{ig}I(x_{i1}^{(m)}=j)}{\sum_{m=1}^{2}\sum_{i=1}^{N_m}\hat{z}_{ig}^{(m)}}, \tag{7.2b}$$

$$\hat{\lambda}_{gjk} = \frac{\sum_{m=1}^{2}\sum_{i=1}^{N_m}\hat{z}_{ig}^{(m)}n_{ijk}^{(m)}}{\sum_{m=1}^{2}\sum_{i=1}^{N_m}\sum_{k'=1}^{J}\hat{z}_{ig}^{(m)}n_{ijk'}^{(m)}}. \tag{7.2c}$$

Note that unsupervised classification (clustering) falls out as the special case when $N_1 = N$.

7.3 Mixture of First-Order Continuous Time Markov Models

We now discuss an extension of the methodology presented in Cadez *et al.* (2003) and Melnykov (2016c) to take into account the amount of time spent in each category. Consider the same scenario as before, except this time we also observe a sequence of times spent in each state before transferring to another state; denote this by $\mathbf{t}_i^{(m)} = (t_{i1}^{(m)}, t_{i2}^{(m)}, \ldots, t_{iL_i}^{(m)})$ for $i = 1, \ldots, N$, where $m = 1$ corresponds to the unlabelled observations and $m = 2$ corresponds to the labelled observations. It is important to note that, unlike the discrete time case, no transitions are made to the same state in continuous time. These data can be modelled using a mixture of continuous time Markov chains, with infinitesimal generators

$$\mathbf{Q}_g = \begin{pmatrix} q_{g11} & q_{g12} & \cdots & q_{g1J} \\ q_{g21} & q_{g22} & \cdots & q_{g2J} \\ \vdots & \vdots & \ddots & \vdots \\ q_{gJ1} & q_{gJ2} & \cdots & q_{gJJ} \end{pmatrix},$$

where $q_{gjk} \geq 0$ for $j \neq k$ and $q_{gjj} = -\sum_{k \neq j} q_{gjk}$ for $g \in \{1, 2, \ldots, G\}$. The first item we note here is that the underlying transition probabilities are given by

$$P(X_{i(l+1)}^{(m)} = x_{i(l+1)}^{(m)} | X_{il}^{(m)} = x_{il}^{(m)}, z_{ig}^{(m)} = 1) = -\frac{q_{gx_{il}^{(m)} x_{i(l+1)}^{(m)}}}{q_{gx_{il}^{(m)} x_{il}^{(m)}}}.$$

The second is that the $T_{il}^{(m)}$ are independent and

$$T_{il}^{(m)} | (X_{il}^{(m)} = x_{il}^{(m)}, z_{ig}^{(m)} = 1) \sim \text{Exp}(-q_{gx_{il}^{(m)} x_{il}^{(m)}}),$$

where $\text{Exp}(a)$ denotes an exponential distribution with rate a. We denote the initial probability vector by $\boldsymbol{\alpha}_g$, as before.

Albert (1960) presents a detailed background for the theory of continuous time Markov chains, and also discusses the likelihood function for a sample of continuous time Markov chains for one component. Modifying this likelihood function for use in the mixture model context with multiple components, we obtain the likelihood

$$\begin{aligned}
\mathcal{L}_{\text{obs}}(\boldsymbol{\vartheta} | \mathcal{D}_{\text{o}}) = \prod_{i=1}^{N_1} \sum_{g=1}^{G} \Bigg\{ \pi_g \left[\prod_{j=1}^{J} \alpha_{gj}^{I(x_{i1}^{(1)} = j)} \right] \left[\prod_{j=1}^{J} \prod_{k \neq j}^{J} q_{gjk}^{n_{ijk}^{(1)}} \right] \left[- \prod_{j=1}^{J} q_{gjj}^{I(x_{iL_i}^{(1)} = j)} \right] \\
\times \exp \left[\sum_{j=1}^{J} \sum_{l=1}^{L} q_{gjj} t_{il}^{(1)} I(x_{il}^{(1)} = j) \right] \Bigg\} \\
\times \prod_{i=1}^{N_2} \prod_{g=1}^{G} \Bigg\{ \pi_g \left[\prod_{j=1}^{J} \alpha_{gj}^{I(x_{i1}^{(2)} = j)} \right] \left[\prod_{j=1}^{J} \prod_{k \neq j}^{J} q_{gjk}^{n_{ijk}^{(2)}} \right] \left[- \prod_{j=1}^{J} q_{gjj}^{I(x_{iL_i}^{(2)} = j)} \right] \\
\times \exp \left[\sum_{j=1}^{J} \sum_{l=1}^{L} q_{gjj} t_{il}^{(2)} I(x_{il}^{(2)} = j) \right] \Bigg\}^{z_{ig}^{(2)}}
\end{aligned}$$

and the complete-data log-likelihood is

$$\ell_{\text{c}} = \sum_{m=1}^{2}\sum_{i=1}^{N_m}\sum_{g=1}^{G} z_{ig}^{(m)} \Bigg[\log \pi_g + \sum_{j=1}^{J} I(x_{i1}^{(m)} = j) \log \alpha_{gj} + \sum_{j=1}^{J}\sum_{k\neq j}^{J} n_{ijk}^{(m)} \log q_{gjk} + \sum_{j=1}^{J} I(x_{iL_i}^{(m)} = j) \log(-q_{gjj}) + \sum_{j=1}^{J}\sum_{l=1}^{L_i} q_{gjj} t_{il}^{(m)} I(x_{il}^{(m)}) \Bigg]. \quad (7.3)$$

In the E step, we update the latent indicator variables $z_{ig}^{(1)}$, and these are the only latent variables for this EM algorithm. At iteration $s+1$, this update is given by

$$\hat{z}_{ig}^{(1)} = \frac{h(\hat{\pi}_g, \hat{\boldsymbol{\alpha}}_g, \hat{\mathbf{Q}}_g, \mathbf{x}_i^{(1)}, \mathbf{t}_i^{(1)})}{\sum_{g'=1}^{G} h(\hat{\pi}_{g'}, \hat{\boldsymbol{\alpha}}_{g'}, \hat{\mathbf{Q}}_{g'}, \mathbf{x}_i^{(1)}, \mathbf{t}_i^{(1)})}, \quad (7.4)$$

where

$$h(\hat{\pi}_g, \hat{\boldsymbol{\alpha}}_g, \hat{\mathbf{Q}}_g, \mathbf{x}_i^{(1)}, \mathbf{t}_i^{(1)}) = \hat{\pi}_g \left[\prod_{j=1}^{J} (\hat{\alpha}_{gj})^{I(x_{i1}^{(1)}=j)}\right] \left[\prod_{j=1}^{J} (-\hat{q}_{gjj})^{I(x_{iL}^{(1)}=j)}\right] \left[\prod_{j=1}^{J}\prod_{k\neq j}^{J} (\hat{q}_{gjk})^{n_{ijk}^{(1)}}\right] \times \exp\left\{\sum_{j=1}^{J}\sum_{l=1}^{L_i} \hat{q}_{gjj} t_{il}^{(1)} I(x_{il}^{(1)} = j)\right\}.$$

In the M step, we update our parameters, $\hat{\pi}_g$, $\hat{\boldsymbol{\alpha}}_g$ and $\hat{\mathbf{Q}}_g$. The updates for the $\hat{\pi}_g$ and $\hat{\boldsymbol{\alpha}}_g$ are the same as those in the discrete case, see (7.2a) and (7.2b). We also update each $\hat{\mathbf{Q}}_g$, and these updates are given by

$$\hat{q}_{gjk} = \begin{cases} \frac{\sum_{i=1}^{N} z_{ig}^{(m)} n_{ijk}}{\hat{\lambda}_{gj}} & \text{if } j \neq k, \\ -\sum_{k\neq j} \hat{q}_{gjk} & \text{if } k = j, \end{cases}$$

where $\hat{\lambda}_{gj} = a/b$ with

$$a = \sum_{m=1}^{2}\sum_{i=1}^{N_m}\left[\sum_{l=1}^{L_i}\hat{z}_{ig}^{(m)}t_{il}^{(m)}I(x_{il}^{(m)}=j) + \sum_{k\neq j}^{J}\hat{z}_{ig}^{(m)}n_{ijk}^{(m)}\right],$$
$$b = \sum_{m=1}^{2}\sum_{i=1}^{N_m}\hat{z}_{ig}^{(m)}I(x_{iL_i}^{(m)}=j) + \sum_{m=1}^{2}\sum_{i=1}^{N_m}\sum_{k\neq j}^{J}\hat{z}_{ig}^{(m)}n_{ijk}^{(m)}.$$

Notably, the number of free parameters in this continuous time model are the same as those in the discrete time model.

7.3.1 Computational Issues

We note that calculating the updates for $\hat{z}_{ig}^{(1)}$ using (7.4) can lead to computational problems. This is due to the calculation of $h(\cdot)$ being computationally equal to 0 for all groups g, which occurs when $\log h(\cdot)$ becomes large and negative, followed by exponentiating this large negative value to get the value of $h(\cdot)$. Taking this into account, we can rewrite this update as

$$\frac{1}{\hat{z}_{ig}^{(1)}} = \sum_{g'=1}^{G}\exp\left\{\log\left(\frac{\hat{\pi}_{g'}}{\hat{\pi}_g}\right) + \sum_{j=1}^{J}I(x_{i1}=j)\log\left(\frac{\hat{\alpha}_{g'j}}{\hat{\alpha}_{gj}}\right) + \sum_{j=1}^{J}\sum_{k=1}^{J}n_{ijk}\log\left(\frac{\hat{q}_{g'jk}}{\hat{q}_{gjk}}\right)\right.$$
$$\left. + \sum_{j=1}^{J}I(x_{iL_i})\log\left(\frac{\hat{q}_{g'jj}}{\hat{q}_{gjj}}\right) + \sum_{j=1}^{J}\sum_{l=1}^{L_i}I(x_{il}=j)t_{il}(\hat{q}_{g'jj}-\hat{q}_{gjj})\right\},$$

which allows the $\hat{z}_{ig}^{(1)}$ to be computed directly instead of having to calculate the $h(\cdot)$ functions for each group separately.

A second computational issue, as discussed in Melnykov (2016c), is the case where there are no transitions present in the data between two states, i.e., $n_{ijk} = 0$ for all i. In this case, the estimates for q_{gjk} would be zero for all g. Firstly, this is a problem

because we can make the reasonable assumption that all states communicate with each other, making an estimate of zero unrealistic. Secondly, this would cause problems with the calculation of the likelihood. We, therefore, set a lower bound of 10^{-6} for all parameter values.

7.4 Analyses

7.4.1 Simulation 7.1

In the first simulation, we simulated from two groups with infinitesimal generators

$$\mathbf{Q}_1 = \begin{pmatrix} -0.100 & 0.050 & 0.020 & 0.020 & 0.010 \\ 0.100 & -1.000 & 0.200 & 0.100 & 0.600 \\ 0.020 & 0.050 & -0.100 & 0.005 & 0.025 \\ 0.050 & 0.050 & 0.050 & -1.000 & 0.850 \\ 0.006 & 0.004 & 0.050 & 0.040 & -0.100 \end{pmatrix},$$

$$\mathbf{Q}_2 = \begin{pmatrix} -0.100 & 0.001 & 0.009 & 0.015 & 0.075 \\ 0.700 & -1.000 & 0.200 & 0.050 & 0.050 \\ 0.010 & 0.005 & -0.100 & 0.030 & 0.055 \\ 0.400 & 0.400 & 0.100 & -1.000 & 0.100 \\ 0.030 & 0.030 & 0.020 & 0.020 & -0.100 \end{pmatrix}.$$

We took sample sizes of $N \in \{50, 100, 200, 400\}$ with clickstream lengths L ranging from 4 to 25 and 25 to 100. We also considered two cases with equal proportions, $\pi_1 = \pi_2 = 0.5$ (Simulation 7.1A) and $\pi_1 = 0.2, \pi_2 = 0.8$ (Simulation 7.1B). The

purpose of the second case is that, when looking for suspicious behaviour, it is highly likely that there are fewer suspicious users than regular users.

In these simulations, we have large separation in the underlying transition probabilities; however, because the diagonal elements are identical between groups, there is no separation in the average amount of time spent in each category. The results for Simulations 7.1A and 7.1B for both the discrete and continuous time models are summarized in Tables 7.1 and 7.2. After fitting the model for $G = 1, 2, \ldots, 5$, we consider the number of times each G was chosen using the BIC as well as the average ARI and the associated standard deviation.

In this case, we see that the results for the continuous and discrete models are almost identical, with only slight variations in the ARI between the two methods. The BIC in all cases with low values of L, correctly finds two groups for both the continuous and discrete models. When L is increased, there is a very slight chance of overfitting the number of groups. It is interesting to note that a higher sample size does not affect the classification performance as much as a longer length of the clickstream. Finally, very little difference is seen when changing the mixing proportions. The similar results between the discrete and continuous time models illustrate the ability of the continuous time model to effectively detect group structure based solely on differences in transition probabilities.

7.4.2 Simulation 7.2

In this simulation, we once again look at clustering. This time, data are simulated from three different groups, with mixing proportions $\pi_1 = \pi_2 = \pi_3 = 1/3$ (Simulation 7.2A) and $\pi_1 = 0.2$, $\pi_2 = 0.4, \pi_3 = 0.4$ (Simulation 7.2B). There are seven states, $\boldsymbol{\alpha}_1$

is taken to be uniform, $\boldsymbol{\alpha}_2$ gives probability 0.1 for all states except state 7, which has probability of 0.4, and $\boldsymbol{\alpha}_3$ gives probability 0.1 to all states except state 3, which has probability 0.4. The infinitesimal generators are taken to be

$$\mathbf{Q}_1 = \begin{pmatrix} -0.14 & 0.05 & 0.02 & 0.02 & 0.01 & 0.02 & 0.02 \\ 0.10 & -1.40 & 0.20 & 0.10 & 0.60 & 0.20 & 0.20 \\ 0.02 & 0.05 & -0.14 & 0.01 & 0.03 & 0.02 & 0.02 \\ 0.05 & 0.05 & 0.05 & -1.40 & 0.80 & 0.25 & 0.20 \\ 0.01 & 0.00 & 0.05 & 0.04 & -0.14 & 0.04 & 0.01 \\ 0.70 & 0.10 & 0.10 & 0.10 & 0.10 & -1.40 & 0.30 \\ 0.50 & 0.50 & 0.05 & 0.05 & 0.10 & 0.20 & -1.40 \end{pmatrix},$$

$$\mathbf{Q}_2 = \begin{pmatrix} -1.40 & 0.40 & 0.30 & 0.15 & 0.15 & 0.25 & 0.15 \\ 0.02 & -0.14 & 0.03 & 0.02 & 0.03 & 0.03 & 0.01 \\ 0.30 & 0.50 & -1.40 & 0.10 & 0.10 & 0.20 & 0.20 \\ 0.01 & 0.01 & 0.01 & -0.14 & 0.05 & 0.03 & 0.03 \\ 0.01 & 0.01 & 0.04 & 0.05 & -0.14 & 0.02 & 0.02 \\ 0.70 & 0.05 & 0.15 & 0.05 & 0.15 & -1.40 & 0.30 \\ 0.05 & 0.05 & 0.01 & 0.01 & 0.01 & 0.01 & -0.14 \end{pmatrix},$$

$$\mathbf{Q}_3 = \begin{pmatrix} -1.40 & 0.20 & 0.70 & 0.20 & 0.10 & 0.10 & 0.10 \\ 0.60 & -1.40 & 0.20 & 0.20 & 0.20 & 0.10 & 0.10 \\ 0.10 & 0.10 & -1.40 & 0.80 & 0.10 & 0.10 & 0.20 \\ 0.05 & 0.03 & 0.03 & -0.14 & 0.01 & 0.01 & 0.01 \\ 0.05 & 0.05 & 0.01 & 0.01 & -0.14 & 0.01 & 0.02 \\ 1.00 & 0.02 & 0.03 & 0.02 & 0.03 & -1.40 & 0.30 \\ 0.20 & 0.20 & 0.20 & 0.20 & 0.20 & 0.40 & -1.40 \end{pmatrix}.$$

In this case, there are two groups with similar underlying transition probabilities, but different amounts of time on average being spent in each state. The third group has a large amount of separation in the underlying transition probabilities in comparison to the first two. Also, the third group is defined by more time spent in states 4 and 5 on average than the rest of the states. From the results (Tables 7.3 and 7.4), we see that the continuous time model outperforms the discrete time model in all cases. Specifically, for short clickstream lengths, the BIC under-fits the true number of groups in all cases for the discrete model. Increasing N for shorter clickstreams helps the discrete model a little, i.e., for small N, the discrete time model finds only one group but, for larger N, two groups are selected, which is closer to the true number of groups. Increasing the length of the clickstream also helps with selecting the correct number of groups for the discrete time model, but still requires a sample size of 600 to choose the correct number of groups in all cases. The continuous time model performs well for all values of L and N—again, increasing the sample size is not as impactful as increasing the clickstream length. It is not surprising that the continuous time model outperforms the discrete time model in this case because there is very little separation in the underlying transition probabilities for two of the groups

but the time spent in each state is fairly well separated between the three groups. Accordingly, the discrete model, being unable to take into account the amount of time in each state, is unable to distinguish between groups 1 and 2.

Table 7.1: Summary of the results from Simulation 7.1A ($\pi_1 = \pi_2 = 0.5$).

L from 4 to 25							
Sample Size	Model	G=1	G=2	G=3	G=4	G=5	$\overline{\text{ARI}}$ (sd)
N=50	Continuous	0	100	0	0	0	0.935 (0.069)
	Discrete	0	100	0	0	0	0.942 (0.06)
N=100	Continuous	0	100	0	0	0	0.955(0.045)
	Discrete	0	100	0	0	0	0.955(0.043)
N=200	Continuous	0	100	0	0	0	0.958(0.029)
	Discrete	0	100	0	0	0	0.958(0.029)
N=400	Continuous	0	100	0	0	0	0.950(0.026)
	Discrete	0	100	0	0	0	0.950(0.025)
L from 25 to 100							
Sample Size	Model	G=1	G=2	G=3	G=4	G=5	$\overline{\text{ARI}}$ (sd)
N=50	Continuous	0	98	2	0	0	0.995 (0.032)
	Discrete	0	98	2	0	0	0.995 (0.035)
N=100	Continuous	0	99	1	0	0	0.999(0.012)
	Discrete	0	96	4	0	0	0.994(0.036)
N=200	Continuous	0	98	2	0	0	0.999(0.008)
	Discrete	0	98	2	0	0	0.997(0.022)
N=400	Continuous	0	99	1	0	0	0.999(0.004)
	Discrete	0	95	5	0	0	0.998(0.012)

Table 7.2: Summary of the results from Simulation 7.1B ($\pi_1 = 0.2, \pi_2 = 0.8$).

L from 4 to 25							
Sample Size	Model	G=1	G=2	G=3	G=4	G=5	$\overline{\text{ARI}}$ (sd)
N=50	Continuous	0	100	0	0	0	0.940 (0.070)
	Discrete	0	100	0	0	0	0.948 (0.067)
N=100	Continuous	0	100	0	0	0	0.955(0.042)
	Discrete	0	100	0	0	0	0.957(0.041)
N=200	Continuous	0	100	0	0	0	0.960(0.032)
	Discrete	0	100	0	0	0	0.957(0.032)
N=400	Continuous	0	100	0	0	0	0.957(0.024)
	Discrete	0	100	0	0	0	0.958(0.025)
L from 25 to 100							
Sample Size	Model	G=1	G=2	G=3	G=4	G=5	$\overline{\text{ARI}}$ (sd)
N=50	Continuous	0	91	9	0	0	0.960 (0.14)
	Discrete	0	91	9	0	0	0.960 (0.14)
N=100	Continuous	0	99	1	0	0	0.999(0.004)
	Discrete	0	89	11	0	0	0.965(0.13)
N=200	Continuous	0	96	4	0	0	0.992(0.056)
	Discrete	0	91	9	0	0	0.970(0.11)
N=400	Continuous	0	96	4	0	0	0.996(0.026)
	Discrete	0	91	9	0	0	0.987(0.067)

Table 7.3: Summary of results for Simulation 7.2A ($\pi_1 = \pi_2 = \pi_3 = 1/3$).

L from 4 to 25							
Sample Size	Model	G=1	G=2	G=3	G=4	G=5	$\overline{\text{ARI}}$ (sd)
N=75	Continuous	0	0	100	0	0	0.934 (0.045)
	Discrete	34	66	0	0	0	0.302 (0.22)
N=150	Continuous	0	0	100	0	0	0.952(0.026)
	Discrete	0	100	0	0	0	0.467(0.043)
N=300	Continuous	0	0	100	0	0	0.958(0.021)
	Discrete	0	100	0	0	0	0.477(0.032)
N=600	Continuous	0	0	100	0	0	0.956(0.014)
	Discrete	0	100	0	0	0	0.476(0.021)
L from 25 to 100							
Sample Size	Model	G=1	G=2	G=3	G=4	G=5	$\overline{\text{ARI}}$ (sd)
N=75	Continuous	0	0	95	5	0	0.994 (0.026)
	Discrete	0	100	0	0	0	0.564 (0.004)
N=150	Continuous	0	0	94	6	0	0.994(0.023)
	Discrete	0	89	11	0	0	0.603(0.10)
N=300	Continuous	0	0	98	2	0	0.992(0.056)
	Discrete	0	2	98	0	0	0.872(0.051)
N=600	Continuous	0	0	97	3	0	0.999(0.006)
	Discrete	0	0	100	0	0	0.882(0.020)

Table 7.4: Summary of results for Simulation 7.2B ($\pi_1 = 0.2, \pi_2 = 0.4, \pi_3 = 0.4$).

