Model-based Learning:
*t*-Families, Variable Selection, and Parameter Estimation

by

Jeffrey L. Andrews

A Thesis
presented to
The University of Guelph

In partial fulfilment of requirements
for the degree of
Doctor of Philosophy
in
Mathematics and Statistics

Guelph, Ontario, Canada
ABSTRACT

Model-based Learning:

t-Families, Variable Selection, and Parameter Estimation

Jeffrey L. Andrews
University of Guelph, 2012

Advisor: Dr. Paul McNicholas

The phrase model-based learning describes the use of mixture models in machine learning problems. This thesis focuses on a number of issues surrounding the use of mixture models in statistical learning tasks: including clustering, classification, discriminant analysis, variable selection, and parameter estimation. After motivating the importance of statistical learning via mixture models, five papers are presented. For ease of consumption, the papers are organized into three parts: mixtures of multivariate t-families, variable selection, and parameter estimation.
Acknowledgments

I extend a heartfelt thank-you to Professor Paul McNicholas for his support, guidance, and friendship during the course of my studies. I also wish to thank my advisory committee members (Professors Zeny Feng and Loong-Tak Lim) and the examining committee (Professors Ryan Browne and Edward Carter) for their time; a special thank you to Professor Geoff McLachlan for agreeing to serve as the external examiner on this work.

Finally, I gratefully acknowledge the financial support I have received through an Ontario Graduate Scholarship in Science and Technology (OGSST) and the Natural Sciences and Engineering Research Council of Canada (NSERC) via a doctoral postgraduate scholarship (PGS-D). Friends, family, Irene, and camaraderie amongst the McNicholas research group ensured my graduate studies at the University of Guelph were a constant pleasure.
Publications

This manuscript-style thesis contains the following articles, in the order in which they appear:


# Table of Contents

1 Introduction 1

I Mixtures of Multivariate \(t\)-Families 5

2 Model-based Clustering, Classification, and Discriminant Analysis via Mixtures of Multivariate \(t\)-Distributions: The \(t\)EIGEN Family 6

2.1 Introduction 6

2.2 Methodology 10

2.2.1 The \(t\)EIGEN Family of Models 10

2.2.2 The EM Algorithm 11

2.2.3 Model-Based Clustering 12

2.2.4 Model-Based Classification 15

2.2.5 Model-based Discriminant Analysis 16

2.2.6 Model Selection 17

2.2.7 Performance Assessment 17

2.3 Data Analyses 18

2.3.1 Three Real Data Sets 18

2.3.2 Model-Based Classification & Discriminant Analysis 18

2.3.3 Model-Based Clustering 21

2.4 Discussion 23

3 \texttt{teigen}: Software for Model-based Clustering and Classification via the Multivariate \(t\)-Distribution 26

3.1 Introduction 26

3.2 The \(t\)EIGEN Family 28

3.2.1 Parameter Estimation 30

3.2.2 Initialization 31

3.2.3 Convergence 32

3.2.4 Model Selection 32

3.3 Code specifics 33

3.4 Examples 37

3.4.1 Model-based clustering: the basics on Old Faithful 37

3.4.2 Model-based classification: the basics on the Iris data set 38

3.4.3 Model-based clustering: specifying initialization on Body 40

3.4.4 Model-based clustering: univariate clustering on Bank Notes 41
## 4 Mixtures of Modified $t$-Factor Analyzers for Model-based Clustering, Classification, and Discriminant Analysis

4.1 Introduction ................................................. 44
4.2 Methodology ................................................. 47
   4.2.1 Factor Analysis & Variations ......................... 47
   4.2.2 Three Likelihoods ..................................... 49
   4.2.3 Parameter Estimation ................................. 51
4.3 Computational Issues ............................... 54
   4.3.1 Model Selection & Performance Assessment .... 54
   4.3.2 The Woodbury Identity ............................ 55
   4.3.3 Initialization ...................................... 56
4.4 Data Analyses ............................................. 57
   4.4.1 Three Real Data Sets ............................... 57
   4.4.2 Model-Based Classification & Discriminant Analysis .... 58
   4.4.3 Model-Based Clustering ............................ 59
4.5 Discussion ................................................. 60

## II Variable Selection

### 5 Variable Selection for Clustering and Classification

5.1 Introduction ............................................. 62
5.2 Background ................................................. 64
5.3 Methodology ................................................. 65
   5.3.1 Introduction ...................................... 65
   5.3.2 A Motivating Example ............................ 67
   5.3.3 The VSCC Method ................................ 68
   5.3.4 Subset Selection .................................. 70
   5.3.5 Clustering .......................................... 72
   5.3.6 Classification ..................................... 73
   5.3.7 Clustering/Classification Performance ........ 74
5.4 Simulations ................................................. 74
   5.4.1 Increased Dimension .............................. 75
   5.4.2 Increased Number of Groups ..................... 76
   5.4.3 Classification Example ......................... 78
5.5 Applications .............................................. 79
   5.5.1 Introduction ...................................... 79
   5.5.2 Leptograpsus Crabs ............................... 80
   5.5.3 Italian Wine ...................................... 81
   5.5.4 Swiss bank notes data ......................... 82
   5.5.5 Coffee data ...................................... 83
5.6 Discussion and Future Work ......................... 83
## III Parameter Estimation

### 6 Using Evolutionary Algorithms for Model-based Clustering

#### 6.1 Introduction

#### 6.2 Background

- 6.2.1 Model-based Clustering Likelihood
- 6.2.2 The EM Algorithm and Extensions Thereof
- 6.2.3 Evolutionary Algorithms

#### 6.3 Methodology

- 6.3.1 Introduction
- 6.3.2 Fitness Function
- 6.3.3 Survival
- 6.3.4 Reproduction
- 6.3.5 Convergence
- 6.3.6 Pseudo-Code

#### 6.4 Applications

- 6.4.1 Introduction
- 6.4.2 Iris
- 6.4.3 Italian Wines
- 6.4.4 Leptograpsus Crabs

#### 6.5 Discussion and Future Work

---

## IV Conclusion

### 7 Summary and Future Work

Bibliography
Chapter 1

Introduction

Machine learning can be viewed as the attempt to teach machines, namely computers, to perform tasks that humans are often quite good at. Take for instance grouping and organizing information: our brains are seemingly hard-wired to do this. We organize life itself into domains, domains into kingdoms, kingdoms into phylums, phylums into classes, orders, families, genera, and species. We organize universities into faculties, faculties into colleges, colleges into departments, and departments into research groups. We organize people into religions, countries, and races; perhaps, in these cases, to our detriment.