L from 4 to 25							
Sample Size	Model	G=1	G=2	G=3	G=4	G=5	$\overline{\text{ARI}}$ (sd)
N=75	Continuous	0	0	100	0	0	0.935 (0.045)
	Discrete	30	70	0	0	0	0.382 (0.26)
N=150	Continuous	0	0	100	0	0	0.948(0.030)
	Discrete	0	100	0	0	0	0.559(0.050)
N=300	Continuous	0	0	100	0	0	0.953(0.023)
	Discrete	0	100	0	0	0	0.566(0.034)
N=600	Continuous	0	0	100	0	0	0.953(0.015)
	Discrete	0	100	0	0	0	0.569(0.025)
L from 25 to 100							
Sample Size	Model	G=1	G=2	G=3	G=4	G=5	$\overline{\text{ARI}}$ (sd)
N=75	Continuous	0	0	98	2	0	0.999 (0.011)
	Discrete	0	100	0	0	0	0.677 (0.007)
N=150	Continuous	0	0	96	4	0	0.996(0.021)
	Discrete	0	99	1	0	0	0.682(0.031)
N=300	Continuous	0	0	96	4	0	0.997(0.018)
	Discrete	0	29	71	0	0	0.842(0.10)
N=600	Continuous	0	0	97	3	0	0.999(0.004)
	Discrete	0	0	100	0	0	0.915(0.018)

7.4.3 Simulation 7.3

In this simulation, semi-supervised classification was considered by simulating under the same circumstances as Simulation 7.1A (Simulation 7.3A) and Simulation 7.2A (Simulation 7.3B). Supervision levels of 20, 40 and 80% were considered, and the average ARI values for observations considered unlabelled with standard deviations are shown in Tables 7.5 and 7.6.

For Simulation 7.3A, we again see that the results are very similar between the discrete and continuous time models. Increasing the level of supervision, unsurprisingly improves the classification performance for short clickstream lengths. When increasing the lengths of the clickstreams, perfect classification performance is achieved. For Simulation 7.3B, again the clickstream model outperforms the discrete model. Moreover, increasing the clickstream length again improves the performance for both models, giving perfect classification for the continuous time model for all levels of supervision, and a much improved ARI for the discrete time model.

7.4.4 Modified MSNBC Data

There is a dearth of publicly available clickstream data that records the amount of time spent in each state. Here, the MSNBC dataset that was analyzed in Melnykov (2016c) is used for illustration, with simulated times added in each state. There are 17 categories in this dataset: (1) frontpage, (2) news, (3) tech, (4) local, (5) opinion, (6) on-air, (7) misc, (8) weather, (9) msn-news, (10) health, (11) living, (12) business, (13) msn-sports, (14) sports, (15) summary, (16) bbs, and (17) travel. The clickstreams in this data contained within-state repetitions, meaning, for example, that a user could look at three different pages within the weather category, and would

be recorded as (5, 5, 5) in the clickstream. To avoid imposing a non-existent group structure with the simulated time points, the times were simulated using exponential time with rate $1/w$, where w is the number of within-state repetitions. Therefore, more within-state transitions would result in longer times on average, which would be a reasonable assumption. The MSNBC data is available in the `ClickClust` package Melnykov (2016a) for R (R Core Team, 2019) and modified MSNBC dataset analyzed herein is available as `mMSNBC` within the `ClickClustCont` package (Gallaugher and McNicholas, 2019a) for R.

Three different cases are considered with the number of groups ranging from $G = 1$ to $G = 5$. The first case is our continuous time model (CM), the second is the discrete model using the data without within-state repetitions (DM), and finally, we fit the discrete model on the data with within-state repetitions (DWM), which is the original dataset. Both the DWM and CM models find three groups, and the DM model finds only two groups. Table 7.7 contains classification tables comparing the classification results of the CM and DWM models as well as the CM and DM models. It is interesting to note that the clusters found by the CM and DWM models are very similar. Moreover, when comparing the CM and DM models, it appears that observations in groups 1 and 2 using the CM are generally combined into one cluster in the DM.

These results are not too surprising because the data with within-state transitions would implicitly take into account the amount of time whereas the data without within-state transitions would not contain this information. This also indicates that if a given clickstream dataset contains the time information but no within-state transitions, the CM model would have the ability to detect, potentially important,

additional groups not found by the discrete model.

Table 7.5: Average ARI values for unlabelled observations over 100 datasets, with standard deviations in parentheses, for Simulation 7.3A.

		L from 4 to 25		
Sample Size	Model	20% Supervision	40% Supervision	80% Supervision
N=50	Continuous	0.946(0.069)	0.967(0.068)	0.984(0.079)
	Discrete	0.945(0.071)	0.967(0.068)	0.984(0.079)
N=100	Continuous	0.948(0.046)	0.960(0.047)	0.990(0.044)
	Discrete	0.947(0.046)	0.964(0.042)	0.992(0.039)
N=200	Continuous	0.954(0.035)	0.969(0.034)	0.987(0.034)
	Discrete	0.954(0.034)	0.969(0.032)	0.987(0.034)
N=400	Continuous	0.963(0.020)	0.971(0.022)	0.991(0.021)
	Discrete	0.963(0.020)	0.972(0.021)	0.991(0.021)
		L from 25 to 100		
N=50	Continuous	1.00(0.00)	1.00(0.00)	1.00(0.00)
	Discrete	1.00(0.00)	1.00(0.00)	1.00(0.00)
N=100	Continuous	1.00(0.00)	1.00(0.00)	1.00(0.00)
	Discrete	1.00(0.00)	1.00(0.00)	1.00(0.00)
N=200	Continuous	1.00(0.00)	1.00(0.00)	1.00(0.00)
	Discrete	1.00(0.00)	1.00(0.00)	1.00(0.00)
N=400	Continuous	1.00(0.00)	1.00(0.00)	1.00(0.00)
	Discrete	1.00(0.00)	1.00(0.00)	1.00(0.00)

Table 7.6: Average ARI values for unlabelled observations over 100 datasets, with standard deviations in parentheses, for Simulation 7.3B.

		L from 4 to 25		
Sample Size	Model	20% Supervision	40% Supervision	80% Supervision
N=75	Continuous	0.959(0.042)	0.968(0.045)	0.988(0.050)
	Discrete	0.501(0.087)	0.594(0.109)	0.828(0.168)
N=150	Continuous	0.960(0.028)	0.969(0.029)	0.988(0.033)
	Discrete	0.526(0.074)	0.638(0.065)	0.882(0.103)
N=300	Continuous	0.966(0.021)	0.976(0.020)	0.989(0.022)
	Discrete	0.578(0.053)	0.670(0.058)	0.890(0.072)
N=600	Continuous	0.967(0.013)	0.975(0.013)	0.993(0.013)
	Discrete	0.616(0.034)	0.702(0.037)	0.890(0.056)
		L from 25 to 100		
N=75	Continuous	1.00(0.00)	1.00(0.00)	1.00(0.00)
	Discrete	0.848(0.080)	0.899(0.069)	0.977(0.071)
N=150	Continuous	1.00(0.00)	1.00(0.00)	1.00(0.00)
	Discrete	0.889(0.043)	0.904(0.052)	0.973(0.051)
N=300	Continuous	1.00(0.00)	1.00(0.00)	1.00(0.00)
	Discrete	0.906(0.033)	0.927(0.038)	0.978(0.033)
N=600	Continuous	1.00(0.00)	1.00(0.00)	1.00(0.00)
	Discrete	0.909(0.021)	0.928(0.020)	0.972(0.027)

Table 7.7: Classification comparison of the CM model with the DWM and DM models for the MSNBC dataset with simulated time stamps.

	DWM			DM	
CWM	1	2	3	1	2
1	90	4	0	78	16
2	14	128	3	139	6
3	3	4	77	2	82

7.5 Summary

An approach was introduced that incorporates continuous time for unsupervised and semi-supervised classification of clickstream data. This approach is based on a mixture of first-order continuous time Markov models. An EM algorithm was outlined for parameter estimation, and the BIC was used to select the number of groups G.

In the analyses that were carried out, we noted that incorporating the amount of time spent in each category allowed for the detection of groups of users that the discrete time model was unable to detect. This was especially true where there was not a lot of separation in the transition probabilities between groups, but differences in the average amount of time spent in each state. Moreover, if the amount of time spent in each state was similar, on average, between groups but there was a lot of separation in the transition probabilities, the continuous time model performance was very similar to the discrete time model. Finally, the real data analysis suggested that if no within-state transitions are considered, but the amount of time is given, then the continuous model would be able to detect, potentially important, additional groups not found when using the discrete model. These results indicate that the continuous time model is possibly more robust than the discrete time model, especially when there are no within-state transitions provided.

Chapter 8

Parsimonious Mixtures of Matrix Variate Bilinear Factor Analyzers

8.1 Introduction

In this chapter, a small extension of the MMVBFA model from Chapter 5. One feature of the MMVBFA model is that each of the resultant scale matrices has the same form as the covariance matrix in the (multivariate) mixture of factor analyzers model. Therefore, MMVBFA lends itself naturally to a matrix variate extension of the parsimonious Gaussian mixture models (PGMMs) developed by McNicholas and Murphy (2008). Specifically, we apply combinations of the constraints $\mathbf{\Lambda}_g = \mathbf{\Lambda}$, $\mathbf{\Sigma}_g = \mathbf{\Sigma}$, $\mathbf{\Sigma}_g = \sigma_g \mathbf{I}_n$ with $\sigma_g \in \mathbb{R}^+$, $\mathbf{\Delta}_g = \mathbf{\Delta}$, $\mathbf{\Psi}_g = \mathbf{\Psi}$, and $\mathbf{\Psi}_g = \psi_g \mathbf{I}_p$ with $\psi_g \in \mathbb{R}^+$. This leads to a total of 64 models, which we refer to as the parsimonious mixtures of matrix variate bilinear factor analyzers (PMMVBFA) family. In Tables 8.1 and 8.2, the models along with the number of scale parameters are presented for the row and column scale matrices. We will refer to these as the row and column models,

Table 8.1: Row models with the respective number of scale parameters.

$\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}$	$\boldsymbol{\Sigma}_g = \boldsymbol{\Sigma}$	$\boldsymbol{\Sigma}_g = \sigma_g \mathbf{I}_n$	Number of Scale Parameters
C	C	C	$[nq + n - q(q-1)/2] + 1$
C	C	U	$[nq + n - q(q-1)/2] + n$
C	U	C	$[nq + n - q(q-1)/2] + G$
C	U	U	$[nq + n - q(q-1)/2] + nG$
U	C	C	$G[nq + n - q(q-1)/2] + 1$
U	C	U	$G[nq + n - q(q-1)/2] + n$
U	U	C	$G[nq + n - q(q-1)/2] + G$
U	U	U	$G[nq + n - q(q-1)/2] + nG$

Table 8.2: Column models with the respective number of scale parameters.

$\boldsymbol{\Delta}_g = \boldsymbol{\Delta}$	$\boldsymbol{\Psi}_g = \boldsymbol{\Psi}$	$\boldsymbol{\Psi}_g = \psi_g \mathbf{I}_r$	Number of Scale Parameters
C	C	C	$[pr + p - r(r-1)/2] + 1$
C	C	U	$[pr + p - r(r-1)/2] + p$
C	U	C	$[pr + p - r(r-1)/2] + G$
C	U	U	$[pr + p - r(r-1)/2] + pG$
U	C	C	$G[pr + p - r(r-1)/2] + 1$
U	C	U	$G[pr + p - r(r-1)/2] + p$
U	U	C	$G[pr + p - r(r-1)/2] + G$
U	U	U	$G[pr + p - r(r-1)/2] + pG$

respectively. Parameter estimation proceeds in the same manner as the MMVBFA model with the exception of the updates for $\boldsymbol{\Sigma}_g$, $\boldsymbol{\Psi}_g$, $\boldsymbol{\Lambda}_g$, and $\boldsymbol{\Delta}_g$. The exact updates are given in Appendix A.

8.2 Simulations

8.2.1 Simulation 8.1

Three simulations were conducted. In the first, we consider $d \times d$ matrices with $d \in \{10, 20\}$, $G = 2$ and $\mathbf{M}_1 = \mathbf{0}, \mathbf{M}_2 = \mathbf{M}_{\text{LT}}^{(\delta)}$, where $\delta \in \{1, 2, 4\}$ and $\mathbf{M}_{\text{LT}}^{(\delta)}$ represents a lower triangular matrix with δ on and below the diagonal. We consider the case where both rows and columns have a CCU model. The parameters for the column factor loading matrices are:

$$\boldsymbol{\Lambda}_1 = \boldsymbol{\Lambda}_2 = \begin{bmatrix} \mathbf{1}_5 & \mathbf{0}_5 & \mathbf{0}_5 \\ \mathbf{0}_2 & \mathbf{1}_2 & \mathbf{0}_2 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{1}_3 \end{bmatrix} (d = 10), \qquad \boldsymbol{\Lambda}_1 = \boldsymbol{\Lambda}_2 = \begin{bmatrix} \mathbf{1}_{10} & \mathbf{0}_{10} & \mathbf{0}_{10} \\ \mathbf{0}_4 & \mathbf{1}_4 & \mathbf{0}_4 \\ \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{1}_6 \end{bmatrix} (d = 20).$$

The row factor loading matrices are

$$\boldsymbol{\Delta}_1 = \boldsymbol{\Delta}_2 = \begin{bmatrix} -\mathbf{1}_{d/2} & \mathbf{0}_{d/2} \\ \mathbf{1}_{d/2} & \mathbf{1}_{d/2} \end{bmatrix},$$

where $\mathbf{1}_c$ and $\mathbf{0}_c$ represent c-dimensional vectors of 1s and 0s, respectively. The error covariance matrices are taken to be

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Psi}_1 = \boldsymbol{\Psi}_2 = \mathbf{D},$$

where $\mathbf{D}$ is a diagonal matrix with diagonal entries $d_{tt} = t/5$ when $d = 10$ and $d_{tt} = t/10$ when $d = 20$.

Finally, sample sizes of $N \in \{100, 200, 400\}$ are considered with $\pi_1 = \pi_2 = 0.5$.

Table 8.3: Number of datasets for which the BIC correctly chose the number of groups (G), column factors (q), row factors (r), row model (RM), column model (CM), and the average ARI over 25 datasets (Simulation 8.1)

		$d = 10$						$d = 20$					
δ	N	G	q	r	RM	CM	$\overline{\text{ARI}}$(sd)	G	q	r	RM	CM	$\overline{\text{ARI}}$(sd)
	100	0	25	25	25	25	0.000(0.00)	25	24	25	25	25	1.000(0.00)
1	200	21	25	25	25	25	0.723(0.33)	25	24	24	25	24	1.000(0.00)
	400	25	25	25	25	25	0.883(0.04)	25	25	25	25	25	1.000(0.002)
	100	25	25	25	25	25	1.000(0.00)	25	24	25	25	25	1.000(0.00)
2	200	25	25	25	25	25	0.999(0.004)	25	25	25	25	25	1.000(0.00)
	400	25	25	25	25	25	1.000(0.002)	25	25	25	25	25	1.000(0.00)
	100	25	25	25	25	25	1.000(0.00)	25	24	25	25	25	1.000(0.00)
4	200	25	25	24	25	25	1.000(0.00)	25	25	25	25	25	1.000(0.00)
	400	25	25	25	25	25	1.000(0.00)	25	25	25	25	25	1.000(0.00)

For each of these combinations, 25 datasets are simulated. The model is fit for $G = 1, \ldots, 4$ groups, 1 to 5 row factors and column factors, and all 64 scale models, leading to a total of 6,400 models fit for each dataset.

In Table 8.3, we display the number of times the correct number of groups, row factors, and column factors are selected by the BIC, as well as the number of times the row and column models were correctly identified. We also include the average ARI over the 25 datasets with associated standard deviations. As expected, as the separation and sample size increase, better classification results are obtained. The correct number of groups, column factors, and row factors are chosen for all 25 datasets in nearly all cases considered. Moreover, the selection of the row and column models is very accurate in all cases considered.

8.2.2 Simulation 8.2

In this simulation, similar conditions to Simulation 8.1 are considered, including using the same mean matrices; however, we place a CUC model on the rows and a UCU model on the columns. The column factor loading matrices are the same as used for Simulation 8.1, $\boldsymbol{\Delta}_1$ is the same as in Simulation 8.1, and the row factor loadings matrix for group 2 is

$$\boldsymbol{\Delta}_2 = \begin{bmatrix} \mathbf{1}_{d/2} & -\mathbf{1}_{d/2} \\ \mathbf{1}_{d/2} & \mathbf{0}_{d/2} \end{bmatrix}.$$

We take $\boldsymbol{\Sigma}_1 = \mathbf{I}_d, \boldsymbol{\Sigma}_2 = 2\mathbf{I}_d$ and $\boldsymbol{\Psi}_1 = \boldsymbol{\Psi}_2 = \mathbf{D}$, where $\mathbf{D}$ is the same as from Simulation 8.1.

Results are displayed in Table 8.4. Overall, we obtain excellent classification results, even when the sample size is small and there is little spatial separation. There is some difficulty in choosing the column model when $d = 10$ but this issue abates for $N = 400$. When $d = 20$, some difficulty is encountered in choosing the correct number of column factors q; however, the classification performance is consistently excellent.

Table 8.4: Number of datasets for which the BIC correctly chose the number of groups (G), column factors (q), row factors (r), row model (RM), column model (CM), and the average ARI over 25 datasets (Simulation 8.2)

		$d = 10$						$d = 20$					
δ	N	G	q	r	RM	CM	$\overline{\text{ARI}}$(sd)	G	q	r	RM	CM	$\overline{\text{ARI}}$(sd)
	100	25	25	25	25	25	0.990(0.02)	25	0	25	25	25	1.000(0.00)
1	200	25	25	25	25	1	0.998(0.007)	25	24	25	25	25	1.000(0.00)
	400	25	25	25	25	25	0.997(0.006)	25	25	25	25	25	1.000(0.00)
	100	25	25	25	25	0	0.998(0.01)	25	0	25	25	25	1.000(0.00)
2	200	25	25	25	25	0	1.000(0.00)	25	24	25	25	25	1.000(0.00)
	400	25	25	25	25	25	0.999(0.003)	25	25	25	25	25	1.000(0.00)
	100	25	25	25	25	0	1.000(0.00)	25	10	25	25	25	1.000(0.00)
4	200	25	25	25	25	2	1.000(0.00)	25	23	25	25	25	1.000(0.00)
	400	25	25	24	25	25	1.000(0.00)	25	5	25	25	20	1.000(0.00)

8.2.3 Simulation 8.3

In the last simulation, the mean matrices are now diagonal with diagonal entries equal to δ. A CCU model is taken for the rows. In the case of $d = 10$, the parameters are

$$\boldsymbol{\Lambda}_1 = \boldsymbol{\Lambda}_2 = \begin{bmatrix} \mathbf{1}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{1}_2 & \mathbf{0}_2 & \mathbf{1}_2 \\ -\mathbf{1}_2 & -\mathbf{1}_2 & -\mathbf{1}_2 \\ -\mathbf{1}_3 & -\mathbf{1}_3 & \mathbf{0}_3 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbf{I}_{d\{\sigma_{2,2}=2,\sigma_{9,9}=4\}}.$$

To clarify this notation, the row scale matrices have 1s on the diagonal except for places 2 and 9 which have values 2 and 4 respectively. The column scale matrices have a UCC model with

$$\boldsymbol{\Delta}_1 = \begin{bmatrix} -\mathbf{1}_5 & \mathbf{0}_5 \\ \mathbf{1}_5 & \mathbf{1}_5 \end{bmatrix}, \qquad \boldsymbol{\Delta}_2 = \begin{bmatrix} -\mathbf{1}_5 & \mathbf{1}_5 \\ \mathbf{1}_5 & \mathbf{0}_5 \end{bmatrix}, \qquad \boldsymbol{\Psi}_1 = \boldsymbol{\Psi}_2 = \mathbf{I}_{10}.$$

In the case of $d = 20$, the parameters are

$$\boldsymbol{\Lambda}_1 = \boldsymbol{\Lambda}_2 = \begin{bmatrix} \mathbf{1}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{1}_4 & \mathbf{0}_4 & \mathbf{1}_4 \\ -\mathbf{1}_4 & -\mathbf{1}_4 & -\mathbf{1}_4 \\ -\mathbf{1}_6 & -\mathbf{1}_6 & \mathbf{0}_6 \end{bmatrix}, \qquad \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbf{I}_{30\{\sigma_{2,2}=4,\sigma_{9,9}=2,\sigma_{12,12}=3,\sigma_{19,19}=5\}},$$

and

$$\boldsymbol{\Delta}_1 = \begin{bmatrix} -\mathbf{1}_{10} & \mathbf{0}_{10} \\ \mathbf{1}_{10} & \mathbf{1}_{10} \end{bmatrix}, \qquad \boldsymbol{\Delta}_2 = \begin{bmatrix} -\mathbf{1}_{10} & \mathbf{1}_{10} \\ \mathbf{1}_{10} & \mathbf{0}_{10} \end{bmatrix}, \qquad \boldsymbol{\Psi}_1 = \boldsymbol{\Psi}_2 = \mathbf{I}_{20}.$$

The results are presented in Table 8.5. In this case, there is more variability in the correct selection of the row and column models, especially the latter. The selection of q and r is generally accurate. The classification performance is generally very good with the exception of the combination of a small sample size N with a low degree of separation δ.

8.3 MNIST Data Analysis

The MNIST digits dataset is again considered. In this chapter, we consider digits 1 and 2. This dataset consists of 60,000 (training) images of Arabic numerals 0 to 9. We consider different levels of supervision and perform either clustering or semi-supervised classification. Specifically we look at 0% (clustering), 25%, and 50% supervision. For each level of supervision, 25 datasets consisting of 200 images each of digits 1 and 2 are taken. As discussed in Gallaugher and McNicholas (2018b),

Table 8.5: Number of datasets for which the BIC correctly chose the number of groups (G), column factors (q), row factors (r), row model (RM), column model (CM), and the average ARI over 25 datasets (Simulation 8.3)

		$d = 10$						$d = 20$					
δ	N	G	q	r	RM	CM	$\overline{\text{ARI}}$(sd)	G	q	r	RM	CM	$\overline{\text{ARI}}$(sd)
	100	0	25	25	25	0	0.000(0.00)	0	25	25	25	0	0.000(0.00)
1	200	0	25	25	20	0	0.000(0.00)	0	25	25	25	0	0.000(0.00)
	400	22	12	25	12	16	0.705(0.27)	22	21	24	19	13	0.833(0.32)
	100	25	24	25	24	17	0.968(0.04)	21	24	25	25	10	0.840(0.37)
2	200	25	25	25	25	11	0.984(0.02)	25	25	25	25	11	1.000(0.00)
	400	25	20	25	18	22	0.988(0.01)	25	25	25	25	20	1.000(0.00)
	100	25	24	25	24	15	1.000(0.00)	25	25	25	25	18	1.000(0.00)
4	200	25	25	25	25	10	1.000(0.00)	25	25	25	25	22	1.000(0.00)
	400	25	24	25	20	23	1.000(0.00)	25	25	25	25	17	1.000(0.00)

Table 8.6: Average ARI values and misclassification rates for each level of supervision, with respective standard deviations in parentheses, for datasets consisting of digits 1 and 2 drawn from the MNIST dataset

Supervision	ARI	Misclassification rate
0% (clustering)	0.652(0.05)	0.0962(0.02)
25%	0.733(0.059)	0.072(0.02)
50%	0.756(0.064)	0.065(0.018)

because of the lack of variability in the outlying rows and columns of the data matrices, random noise is added to ensure non-singularity of the scale matrices. In Table 8.6, we present the average ARIs and misclassification rates along with respective standard deviations.

As expected, as the level of supervision is increased, better classification performance is obtained. Specifically, the MCR decreases to around 6.5% with an ARI of 0.756 when the level of supervision is raised to 50%. Moreover, the performance in the completely unsupervised case is fairly good. In Figure 8.1, heatmaps for the estimated mean matrices, for one dataset, for each digit and level of supervision are

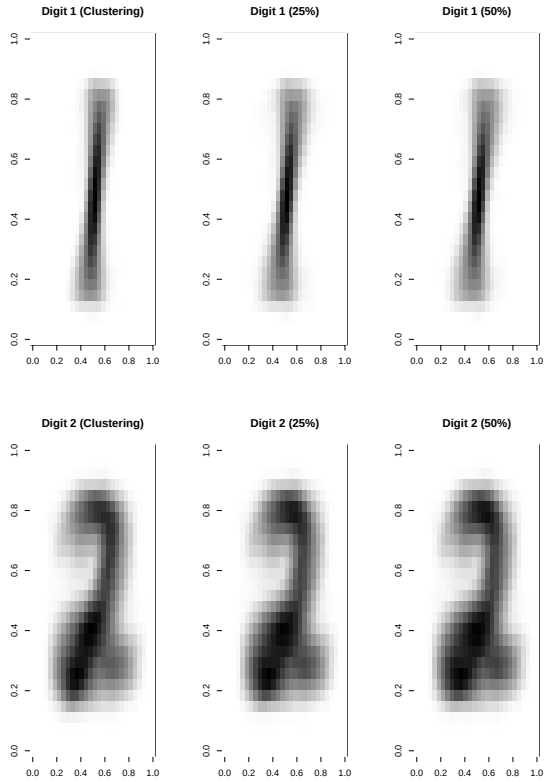


Figure 8.1: Heatmaps of the mean matrices, from one of the datasets, for each digit at each level of supervision.

presented. Although barely perceptible, there is a slight increase in clarity as the supervision is raised to 50%. For all levels of supervision, the UUU row model is chosen for all 25 datasets. The chosen model for the columns is the UCU model for 7 of the 25 datasets for 0% and 50% supervision, and 10 datasets for 25% supervision.

Chapter 9

Skewed Distributions or Transformations? Incorporating Skewness in a Cluster Analysis

9.1 Introduction

Due to its mathematical tractability, the Gaussian mixture model holds a special place in the clustering literature. For all its benefits, however, the Gaussian mixture model poses problems when dealing with data that is either skewed, or contains outliers. Specifically, in the presence of skewness and/or outliers, the Gaussian model tends to over fit the number of groups. Therefore, many methods have been proposed over the years to alleviate this issue; however, they fall within two main classes of methods. The first is to fit a mixture of more flexible distributions such as those that model skewness and or kurtosis. The second is to perform a suitable transformation to near normality and then fit a Gaussian mixture based on the transformed data.

The second is transformation-based mixture models. It assumes that after applying suitable transformation marginally to each component, data groups follow approximate normal distributions. The transformation-based mixture model is then derived based on back-transformation from the Gaussian mixture model. Although these methods have been compared to a certain extent in their respective papers, there is still uncertainty as to when one method might be preferred over another. Herein we aim to fill this gap by performing an extensive study on well known benchmarking clustering datasets with an extensive set of initialization partitions. In addition to this extensive comparison between these two classes of methods, a new method for determining cluster separation is also proposed.

9.2 Mixtures of Skewed Distributions

The first class of methods for dealing with skewness is to consider mixtures of more flexible distributions, many of which have already been discussed. For purposes of this chapter, we consider two representatives of these aforementioned distributions, specifically the variance gamma, and generalized hyperbolic distributions. The main reason for this selection is because both of these distributions are derived from the variance-mean mixture model.

If $W \sim \text{Gamma}(\gamma, \gamma)$, then the result is the variance gamma distribution (McNicholas *et al.*, 2017) and its density is

$$\begin{aligned} f_{\text{VG}}(\mathbf{x}|\boldsymbol{\vartheta}) =& \frac{2\gamma^{\gamma}\exp\left\{(\mathbf{x}-\mathbf{M})'\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha})\right\}}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}\Gamma(\gamma)}\left(\frac{\delta(\mathbf{x};\mathbf{M},\boldsymbol{\Sigma})}{\rho(\boldsymbol{\alpha},\boldsymbol{\Sigma})+2\gamma}\right)^{\frac{(\gamma-p/2)}{2}} \\ & \times K_{\left(\gamma-\frac{p}{2}\right)}\left(\sqrt{\left[\rho(\boldsymbol{\alpha},\boldsymbol{\Sigma})+2\gamma\right]\left[\delta(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})\right]}\right), \end{aligned}$$

where $\delta(\mathbf{X};\mathbf{M},\boldsymbol{\Sigma}) = (\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$, $\rho(\boldsymbol{\alpha},\boldsymbol{\Sigma}) = \boldsymbol{\alpha}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}$ and $\gamma > 0$. Likewise if $W \sim \text{GIG}(\omega, 1, \lambda)$, where GIG represents the generalized inverse Gaussian distribution with the parameterization used by Browne and McNicholas (2015), then the result is the generalized hyperbolic distribution, with density

$$\begin{aligned} f_{\text{GH}}(\mathbf{x}|\boldsymbol{\vartheta}) =& \frac{\exp\left\{(\mathbf{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}\boldsymbol{\alpha}'\right\}}{(2\pi)^{\frac{np}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}K_{\lambda}(\omega)}\left(\frac{\delta(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})+\omega}{\rho(\boldsymbol{\alpha},\boldsymbol{\Sigma})+\omega}\right)^{\frac{(\lambda-\frac{p}{2})}{2}} \\ & \times K_{(\lambda-p/2)}\left(\sqrt{[\rho(\boldsymbol{\alpha},\boldsymbol{\Sigma})+\omega]\,[\delta(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})+\omega]}\right), \end{aligned}$$

$\lambda \in \mathbb{R}$, $\omega > 0$.

Likewise the skew-t and NIG distributions can be derived in a similar manner. Parameter estimation is performed using an expectation conditional maximization (ECM) algorithm, and the details for the VG and GH distributions can be found in McNicholas *et al.* (2017) and Browne and McNicholas (2015), respectively.

Skewed distributions that are derived using hidden truncation such as the skew normal and the skew-t used in Lee and McLachlan (2014) were not considered due to the computational time associated with these distributions, and the extensive number of initializations used in the analyses.

The Infinite Likelihood Problem

One numerical aspect of the variance gamma distribution that must be considered is the infinite likelihood problem. This was discussed by Franczak *et al.* (2014) for the skewed asymmetric Laplace (SAL) distribution, and occurs when $\hat{\boldsymbol{\mu}}_g \to \mathbf{x}_i$. As the SAL distribution is a special case of the variance gamma with $\gamma = 1$, it is not surprising that this also occurs for the variance gamma distribution. This is due to the

density being unbounded when $\hat{\boldsymbol{\mu}}_g \to \mathbf{x}_i$ and $\gamma < p/2$, as discussed in Nitithumbundit and Chan (2015). This is due to both the Bessel function going to infinity as the argument approaches 0, and

$$\left(\frac{\delta(\mathbf{x};\mathbf{M},\boldsymbol{\Sigma})}{\rho(\boldsymbol{\alpha},\boldsymbol{\Sigma})+2\gamma}\right)^{\frac{(\gamma-p/2)}{2}} \to \infty$$

if $\boldsymbol{\mu} \to \mathbf{x}_i$ and $\gamma < p/2$.

The solution to this problem is not trivial. In Franczak *et al.* (2014), the authors propose running the ECM algorithm to the point where this occurs, go back one iteration and set $\hat{\boldsymbol{\mu}}_g$ to be the value at the preceding iteration, and update the skewness accordingly. A second possible solution is to bound the density in this situation, Nitithumbundit and Chan (2015). The problem with both of these solutions, is that at that point in the algorithm, the parameter estimates have most likely entered an unstable part of the parameter space. This could in turn dramatically affect the

Therefore, with all of this taken into consideration, we propose restricting γ so that if $\hat{\gamma}_g < p/2$ then we let $\hat{\gamma}_g = p/2$.

It is important to note that this scenario does not occur for the generalized hyperbolic, skew-t and NIG distributions, because in all of these cases, the $\delta(\cdot)$ term is accompanied by a positive value, ω in the generalized hyperbolic case, thus ensuring boundedness of the density function for all values of λ.

9.3 Transformation Methods

The second class of methods to handle skewness in a cluster analysis is the use of some transformation to near normality, introduced by Zhu and Melnykov (2018); Melnykov

and Zhu (2018, 2019). The basic idea is to assume that there exists a transformation, $\mathcal{T}(\mathbf{X} \mid \mathbf{\Lambda})$, where $\mathbf{X}$ is the original data vector, and $\mathbf{\Lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_p)'$ is a transformation vector, such that

$$\mathcal{T}(\mathbf{X} \mid \mathbf{\Lambda}) = (\mathcal{T}(\mathbf{x}_1|\lambda_1), \mathcal{T}(\mathbf{x}_2|\lambda_2), \ldots, \mathcal{T}(\mathbf{x}_p|\lambda_p)) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}),$$

where $\mathcal{N}(\cdot)$ represents the p-variate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$. $\lambda_1, \lambda_2, \ldots, \lambda_p$ are marginal transformation parameters responsible for each one of the p dimensions, respectively. Herein are considered two different univariate transformations, namely the power and Manly transformations.

The power transformation, proposed by Yeo and Johnson (2000), is defined as

$$\mathcal{T}(x|\lambda) = \begin{cases} [(x+1)^\lambda - 1]/\lambda & \text{if} \quad (x \geq 0, \lambda \neq 0), \\ \log(x+1) & \text{if} \quad (x \geq 0, \lambda = 0), \\ -[(-x+1)^{2-\lambda} - 1]/(2-\lambda) & \text{if} \quad (x < 0, \lambda \neq 2), \\ -\log(-x+1) & \text{if} \quad (x < 0, \lambda = 2). \end{cases}$$

The Manly transformation (also called the exponential transformation), introduced first by Manly (1976) is defined as

$$\mathcal{T}(x|\lambda) = \begin{cases} [\exp\{\lambda x\} - 1]/\lambda & \text{if} \quad \lambda \neq 0, \\ x & \text{otherwise.} \end{cases}$$

Both of these transformations have been shown to handle both positive and negative skewness. By applying the back-transformation from p-variate normal distribution, the corresponding transformation-based density can be written as

$$f_{\mathcal{T}}(\mathbf{x} \mid \boldsymbol{\vartheta}) = \phi(\mathcal{T}(\mathbf{x} \mid \boldsymbol{\Lambda}); \boldsymbol{\mu}, \boldsymbol{\Sigma}) J_{\mathcal{T}}(\mathbf{x} \mid \boldsymbol{\Lambda}),$$

where $J_{\mathcal{T}}(\mathbf{x} \mid \boldsymbol{\Lambda}) = |\partial \mathcal{T}(\mathbf{x} \mid \boldsymbol{\Lambda})/\partial \mathbf{x}'|$ represents the Jacobian derived based on the back-transformation from normal distribution. For the Manly transformation, its Jacobian can be written as $J_{\mathcal{T}}(\mathbf{x} \mid \boldsymbol{\Lambda}) \equiv \exp\{\boldsymbol{\Lambda}'\mathbf{x}\}$. For the power transformation,

$$J_{\mathcal{T}}(\mathbf{x} \mid \boldsymbol{\Lambda}) \equiv \prod_{j=1}^{p} (|x_j| + 1)^{\text{sgn}(x_j)(\lambda_j - 1)},$$

where

$$\text{sgn}(x) = \begin{cases} 1 & \text{if} \quad x > 0, \\ 0 & \text{if} \quad x = 0, \\ -1 & \text{if} \quad x < 0. \end{cases}$$

9.4 Measures used for Comparison

9.4.1 Multivariate Skewness and Kurtosis

For the purposes considered here, an assessment of component skewness and kurtosis is desirable. It is possible to consider the univariate skewness and kurtosis for each dimension; however, for higher dimensions this becomes difficult to assess. Therefore, the use of Mardia's multivariate skewness and kurtosis (Mardia, 1970) is employed, as this gives a single measure for skewness and kurtosis, and it provides a test for assessing multivariate normality.

The multivariate skewness, $\beta_{1,p}$ for a multivariate random vector $\mathbf{X}$ with mean vector $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_p)$ and covariance $\boldsymbol{\Sigma}$ is defined as

$$\beta_{1,p} = \sum_{r,s,t} \sum_{r',s',t'} \sigma^{rr'} \sigma^{ss'} \sigma^{tt'} \mu_{111}^{(rst)} \mu_{111}^{(r's't')},$$

where σ^{ij} is the i, j element of the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ and

$$\mu_{111}^{rst} = \mathbb{E}[(X_r - \mu_r)(X_s - \mu_s)(X_t - \mu_t)].$$

Under multivariate normality, $\beta_{1,p} = 0$.

The multivariate kurtosis is similar and is given by

$$\beta_{2,p} = \mathbb{E}[(\mathbf{X} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \boldsymbol{\mu})]^2,$$

and under multivariate normality has value $p(p+2)$. Therefore, we report the kurtosis as $\hat{\beta}_{2,p} - p(p+2)$, so that negative kurtosis corresponds to lighter tails and positive kurtosis corresponds to heavier tails. In addition to these values, Mardia (1970) also provides tests to determine if the skewness and kurtosis are significantly different from what is expected under normality.

These values, along with their p-values for the tests can be calculated using the R package `psych` (Revelle, 2018).

9.4.2 Cluster Overlap

One property of a dataset we consider for comparing the two classes of methods is cluster separation. An outline of the method we propose is described as follows.

1. **Density Estimation**:

   In this step, a density function $\hat{f}_g(\mathbf{x})$ is estimated for each group $g \in \{1, 2, \ldots, G\} = \mathcal{G}$ with mixing proportions $\pi_1, \pi_2, \ldots, \pi_G$.

2. **Simulation**:

   For each group g, simulate N observations. Denote these observation matrices by $\mathbf{x}_g = (\mathbf{x}_{g1}, \mathbf{x}_{g2}, \ldots, \mathbf{x}_{gN})$.

3. **Calculation**:

   For each $\mathbf{x}_g$ and $g \in \mathcal{G}$, let $\mathbf{C}_g$ denote an N dimensional vector with entry i being

   $$\mathbf{C}_g\{i\} = \text{argmax}_{h \in \{1,2,\ldots,G\}} \pi_h \hat{f}_g(\mathbf{x}_{ih})$$

4. **Map**:

   For each $g, h \in \mathcal{G}$, let

   $$p_{gh} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\mathbf{C}_g = h)$$

   and let the $G \times G$ map matrix P to be defined as $P\{g, h\} = p_{gh}$.

This general method has been used, for example, in Melnykov (2016b) after fitting the model of interest, and considered pairwise overlap between clusters.

For the purposes considered here, it is desirable that the estimated density captures the true nature of the component and for simulation to be computationally feasible. Herein, two different methods are proposed. The first is to consider a mixture of Gaussian distributions,

$$\hat{f}_g(\mathbf{x}|\boldsymbol{\vartheta}) = \sum_{j=1}^{J} \pi_j \phi_p(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j).$$

Although not effective for modelling multiple skewed components, a mixture of Gaussian distributions is effective for modelling a single skewed component. Moreover, it is clear that it is simple to simulate from this density function, generally quite flexible, and can effectively captures the nature of a component.

Another method we propose for density estimation is to consider kernel density estimation (KDE). This assumes that the estimate of the density can be written

$$\hat{f}_g(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i),$$

where n is the sample size, $\mathbf{H}$ is a smoothing matrix, and, in this case, $K(\cdot)$ is the Gaussian kernel.

The choice of a smoother matrix is not trivial, especially in the multivariate case, and many such matrices have been proposed. However, for our purposes, we consider a diagonal smoother matrix with elements

$$h_j = \left(\frac{4}{p+2}\right)^{1/(p+4)} n^{-1/(p+4)}\hat{\sigma}_j,$$

where $\hat{\sigma}_j$ is the standard deviation of variable j (Härdle and Müller, 1997).

Example: Iris Dataset

An example of the cluster overlap procedure described above is now presented on the well known Iris dataset, Anderson (1935). This dataset provides four measurements on three different species of iris. Figure 9.1 displays a pairs plot for the original dataset, and Figure 9.2 shows 1000 simulated points for each cluster using the two different density estimation methods described above. Table 9.1 shows the resultant

Table 9.1: Gaussian mixture and KDE misclassification maps for the iris dataset.

Gaussian Mixture Map:	$\begin{pmatrix} 1.00 & 0.00 & 0.00 \\ 0.00 & 0.98 & 0.02 \\ 0.00 & 0.03 & 0.97 \end{pmatrix}$			KDE Map:	$\begin{pmatrix} 1.00 & 0.00 & 0.00 \\ 0.03 & 0.77 & 0.21 \\ 0.00 & 0.17 & 0.83 \end{pmatrix}$		

misclassification maps for the two different density estimation methods.

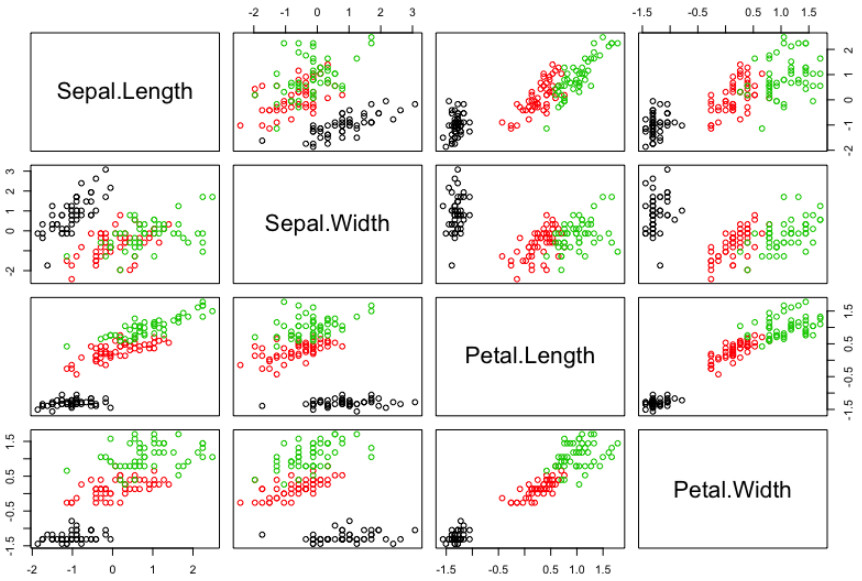
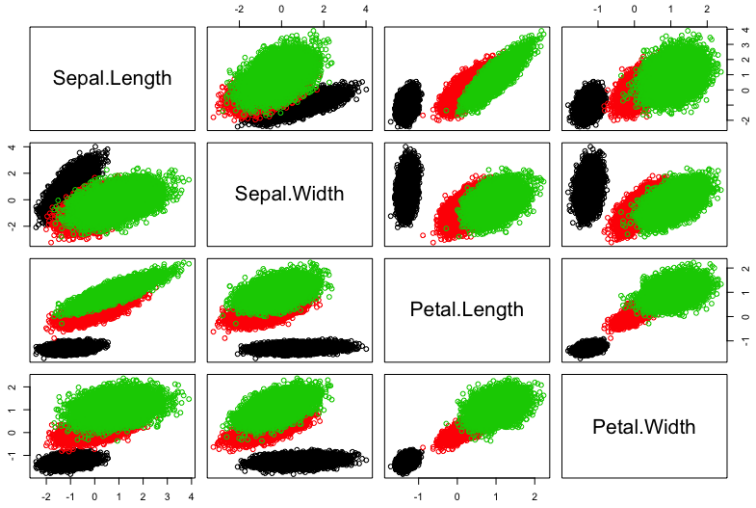


Figure 9.1: Pairs plot of the iris dataset.

It is interesting to note that when using KDE, there is more overlap than when using a Gaussian mixture. The reason for this is simple. When using a mixture of Gaussian distributions for each component, naturally more points will be simulated closer to the centre of each mode in the mixture. In the case of KDE estimation, the points in the tail of the distribution, more importantly the points which lie on the border of the two clusters, have the same probability as the points in centre of being chosen as the point around which to simulate.