Nowadays, most of what we consider information is stored on computers; and the amount of data being collected is increasing. This is where the ability of humans to distinguish groups degrades. Information for us is three visual dimensions plus a few other sensory dimensions; we are not prepared to organize information contained in thousands of dimensions. Computers, on the other hand, are built for one task: fast, binary computation. Though extreme high-dimensionality still presents a challenge to computers, they are built to have a considerably higher threshold than human ability — with the threshold constantly increasing in accordance with hardware advancements. In contrast, their ability to perform human tasks (a.k.a.
artificial intelligence or AI) is traditionally quite limited; although whether this is
due to the inherent nature of a computer or our inability to properly develop AI
algorithms is up for debate.

Machine learning problems arise in countless disciplines and data sets. The
research contained in this thesis concerns itself with the statistical learning approach
on continuous valued data sets. In fact, its focus is on one particular statistical
learning problem — clustering and classification — utilizing one particular statistical
approach — mixture models. The term ‘clustering’ refers to the grouping of data
without any a priori knowledge of what groups are present in the data. ‘Classification’,
on the other hand, refers to grouping when some subset of the data has known

group membership.

Mixture model-based clustering and classification is attractive for a number
of theoretical reasons. A mathematical description of mixture models will follow in
the body of the thesis, but succinctly: a mixture model is the convex combination
of some number of statistical densities such that the combined mixture model den-
sity still holds the properties of a statistical distribution. Using mixture models to
group data is thus a natural application: we assume that groups of data arise with
differing characteristics which can be described as arising from differing statistical
distributions.

Due to the statistical nature of a mixture model, “clustering methods based
on mixture models allow estimation and hypothesis testing within the framework of
standard statistical theory” (Aitkin et al., 1981, pg 421). This fact makes a strong
case, at least in the minds of many statisticians, for adopting model-based clustering
over more heuristic clustering techniques. Some proponents of model-based clustering go further, stating that mixture-based is the “only clustering technique that is entirely satisfactory from the mathematical point of view. It assumes a well-defined mathematical model, investigates it by well-established statistical techniques, and provides a test of significance for the results” (Marriott, 1974, pg 70). Note that both of these quotes are over three decades old: only recently, with modern-day computer hardware improvements, have we been able to truly implement computationally intensive algorithms to fit multivariate mixture models and expect them to give us answers within a reasonable timeframe.

This introduction concludes with a summary of what is being presented; but first, a disclaimer. Due to the nature of licensing agreements with publishers, some of the work contained herein is required to be reprinted in its entirety. For the sake of consistency, the works with which we still hold copyright are also reprinted in their entirety. Because of this, similar material may be discussed in multiple sections of the thesis. I humbly request the patience of the reader in this regard.

Each paper is given its own chapter and the chapter titles match the published or working titles of the papers. Note that the numbered references to figures, tables, and sections will differ from the original papers in order to avoid redundancy. Also, the abstracts and author contact information do not appear — I am the first author on each paper, with second authorship belonging to my advisor.

Part I deals with the use of the multivariate t-distribution for model-based clustering and classification. Three articles appear in Part I, and they further serve as an introduction to model-based clustering/classification. In the first article, the
tEIGEN family is introduced via constraints on an eigen-decomposed covariance structure. The second article introduces open-source software for the R computing environment that implements tEIGEN for clustering and classification. The final paper in Part I introduces the MODt family, which is developed through constraints on a modified factor analysis covariance structure.

In Part II, a novel variable selection technique is introduced to facilitate model-based clustering and classification. Its performance, both in speed and clustering accuracy, is shown to be superior with respect to comparable techniques.

Finally, Part III sees a new parameter estimation technique for mixture models put forth that makes use of both the expectation-maximization algorithm and evolutionary algorithms.
Part I

Mixtures of Multivariate $t$-Families
Chapter 2

Model-based Clustering, Classification, and Discriminant Analysis via Mixtures of Multivariate $t$-Distributions: The $t$EIGEN Family

The following work by Andrews and McNicholas (2012) is slated to appear in Statistics and Computing and is reprinted in its entirety, with kind permission from Springer Science and Business Media.

2.1 Introduction

The objective of cluster analysis, in general, is to devise a classification of a set of observations into subsets that are, in some sense, interesting (Gordon, 1981; Titterington et al., 1985). Since they allow for each cluster to be modelled using one of a number of component densities, finite mixture models present a somewhat natural approach to clustering data. Of course, as with any clustering approach, a number of challenges present; perhaps the most apparent of which is the fact that a mixture component will not necessarily directly correspond to a cluster.

A random vector $X$ is said to arise from a parametric finite mixture distri-
bution if, for all $x \subset X$, we can write its density as

$$f(x \mid \vartheta) = \sum_{g=1}^{G} \pi_{g} p_{g}(x \mid \theta_{g}),$$

where $\pi_{g} > 0$, such that $\sum_{g=1}^{G} \pi_{g} = 1$, are called mixing proportions, the $p_{g}(x \mid \theta_{g})$ are the component densities, and $\vartheta = (\pi_{1}, \ldots, \pi_{G}, \theta_{1}, \ldots, \theta_{G})$. In their seminal review of work up to that time, Fraley and Raftery (2002) traced the use of finite mixture models for clustering back to the 1960s. In doing so they cited, amongst others, the work of Wolfe (1965), Edwards and Cavalli-Sforza (1965), and Day (1969). Though reviews of various types of model-based clustering work, carried out up to their respective times, are given by McLachlan (1982), Titterington et al. (1985), McLachlan and Basford (1988), Banfield and Raftery (1993), Dasgupta and Raftery (1998), McLachlan and Peel (2000b) and Fraley and Raftery (2002). The vast majority of this work was based on Gaussian mixture models.