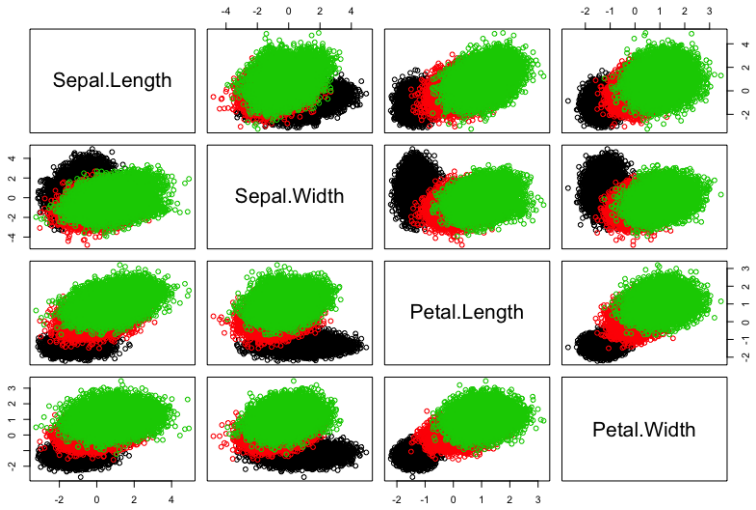


Figure 9.2: 1000 simulated points for each component of the iris dataset simulated using (a) a fitted mixture of Gaussian distributions and (b) using KDE.

General Applicability

It is important to note that in the scenario of a true cluster analysis this method for measuring cluster overlap is not applicable because the true group labels are not known. However, this is useful for the purposes presented here as we wish to determine if cluster separation does result in different performance for the two different methods. Moreover, this overlap procedure would be useful in other scenarios. One example would be in the case of a discriminant analysis, where the true labels are known. Another would be in the case determining separation of classes of categorical variables such as gender or race in regardless of the method of analysis.

9.4.3 Initialization and Convergence

In order to increase the chances of obtaining the maximum likelihood, many different starting values are considered and then the algorithm is ran to full convergence. Specifically, up to eleven k-means initializations, 1000 soft partitions, up to $100G$ (where G is the number of groups) unique hard initializations by running 1 iteration of the k-means algorithm, see Melnykov and Melnykov (2012) for details, and a hierarchical partition using Ward's linkage. Note that only $100G$ hard initializations are considered because the number of unique hard partitions increases with G. In addition, although not applicable in practice, but useful for comparison purposes, initializing with the true labels is considered. The final results are obtained by taking the largest likelihood over all these initializations, with the exception of the true labels as this is not applicable in a true cluster analysis.

The same convergence criterion was used for both methods. Specifically, the EM algorithm is terminated when $(\ell^{(t+1)} - \ell^{(t)})/|\ell^{(t+1)}| < 0.0001$.

9.5 Comparison

9.5.1 Some Technical Differences

One main difference between the use of skewed distributions and transformation methods is how the skewness is how the skewness and kurtosis is modelled. In the case of the skewed distributions considered here, and many others, the skewness is modelled explicitly by means of the skewness vector. Moreover, the concentration parameter also allows for the direct modelling of kurtosis. In the case of transformation methods, the skewness and kurtosis are modelled implicitly by means of the transformation vector $\boldsymbol{\Lambda}$. Therefore, it can be argued that the use of skewed distributions allows for slightly increased interpretability concerning skewness and kurtosis.

On the other hand, the transformation methods are more parsimonious in terms of the number of free parameters. Specifically, transformation methods have a total of pG additional parameters, when compared to the Gaussian mixture model, from the transformation vector $\boldsymbol{\Lambda}$. In the case of the skewed distributions considered herein, as well as many others, there are a total of $pG + G$, or in the case of the generalized hyperbolic distribution, $pG + 2G$ additional parameters. This is due to the addition of the concentration/index parameter(s). Although not significant for small G and large n, this could become significant in the case of larger G and smaller n.

9.5.2 Comparison Using Multiple Datasets

Using the initialization, model selection and convergence criteria outlined previously, we perform a comparison based on multiple real benchmarking datasets. The primary reason for this is to get a better sense of situations on which to base simulations.

However, as will be seen presently, simulations are most likely not necessary.

The first dataset we consider is the iris dataset described previously. Again, this dataset considers 150 observations from three different species of iris, on four variables. The results are summarized in Table 9.2 along with the skewness, kurtosis with their respective p-values, and the classification maps. Both the skewness and kurtosis are not significantly different than what would be expected under multivariate normality. Moreover, as discussed previously, and as is well known with this dataset, the first species is well separated from species 2 and 3, and species 2 and three have a fair amount of overlap. The overall performance over these methods is identical in terms of the number of groups chosen, and the classification performance. The likelihood values are very comparable over all methods, and the lowest BIC is obtained by the power transformation; however, given the comparable likelihood values and the additional parameters, this difference is not really significant.

Table 9.2: Results of the skewed models and transformation methods for the Iris dataset.

	Skewed Models		Transformations	
	VG	GH	Manly	Power
Log-Likelihood	-307.31	-311.04	-308.24	$\mathbf{-306.81}$
BIC	810.03	827.51	801.86	**799.01**
$\mathcal{M}$	39	41	37	37
G	2	2	2	2
ARI	0.568	0.568	0.568	0.568
Confusion	$\begin{pmatrix} 50 & 0 \\ 0 & 50 \\ 0 & 50 \end{pmatrix}$	$\begin{pmatrix} 50 & 0 \\ 0 & 50 \\ 0 & 50 \end{pmatrix}$	$\begin{pmatrix} 50 & 0 \\ 0 & 50 \\ 0 & 50 \end{pmatrix}$	$\begin{pmatrix} 50 & 0 \\ 0 & 50 \\ 0 & 50 \end{pmatrix}$
$G = 3 \quad p = 4 \quad n : 50 + 50 + 50 = 150$				
Skewness: (2.90(0.24), 2.84(0.26), 2.97(0.21))				
Kurtosis: (1.49(0.45), -2.03(0.30), -0.66(0.74))				
KDE Map: $\begin{pmatrix} 1.00 & 0.00 & 0.00 \\ 0.03 & 0.77 & 0.21 \\ 0.00 & 0.17 & 0.83 \end{pmatrix}$			Gaussian Map: $\begin{pmatrix} 1.00 & 0.00 & 0.00 \\ 0.00 & 0.98 & 0.02 \\ 0.00 & 0.03 & 0.97 \end{pmatrix}$	

The second dataset considered was the 13 variable wine dataset from the R package `rattle`, Williams (2011), which measures 13 chemical properties of 3 three different types of wine. Results are shown in Table 9.3. All four methods under fitted the true number of groups. This could be due to one of two reasons. The first is the dimensionality of the data and the fact that we are fitting an unconstrained covariance/scale matrix. However, this could also be due to the significant negative kurtosis for groups 1 and 3. Again, however, there is very little difference in the likelihood and BIC values. Moreover, in terms of classification performance, the results are very similar across methods.

Table 9.3: Results of the skewed models and transformation methods for the Wine dataset.

	Skewed Models		Transformations	
	VG	GH	Manly	Power
Log-Likelihood	-2188.94	-2189.30	-2153.58	-2144.87
BIC	5605.96	5617.04	5524.88	5507.45
$\mathcal{M}$	237	239	235	235
G	2	2	2	2
ARI	0.461	0.461	0.454	0.469
Confusion	$\begin{pmatrix} 59 & 0 \\ 66 & 5 \\ 0 & 48 \end{pmatrix}$	$\begin{pmatrix} 59 & 0 \\ 66 & 5 \\ 0 & 48 \end{pmatrix}$	$\begin{pmatrix} 59 & 0 \\ 65 & 6 \\ 0 & 48 \end{pmatrix}$	$\begin{pmatrix} 59 & 0 \\ 67 & 4 \\ 0 & 48 \end{pmatrix}$
$G = 3 \quad p = 13 \quad n : 59 + 71 + 48 = 178$				
Skewness: (47.27(0.37), 57.68(2.35e-11), 50.44(0.96))				
Kurtosis: $(-13.62(8.05\text{e}-3), 10.22(0.029), -16.63(3.52\text{e}-3))$				
KDE Map: $\begin{pmatrix} 0.91 & 0.08 & 0.00 \\ 0.06 & 0.90 & 0.04 \\ 0.00 & 0.13 & 0.87 \end{pmatrix}$			Gaussian Map: $\begin{pmatrix} 1.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{pmatrix}$	

The next dataset considered was the bankruptcy dataset from the R package `MixGHD` (Tortora *et al.*, 2015) with the results shown in Table 9.4. For all four methods, the BIC choose one component, and the BIC values are once again comparable. From the skewness and kurtosis measures and from looking at the plot of the data, it is clear

that the first component (bankrupt firms) are non-Gaussian and the second group are approximately Gaussian. Moreover, the misclassification maps suggest overlap between these two clusters. This actually displays a general issue in the context of clustering. If a symmetric cluster overlaps with a symmetric cluster and there is no information about the underlying group structure, then it will be very difficult to determine if there is one cluster or two. Moreover, as seen here, using skewed methods do not help in this scenario, as all four fail to capture the two group solution, and as mentioned before the Gaussian mixture over fits the number of groups. This is an area that should be addressed in future work, as it is quite possible using these methods can result in missing a possibly very important group.

Table 9.4: Results of the skewed models and transformation methods for the Bankruptcy dataset.

	Skewed Models		Transformations	
	VG	GH	Manly	Power
Log-Likelihood	-114.84	-110.85	-108.94	-106.54
BIC	263.20	259.40	247.21	242.41
$\mathcal{M}$	8	9	7	7
$\hat{G}$	1	1	1	1
ARI	0.000	0.000	0.000	0.000
Confusion	$\begin{pmatrix} 33 \\ 33 \end{pmatrix}$	$\begin{pmatrix} 33 \\ 33 \end{pmatrix}$	$\begin{pmatrix} 33 \\ 33 \end{pmatrix}$	$\begin{pmatrix} 33 \\ 33 \end{pmatrix}$
$G = 2 \quad p = 2 \quad n : 33 + 33 = 66$				
Skewness: $(15.33(< 1e-16), 0.54(0.56))$				
Kurtosis: $(15.42(< 1e-16), -1.43(0.30))$				
KDE Map: $\begin{pmatrix} 0.84 & 0.16 \\ 0.12 & 0.88 \end{pmatrix}$		Gaussian Map: $\begin{pmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{pmatrix}$		

The diabetes dataset, from Fraley *et al.* (2012) considers three measurements on 145 non-obese diabetes patients, with three types of diabetes which were classified as normal, overt and chemical, with results shown in Table 9.5. Again, very little difference is seen in the performance of the methods.

Table 9.5: Results of the skewed models and transformation methods for the Diabetes dataset.

	Skewed Models		Transformations	
	VG	GH	Manly	Power
Log-Likelihood	-178.75	-176.62	-171.68	-168.24
BIC	491.88	497.57	467.77	480.80
$\mathcal{M}$	27	29	25	25
$\hat{G}$	2	2	2	2
ARI	0.465	0.450	0.465	0.488
Confusion	$\begin{pmatrix} 32 & 4 \\ 76 & 0 \\ 2 & 31 \end{pmatrix}$	$\begin{pmatrix} 32 & 4 \\ 75 & 1 \\ 2 & 31 \end{pmatrix}$	$\begin{pmatrix} 32 & 4 \\ 76 & 0 \\ 2 & 31 \end{pmatrix}$	$\begin{pmatrix} 32 & 4 \\ 76 & 0 \\ 1 & 32 \end{pmatrix}$
$G = 3 \quad p = 3 \quad n : 36 + 76 + 33 = 145$				
Skewness: $(9.74(7.09\text{e}-9), 3.45(3.68\text{e}-6), 7.22(1.92\text{e}-5))$				
Kurtosis: $(8.73(1.72\text{e}-06), 3.22(0.010), 3.28(0.085))$				
KDE Map: $\begin{pmatrix} 0.47 & 0.40 & 0.13 \\ 0.13 & 0.87 & 0.01 \\ 0.09 & 0.25 & 0.66 \end{pmatrix}$			Gaussian Map: $\begin{pmatrix} 0.91 & 0.08 & 0.01 \\ 0.02 & 0.98 & 0.00 \\ 0.02 & 0.00 & 0.98 \end{pmatrix}$	

We also considered the AIS dataset, Azzalini (2018), which considers 11 measures from 100 female and 102 male athletes. The analysis on the full dataset is given in the supplementary material; however, here we present the results for the three commonly used variables for this dataset, namely the BMI, body fat and lean body mass with the results in Table 9.6. Again, very little difference was seen in performance for the generalized hyperbolic and both transformation methods. What is interesting, however, is where these misclassifications lie. Specifically, the variance-gamma misclassifies more men than women whereas the generalized hyperbolic misclassifies more women than men. Moreover, when comparing the transformation methods, equal numbers of men and women are misclassified.

The last dataset considered herein was the famous crabs dataset, Venables and Ripley (2002). This considers two species of crab (blue and orange) and males and females within each species. The first group are the blue males, the second the blue

Table 9.6: Results of the skewed models and transformation methods for the AIS dataset.

	Skewed Models		Transformations	
	VG	GH	Manly	Power
Log-Likelihood	-619.22	-618.18	-620.77	-604.21
BIC	1381.76	1390.29	1341.12	1347.50
$\mathcal{M}$	27	29	25	25
G	2	2	2	2
ARI	0.847	0.922	0.922	0.922
Confusion	$\begin{pmatrix} 99 & 1 \\ 7 & 95 \end{pmatrix}$	$\begin{pmatrix} 97 & 3 \\ 1 & 101 \end{pmatrix}$	$\begin{pmatrix} 98 & 2 \\ 2 & 100 \end{pmatrix}$	$\begin{pmatrix} 98 & 2 \\ 2 & 100 \end{pmatrix}$
$G = 2 \quad p = 3 \quad n : 100 + 102 = 202$				
Skewness: $(2.54(6.53\text{e}-6), 5.66(3.33\text{e}-16))$				
Kurtosis: $(1.69(0.12), 7.97(2.00\text{e}-13))$				
KDE Map: $\begin{pmatrix} 0.89 & 0.11 \\ 0.09 & 0.91 \end{pmatrix}$		Gaussian Map: $\begin{pmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{pmatrix}$		

females, the third the orange males, and the fourth the orange females. The results are shown in Table 9.7. What is very interesting, but not entirely clear from the overlap, skewness, and kurtosis from the four groups is that the skewed distribution methods separate the species perfectly, whereas the transformation methods discriminate based on gender. As all methods were run to convergence on many different initialization values, and the initializations were the same for each method, it is unlikely that it is due to the initialization. However, Table 9.8 shows the skewness and kurtosis values and their respective p-values based on sex and species, and it appears that the transformation methods found one skewed component and a symmetric component, whereas the skewed distribution methods found two skewed components. This might suggest that the transformation methods might be slightly more likely to find symmetric components than the skewed distributions.

Table 9.7: Results of the skewed models and transformation methods for the Crabs dataset.

	Skewed Models		Transformations	
	VG	GH	Manly	Power
Log-Likelihood	165.25	162.12	144.68	144.62
BIC	−49.69	−32.83	−19.16	2.17
$\mathcal{M}$	53	55	51	51
G	2	2	2	2
ARI	0.496	0.496	0.374	0.374
Confusion	$\begin{pmatrix} 50 & 0 \\ 50 & 0 \\ 0 & 50 \\ 0 & 50 \end{pmatrix}$	$\begin{pmatrix} 50 & 0 \\ 50 & 0 \\ 0 & 50 \\ 0 & 50 \end{pmatrix}$	$\begin{pmatrix} 47 & 3 \\ 6 & 44 \\ 50 & 0 \\ 4 & 46 \end{pmatrix}$	$\begin{pmatrix} 47 & 3 \\ 6 & 44 \\ 50 & 0 \\ 4 & 46 \end{pmatrix}$

$G = 4 \quad p = 5 \quad n : 50 + 50 + 50 + 50 = 200$

Skewness: $(4.71(0.28), 5.01(0.20), 2.82(0.93), 5.10(0.18))$

Kurtosis: $(-1.26(0.60), -0.80(0.73), -3.17(0.18), -0.18(0.94))$

KDE Map: $\begin{pmatrix} 0.33 & 0.25 & 0.29 & 0.13 \\ 0.20 & 0.38 & 0.20 & 0.22 \\ 0.26 & 0.18 & 0.37 & 0.19 \\ 0.15 & 0.31 & 0.22 & 0.32 \end{pmatrix}$ Gaussian Map: $\begin{pmatrix} 0.95 & 0.05 & 0.00 & 0.00 \\ 0.04 & 0.96 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.99 & 0.01 \\ 0.00 & 0.00 & 0.01 & 0.98 \end{pmatrix}$

9.6 Discussion

From the analyses performed on a variety of datasets and the extensive number and type of initializations performed, it appears that no one method consistently outperforms the others, and usually the performance is very similar if not identical. Moreover, it does not appear that skewness, kurtosis and cluster overlap completely

Table 9.8: Skewness and kurtosis for the crabs dataset based on sex and species separately.

	Skewness (p-value)	Kurtosis (p-value)
Males	2.7(0.12)	−2.38(0.15)
Females	3.64(0.0046)	−0.47(0.78)
Blue	4.9(1.30e − 5)	0.87(0.6)
Orange	4.01(9.50e − 4)	0.37(0.83)

determines the relative performance of these methods. This is not to say, however, that there are no differences between the two methods. As seen with the crabs, the skewed models discriminate based on species, and the transformation methods on sex. Although not shown herein, the skewed models also discriminate based on sex when using k-means initializations, but when run using the more flexible initialization methods, found the species structure. The transformation methods, on the other hand, always found the sex structure. Therefore, considering that the male crabs do not display skewness, it is possible that although transformation methods are capable of modelling skewed data, they are more likely to find symmetric clusters.

In terms of actual properties of the methods, transformation methods are more parsimonious, controlling for the number of groups, due to the lack of a concentration (or index) parameter. On the other hand, the use of skewed distributions allows for the direct calculation of the skewness and concentration which may be of interest in some scenarios. Therefore, it may not be a question of when one method might be preferable to another, but rather why one method might be preferable to another in the context of the analysis in question.

Chapter 10

Parameter-Wise Co-Clustering for High-Dimensional Data

10.1 Limitations of Co-Clustering

Co-Clustering is a very useful tool for analyzing high-dimensional data. This method considers simultaneous partitions of rows and columns, which are then used to organize the data into homogenous blocks. For traditional co-clustering, as in clustering, data are assumed to come in the form of an $n \times p$ matrix $\mathbf{x}$ with rows represented by $\mathbf{x}_i'$. Each individual element of $\mathbf{x}_i$ is denoted by x_{ij}, so that x_{ij} is the observation in row i and column j.

In co-clustering, there is an unknown partition of the rows into G clusters, from this point onwards referred to as row-clusters, represented by the indicator vector $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG}) \sim \text{Multinomial}(1; \boldsymbol{\pi})$, where z_{ig} is as defined previously, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_2)$ and Multinomial$(\cdot)$ represents the multinomial distribution. Unlike traditional co-clustering, however, there is also a partition of the columns into L clusters, referred

to as column-clusters, represented by the indicator vector $\mathbf{w}_j = (w_{j1}, \ldots, w_{jL}) \sim$ Multinomial$(1; \boldsymbol{\rho})$, where $w_{jl} = 1$ if column j belongs to column-cluster l and $w_{jl} = 0$ otherwise, and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_L)$. It is assumed that each data point x_{ij} is independent once the $\mathbf{z}_i$ and $\mathbf{w}_j$ are fixed. If, in addition, all $\mathbf{z}_i$ and $\mathbf{w}_j$ are assumed independent, and the latent block model is utilized in the same manner as Nadif and Govaert (2010), then the joint density of $\mathbf{x}$ becomes $f(\mathbf{x}; \boldsymbol{\vartheta}) = \sum_{\mathbf{z}\in\mathcal{Z}} \sum_{\mathbf{w}\in\mathcal{W}} p(\mathbf{z}; \boldsymbol{\pi}) p(\mathbf{w}; \boldsymbol{\rho}) f(\mathbf{x}|\mathbf{z}, \mathbf{w}; \boldsymbol{\theta})$, where

$$p(\mathbf{z}; \boldsymbol{\pi}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \pi_g^{z_{ig}}, \qquad p(\mathbf{w}; \boldsymbol{\rho}) = \prod_{j=1}^{p} \prod_{l=1}^{L} {\rho_l}^{w_{jl}}, \qquad \text{and}$$

$$f(\mathbf{x}|\mathbf{z}, \mathbf{w}^{\mu}, \mathbf{w}^{\Sigma}; \boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \prod_{j=1}^{d} \prod_{l=1}^{L} \left[\frac{1}{\sqrt{2\pi}\sigma_{gl}} \exp\left\{ -\frac{1}{2\sigma_{gl}^2}(x_{ij} - \mu_{gl})^2 \right\} \right]^{z_{ig} w_{jl}},$$

where μ_{gl} and σ_{gl}^2 are the mean and variance, respectively, for row-cluster g and column-cluster l, $\boldsymbol{\theta}$ is the set of all μ_{gl} and σ_{gl}^2, and $\boldsymbol{\vartheta} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\theta})$. The total number of free parameters in this traditional co-clustering model is

$$\#\text{Params}_{\text{trad coclust}} = G + L + 2(GL - 1). \tag{10.1}$$

Note that (10.1) does not depend on the dimension, making it a very parsimonious model. Moreover, co-clustering is still possible to perform when $p > n$.

There are two different ways that one can view co-clustering. The first is that the main goal is the clustering of rows, and the clustering of columns is solely a way to solve the problem of dimensionality. However, in certain applications, the clustering of the columns might also be of interest.