The idiom ‘model-based clustering’ has routinely been used when a family of mixture models is fitted to data for clustering and the best model from among this family is selected via some criterion, most often the Bayesian information criterion (BIC Schwarz, 1978). However, the term can also be used to describe the application of a single mixture model for clustering. A family of mixture models is said to arise when various constraints are imposed upon the component densities; most often on the component covariance structure. The result is a flexible modelling paradigm that incorporates more and less parsimonious models.

The most famous family of mixture models within the literature is the MCLUST family (Banfield and Raftery, 1993; Celeux and Govaert, 1995; Fraley and
Raftery, 1998, 2002), which is supported by the mclust package (Fraley and Raftery, 2006) for the R software (R Development Core Team, 2012). The MCLUST family of models utilizes an eigen-decomposition of the component covariance matrices which leads to 10–14 mixture models (cf. Section 2.2.1). More recently, a number of other families of Gaussian mixtures models have been introduced, including work by Bouveyron et al. (2007) and McNicholas and Murphy (2008, 2010a). Interesting work on variable selection for clustering has also been carried out within the Gaussian mixture modelling framework (cf. Raftery and Dean, 2006; Maugis et al., 2009).

The use of mixtures of multivariate t-distributions for model-based clustering has received considerably less attention to date. McLachlan and Peel (1998) motivated the multivariate t-distribution as a heavy-tailed alternative to the multivariate Gaussian distribution, by first considering the Gaussian scale mixture model

\[(1 - \epsilon)\phi(x | \mu, \Sigma) + \epsilon\phi(x | \mu, c\Sigma),\]

where \(c\) is large, \(\epsilon\) is small, and \(\phi(x | \mu, \Sigma)\) is the density of a multivariate Gaussian distribution with mean \(\mu\) and covariance matrix \(\Sigma\). Then, noting that this mixture model can be written

\[\int \phi(x | \mu, \Sigma/u)dH(u),\]

for an appropriately defined probability distribution \(H\), the multivariate t-distribution is obtained when \(H\) is replaced by the probability density of a random variable \(U \sim \text{gamma}(\nu/2, \nu/2)\), where \(\nu\) denotes degrees of freedom. The model density for mixtures of multivariate t-distributions is

\[g(x | \vartheta) = \sum_{g=1}^{G} \pi_g f_t(x | \mu_g, \Sigma_g, \nu_g),\]
where $\pi_g$ are the mixing proportions and $f_t(x \mid \mu_g, \Sigma_g, \nu_g)$ is the density of a $p$-dimensional multivariate $t$-distribution with mean $\mu_g$, covariance matrix $\Sigma_g$ and degrees of freedom $\nu_g$. Note that this version of the multivariate $t$-distribution models each variable with the same $\nu_g$; a less restrictive model is possible but not considered herein. Some recent work using the multivariate $t$-distribution for model-based clustering has been put forth by McLachlan et al. (2007), Greselin and Ingrassia (2010a,b), and Andrews and McNicholas (2011a,b).

McLachlan (1992, Section 2.1) describes how mixture models can be used for discriminant analysis, with parameters estimated based on the training data and then used to classify observations with unknown component memberships. Model-based classification, or partial classification (cf. McLachlan, 1992, Section 2.7), can be thought of as a generalization of model-based clustering, in the sense that the modelling is carried out in the same fashion, except that a number of observations have known component memberships (cf. Section 2.2.4). Dean et al. (2006) extended the MCLUST family in this latter fashion and demonstrated that the resulting modelling framework, whereby the parameter estimates are arrived at using both the observations with known component memberships and the observations to be classified, can deliver superior classification performance to discriminant analysis.

In Section 2.2 we outline the methodology for the multivariate $t$ analog of the MCLUST family, and its application to clustering, classification, and discriminant analysis. A simulation study investigating the efficacy of our model-based classification and discriminant analysis approaches, along with a collection of real data analyses, is given in Section 2.3. We conclude with discussion and summary in
2.2 Methodology

2.2.1 The tEIGEN Family of Models

The eigen-decomposition of a matrix is an oft-used procedure in mathematics with close ties to common multivariate statistical analyses, such as principal component analysis. Applying this parameterization to the mixture model covariance structure gives component covariance matrices $\Sigma_g = \lambda_g D_g A_g D'_g$, where $D_g$ is the matrix of eigenvectors, $A_g$ is the diagonal matrix with entries proportional to the eigenvalues, and $\lambda_g$ are the associated constants of proportionality.

Utilizing the eigen-decomposition of the multivariate Gaussian distribution covariance structure can lead to a family of 14 Gaussian parsimonious clustering models (GPCMs, Celeux and Govaert, 1995) wherein the following constraints are considered and combined if possible: $\lambda_g = \lambda$, $A_g = A$, $D_g = D$, $D_g = I$, and $A_g = I$, where $I$ is the identity matrix of appropriate dimension. A subset of 10 of these models is available as the MCLUST family (Fraley and Raftery, 2002). Considering the same constraints as the MCLUST family of models, in addition to constraining or not the degrees of freedom parameter to be equal across groups (cf. Andrews and McNicholas, 2011a), leads to a family of 20 mixtures of multivariate $t$-distributions with eigen-decomposed covariance, which we shall call the tEIGEN family. In fact, the tEIGEN family of models introduced herein also incorporates two further GPCM models which, again taking into account the degrees of freedom constraint, brings the
total number of tEIGEN models to 24. A summary of these models is given in (see Table 2.4, Appendix 2.4).

2.2.2 The EM Algorithm

The expectation-maximization (EM) algorithm (Dempster et al., 1977) is an iterative procedure for finding maximum likelihood estimates when data are incomplete. Although the consummate reference for the EM algorithm is that of Dempster et al. (1977), Titterington et al. (1985, Section 4.3.2) point out that similar treatments had previously been employed by Baum et al. (1970), Orchard and Woodbury (1972), and Sundberg (1974). The EM algorithm involves the iteration of two steps until convergence is reached. In the expectation step (E-step) the expected value of the complete-data log-likelihood is computed and then, in the maximization step (M-step), the expected value of the complete-data log-likelihood is maximized with respect to the model parameters. Note that the ‘complete-data’ refers to the missing plus observed data.

One variant of the EM algorithm, which is important for our purposes, is the expectation-conditional maximization (ECM) algorithm (Meng and Rubin, 1993). The ECM algorithm replaces the M-step with a number of conditional maximization steps that can be more computationally efficient. For our purposes, the missing data consist of the component indicator variables $z_{ig}$ and the characteristic weights $u_{ig}$. 
The component membership labels are represented by $z_{ig}$, where

$$
z_{ig} = \begin{cases} 
1 & \text{if observation } x_i \text{ belongs to component } g, \\
0 & \text{otherwise,}
\end{cases}
$$

and the estimation of these $z_{ig}$ can be considered the principal objective of model-based clustering, classification, and discriminant analysis.

### 2.2.3 Model-Based Clustering

Under a model-based clustering framework, no observations have known group membership. In order to run the ECM algorithm, initialization of these $z_{ig}$ must take place. Fraley and Raftery (2002) utilized a Gaussian model-based agglomerative hierarchical clustering procedure to obtain starting values when using MCLUST. McNicholas and Murphy (2008) used a number of random starting values on their most constrained model ($\Sigma_g = \Lambda \Lambda' + \Psi$) and utilized the results of the run with the largest BIC value to initialize for each value of $G$. Experience suggests that random starting values pose a particularly grave threat to algorithm failure when using mixtures of multivariate $t$-distributions. In order to avoid this pitfall, the agglomerative hierarchical initialization method used by MCLUST will be used for the clustering analyses herein.

In the case of the $t$EIGEN family, parameter estimation for these models is carried out via the multicycle ECM algorithm and is comparable to the Gaussian case, for which extensive details are given by Celeux and Govaert (1995). The principal differences are the incorporation of the weights $u_{ig}$ and the estimation of the degrees of freedom parameter. For $n \ p$-dimensional observations, the complete-data
log-likelihood for the general tEIGEN model-based clustering case can be written as

\[ l_c(\theta) = \sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} \log \left[ \pi_g \gamma \left( u_{ig} | \nu_g/2, \nu_g/2 \right) \right] \times \phi(x_i | \mu_g, \Sigma_g/u_{ig}) \],

where \( \gamma \left( u_{ig} | \nu_g/2, \nu_g/2 \right) \) models the weights \( u_{ig} \) as the gamma density with shape and rate parameters \( \nu_g/2 \). At each E-step, the component indicator variables are updated by their conditional expected values

\[ \hat{z}_{ig} = \frac{\pi_g f_t(x | \mu_g, \Sigma_g)}{\sum_{h=1}^{G} \pi_h f_t(x | \mu_h, \Sigma_h, \nu_h)} \],

and the characteristic weights are updated using theirs

\[ \hat{u}_{ig} = \frac{\nu_g + p}{\nu_g + \delta(x_i, \mu_g | \Sigma_g)} \],

where \( \delta(x_i, \mu_g | \Sigma_g) \) is the squared Mahalanobis distance between \( x_i \) and \( \mu_g \). In the first CM-step, we update the mixing proportions and component means using

\[ \hat{\pi}_g = \frac{n_g}{n} \quad \text{and} \quad \hat{\mu}_g = \frac{\sum_{i=1}^{n} \hat{z}_{ig} \hat{u}_{ig} x_i}{\sum_{i=1}^{n} \hat{z}_{ig} \hat{u}_{ig}} \],

with \( n_g = \sum_{i=1}^{n} \hat{z}_{ig} \). We also update the degrees of freedom; in the case where \( \nu_g = \nu \), this update is given by solving the equation

\[ 1 - \varphi \left( \frac{\hat{\nu}_{\text{new}}}{2} \right) + \frac{1}{n} \sum_{g=1}^{G} \sum_{i=1}^{n} \hat{z}_{ig} (\log \hat{u}_{ig} - \hat{u}_{ig}) \\
+ \log \left( \frac{\hat{\nu}_{\text{new}}}{2} \right) + \varphi \left( \frac{\hat{\nu}_{\text{old}}}{2} + p \right) - \log \left( \frac{\hat{\nu}_{\text{old}}}{2} + p \right) = 0, \]

for \( \hat{\nu}_{\text{new}} \), where \( \hat{\nu}_{\text{old}} \) is the estimate from the previous iteration (Andrews and McNicholas, 2011a). Updates in the unconstrained case are similar (cf. McLachlan and Peel, 1998).
In the second CM-step, we update the covariance structure. In general, these updates are similar to their Gaussian analogues (cf. Celeux and Govaert, 1995). For illustrative purposes, consider the member of the tEIGEN family with \( D_g = D \) and \( A_g = A \). In this case, the complete-data log-likelihood is given by

\[
l_c(\lambda_g, D, A) = K + \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} p \log(\lambda_g) - \sum_{i=1}^{n} \sum_{g=1}^{G} \frac{z_{ig} u_{ig}}{\lambda_g} \| x_i - \mu_g \|_2^2 \{DAD'\}^{-1},
\]

where \( K \) is a constant with respect to \( \lambda_g, D, \) and \( A \). Setting \( DAD' = C \) and taking the expected value of \( l_c(\lambda_g, D, A) \) gives

\[
Q(\lambda_g, D, A) = K + p \sum_{g=1}^{G} n_g \log(\lambda_g) + \sum_{g=1}^{G} \frac{n_g S_g C^{-1}}{\lambda_g}.
\]

Differentiating \( Q(\lambda_g, D, A) \) with respect to \( \lambda_g \) and setting the result equal to zero gives

\[
\hat{\lambda}_g^{\text{new}} = \frac{1}{pn_g} \text{tr}\{S_g C^{-1}\},
\]

where \( S_g = (1/n_g) \sum_{i=1}^{n} \hat{z}_{ig} \hat{u}_{ig} \| x_i - \hat{\mu}_g \|_2^2 \). Utilizing Theorem A.1 from Celeux and Govaert (1995), the update for \( C \) can be written

\[
C = \frac{(1/\lambda_g) \sum_{g=1}^{G} S_g}{\| (1/\lambda_g) \sum_{g=1}^{G} S_g \|^{1/p}},
\]

which gives \( \hat{D}^{\text{new}} \) as the matrix of eigenvectors of \( C \) and \( \hat{A}^{\text{new}} \) as a diagonal matrix of the eigenvalues of \( C \).