Although co-clustering has advantages over other high dimensional techniques (especially in the number of free parameters), the model is fairly restrictive because all observations in a block are realizations of independent and identically distributed Gaussian random variables with mean μ_{gl} and variance σ^2_{gl}. More flexibility is obtained by fitting more column-clusters and row-clusters, which is not always possible or advisable. What we propose in the present work is a parameter-wise co-clustering method by clustering columns according to both means and variances. This is the reason why we adopt hereafter the denomination "parameter-wise" co-clustering, which is now presented in detail.

10.2 Parameter-Wise Gaussian Co-Clustering

10.2.1 Model to Combine Two Latent Variables in Columns

Recall that traditional co-clustering aims to cluster data such that observations in the same block have the same distribution. An extension of traditional co-clustering for data treated as realizations of a Gaussian random variable is now considered. Similar to traditional co-clustering, there is a partition in rows and columns. However, now there are two partitions in the columns; specifically, a partition with respect to means and a partition with respect to variances.

Recall also that the data, which are treated as realizations of a continuous random variable, are represented as an $n \times p$ matrix, $\mathbf{x} = (x_{ij})_{1 \leq i \leq n, 1 \leq j \leq p}$. The partition in rows is again represented by $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)$.

Two Partitions in Columns The partition in columns by means is represented by $\mathbf{w}^{\mu} = (\mathbf{w}_1^{\mu}, \mathbf{w}_2^{\mu}, \ldots, \mathbf{w}_p^{\mu})$, where

$$\mathbf{w}_j^{\mu} = (w_{j1}^{\mu}, w_{j2}^{\mu}, \ldots, w_{jL^{\mu}}^{\mu}) \sim \text{Multinomial}(1; \boldsymbol{\rho}^{\mu})$$

with $\boldsymbol{\rho}^{\mu} = (\rho_1^{\mu}, \rho_2^{\mu}, \ldots, \rho_{L^{\mu}}^{\mu})$ and the partition in columns by variances is denoted by $\mathbf{w}^{\Sigma} = (\mathbf{w}_1^{\Sigma}, \mathbf{w}_2^{\Sigma}, \ldots, \mathbf{w}_p^{\Sigma})$, where

$$\mathbf{w}_j^{\Sigma} = (w_{j1}^{\Sigma}, w_{j2}^{\Sigma}, \ldots, w_{jL^{\Sigma}}^{\Sigma}) \sim \text{Multinomial}(1; \boldsymbol{\rho}^{\Sigma})$$

with $\boldsymbol{\rho}^{\Sigma} = (\rho_1^{\Sigma}, \rho_2^{\Sigma}, \ldots, \rho_{L^{\Sigma}}^{\Sigma})$. These two partitions in the columns is where the main novelty lies. Note that G, L^{μ} and L^{Σ} are the number of row-clusters, column-clusters by means, and column-clusters by variances, respectively.

Log-Likelihood Using a simple extension of the latent block model the observed log-likelihood is then

$$f(\mathbf{x}; \boldsymbol{\vartheta}) = \sum_{\mathbf{z} \in \mathcal{Z}} \sum_{\mathbf{w}^{\mu} \in \mathcal{W}^{\mu}} \sum_{\mathbf{w}^{\Sigma} \in \mathcal{W}^{\Sigma}} p(\mathbf{z}; \boldsymbol{\pi}) p(\mathbf{w}^{\mu}; \boldsymbol{\rho}^{\mu}) p(\mathbf{w}^{\Sigma}; \boldsymbol{\rho}^{\Sigma}) f(\mathbf{x} | \mathbf{z}, \mathbf{w}^{\mu}, \mathbf{w}^{\Sigma}; \boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where

$$p(\mathbf{z}; \boldsymbol{\pi}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \pi_g^{z_{ig}}, \quad p(\mathbf{w}^{\mu}; \boldsymbol{\rho}^{\mu}) = \prod_{j=1}^{p} \prod_{l^{\mu}=1}^{L^{\mu}} \left(\rho_{l^{\mu}}^{\mu}\right)^{w_{jl^{\mu}}^{\mu}}, \quad p(\mathbf{w}^{\Sigma}; \boldsymbol{\rho}^{\Sigma}) = \prod_{j=1}^{p} \prod_{l^{\Sigma}=1}^{L^{\Sigma}} \left(\rho_{l^{\Sigma}}^{\Sigma}\right)^{w_{jl^{\Sigma}}^{\Sigma}},$$

and

$$f(\mathbf{x} | \mathbf{z}, \mathbf{w}^{\mu}, \mathbf{w}^{\Sigma}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \prod_{j=1}^{p} \prod_{l^{\mu}=1}^{L^{\mu}} \prod_{l^{\Sigma}=1}^{L^{\Sigma}} \left[\frac{1}{\sqrt{2\pi}\sigma_{gl^{\Sigma}}} \exp\left\{ -\frac{1}{2\sigma_{gl^{\Sigma}}^2} (x_{ij} - \mu_{gl^{\mu}})^2 \right\} \right]^{z_{ig} w_{jl^{\mu}}^{\mu} w_{jl^{\Sigma}}^{\Sigma}}.$$

In terms of notation, $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_G)$, where $\boldsymbol{\mu}_g = (\mu_{g1}, \mu_{g2}, \ldots, \mu_{gL^\mu})$. Note that μ_{gl^μ} is the mean for row-cluster g and column-cluster by means l^μ. Likewise, $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \ldots, \boldsymbol{\Sigma}_G)$, where $\boldsymbol{\Sigma}_g = (\sigma^2_{g1}, \sigma^2_{g2}, \ldots, \sigma^2_{gL^\Sigma})$ and $\sigma^2_{gl^\Sigma}$ is the variance for row-cluster g and column-cluster by variances l^Σ. Finally, the complete-data log-likelihood is

$$\begin{aligned} p(\mathbf{x}, \mathbf{z}, \mathbf{w}^\mu, \mathbf{w}^\Sigma; \boldsymbol{\vartheta}) = C + \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig} \log \pi_g + \sum_{j=1}^{p}\sum_{l^\mu=1}^{L^\mu} w^\mu_{jl^\mu} \log \rho^\mu_{l^\mu} + \sum_{j=1}^{p}\sum_{l^\Sigma=1}^{L^\Sigma} w^\Sigma_{jl^\Sigma} \log \rho^\Sigma_{l^\Sigma} \\ - \frac{1}{2}\sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{j=1}^{p}\sum_{l^\mu=1}^{L^\mu}\sum_{l^\Sigma=1}^{L^\Sigma} z_{ig} w^\mu_{jl^\mu} w^\Sigma_{jl^\Sigma} \left[\log \sigma^2_{gl^\Sigma} + \frac{(x_{ij} - \mu_{gl^\mu})^2}{\sigma^2_{gl^\Sigma}}\right], \end{aligned}$$

where C is a constant with respect to the parameters and $\boldsymbol{\vartheta} = (\boldsymbol{\pi}, \boldsymbol{\rho}^\mu, \boldsymbol{\rho}^\Sigma, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. From this point on, we refer to this model as parameter-wise co-clustering.

Number of Free Parameters The number of free parameters in the parameter-wise co-clustering model is

$$\begin{aligned} \#\text{Params}_{\text{new coclust}} &= G - 1 + L^\mu - 1 + L^\Sigma - 1 + GL^\mu + GL^\Sigma \\ &= G + (L^\mu + L^\Sigma)(G + 1) - 3. \end{aligned}$$

There are a few comparisons with traditional co-clustering that are now discussed. First, similar to traditional co-clustering, the number of free parameters for the proposed parameter-wise method is independent of the dimension, meaning a high degree of parsimony is still maintained. Before mentioning the second point, note that the column-clusters by means and column-clusters by variances can be combined. For example, columns in column-cluster 1 by means and column-cluster 1 by variances can be combined to form one column-cluster. In general, columns in column-cluster l^μ by

means and column-cluster l^{Σ} by variances can be combined to form one column-cluster for any combination of l^{μ} and l^{Σ}, leading to a maximum of $L^{\mu}L^{\Sigma}$ column-clusters. There can, however, be fewer than $L^{\mu}L^{\Sigma}$ combined column-clusters because it is possible, for example, that no columns are clustered into column-cluster 3 by means and column-cluster 2 by variances. Now, assuming G is equal for both parameter-wise and traditional co-clustering, and $L^{\mu} = L^{\Sigma} = L$, then there are only an additional $L-1$ free parameters when using the parameter-wise model. Although there are these additional free parameters, there is the possibility of L^2 combined column-clusters, allowing for a finer partition of the columns and increased flexibility.

There is also the possibility that the parameter-wise model has fewer free parameters than traditional co-clustering while still maintaining similar flexibility. For example, if traditional co-clustering is considered with $G = 4$ and $L = 5$, then the total number of free parameters is 47. In the parameter-wise case, if $G = 4$, $L^{\mu} = 3$, $L^{\Sigma} = 3$, then the total number of free parameters is 31. In this case, there is a possibility of a total of nine column-clusters compared to five column-clusters when using traditional co-clustering.

10.2.2 Parameter Estimation Using the SEM Gibbs Algorithm

The SEM algorithm after initialization at iteration q proceeds as follows.

SE Step: Generate the row partition $\mathbf{z}^{(q+1)}$ according to

$$P(z_{ig} = 1|\mathbf{x}, \mathbf{w}^{\mu(q)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}^{(q)}, \boldsymbol{\Sigma}^{(q)}, \boldsymbol{\pi}^{(q)}) = \frac{\pi_g^{(q)} f(\mathbf{x}_i|\mathbf{w}^{\mu(q)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}_g^{(q)}, \boldsymbol{\Sigma}_g^{(q)})}{\sum_{g'}^{G} \pi_{g'}^{(q)} f(\mathbf{x}_i|\mathbf{w}^{\mu(q)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}_{g'}^{(q)}, \boldsymbol{\Sigma}_{g'}^{(q)})},$$

where

$$f(\mathbf{x}_i|\mathbf{w}^{\mu(q)},\mathbf{w}^{\Sigma(q)};\boldsymbol{\mu}_g^{(q)},\boldsymbol{\Sigma}_g^{(q)}) = \prod_{j=1}^{p}\prod_{l^\mu=1}^{L^\mu}\prod_{l^\Sigma=1}^{L^\Sigma}\left[\frac{1}{\sqrt{2\pi}\sigma_{gl^\Sigma}^{(q)}}\exp\left\{-\frac{1}{2\sigma_{gl^\Sigma}^{2(q)}}(x_{ij}-\mu_{gl^\mu}^{(q)})^2\right\}\right]^{w_{jl^\mu}^{\mu(q)}w_{jl^\Sigma}^{\Sigma(q)}}.$$

Generate the column partition by means $\mathbf{w}^{\mu(q+1)}$ according to

$$P(w_{jl^\mu}^{\mu}=1|\mathbf{x},\mathbf{z}^{(q+1)},\mathbf{w}^{\Sigma(q)};\boldsymbol{\mu}^{(q)},\boldsymbol{\Sigma}^{(q)},\boldsymbol{\rho}^{\mu(q)}) = \frac{\rho_{l^\mu}^{\mu(q)}f(\mathbf{x}_{\cdot j}|\mathbf{z}^{(q+1)},\mathbf{w}^{\Sigma(q)};\boldsymbol{\mu}_{l^\mu}^{(q)},\boldsymbol{\Sigma}^{(q)})}{\sum_{l^{\mu\prime}}^{L^\mu}\rho_{l^{\mu\prime}}^{\mu(q)}f(\mathbf{x}_{\cdot j}|\mathbf{z}^{(q+1)},\mathbf{w}^{\Sigma(q)};\boldsymbol{\mu}_{l^{\mu\prime}}^{(q)},\boldsymbol{\Sigma}^{(q)})},$$

where $\mathbf{x}_{\cdot j}=(x_{1j},x_{2j},\ldots,x_{nj})$, $\boldsymbol{\mu}_{l^\mu}^{(q)}=(\mu_{1l^\mu}^{(q)},\mu_{2l^\mu}^{(q)},\ldots,\mu_{Gl^\mu}^{(q)})$, and

$$f(\mathbf{x}_{\cdot j}|\mathbf{z}^{(q+1)},\mathbf{w}^{\Sigma(q)};\boldsymbol{\mu}_{l^\mu}^{(q)},\boldsymbol{\Sigma}^{(q)}) = \prod_{i=1}^{n}\prod_{g=1}^{G}\prod_{l^\Sigma=1}^{L^\Sigma}\left[\frac{1}{\sqrt{2\pi}\sigma_{gl^\Sigma}^{(q)}}\exp\left\{-\frac{1}{2\sigma_{gl^\Sigma}^{2(q)}}(x_{ij}-\mu_{gl^\mu}^{(q)})^2\right\}\right]^{z_{ig}^{(q+1)}w_{jl^\Sigma}^{\Sigma(q)}}.$$

Generate the column partition by variances $\mathbf{w}^{\Sigma(q+1)}$ according to

$$P(w_{jl^\Sigma}^{\Sigma}=1|\mathbf{x},\mathbf{z}^{(q+1)},\mathbf{w}^{\mu(q+1)};\boldsymbol{\mu}^{(q)},\boldsymbol{\Sigma}^{(q)},\boldsymbol{\rho}^{\Sigma(q)}) = \frac{\rho_{l^\Sigma}^{\Sigma(q)}f(\mathbf{x}_{\cdot j}|\mathbf{z}^{(q+1)},\mathbf{w}^{\mu(q+1)};\boldsymbol{\mu}^{(q)},\boldsymbol{\Sigma}_{l^\Sigma}^{(q)})}{\sum_{l^{\Sigma\prime}}^{L^\Sigma}\rho_{l^{\Sigma\prime}}^{\Sigma(q)}f(\mathbf{x}_{\cdot j}|\mathbf{z}^{(q+1)},\mathbf{w}^{\mu(q+1)};\boldsymbol{\mu}^{(q)},\boldsymbol{\Sigma}_{l^{\Sigma\prime}}^{(q)})},$$

where $\boldsymbol{\Sigma}_{l^\Sigma}^{(q)}=(\sigma_{1l^\Sigma}^{2(q)},\sigma_{2l^\Sigma}^{2(q)},\ldots,\sigma_{Gl^\Sigma}^{2(q)})$ and

$$f(\mathbf{x}_{\cdot j}|\mathbf{z}^{(q+1)},\mathbf{w}^{\mu(q+1)};\boldsymbol{\mu}^{(q)},\boldsymbol{\Sigma}_{l^\Sigma}^{(q)}) = \prod_{i=1}^{n}\prod_{g=1}^{G}\prod_{l^\mu=1}^{L^\mu}\left[\frac{1}{\sqrt{2\pi}\sigma_{gl^\Sigma}^{(q)}}\exp\left\{-\frac{1}{2\sigma_{gl^\Sigma}^{2(q)}}(x_{ij}-\mu_{gl^\mu}^{(q)})^2\right\}\right]^{z_{ig}^{(q+1)}w_{jl^\mu}^{\mu(q+1)}}.$$

M Step: Update the parameters according to

$$\pi_g^{(q+1)} = \frac{\sum_{i=1}^{n} z_{ig}^{(q+1)}}{n}, \qquad \rho_{l^\mu}^{\mu\ (q+1)} = \frac{\sum_{j=1}^{p} w_{jl^\mu}^{\mu\ (q+1)}}{p}, \qquad \rho_{l^\Sigma}^{\Sigma\ (q+1)} = \frac{\sum_{j=1}^{p} w_{jl^\Sigma}^{\Sigma\ (q+1)}}{p},$$

$$\mu_{gl^\mu}^{(q+1)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{l^\Sigma=1}^{L^\Sigma} z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)} w_{jl^\Sigma}^{\Sigma\ (q+1)} x_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{l^\Sigma=1}^{L^\Sigma} z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)} w_{jl^\Sigma}^{\Sigma\ (q+1)}} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{p} z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)} x_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{p} z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)}},$$

$$\sigma_{gl^\Sigma}^{2\ (q+1)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{l^\mu=1}^{L^\mu} z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)} w_{jl^\Sigma}^{\Sigma\ (q+1)} (x_{ij} - \mu_{gl^\mu}^{(q+1)})^2}{\sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{l^\mu=1}^{L^\mu} z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)} w_{jl^\Sigma}^{\Sigma\ (q+1)}}.$$

After a burn-in period of the algorithm, the estimates of each of the parameters are just the mean of the runs of the SEM algorithm (the number of runs are assessed experimentally in Section 4). We denote these final estimates by $\hat{\boldsymbol{\vartheta}} = (\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\rho}}^{\hat{\mu}}, \hat{\boldsymbol{\rho}}^{\hat{\Sigma}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$. For the final partition of rows, columns by means, and columns by variances, we fix the parameters at their estimates and run more iterations of the SE step. We then assign each row to the row-cluster to which it is assigned most often over these additional SE steps. Likewise, each column is assigned to the column-cluster by means to which it is assigned most often over the additional SE steps, and finally each column is assigned to the column-cluster by variances to which it is assigned most often over the additional SE iterations. For our simulations and real data analyses, we take 20 such runs to obtain the final partitions $\hat{\mathbf{z}}, \hat{\mathbf{w}^{\mu}}$, and $\hat{\mathbf{w}^{\Sigma}}$.

10.2.3 Model Selection

ICL–BIC As is the case in any clustering scenario, the number of row-clusters, column-clusters by means, and column-clusters by variances are not known *a priori*

and, therefore, a model selection criterion is required. Similar to traditional co-clustering, the observed log-likelihood is intractable and so the BIC cannot be used. Therefore, we propose using the integrated complete log-likelihood (ICL; Biernacki *et al.*, 2000), which relies on the complete data log-likelihood instead of the observed log-likelihood. This criterion is called the ICL–BIC, similar to that used by Jacques and Biernacki (2018) and is given by

$$\text{ICL–BIC} = p(\mathbf{x}, \hat{\mathbf{z}}, \hat{\mathbf{w}}^{\mu}, \hat{\mathbf{w}^{\Sigma}}; \hat{\boldsymbol{\vartheta}}) - \frac{G-1}{2}\log n - \frac{L^{\mu}+L^{\Sigma}-2}{2}\log p - \frac{G(L^{\mu}+L^{\Sigma})}{2}\log np.$$

From the property proven by Brault *et al.* (2017), the BIC and ICL–BIC exhibit the same behaviour for large values of n and/or p, thus the number of blocks chosen by this criterion is consistent (under some conditions not mentioned here). The model with the largest ICL–BIC is retained.

Search Algorithm Because an extra layer of complexity is introduced with the parameter-wise model by considering two column partitions, it may take a very long time to perform an exhaustive search of all possible combinations of G, L^{μ} and L^{Σ} in a pre-defined range. This has been discussed in the literature, specifically by Robert (2017), and a non-exhaustive search algorithm for the parameter-wise model is now presented. Specifically, the algorithm begins with the parameters $(G, L^{\mu}, L^{\Sigma}) = (G_1, L_1^{\mu}, L_1^{\Sigma})$. Three models with parameters $(G_1+1, L^{\mu}, L^{\Sigma})$, $(G_1, L^{\mu}+1, L^{\Sigma})$ and $(G_1, L^{\mu}, L^{\Sigma}+1)$ are then fit. The set with the highest ICL–BIC is retained and we obtain the set $(G_2, L_2^{\mu}, L_2^{\Sigma})$. The procedure is then repeated until a maximum threshold is reached for these parameters or the ICL–BIC no longer

increases. Although not as pertinent for traditional co-clustering, a similar non-exhaustive search algorithm can be used for traditional co-clustering.

10.3 Numerical Experiments on Artificial Data

10.3.1 Algorithm and Parameter Estimation Evaluation

Two different simulations are performed to evaluate the algorithm, parameter estimation, and classification performance.

Simulation 10.1

50 datasets are simulated according to the following parameters. $n = 1000$, $p = 100$, $G = 3$, $L^{\mu} = 2$, $L^{\Sigma} = 3$,

$$\boldsymbol{\mu} = \begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & -3 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.75 \\ 2 & 1.75 & 0.25 \\ 1.5 & 2.25 & 2.5 \end{pmatrix},$$

and mixing proportions

$$\boldsymbol{\pi} = (0.3, 0.3, 0.4), \qquad \boldsymbol{\rho}^{\mu} = (0.4, 0.6), \qquad \boldsymbol{\rho}^{\Sigma} = (0.3, 0.3, 0.4).$$

To clarify notation, the cell gl^{μ} in the matrix μ corresponds to the mean of an observation from row-cluster g and column-cluster by means l^{μ}, i.e., $\mu_{gl^{\mu}}$. Likewise, the cell gl^{Σ} in the matrix $\boldsymbol{\Sigma}$ corresponds to the variance of an observation from row-cluster g and column-cluster by variances l^{Σ}, i.e., $\sigma^2_{gl^{\Sigma}}$.

A burn-in of 20 iterations for the SEM-Gibbs algorithm is used, followed by 100 iterations, followed by 20 iterations of the SE-step to obtain the final partitions.

The error in the mean estimates is calculated using

$$\Delta\boldsymbol{\mu} = \sum_{g,l^{\mu}} |\hat{\mu}_{gl^{\mu}} - \mu_{gl^{\mu}}|.$$

The errors for the other parameters are calculated in a similar fashion and are denoted by $\Delta\boldsymbol{\Sigma}$, $\Delta\boldsymbol{\pi}$, $\Delta\boldsymbol{\rho}^{\mu}$ and $\Delta\boldsymbol{\rho}^{\Sigma}$, respectively. The averaged errors (and their standard deviations) over the 50 datasets are shown in Table 10.1. The average errors are low for all variables indicating good parameter recovery.

Table 10.2 displays the average ARI, with standard deviations, for the row, column by means, and column by variances partitions over the 50 simulated datasets. Notice that the classification is perfect for both partitions by columns for all simulated datasets. Moreover, the average ARI for the rows is very high.

Table 10.1: Average error (and standard deviation) of the parameter estimates over the 50 datasets for Simulation 10.1.