In our analyses we use an asymptotic approximation of the log-likelihood, obtained based on Aitken’s acceleration, to determine convergence of our EM algorithms (cf. Lindsay, 1995; McNicholas et al., 2010). Once the algorithm has converged,
cluster memberships are estimated via the maximum \textit{a posteriori} (MAP) classification, which is given by

$$\text{MAP}\{\hat{z}_{ig}\} = \begin{cases} 
1 & \text{if } \max_g \{\hat{z}_{ig}\} \text{ occurs at component } g, \\
0 & \text{otherwise.}
\end{cases}$$

Note that, for the fully unconstrained \textit{tEIGEN} model we have $\Sigma_g = \lambda_g D_g A_g D_g'$ which can be estimated with $\hat{\Sigma}_g = S_g$ and, similarly, the fully constrained model has $\Sigma_g = \lambda DAD'$ and so $\hat{\Sigma}_g = S$, where $S = \sum_{g=1}^G \pi_g S_g$. As such, four of the twenty four members of the \textit{tEIGEN} family have appeared previously in statistical literature (McLachlan and Peel, 1998; Andrews et al., 2011).

\subsection{Model-Based Classification}

In classification procedures, some subset of the data has known group memberships and the objective is to estimate the group memberships for the observations with unknown group memberships. Model-based classification utilizes both the known and unknown data to update the model parameters. A review of model-based classification procedures up to that time can be found in McLachlan (1992, Section 2.7). More recently, relevant papers have been put forth by Dean et al. (2006), McNicholas (2010), and Andrews et al. (2011).

Suppose that we observe $n$ $p$-dimensional observations and that $k < n$ of these have known classifications. Now, ordering these observations so that it is the first $k$ that have known $z_{ig}$, the likelihood for the model-based classification framework
\begin{align*}
\mathcal{L}(\theta \mid \mathbf{x}, \mathbf{z}) &= \prod_{i=1}^{k} \prod_{g=1}^{G} \left[ \pi_g f_i(x_i \mid \mu_g, \Sigma_g, \nu_g) \right]^{z_{ig}} \\
&\quad \times \prod_{j=k+1}^{n} \sum_{h=1}^{G} \pi_h f_i(x_j \mid \mu_h, \Sigma_h, \nu_h).
\end{align*}

Following the imposition of any desired constraints on \( \Sigma_g \), parameter estimation proceeds in an analogous fashion to the model-based clustering paradigm (cf. Section 2.2.3). As in the model-based clustering scenario, the unknown \( z_{ig} \) must be initialized. Andrews et al. (2011) introduce a uniform initialization procedure (\( \hat{z}_{ig} = 1/G \)) and compare it with the MCLUST hierarchical agglomerative initialization procedure. Both procedures will be used for the analyses herein, thus facilitating further comparison.

### 2.2.5 Model-based Discriminant Analysis

Discriminant analysis is a classification technique that utilizes the data with known group memberships to set classification rules for the unknown data. Thus, in a mixture discriminant analysis (Hastie and Tibshirani, 1996) framework, the data with known group memberships are used to estimate the model density and the data with unknown group memberships are classified according to their \( \hat{z}_{ig} \). Fraley and Raftery (2002) discuss MCLUST DA, where they allow multiple components per known group by using the BIC to choose the number of components and the best MCLUST model for each group; we will take a similar approach with the \textit{t}EIGEN family.
2.2.6 Model Selection

When using a family of mixture models, some objective criterion is necessary for selecting the ‘best’ model for the data under consideration. The Bayesian information criterion (BIC, Schwarz, 1978) is most commonly used for these purposes, and is given by

\[
\text{BIC} = 2l(x, \hat{\vartheta}) - m \log n,
\]

where \(m\) is the number of free parameters, \(n\) is the sample size, \(l(x, \hat{\vartheta})\) is the maximized log-likelihood, and \(\hat{\vartheta}\) is the maximum likelihood estimate of \(\vartheta\). The use of the BIC for model selection in model-based clustering is well established and has some theoretical and applied bases for its use (Leroux, 1992; Kass and Raftery, 1995; Kass and Wasserman, 1995; Keribin, 2000).

2.2.7 Performance Assessment

Although we will treat the data analyses of Section 2.3 as genuine classification and clustering examples, the true classifications are in fact known for these data. In these examples, the adjusted Rand index (Hubert and Arabie, 1985) is used to measure class agreement. The Rand index (Rand, 1971) can be expressed as

\[
\frac{\text{number of pairwise agreements}}{\text{number of pairs}},
\]

and takes a value on \([0, 1]\), where 1 indicates perfect agreement. One problem with the Rand index is that its expected value is greater than 0 under random classification, making smaller values of the Rand index difficult to interpret. The adjusted Rand index corrects the Rand index for chance by accounting for the fact that classification
performed randomly would probably correctly classify some cases. The adjusted Rand index has an expected value of 0 under random classification and perfect classification results in a value of 1.

2.3 Data Analyses

2.3.1 Three Real Data Sets

Forina et al. (1986) recorded 28 chemical and physical properties of three types of wine (Barolo, Grignolino, Barbera) from the Piedmont region of Italy. A subset of 13 of these variables is available in the gclus package (Hurley, 2004) for R.

Mangasarian et al. (1995) describe a study wherein fluid was extracted from a breast mass using a procedure known as fine needle aspirate (FNA). There were 30 attributes for each of 569 cases, of which 176 developed breast cancer.

The leptograpsus crabs data set can be found in the MASS package (Venables and Ripley, 1999) in R. These data contain five physical measurements on two different colour forms of the crabs, further separated into the two genders. We consider all four groups in our analyses, taking account of both gender and colour.

2.3.2 Model-Based Classification & Discriminant Analysis

Simulated Data

A two-group bivariate data set was simulated to investigate the effect of biased training data on both model-based classification and model-based discriminant analysis. Two-hundred observations were generated from a multivariate Gaussian
distribution for each group, so that \( n = 400 \). The training data were selected such that all those points below a threshold were considered known for the first group and all those points above the same threshold were selected for the second group. The data set, training set, and discriminant analysis results can be viewed graphically in Figure 2.1. A summary of the misclassifications for each technique is given in Table 2.1.

![True Classifications](image1)

![Training Set](image2)

![MCLUST DA](image3)

![tEIGEN DA](image4)

Figure 2.1: Raw data, biased training data, and discriminant analysis results for the simulation that was conducted to compare model-based discriminant analysis to model-based classification.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Training Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Biased</td>
</tr>
<tr>
<td>tEIGEN model-based class.</td>
<td>1.5%</td>
</tr>
<tr>
<td>tEIGEN model-based disc.</td>
<td>49.0%</td>
</tr>
<tr>
<td>MCLUST model-based class.</td>
<td>1.5%</td>
</tr>
<tr>
<td>MCLUST model-based disc.</td>
<td>32.0%</td>
</tr>
</tbody>
</table>

From Figure 2.1 and Table 2.1, it is apparent that both discriminant analyses
have misclassified many of the data points (32% and 49% for MCLUST and \( t \)EIGEN, respectively) due to the biased training data. In stark contrast to these results, model-based classification with the \( t \)EIGEN and MCLUST families misclassified merely three observations using the same training data. When the bias is taken out of the training data by selecting points to be known at random, both model-based classification and discriminant analysis give very good performance with either perfect, or near-perfect, classifications.

**Real Data**

We analyze each of the data sets introduced in Section 2.3.1. To simulate a classification scenario, we randomly chose 25 different subsets of 50% of the observations for each data set to be taken as having known group memberships. We then analyzed these data using model-based classification and discriminant analysis within the \( t \)EIGEN framework. As mentioned in Section 2.2.4, we used two different initialization techniques within the model-based classification scenario: the agglomerative approach used by MCLUST and the uniform approach (\( \hat{z}_{ig} = 1/G, \text{ for } i = k+1, \ldots, n \)). For comparison, we also ran MCLUST on these data. The classification performance for all methods, and for different initialization procedures, where relevant, are given in Table 2.2.

For two of the three data sets, the top performer over the 25 runs was a model-based classification technique, rather than a discriminant analysis technique. The \( t \)EIGEN family with model-based classification gave perfect classification for the Italian wine data set and similar performance to MCLUST model-based classification
Table 2.2: Adjusted Rand indices for model-based classification (MBC) and model-based discriminant analysis (MBDA) techniques from the tEIGEN and MCLUST families, over all 25 runs. For tEIGEN MBC, results are given for both agglomerative hierarchical (agg.) and uniform (uni.) initializations.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Wine</th>
<th>Breast Cancer</th>
<th>Crabs</th>
</tr>
</thead>
<tbody>
<tr>
<td>tEIGEN MBC (agg.)</td>
<td>1.00</td>
<td>0.75</td>
<td>0.87</td>
</tr>
<tr>
<td>tEIGEN MBC (uni.)</td>
<td>1.00</td>
<td>0.75</td>
<td>0.87</td>
</tr>
<tr>
<td>tEIGEN MBDA</td>
<td>0.89</td>
<td>0.79</td>
<td>0.80</td>
</tr>
<tr>
<td>MCLUST MBC</td>
<td>0.91</td>
<td>0.78</td>
<td>0.87</td>
</tr>
<tr>
<td>MCLUST DA</td>
<td>0.73</td>
<td>0.82</td>
<td>0.71</td>
</tr>
</tbody>
</table>

for the other two data sets. Model-based discriminant analysis results were comparable across both families and all three data sets. The tEIGEN family with model-based classification gave the same results, to two decimal places, under the two initialization procedures under consideration. This lends further support to the argument for the use of the much more efficient uniform initialization procedure in model-based classification analyses (cf. Andrews et al., 2011).

2.3.3 Model-Based Clustering

The MCLUST and tEIGEN families were applied to the three real data sets from Section 2.3.1.

Table 2.3: Adjusted Rand indices, covariance structure, number of components, and estimated degrees of freedom for the chosen model-based clustering models for the three real data sets using both the MCLUST and tEIGEN families.

<table>
<thead>
<tr>
<th></th>
<th>Wine</th>
<th>Breast Cancer</th>
<th>Crabs</th>
</tr>
</thead>
<tbody>
<tr>
<td>tEIGEN</td>
<td>0.52</td>
<td>CICC (6, 9.7)</td>
<td>0.69</td>
</tr>
<tr>
<td>MCLUST</td>
<td>0.48</td>
<td>VEI (8, –)</td>
<td>0.59</td>
</tr>
</tbody>
</table>

The results (Table 2.3) show that the tEIGEN family consistently outperforms the MCLUST family on these data sets. This is due, in large part, to the
tendency of MCLUST to choose an excess of components. Interestingly, the best \textit{tEIGEN} model has constrained degrees of freedom in each case and the estimated values $\hat{\nu}$ are low (9.7 and 3.9) for two of the three data sets, indicating clearly non-Gaussian component densities.

Note the superiority of the \textit{tEIGEN} family’s performance on the crabs data set. This is particularly interesting due to the relatively large estimate of the degrees of freedom ($\hat{\nu} = 40.6$), which may lead one to regard this model as being effectively Gaussian. However, our best \textit{tEIGEN} model better fits the data, with a BIC value of 133.2, compared to 13.6 for the best MCLUST model. Furthermore, the MCLUST model (EEV) that is ‘essentially equivalent’ to the chosen \textit{tEIGEN} model has a BIC of -71.6 and gives poor clustering performance; ‘essentially equivalent’ here means that the covariance structures are identical and $\hat{\nu}$ is large.