$\overline{\Delta\boldsymbol{\mu}}$	$\overline{\Delta\boldsymbol{\Sigma}}$	$\overline{\Delta\boldsymbol{\pi}}$	$\overline{\Delta\boldsymbol{\rho}^{\mu}}$	$\overline{\Delta\boldsymbol{\rho}^{\Sigma}}$
0.14 (0.70)	0.24 (0.75)	0.012 (0.082)	1.44e-15 (5.61e-16)	1.33e-15 (4.59e-16)

Table 10.2: Average ARI (and standard deviation) for the row ($\overline{\text{ARI}}_r$), column by means ($\overline{\text{ARI}}_{c\mu}$), and column by variances ($\overline{\text{ARI}}_{c\Sigma}$) partitions over the 50 datasets for Simulation 10.1.

$\overline{\text{ARI}}_r$	$\overline{\text{ARI}}_{c\mu}$	$\overline{\text{ARI}}_{c\Sigma}$
0.99 (0.068)	1.00 (0.00)	1.00 (0.00)

In Figure 10.1, the progression of the parameter estimates over the course of the SEM-Gibbs algorithm is shown for one of the datasets (the other datasets exhibit

similar behaviour). From these plots, it is clear that a burn-in of 20 iterations is sufficient to obtain a stable chain.

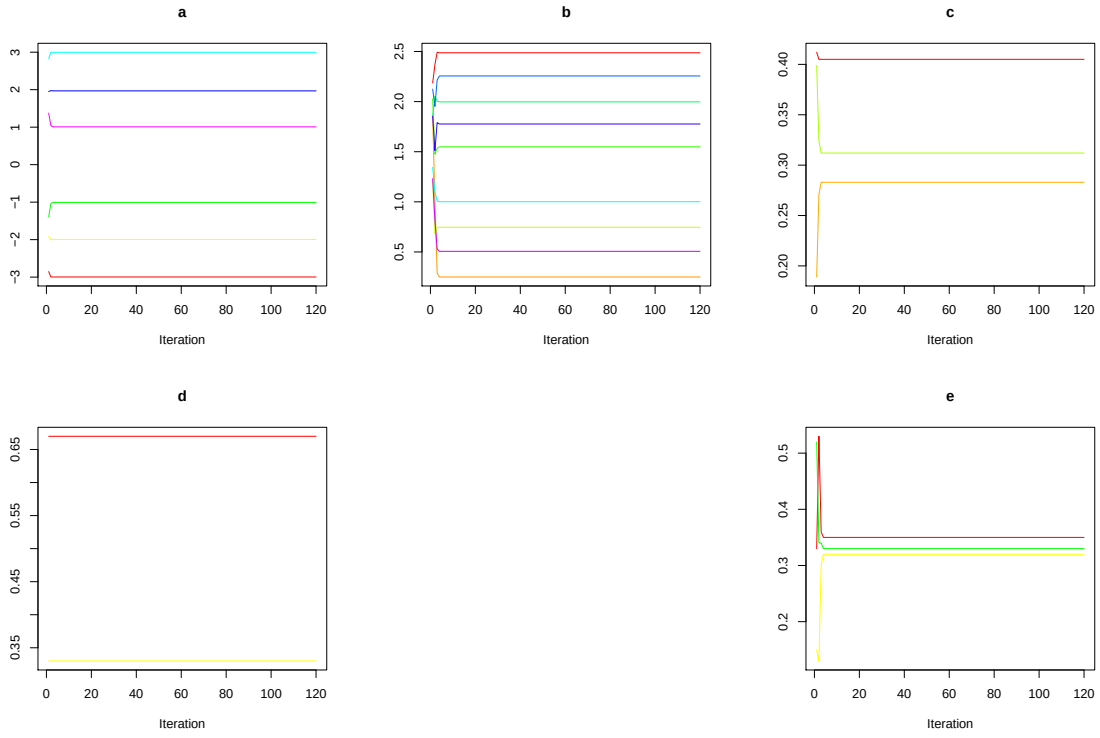


Figure 10.1: SEM algorithm parameter estimation progression for one dataset for (a) the mean parameters $\mu_{gl^{\mu}}$, (b) the variance parameters $\sigma^2_{gl^{\Sigma}}$, (c) the row mixing proportions π_g, (d) the column by means mixing proportions $\rho^{\mu}_{l^{\mu}}$, and (e) the column by variances mixing proportions $\rho^{\Sigma}_{l^{\Sigma}}$ for Simulation 10.1.

Finally, in Figure 10.2, the co-clustering results for one of the 50 datasets is displayed. Note, in this case, the estimated co-clustering result is the same as the true co-clustering solution. In the top left panel, a heatmap of the original data is displayed. In the co-clustering by means panel (bottom left), the co-clustering results for the row-clusters and the column-clusters by means is shown. The co-clustering by variances panel (bottom right) shows the co-clustering results for the row-clusters and the column-clusters by variances. Finally, the combined co-clustering (top right)

displays the co-clustering solution with all combined column-clusters. Specifically, going from left to right, the first combined column-cluster consists of the columns partitioned into column-cluster 1 for the means and column-cluster 1 for the variances, the second combined column-cluster are the columns clustered into column-cluster 2 for the means and column-cluster 1 for the variances and so on. Combining the column-clusters by means and variances in this manner results in a maximum of $L^{\mu}L^{\Sigma}$ combined column-clusters (as is the case here) thus allowing more flexibility. It is important to note, however, that there may be cases, as we will see with the real dataset, when no columns are clustered into a particular pair l^{μ} and l^{Σ}, and thus the combined co-clustering result might have fewer than $L^{\mu}L^{\Sigma}$ combined column-clusters but never more.

Simulation 10.2

In Simulation 10.2, less separation between groups is considered. A total of 50 datasets are again considered with the parameters $n = 200$, $p = 500$, $G = 3$, $L^{\mu} = 3$, $L^{\Sigma} = 2$,

$$\boldsymbol{\mu} = \begin{pmatrix} 1 & 1.25 & 0 \\ 2 & 1.2 & 1 \\ 1.5 & 1.9 & 0.5 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 \\ 2 & 1.75 \\ 1.5 & 2.25 \end{pmatrix},$$

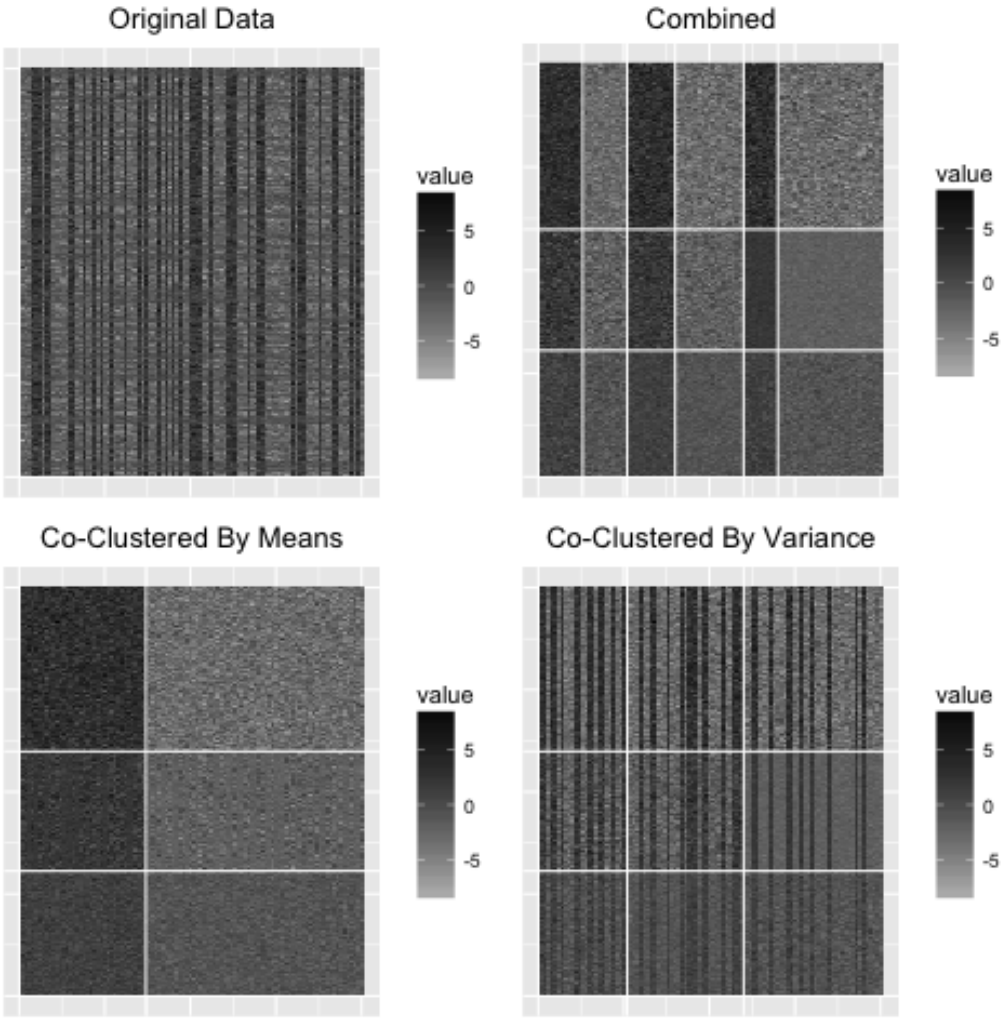


Figure 10.2: Estimated co-clustering solution for one of the fifty datasets from Simulation 10.1.

and the mixing proportions

$$\boldsymbol{\pi} = (0.3, 0.3, 0.4), \qquad \boldsymbol{\rho}^{\mu} = (0.3, 0.5, 0.2), \qquad \boldsymbol{\rho}^{\Sigma} = (0.4, 0.6).$$

Table 10.3 shows the average error of the estimates over the 50 datasets, and the average ARI values over the 50 datasets for each partition are shown in Table 10.4.

Again, we obtain very good classification performance for all three partitions. The progression of the parameter estimates is shown in Figure 10.3. Similar to Simulation 10.1, a burn-in period of 20 iterations is still sufficient to obtain a stable chain. Finally, Figure 10.4 displays the co-clustering solutions for one of the 50 datasets. Unlike in the first simulation, there is very little spatial separation between blocks.

Table 10.3: Average error (and standard deviation) of the estimates over the 50 datasets for Simulation 10.2.

$\overline{\Delta\boldsymbol{\mu}}$	$\overline{\Delta\boldsymbol{\Sigma}}$	$\overline{\Delta\boldsymbol{\pi}}$	$\overline{\Delta\boldsymbol{\rho}^{\mu}}$	$\overline{\Delta\boldsymbol{\rho}^{\Sigma}}$
0.15 (0.50)	0.085 (0.046)	1.29e-15 (3.91e-16)	0.015 (0.088)	0.0079 (0.0054)

Table 10.4: Average ARI (and standard deviation) for the row ($\overline{\text{ARI}}_r$), column by means ($\overline{\text{ARI}}_{c\mu}$), and column by variances ($\overline{\text{ARI}}_{c\Sigma}$) partitions over the 50 datasets for Simulation 10.2.

$\overline{\text{ARI}}_r$	$\overline{\text{ARI}}_{c\mu}$	$\overline{\text{ARI}}_{c\Sigma}$
1.00 (0.00)	0.98 (0.080)	0.96 (0.018)

10.3.2 Simulation 10.3

In this simulation, the performance of the ICL–BIC selection criterion is considered. Again, 50 datasets are simulated with $n = 2000$, $p = 500$, $G = L^{\mu} = L^{\Sigma} = 3$,

$$\boldsymbol{\mu} = \begin{pmatrix} 1 & 1.25 & 0 \\ 2 & 1.2 & 1 \\ 1.5 & 1.9 & 0.5 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.25 \\ 2 & 1.75 & 0.5 \\ 1.5 & 2.25 & 1 \end{pmatrix},$$

and mixing proportions

$$\boldsymbol{\pi} = (0.3, 0.3, 0.4), \qquad \boldsymbol{\rho}^{\mu} = (0.3, 0.4, 0.3), \qquad \boldsymbol{\rho}^{\Sigma} = (0.4, 0.3, 0.3).$$

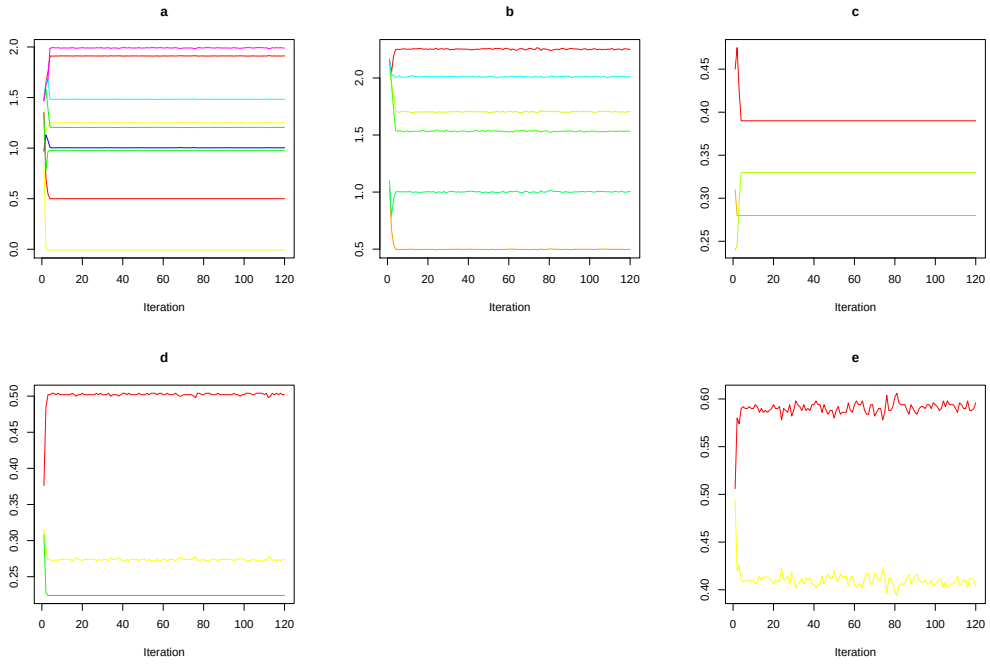


Figure 10.3: Simulation 10.2 SEM algorithm parameter estimation progression for one dataset for (a) the mean parameters $\mu_{gl^{\mu}}$, (b) the variance parameters $\sigma^2_{gl^{\Sigma}}$, (c) the row mixing proportions π_g, (d) the column by means mixing proportions $\rho^{\mu}_{l^{\mu}}$, and (e) the column by variances mixing proportions $\rho^{\Sigma}_{l^{\Sigma}}$.

An exhaustive search is performed considering each of combination of $G, L^{\mu}, L^{\Sigma} \in \{2, 3, 4\}$. In Table 10.5, the number of times each value of G, L^{μ} and L^{Σ} is chosen by the ICL–BIC is displayed. For the vast majority of the datasets, the correct model is chosen by the ICL–BIC.

Table 10.5: Frequency of the number of row-clusters, column-clusters by means, and column-clusters by variances chosen by the ICL–BIC over the 50 simulated datasets when using the exhaustive search in Simulation 10.3.

	2	3	4
G	0	49	1
L^{μ}	0	48	2
L^{Σ}	0	48	2

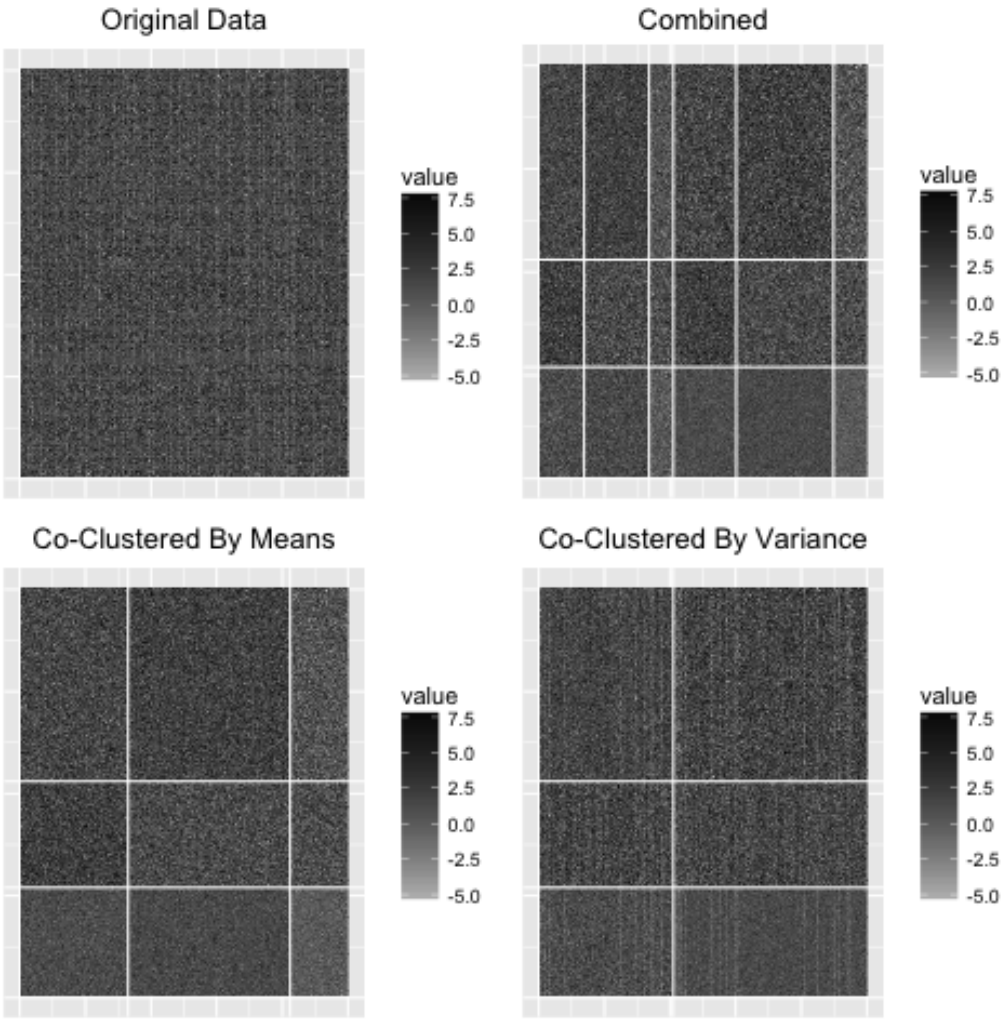


Figure 10.4: Estimated co-clustering solution for one of the fifty datasets from Simulation 10.2.

10.3.3 Simulation 10.4

In the last simulation, the performance of the non-exhaustive search algorithm described in Section 3.3 is addressed. In all, 25 datasets are simulated according to the parameters $n = 100, p = 200, G = L^{\Sigma} = 3, L^{\mu} = 4$,

$$\boldsymbol{\mu} = \begin{pmatrix} 1 & -0.25 & 0.3 & -1 \\ 1.25 & 0 & 0.1 & -0.3 \\ 0.5 & -1 & 0 & 0.1 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.25 \\ 2 & 1.75 & 0.5 \\ 1.5 & 2.25 & 1 \end{pmatrix},$$

and

$$\boldsymbol{\pi} = (0.3, 0.3, 0.4), \qquad \boldsymbol{\rho}^{\mu} = (0.2, 0.3, 0.25, 0.25), \qquad \boldsymbol{\rho}^{\Sigma} = (0.5, 0.25, 0.25).$$

The initial values are taken to be $(G_1, L_1^{\mu}, L_1^{\Sigma}) = (1, 1, 1)$ and the maximum values for all three are set to five. In Table 10.6, the number of times each value of G, L^{μ} and L^{Σ} is chosen by the ICL–BIC is shown. Notice that the procedure performs quite well for choosing the correct model.

Table 10.6: Frequency of the number of row-clusters, column-clusters by means, and column-clusters by variances chosen by the ICL–BIC over the 25 simulated datasets when using the non-exhaustive search method for Simulation 10.4.

	2	3	4
G	0	24	1
L^{μ}	0	0	25
L^{Σ}	1	24	0

10.4 Real Data Analyses

10.4.1 Comparing Parameter-Wise and Traditional Co-Clustering Under Similar Conditions

A subset of the Jester dataset used by Goldberg *et al.* (2001) is used to compare parameter-wise co-clustering and traditional co-clustering. The data consist of 100 jokes rated on a "continuous" scale from -10 to 10. A total of 7200 users rated all 100 jokes, and a random sample of 2000 of these users is considered herein.

The non-exhaustive search algorithm is performed for traditional co-clustering with the number of row-clusters ranging from one to 25 and the number of column-clusters ranging from one to seven. This results in choosing seven row-clusters and three column-clusters and the resultant ICL–BIC is -569487.0. With these values for G and L, the total number of free parameters is 50. In the next section, the non-exhaustive search algorithm is used for the proposed parameter-wise method; however, it is interesting to consider the performance of the parameter-wise method under similar conditions to the results obtained with traditional co-clustering. Specifically, the parameter-wise method is performed on this dataset with $G = 7, L^{\mu} = L^{\Sigma} = 3$. Under this model, the ICL–BIC is -569010.4, and the total number of free parameters is 52. Note that the ICL–BIC values for both traditional and parameter-wise co-clustering are quite similar, with a slightly higher value obtained when using parameter-wise co-clustering. In Figure 10.5, the original data (left panel) and the traditional co-clustering solution (right panel), are shown, and the co-clustering solutions for parameter-wise co-clustering are displayed (Figure 10.6) in the same format as the simulations. Notice that a total of seven combined column-clusters are obtained

when using parameter-wise co-clustering.