Looking at the results (Table 2.3), it is apparent that neither the MCLUST nor the \textit{tEIGEN} family give particularly good clustering performance on the Italian wine data set. It is well known that eigen-decomposed covariance structures do not do as well as the latent factor structures on certain data sets, including this one (cf. McNicholas and Murphy, 2008, 2010b; Andrews and McNicholas, 2011b). There are also other approaches that are known to give better clustering performance on some of these data sets (see Scrucca, 2010, for example). However, the main purpose of our analyses was to compare the MCLUST and \textit{tEIGEN} families and, in this sense, the analyses are effective.
2.4 Discussion

The body of literature on model-based clustering, classification, and discriminant analysis was reviewed. Then, the toolkit of clustering, classification, and discriminant analysis methodologies that use mixture models was extended by the introduction of a new family of models based on mixtures of multivariate $t$-distributions; the $t$EIGEN family.

The performance of this family of mixture models was evaluated on real and simulated data. In the model-based clustering paradigm, the $t$EIGEN family performed at least as well as its Gaussian counterpart on each of three real data sets. In the model-based classification analyses, the $t$EIGEN family gave better classification performance when compared to MCLUST on the wine data and similar performance on the other two data sets. Interestingly, the $t$EIGEN family gave perfect classification for the wine data set (Table 2.2) despite the fact that such models generally do not do very well when clustering these data (Table 2.3). Perhaps more importantly, our simulation results support the argument that classification within the semi-supervised model-based classification framework is somewhat safer than model-based discriminant analysis.

Future work will investigate efficient implementations of these models, both in serial and in parallel. This will facilitate the dissemination of this work to the wider scientific community. Work will also be conducted on restricting the estimated degrees of freedom to one of a few values during parameter estimation for the $t$EIGEN family (cf. Besag et al., 1995), and on the use of alternative model selection approaches.
Acknowledgements

The authors gratefully acknowledge the helpful comments of an associate editor and two anonymous reviewers. Thanks to Professors Adrian Raftery, Gilles Celeux, Geoff McLachlan, Brendan Murphy and the other members of the University of Washington Working Group on Model-Based Clustering for helpful comments. This work was supported by a Postgraduate Doctoral Scholarship (Andrews) and a Discovery Grant (McNicholas) from the Natural Sciences & Engineering Research Council of Canada. The computing equipment that was used was purchased using grants from the Canada Foundation for Innovation Leaders Opportunity Fund and from the Ontario Ministry for Research & Innovation Small Infrastructure Fund.

Members of the tEIGEN Family

The nomenclature, covariance structure, and number of free parameters in the covariance matrix are given, for all members of the tEIGEN family, in Table 2.4.
Table 2.4: Nomenclature and number of covariance parameters for each member of the rEIGEN family; 'C' denotes that a constraint is imposed, 'U' denotes that a constraint is not imposed, and 'I' denotes that the matrix in question is taken to be the identity matrix of suitable dimension. Superscripts correspond to the article where a model was previously used: 1McLachlan and Peel (1998), 2Andrews et al. (2011).

<table>
<thead>
<tr>
<th>Model</th>
<th>$\lambda_g = \lambda$</th>
<th>$D_g = D$</th>
<th>$A_g = A$</th>
<th>$\nu_g = \nu$</th>
<th>Free Covariance Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIIC</td>
<td>C</td>
<td>I</td>
<td>I</td>
<td>C</td>
<td>$1 + 1$</td>
</tr>
<tr>
<td>CIIU</td>
<td>C</td>
<td>I</td>
<td>I</td>
<td>U</td>
<td>$1 + G$</td>
</tr>
<tr>
<td>UIIC</td>
<td>U</td>
<td>I</td>
<td>I</td>
<td>C</td>
<td>$(G - 1) + 1$</td>
</tr>
<tr>
<td>UIIU</td>
<td>U</td>
<td>I</td>
<td>I</td>
<td>U</td>
<td>$(G - 1) + G$</td>
</tr>
<tr>
<td>CICC</td>
<td>C</td>
<td>I</td>
<td>C</td>
<td>C</td>
<td>$p + 1$</td>
</tr>
<tr>
<td>CICU</td>
<td>C</td>
<td>I</td>
<td>C</td>
<td>U</td>
<td>$p + G$</td>
</tr>
<tr>
<td>UICC</td>
<td>U</td>
<td>I</td>
<td>C</td>
<td>C</td>
<td>$p + (G - 1) + 1$</td>
</tr>
<tr>
<td>UCIC</td>
<td>U</td>
<td>I</td>
<td>C</td>
<td>U</td>
<td>$p + (G - 1) + G$</td>
</tr>
<tr>
<td>CIUC</td>
<td>C</td>
<td>I</td>
<td>U</td>
<td>C</td>
<td>$Gp - (G - 1) + 1$</td>
</tr>
<tr>
<td>CIUU</td>
<td>C</td>
<td>I</td>
<td>U</td>
<td>U</td>
<td>$Gp - (G - 1) + G$</td>
</tr>
<tr>
<td>UIUC</td>
<td>U</td>
<td>I</td>
<td>U</td>
<td>C</td>
<td>$Gp + 1$</td>
</tr>
<tr>
<td>UIUU</td>
<td>U</td>
<td>I</td>
<td>U</td>
<td>U</td>
<td>$Gp + G$</td>
</tr>
<tr>
<td>CCCC$^2$</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>$[p(p + 1)/2] + 1$</td>
</tr>
<tr>
<td>CCCU$^2$</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>U</td>
<td>$[p(p + 1)/2] + G$</td>
</tr>
<tr>
<td>UCCC</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>$[p(p + 1)/2] + (G - 1) + 1$</td>
</tr>
<tr>
<td>UCCU</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>U</td>
<td>$[p(p + 1)/2] + (G - 1) + G$</td>
</tr>
<tr>
<td>CUCC</td>
<td>C</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>$G[p(p + 1)/2] - (G - 1)(p) + 1$</td>
</tr>
<tr>
<td>CUCU</td>
<td>C</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td>$G[p(p + 1)/2] - (G - 1)(p) + G$</td>
</tr>
<tr>
<td>UUCC</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>$G[p(p + 1)/2] - (G - 1)(p - 1) + 1$</td>
</tr>
<tr>
<td>UUCU</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td>$G[p(p + 1)/2] - (G - 1)(p - 1) + G$</td>
</tr>
<tr>
<td>CUUC</td>
<td>C</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>$G[p(p + 1)/2] - (G - 1) + 1$</td>
</tr>
<tr>
<td>CUUU</td>
<td>C</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>$G[p(p + 1)/2] - (G - 1) + G$</td>
</tr>
<tr>
<td>UUUC$^2$</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>$G[p(p + 1)/2] + 1$</td>
</tr>
<tr>
<td>UUUU$^1$</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>$G[p(p + 1)/2] + G$</td>
</tr>
</tbody>
</table>
Chapter 3