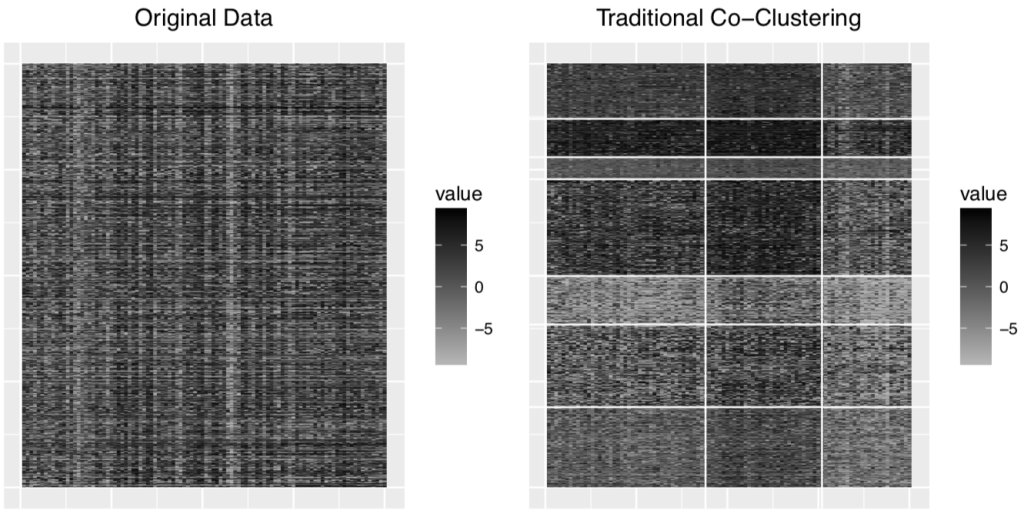


Figure 10.5: Traditional co-clustering results for the Jester data.

In Table 10.7, we show a classification table comparing the column-clusters by means and column-clusters by variances found using parameter-wise co-clustering and the column-clusters found using traditional co-clustering. There is almost perfect agreement between the column-clusters from traditional co-clustering and the column-clusters by means from parameter-wise co-clustering. This, however, is not true for the column-clusters by variances. This result is somewhat perceptible in the images of the co-clustering solutions. In Table 10.8, the classification table comparing row-clusters from traditional and parameter-wise co-clustering is displayed. It is clear that the row-clusters found by both of these methods are quite comparable — the ARI when comparing these two partitions is 0.86.

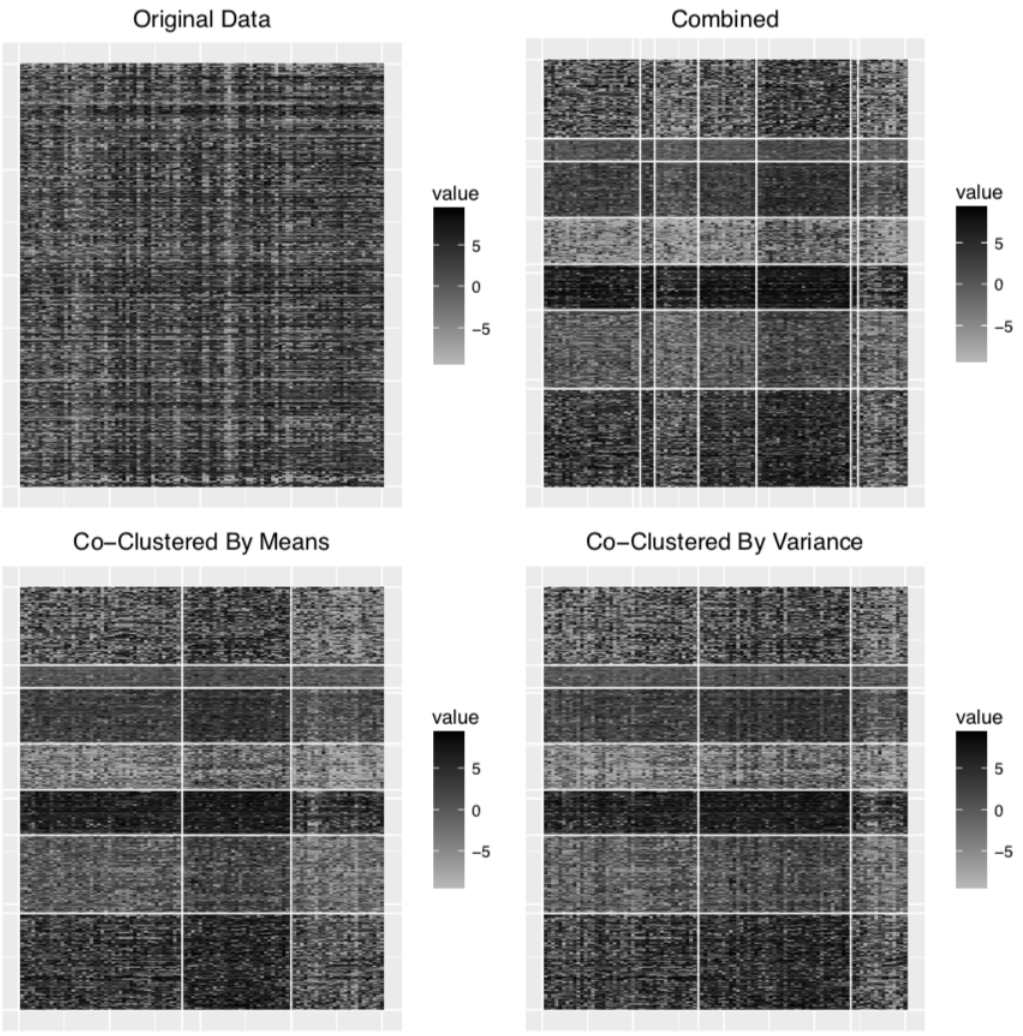


Figure 10.6: Parameter-wise co-clustering results for the Jester dataset under similar conditions to the traditional co-clustering solution.

10.4.2 Further Analysis with Parameter-Wise Co-Clustering

The non-exhaustive search algorithm is now performed for parameter-wise co-clustering. The range of values was one to 25 row-clusters, and one to seven column-clusters by means and column-clusters by variances resulting in the ICL–BIC choosing a model

Table 10.7: Classification table comparing the column-clusters by means and column-clusters by variances for parameter-wise co-clustering and column-clusters from traditional co-clustering for the Jester dataset.

	Means			Variances		
Traditional	1	2	3	1	2	3
1	43	0	1	28	14	2
2	2	30	0	4	28	0
3	0	0	24	11	0	13

Table 10.8: Classification table comparing row-clusters for parameter-wise and traditional co-clustering.

	Traditional						
Parameter-Wise	1	2	3	4	5	6	7
1	427	10	1	0	3	0	16
2	0	350	0	9	0	0	11
3	18	0	180	0	16	0	0
4	0	0	0	216	0	0	3
5	10	11	0	0	241	1	0
6	0	5	0	0	0	103	0
7	2	3	0	4	0	0	360

with 17 row-clusters, six column-clusters by means, and four column-clusters by variances. The resulting ICL–BIC is -561099.0 and a total of 15 combined column-clusters are obtained. Notice that there is significant improvement in the ICL–BIC in this case. In Figure 10.7, we show the parameter-wise co-clustering solution. Because more row-clusters are obtained, it is far more difficult to visualize the row-clusters. Moreover, the combined co-clustering solution is very difficult to interpret in this scenario, which displays the benefit of visualizing the column-clusters by means and column-clusters by variances separately.

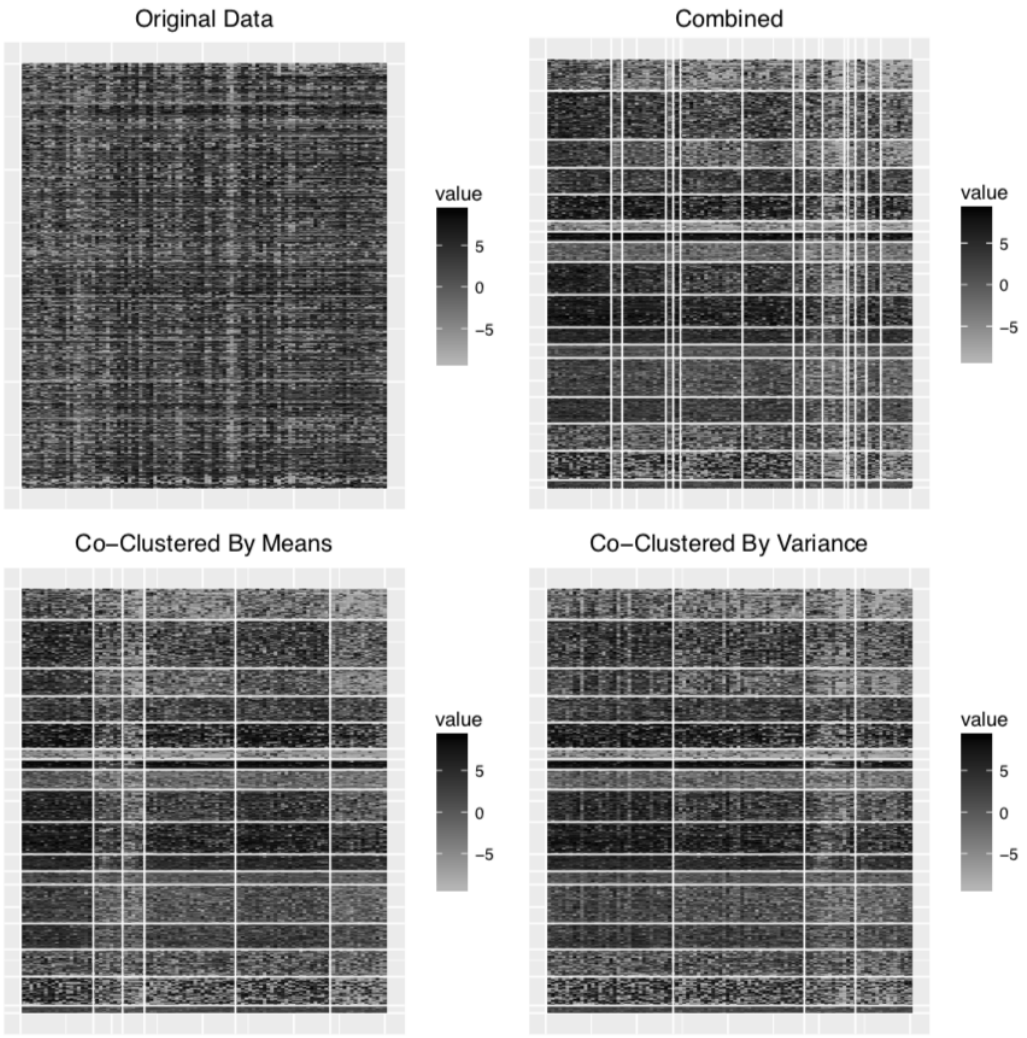


Figure 10.7: Parameter-wise co-clustering results for the Jester data after performing the non-exhaustive search algorithm.

Finally, the exhaustive search algorithm is performed for both traditional and parameter-wise co-clustering. For each value of $G \in \{1, 2, \ldots, 25\}$, the maximum ICL–BIC over all values of L for traditional co-clustering, and L^{μ} and L^{Σ} for parameter-wise co-clustering is considered. In Figure 10.8, we display a plot of this maximum ICL–BIC against G. For both traditional and parameter-wise co-clustering, the ICL–BIC begins to plateau around $G = 10$. Moreover, the ICL–BIC for parameter-wise

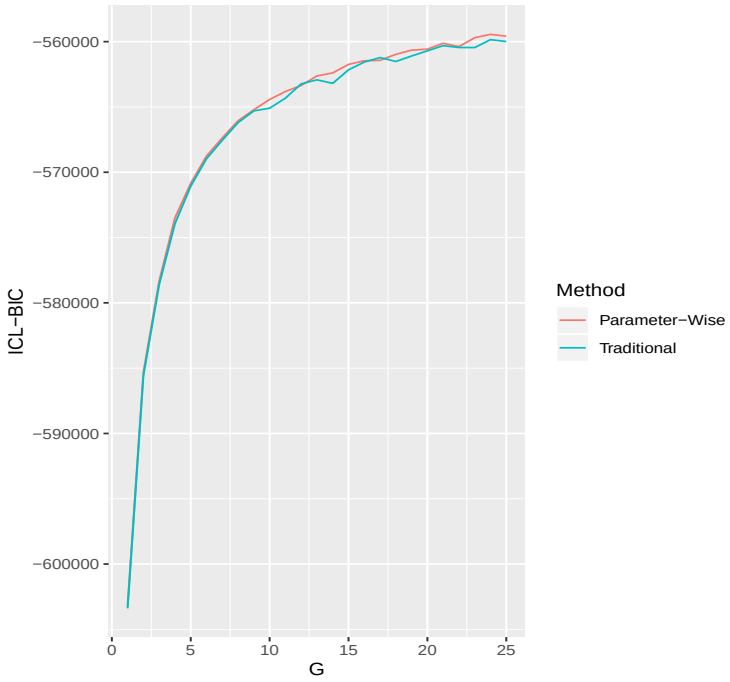


Figure 10.8: Maximum ICL–BIC over L for traditional co-clustering (turquoise), and L^{μ} and L^{Σ} for parameter-wise co-clustering (red) for each value of G, against G.

co-clustering is oftentimes, if only very slightly, higher than traditional co-clustering. Finally, we note that it is very computationally expensive to run the exhaustive search with parameter-wise co-clustering taking around 24 hours using 25 1200MHz cores running continuously.

10.5 Summary

A parameter-wise co-clustering algorithm was developed for high-dimensional data. This parameter-wise method allowed for two partitions of the columns based on both means and variances, as well as a combined co-clustering solution. This, in essence, provides more flexibility than traditional co-clustering, while maintaining the high

degree of parsimony inherent to traditional co-clustering. An SEM Gibbs algorithm was used for parameter estimation, and evaluated by two simulations. An ICL–BIC criterion, as well as a non-exhaustive search algorithm, were developed for model selection.

A subset of the Jester dataset was considered for comparison purposes between traditional and parameter-wise co-clustering. After applying traditional co-clustering to the data, parameter-wise co-clustering was performed using similar parameters, i.e., same G and $L^{\mu} = L^{\Sigma} = L$. This resulted in similar row-clusters between the two methods. Furthermore, the column-clusters by means using parameter-wise co-clustering were almost identical to the column-clusters from traditional co-clustering. This was not true, however, when comparing the column-clusters by variances and the column-clusters obtained from traditional co-clustering. Parameter-wise co-clustering also had a marginally higher ICL–BIC in this case. Using the non-exhaustive search algorithm for parameter-wise co-clustering resulted in far more row-clusters, and many more combined column-clusters, which displayed the utility of considering the co-clustering by means, and co-clustering by variances separately from the combined co-clustering solution.

Chapter 11

Conclusions

11.1 Discussion

In this thesis, multiple topics in the area of model-based clustering and classification were considered. The first topic, and main component of the thesis, was the development of model-based clustering and classification methodology for three way data. Specifically, four skewed matrix variate distributions were derived, and then used in the mixture model context for clustering and classification of three-way data. A matrix variate extension of the mixture of factor analyzers model was then developed for clustering high-dimensional three-way data using both the matrix variate normal distribution, and then the four skewed matrix variate distributions.

A second topic involved mixtures of first-order continuous time Markov chains for clustering clickstream data. This allowed for the amount of time spent in each category to be taken into consideration. This in turn allowed the detection of groups of users that the discrete time model was unable to detect.

The third topic, based on the MMVBFA model presented in Chapter 5, imposed

constraints on the scale and factor loading matrices to develop a family of 64 parsimonious models.

The fourth topic considered a detailed comparison between two different methods for clustering data with skewed components, namely using skewed component densities, or transformation methods. In general, the results of the two methods were very similar, but they did have a few differences. Most notably, for the crabs data, the skewed methods discriminated based on species, whereas the transformation methods discriminated based on sex.

Finally, a parameter-wise co-clustering model was developed for clustering high dimensional data. This allowed for more resultant column-clusters, and thus increasing the flexibility of the co-clustering model. Moreover, this additional flexibility is obtained with very few extra parameters, and in some cases fewer parameters than traditional co-clustering.

Possible future directions for some of these topics are now addressed.

11.2 Future Work

11.2.1 Three-Way Data Analysis

One problem that has yet to be addressed in the area of three-way data is mixed data in multivariate longitudinal analysis. In the case of multivariate longitudinal analysis, there are most likely variables that can not be considered realizations of continuous random variables and, in some cases, variables that will not change over time. For example, variables like gender, eye colour and race are variables that will be the same at all time points. Moreover, variables such as age or number of children might be

best treated as count variables. In these cases, the use of a continuous matrix variate distribution would not be advisable. One way forward in this regard, is in a similar manner to the multivariate case, where the variables (rows or columns of the data matrices) are partitioned into continuous, categorical and stationary components. Independence could then be used, as in the multivariate case, to greatly simplify the problem. This, however, would completely disregard any relationships between the different types of variables, and in many cases would be an unreasonable assumption. Therefore, methods similar to the multivariate case could be considered. This will in turn allow more complex three-way data to be analyzed in the future with applications in clinical studies, spatial temporal data, etc. Another aspect of three-way data not yet considered is unbalanced data. For example, it may be the case that a variable is only measured at certain time points, while others are measured at all time points. In this case, it is not entirely clear how one might approach this, and will be a consideration in this project.

A second future direction is in the area of multiway data analysis. Examples of such data types are black and white video clips, which can be represented as a third order tensor, coloured images which can also be represented as third order tensors, and finally coloured video clips which would be fourth order tensors. Work has already been completed in this area using the tensor variate normal distribution (Tait and McNicholas, 2019), but can also be extended to skewed tensor distributions. In addition, a tensor extension of the matrix variate bilinear factor analysis model can also be considered.

11.2.2 Clustering Clickstream Data

There are a few potential directions for future work for model based clustering of clickstream data. For example, a different distribution for the holding time could be considered. Although the classical approach is to use an exponential holding time in each state, this may not be realistic in some real applications. This issue with the exponential distribution is that the model allows for almost immediate or unrealistically long transition times. This leads us to another issue, i.e., that the continuous time model is time unit dependent, which also needs to be considered in a real application. Therefore, the of use a truncated exponential distribution for the holding time will be a topic of future work. Finally, although conceived in the context of clickstream data, this methodology could be used in other applications that look at state transitions, such life events, illnesses, and migration patterns.

11.2.3 Extensions of Parameter-Wise Co-Clustering

Although the parameter-wise co-clustering method presented herein only considered the use of the Gaussian distribution, it can be extended in various ways. One example would be to use other continuous distributions with more than one parameter. For example, one could consider the skew-t distribution and cluster columns based on location, scale, concentration and skewness. This could also be extended to data that cannot be considered a realization of a continuous random variable such as ordinal data where the columns could be partitioned according to mode and precision. The number of free parameters in each of these cases will not depend on the dimensionality of the data thus preserving the parsimony inherent to co-clustering.

11.2.4 Model Averaging

One final area of future work is to consider model averaging. Wei and McNicholas (2015) introduce mixture model averaging based on the approach of Madigan and Raftery (1994) using Occam's window. Mixture model averaging may be applied in this manner to any of the methodological approaches presented herein to possibly increase predictive performance.

Appendix A

Updates for Scale Matrices and Factor Loadings

The updates for the scale matrices and the factor loading matrices in the AECM algorithm for parsimonious MMVBFA are dependent on the model. The exact updates for each model are presented here.

Row Model Updates

CCC:

$$\hat{\boldsymbol{\Lambda}} = \left(\sum_{g=1}^{G}\sum_{i=1}^{N}\hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}}a_{ig}^{A\prime}\right)\left(\sum_{g=1}^{G}\sum_{i=1}^{N}\hat{z}_{ig}b_{ig}^{B}\right)^{-1}, \quad \hat{\sigma} = \frac{1}{Nnp}\operatorname{tr}\{\mathbf{S}^{(1)}\}.$$

where

$$\mathbf{S}^{(1)} = \sum_{g=1}^{G}\sum_{i=1}^{N}\hat{z}_{ig}\left[(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)' - \hat{\boldsymbol{\Lambda}}a_{ig}^{B\prime}\hat{\boldsymbol{\Psi}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\right].$$

CCU:

$$\hat{\boldsymbol{\Lambda}} = \left(\sum_{g=1}^{G}\sum_{i=1}^{N}\hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}}a_{ig}^{B\prime}\right)\left(\sum_{g=1}^{G}\sum_{i=1}^{N}\hat{z}_{ig}b_{ig}^{B}\right)^{-1}, \quad \hat{\boldsymbol{\Sigma}} = \frac{1}{Np}\operatorname{diag}\{\mathbf{S}^{(1)}\},$$

CUU:

For this model, the update for $\boldsymbol{\Lambda}$ needs to be performed row by row. Specifically, the updates are:

$$\hat{\boldsymbol{\Lambda}}_{(j)} = \left(\sum_{i=1}^{N}\hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}}a_{ig}^{B\prime}\right)_{(j)}\left(\sum_{g=1}^{G}\frac{1}{\sigma_{g_{(jj)}}}\sum_{i=1}^{N}\hat{z}_{ig}b_{ig}^{B}\right)^{-1},$$
$$\hat{\boldsymbol{\Sigma}}_g = \frac{1}{N_g p}\operatorname{diag}\{\mathbf{S}_g^{(2)}\},$$

where

$$\mathbf{S}_g^{(2)} = \sum_{i=1}^{N}\hat{z}_{ig}\left[(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)' - 2\hat{\boldsymbol{\Lambda}}a_{ig}^{B}\hat{\boldsymbol{\Psi}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)' + \hat{\boldsymbol{\Lambda}}b_{ig}^{B}\hat{\boldsymbol{\Lambda}}'\right].$$

CUC:

$$\hat{\boldsymbol{\Lambda}} = \left(\sum_{g=1}^{G}\frac{1}{\hat{\sigma}_g}\sum_{i=1}^{N}\hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}}a_{ig}^{B\prime}\right)\left(\sum_{g=1}^{G}\frac{1}{\hat{\sigma}_g}\sum_{i=1}^{N}\hat{z}_{ig}b_{ig}^{B}\right)^{-1},$$
$$\hat{\sigma}_g = \frac{1}{N_g np}\operatorname{tr}\{\mathbf{S}_g^{(2)}\}.$$

UCC:

$$\hat{\boldsymbol{\Lambda}}_g = \left(\sum_{i=1}^{N}\hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}}a_{ig}^{B\prime}\right)\left(\sum_{i=1}^{N}\hat{z}_{ig}b_{ig}^{B}\right)^{-1}, \quad \hat{\sigma} = \frac{1}{Nnp}\operatorname{tr}\{\mathbf{S}^{(3)}\},$$

where

$$\mathbf{S}^{(3)} = \sum_{g=1}^{G}\sum_{i=1}^{N} \hat{z}_{ig}\big[(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)' - \hat{\boldsymbol{\Lambda}}_g a_{ig}^{B\prime}\hat{\boldsymbol{\Psi}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\big].$$

UCU:

$$\hat{\boldsymbol{\Lambda}}_g = \left(\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}} a_{ig}^{B\prime}\right)\left(\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}^{B}\right)^{-1}, \quad \hat{\boldsymbol{\Sigma}} = \frac{1}{Np}\operatorname{diag}\{\mathbf{S}^{(3)}\}.$$

UUC:

$$\hat{\boldsymbol{\Lambda}}_g = \left(\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}} a_{ig}^{B\prime}\right)\left(\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}^{B}\right)^{-1}, \quad \hat{\sigma}_g = \frac{1}{N_g np}\operatorname{tr}\{\mathbf{S}_g^{(4)}\},$$

where

$$\mathbf{S}_g^{(4)} = \sum_{i=1}^{N} \hat{z}_{ig}\big[(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)' - \hat{\boldsymbol{\Lambda}}_g a_{ig}^{B}\hat{\boldsymbol{\Psi}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\big].$$

UUU:

$$\hat{\boldsymbol{\Lambda}}_g = \left(\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\hat{\boldsymbol{\Psi}}_g^{*^{-1}} a_{ig}^{B\prime}\right)\left(\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}^{B}\right)^{-1}, \quad \hat{\boldsymbol{\Sigma}}_g = \frac{1}{N_g p}\operatorname{diag}\{\mathbf{S}_g^{(4)}\}.$$

Column Model Updates

CCC:

$$\hat{\boldsymbol{\Delta}} = \left(\sum_{g=1}^{G}\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}} a_{ig}^{A}\right)\left(\sum_{g=1}^{G}\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}^{A}\right)^{-1}, \quad \hat{\psi} = \frac{1}{Nnp}\operatorname{tr}\{^{(1)}\},$$

where

$$^{(1)} = \sum_{g=1}^{G}\sum_{i=1}^{N} \hat{z}_{ig}\big[(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g) - \hat{\boldsymbol{\Delta}}a_{ig}^{A\prime}\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\big].$$

CCU:

$$\hat{\boldsymbol{\Delta}} = \left(\sum_{g=1}^{G}\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}a_{ig}^{A}\right)\left(\sum_{g=1}^{G}\sum_{i=1}^{N} \hat{z}_{ig}b_{ig}^{A}\right)^{-1}, \quad \hat{\boldsymbol{\Psi}} = \frac{1}{Nn}\operatorname{diag}\{^{(1)}\}.$$

CUU:

For this model, the update for $\boldsymbol{\Delta}$ needs to be performed row by row. Specifically the updates are:

$$\hat{\boldsymbol{\Delta}}_{(j)} = \left(\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}a_{ig}^{A}\right)_{(j)}\left(\sum_{g=1}^{G}\frac{1}{\psi_{g_{(jj)}}}\sum_{i=1}^{N} \hat{z}_{ig}b_{ig}^{A}\right)^{-1},$$
$$\hat{\boldsymbol{\Psi}}_g = \frac{1}{N_g n}\operatorname{diag}\{^{(2)}_{g}\},$$

where

$$^{(2)}_{g} = \sum_{i=1}^{N} \hat{z}_{ig}\big[(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g) - 2\hat{\boldsymbol{\Delta}}a_{ig}^{A\prime}\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g) + \hat{\boldsymbol{\Delta}}b_{ig}^{A}\hat{\boldsymbol{\Delta}}'\big].$$

CUC:

$$\hat{\boldsymbol{\Delta}} = \left(\sum_{g=1}^{G}\frac{1}{\hat{\psi}_g}\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}a_{ig}^{A}\right)\left(\sum_{g=1}^{G}\frac{1}{\hat{\psi}_g}\sum_{i=1}^{N} \hat{z}_{ig}b_{ig}^{A}\right)^{-1},$$
$$\hat{\psi}_g = \frac{1}{N_g np}\operatorname{tr}\{^{(2)}_{g}\}.$$

UCC:

$$\hat{\boldsymbol{\Delta}}_g = \left(\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}} a_{ig}^A\right)\left(\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}^A\right)^{-1}, \quad \hat{\psi} = \frac{1}{Nnp}\operatorname{tr}\{{}^{(3)}\},$$

where

$${}^{(3)} = \sum_{g=1}^{G}\sum_{i=1}^{N} \hat{z}_{ig}\left[(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g) - \hat{\boldsymbol{\Delta}}_g a_{ig}^{A\prime}\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\right].$$

UCU:

$$\hat{\boldsymbol{\Delta}}_g = \left(\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}} a_{ig}^A\right)\left(\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}^A\right)^{-1}, \quad \hat{\boldsymbol{\Psi}} = \frac{1}{Nn}\operatorname{diag}\{{}^{(3)}\}.$$

UUC:

$$\hat{\boldsymbol{\Delta}}_g = \left(\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}} a_{ig}^A\right)\left(\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}^A\right)^{-1}, \quad \hat{\psi}_g = \frac{1}{N_g np}\operatorname{tr}\{{}_g^{(4)}\},$$

where

$${}_g^{(4)} = \sum_{i=1}^{N} \hat{z}_{ig}\left[(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g) - \hat{\boldsymbol{\Delta}}_g a_{ig}^{A\prime}\hat{\boldsymbol{\Sigma}}_g^{*^{-1}}(\mathbf{X}_i - \hat{\mathbf{M}}_g)\right].$$

UUU:

$$\hat{\boldsymbol{\Delta}}_g = \left(\sum_{i=1}^{N} \hat{z}_{ig}(\mathbf{X}_i - \hat{\mathbf{M}}_g)'\hat{\boldsymbol{\Sigma}}_g^{*^{-1}} a_{ig}^A\right)\left(\sum_{i=1}^{N} \hat{z}_{ig} b_{ig}^A\right)^{-1}, \quad \hat{\boldsymbol{\Psi}}_g = \frac{1}{N_g n}\operatorname{diag}\{{}_g^{(4)}\}.$$

Bibliography

Aggarwal, C. C., Han, J., Wang, J., and Yu, P. S. (2003). A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 81–92. VLDB Endowment.

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

Albert, A. (1960). Estimating the infinitesimal generator of a finite state continuous-time markov process. *Annals of Mathematical Statistics*, **31**(3), 811–811.

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

Anderson, E. (1935). The irises of the Gaspé Peninsula. *Bulletin of the American Iris Society*, **59**, 2–5.

Andrews, J. L. and McNicholas, P. D. (2011). Extending mixtures of multivariate t-factor analyzers. *Statistics and Computing*, **21**(3), 361–373.

Andrews, J. L. and McNicholas, P. D. (2012). Model-based clustering, classification,

and discriminant analysis via mixtures of multivariate t-distributions: The tEIGEN family. *Statistics and Computing*, **22**(5), 1021–1029.

Andrews, J. L., McNicholas, P. D., and Subedi, S. (2011). Model-based classification via mixtures of multivariate t-distributions. *Computational Statistics and Data Analysis*, **55**(1), 520–529.

Azzalini, A. (2018). *The R package* `sn`*: The Skew-Normal and Related Distributions such as the Skew-t (version 1.5-3).* Università di Padova, Italia.

Banerjee, A. and Ghosh, J. (2000). Concept-based clustering of clickstream data.

Banerjee, A. and Ghosh, J. (2001). Clickstream clustering using weighted longest common subsequences. In *Proceedings of the web mining workshop at the 1st SIAM conference on data mining*, volume 143, page 144. Citeseer.

Baricz, A. (2010). Turn type inequalities for some probability density functions. *Studia Scientiarum Mathematicarum Hungarica*, **47**, 175–189.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, **41**, 164–171.

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

Bhattacharya, S. and McNicholas, P. D. (2014). A LASSO-penalized BIC for mixture model selection. *Advances in Data Analysis and Classification*, **8**(1), 45–61.

Biernacki, C. and Maugis, C. (2017). High-dimensional clustering. In *Choix de modèles et agrégation, Sous la direction de J-J. Droesbeke, G. Saporta, C. Thoma-Agnan Edition: Technip.*

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

Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics and Data Analysis*, **41**, 561–575.

Böhning, D., Dietz, E., Schaub, R., Schlattmann, P., and Lindsay, B. (1994). The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family. *Annals of the Institute of Statistical Mathematics*, **46**, 373–388.

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

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

Brault, V., Keribin, C., and Mariadassou, M. (2017). Consistency and asymptotic normality of latent blocks model estimators. arXiv preprint arXiv:1704.06629.

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

Cadez, I., Heckerman, D., Meek, C., Smyth, P., and White, S. (2003). Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery*, **7**(4), 399–424.

Celeux, G., Hurn, M., and Robert, C. P. (2000). Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, **95**, 957–970.

Chen, J. T. and Gupta, A. K. (2005). Matrix variate skew normal distributions. *Statistics*, **39**(3), 247–253.

Dang, U. J., Browne, R. P., and McNicholas, P. D. (2015). Mixtures of multivariate power exponential distributions. *Biometrics*, **71**(4), 1081–1089.

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*, **39**(1), 1–38.

Doğru, F. Z., Bulut, Y. M., and Arslan, O. (2016). Finite mixtures of matrix variate t distributions. *Gazi University Journal of Science*, **29**(2), 335–341.

Domínguez-Molina, J. A., González-Farías, G., Ramos-Quiroga, R., and Gupta, A. K. (2007). A matrix variate closed skew-normal distribution with applications to stochastic frontier analysis. *Communications in Statistics – Theory and Methods*, **36**(9), 1691–1703.

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

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 597, Department of Statistics, University of Washington, Seattle, WA.

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

Gallaugher, M. P. B. and McNicholas, P. D. (2017). A matrix variate skew-t distribution. *Stat*, **6**(1), 160–170.

Gallaugher, M. P. B. and McNicholas, P. D. (2018a). Clustering and semi-supervised classification for clickstream data via mixture models. arXiv preprint arXiv:1802.04849.

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

Gallaugher, M. P. B. and McNicholas, P. D. (2018c). Mixtures of matrix variate bilinear factor analyzers. In *Proceedings of the Joint Statistical Meetings*, Alexandria, VA. American Statistical Association. Preprint available as arXiv:1712.08664.

Gallaugher, M. P. B. and McNicholas, P. D. (2019a). *ClickClustCont: Mixtures of Continuous Time Markov Models*. R package version 0.1.7.

Gallaugher, M. P. B. and McNicholas, P. D. (2019b). Mixtures of skewed matrix variate bilinear factor analyzers. *Advances in Data Analysis and Classification*. DOI:10.1007/s11634-019-00377-4.

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

Gallaugher, M. P. B. and McNicholas, P. D. (2020). Parsimonious mixtures of matrix variate bilinear factor analyzers. In T. Imaizumi *et al.*, editors, *Advanced Studies in Behaviormetrics and Data Science*. Springer, Singapore.

Gallaugher, M. P. B., Biernacki, C., and McNicholas, P. D. (2018). Relaxing the identically distributed assumption in gaussian co-clustering for high dimensional data. arXiv preprint arXiv:1808.08366.

Ghahramani, Z. and Hinton, G. E. (1997). The EM algorithm for factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, Toronto, Canada.

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, **4**(2), 133–151.

Gupta, A. K. and Nagar, D. K. (1999). *Matrix variate distributions*. Chapman & Hall/CRC Press, Boca Raton.

Härdle, W. and Müller, M. (1997). Multivariate and semiparametric kernel regression. SFB 373 discussion paper, Humboldt University of Berlin, Berlin.

Harrar, S. W. and Gupta, A. K. (2008). On matrix variate skew-normal distributions. *Statistics*, **42**(2), 179–194.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, **2**(1), 193–218.

Jacques, J. and Biernacki, C. (2018). Model-based co-clustering for ordinal data. *Computational Statistics & Data Analysis*, **123**, 101–115.

Karlis, D. and Santourian, A. (2009). Model-based clustering with non-elliptically contoured distributions. *Statistics and Computing*, **19**(1), 73–83.

Lawley, D. N. and Maxwell, A. E. (1962). Factor analysis as a statistical method. *Journal of the Royal Statistical Society: Series D*, **12**(3), 209–229.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.

Lee, S. and McLachlan, G. J. (2014). Finite mixtures of multivariate skew t-distributions: some recent and new results. *Statistics and Computing*, **24**, 181–202.

Lin, T., McLachlan, G. J., and Lee, S. X. (2016). Extending mixtures of factor models using the restricted multivariate skew-normal distribution. *Journal of Multivariate Analysis*, **143**, 398–413.

Lin, T.-I. (2010). Robust mixture modeling using multivariate skew t distributions. *Statistics and Computing*, **20**(3), 343–356.

Lin, T.-I., McNicholas, P. D., and Hsiu, J. H. (2014). Capturing patterns via parsimonious t mixture models. *Statistics and Probability Letters*, **88**, 80–87.

Lindsay, B. G. (1995). Mixture models: Theory, geometry and applications. In *NSF-CBMS Regional Conference Series in Probability and Statistics*, volume 5. Hayward, California: Institute of Mathematical Statistics.

Madigan, D. and Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, **89**(428), 1535–1546.

Manly, B. (1976). Exponential data transformations. *Journal of the Royal Statistical Society: Series D (The Statistician)*, **25**(1), 37–42.

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

McLachlan, G. and Peel, D. (2000a). Mixtures of factor analyzers. In *In Proceedings of the Seventeenth International Conference on Machine Learning*, pages 599–606. Morgan Kaufmann, San Francisco.

McLachlan, G. J. and Peel, D. (2000b). *Finite Mixture Models*. John Wiley & Sons, New York.

McNeil, A. J., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques and Tools*. 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, P. D. and Murphy, T. B. (2010). Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics*, **26**(21), 2705–2712.

McNicholas, P. D. and Tait, P. (2019). *Data Science with Julia*. Chapman & Hall/CRC Press, Boca Raton.

McNicholas, P. D., Murphy, T. B., McDaid, A. F., and Frost, D. (2010). Serial and parallel implementations of model-based clustering via parsimonious Gaussian mixture models. *Computational Statistics and Data Analysis*, **54**(3), 711–723.

McNicholas, S. M., McNicholas, P. D., and Browne, R. P. (2017). A mixture of variance-gamma factor analyzers. In S. E. Ahmed, editor, *Big and Complex Data Analysis, Contributions to Statistics*, pages 369–385. Springer International Publishing, Cham.

Melnykov, V. (2016a). ClickClust: An R package for model-based clustering of categorical sequences. *Journal of Statistical Software*, **74**(9), 1–34.

Melnykov, V. (2016b). Merging mixture components for clustering through pairwise overlap. *Journal of Computational and Graphical Statistics*, **25**(1), 66–90.

Melnykov, V. (2016c). Model-based biclustering of clickstream data. *Computational Statistics and Data Analysis*, **21**, 31–45.

Melnykov, V. and Melnykov, I. (2012). Initializing the EM algorithm in gaussian mixture models with an unknown number of components. *Computational Statistics & Data Analysis*, **56**(6), 1381–1395.

Melnykov, V. and Zhu, X. (2018). On model-based clustering of skewed matrix data. *Journal of Multivariate Analysis*, **167**, 181–194.

Melnykov, V. and Zhu, X. (2019). Studying crime trends in the USA over the years 2000–2012. *Advances in Data Analysis and Classification*, **13**(1), 325–341.

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

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

Meynet, C. and Maugis-Rabusseau, C. (2012). A sparse variable selection procedure in model-based clustering. Research report.

Montgomery, A. L., Li, S., Srinivasan, K., and Liechty, J. C. (2004). Modeling online browsing and path analysis using clickstream data. *Marketing Science*, **23**(4), 579–595.

Morris, K. and McNicholas, P. D. (2013). Dimension reduction for model-based clustering via mixtures of shifted asymmetric Laplace distributions. *Statistics and Probability Letters*, **83**(9), 2088–2093.

Murray, P. M., McNicholas, P. D., and Browne, R. B. (2014a). A mixture of common skew-t factor analyzers. *Stat*, **3**(1), 68–82.

Murray, P. M., Browne, R. B., and McNicholas, P. D. (2014b). Mixtures of skew-t factor analyzers. *Computational Statistics and Data Analysis*, **77**, 326–335.

Murray, P. M., Browne, R. B., and McNicholas, P. D. (2017). A mixture of SDB skew-t factor analyzers. *Econometrics and Statistics*, **3**, 160–168.

Nadif, M. and Govaert, G. (2010). Model-based co-clustering for continuous data. In *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*, pages 175–180. IEEE.

Nitithumbundit, T. and Chan, J. S. K. (2015). An ECM algorithm for skewed multivariate variance gamma distribution in normal mean-variance representation. arXiv preprint arXiv:1504.01239.

Pan, W. and Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, **8**(May), 1145–1164.

Pau, G., Fuchs, F., Sklyar, O., Boutros, M., and Huber, W. (2010). EBImage—an R package for image processing with applications to cellular phenotypes. *Bioinformatics*, **26**(7), 979–981.

Peel, D. and McLachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and Computing*, **10**(4), 339–348.

Počuča, N., Gallaugher, M. P. B., and McNicholas, P. D. (2019). Matrixvariate.jl: A complete statistical framework for analyzing matrix variate data. http://github.com/nikpocuca/MatrixVariate.jl. Julia package version 0.2.0.

R Core Team (2019). *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.

Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, **26**(2), 195–239.

Revelle, W. (2018). *psych: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University, Evanston, Illinois. R package version 1.8.12.

Robert, V. (2017). *Coclustering for the analysis of pharmacovigilance massive datasets*. Ph.D. thesis, Université Paris-Saclay. Hal preprint: tel-01806330.

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

Scott, A. J. and Symons, M. J. (1971). Clustering methods based on likelihood ratio criteria. *Biometrics*, **27**, 387–397.

Steinley, D. (2004). Properties of the Hubert-Arabie adjusted Rand index. *Psychological Methods*, **9**, 386–396.

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components — an alternative to reversible jump methods. *The Annals of Statistics*, **28**(1), 40–74.

Subedi, S. and McNicholas, P. D. (2014). Variational Bayes approximations for clustering via mixtures of normal inverse Gaussian distributions. *Advances in Data Analysis and Classification*, **8**(2), 167–193.

Tait, P. A. and McNicholas, P. D. (2019). Clustering higher order data: Finite mixtures of multidimensional arrays. arXiv preprint arXiv:1907.08566.

Thabane, L. and Safiul Haq, M. (2004). On the matrix-variate generalized hyperbolic distribution and its bayesian applications. *Statistics*, **38**(6), 511–526.

Tiedeman, D. V. (1955). On the study of types. In S. B. Sells, editor, *Symposium on Pattern Analysis*. Air University, U.S.A.F. School of Aviation Medicine, Randolph Field, Texas.

Tipping, M. E. and Bishop, C. M. (1999a). Mixtures of probabilistic principal component analysers. *Neural Computation*, **11**(2), 443–482.

Tipping, M. E. and Bishop, C. M. (1999b). Probabilistic principal component analysers. *Journal of the Royal Statistical Society. Series B*, **61**, 611–622.

Tortora, C., Browne, R. P., Franczak, B. C., and McNicholas, P. D. (2015). *MixGHD: Model Based Clustering, Classification and Discriminant Analysis Using the Mixture of Generalized Hyperbolic Distributions*. R package version 1.8.

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.

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.

Vrbik, I. and McNicholas, P. D. (2012). Analytic calculations for the EM algorithm for multivariate skew-t mixture models. *Statistics and Probability Letters*, **82**(6), 1169–1174.

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

Waddell, A. R. and Oldford, R. W. (2013). *RnavGraphImageData: Some image data used in the RnavGraph package demos*. R package version 0.0.3.

Wei, J., Shen, Z., Sundaresan, N., and Ma, K.-L. (2012). Visual cluster exploration of web clickstream data. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 3–12. IEEE.

Wei, Y. and McNicholas, P. D. (2015). Mixture model averaging for clustering. *Advances in Data Analysis and Classification*, **9**(2), 197–217.

Williams, G. (2011). *Data mining with Rattle and R: The art of excavating data for knowledge discovery*. Springer Science & Business Media.

Wishart, J. (1928). The generalised product moment distribution in samples from a normal multivariate population. *Biometrika*, pages 32–52.

Wolfe, J. H. (1965). A computer program for the maximum likelihood analysis of types. Technical Bulletin 65-15, U.S. Naval Personnel Research Activity.

Xie, X., Yan, S., Kwok, J. T., and Huang, T. S. (2008). Matrix-variate factor analysis and its applications. *IEEE Transactions on Neural Networks*, **19**(10), 1821–1826.

Yeo, I.-K. and Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, **87**(4), 954–959.

Yu, S., Bi, J., and Ye, J. (2008). Probabilistic interpretations and extensions for a family of 2D PCA-style algorithms. In *Workshop Data Mining Using Matrices and Tensors (DMMT '08): Proceedings of a Workshop held in Conjunction with the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD 2008)*.

Zhao, J., Philip, L., and Kwok, J. T. (2012). Bilinear probabilistic principal component analysis. *IEEE Transactions on Neural Networks and Learning Systems*, **23**(3), 492–503.

Zhou, H., Pan, W., and Shen, X. (2009). Penalized model-based clustering with unconstrained covariance matrices. *Electronic Journal of Statistics*, **3**, 1473.

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