teigen: Software for Model-based Clustering and Classification via the Multivariate t-Distribution

The following article introduces open-source software in order to implement the family developed in Chapter 2. The paper in this chapter also develops four univariate mixture models to include in the tEIGEN family. A number of examples are given to further illustrate the usage of the software package.

The paper is currently under peer-review and the software package referred to throughout is currently available through CRAN:

http://cran.r-project.org/web/packages/teigen/index.html.

3.1 Introduction

The usage of mixture models for cluster analyses is commonly referred to as model-based clustering. A random vector $\mathbf{X}$ arises from a parametric finite mixture model if the density of origin can be written as $f(\mathbf{x} \mid \vartheta) = \sum_{g=1}^{G} \pi_g p_g(\mathbf{x} \mid \theta_g)$, where $\vartheta$ is the parameter space, $G$ is the number of components, $\pi_g$ are the mixing weights such that $\sum_{g=1}^{G} \pi_g = 1$ and all $\pi_g > 0$, and $p_g(\mathbf{x} \mid \theta_g)$ are parametric densities with parameters $\theta_g$. 
The multivariate Gaussian distribution has received the bulk of researchers’ attention over the past couple of decades (Banfield and Raftery, 1993; Celeux and Govaert, 1995; Fraley and Raftery, 2002; McNicholas and Murphy, 2008). More recently, the focus has been shifting to non-Gaussian mixture models (McLachlan and Peel, 1998; Lin, 2010; Karlis and Santourian, 2009; Andrews and McNicholas, 2011b,a; Vrbik and McNicholas, 2012). The most common method of fitting these models is by using the expectation-maximization algorithm, or a closely related variant.

The number of parameters requiring estimation can be computationally crippling and in many cases increases quadratically with the dimensionality of the data. The bulk of these parameters lie in the covariance structure of the density. One way of reducing this hindrance is the development of families of mixture models that arise from constraints on the covariance structure — indeed, many of the previously noted references are focused on this task. The flagship mixture model family is the MCLUST family (Fraley and Raftery, 1999, 2003, 2006), which is derived from the multivariate Gaussian distribution with eigen-decomposed covariance structure (Banfield and Raftery, 1993; Celeux and Govaert, 1995).

In the section that follows, we review the multivariate $t$-distributional equivalent of the MCLUST family — the $t$EIGEN family (Andrews and McNicholas, 2012) — for both clustering and classification; we also introduce four univariate $t$-distribution mixture models. Section 3.3 includes specifics on the R package introduced, while Section 3.4 gives examples for application of the package. We conclude in Section 3.5 with a summary.
3.2 The tEIGEN Family

Andrews and McNicholas (2012) introduce the 24 member tEIGEN mixture model family for model-based clustering and classification. The tEIGEN family is derived from mixtures of multivariate $t$-distributions, whose density is

$$f(x \mid \vartheta) = \sum_{g=1}^{G} \pi_g f_t(x \mid \mu_g, \Sigma_g, \nu_g),$$

where $f_t(x \mid \mu_g, \Sigma_g, \nu_g)$ is the multivariate $t$ density

$$f_t(x \mid \mu_g, \Sigma_g, \nu_g) = \frac{\Gamma \left( \frac{\nu_g + p}{2} \right) \left| \Sigma_g \right|^{-\frac{1}{2}}}{(\pi \nu_g)^{\frac{p}{2}} \Gamma \left( \frac{\nu_g}{2} \right)} \left[ 1 + \frac{\delta(x, \mu_g, \Sigma_g)}{\nu_g} \right]^{-\frac{\nu_g + p}{2}},$$

with mean vector $\mu_g$, scale matrix $\Sigma_g$, and degrees of freedom $\nu_g$.

Following the work of Banfield and Raftery (1993) and Celeux and Govaert (1995), an eigen-decomposition is imposed on the covariance matrix $\Sigma_g = \lambda_g D_g A_g D_g'$, where $D_g$ is the matrix of eigenvectors, $A_g$ is the diagonal matrix of scaled eigenvalues, and $\lambda_g$ are the associated constants of proportionality. These individual eigen-decomposition components can then be constrained to be equal across mixture components, or in some cases constrained to be the identity matrix: $\Sigma_g = \lambda_g I A I' = \lambda_g A$, for instance. In addition, following Andrews and McNicholas (2011a) constraints are also imposed on the degrees of freedom $\nu_g$. Table 3.1 gives the 24 tEIGEN models from Andrews and McNicholas (2012).

All of the previously considered tEIGEN models are applicable only to multivariate data. In order for the package introduced herein to be applicable to univariate data, we introduce four more models into the tEIGEN family. The univariate mixture
Table 3.1: $t$EIGEN model names and the number of covariance and degrees of freedom parameters requiring estimation. ‘C’ denotes constrained, ‘U’ denotes unconstrained, and ‘I’ denotes the identity matrix.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\lambda_g = \lambda$</th>
<th>$D_g = D$</th>
<th>$A_g = A$</th>
<th>$\nu_g = \nu$</th>
<th>Free Covariance and DF Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIIC</td>
<td>C</td>
<td>I</td>
<td>I</td>
<td>C</td>
<td>1 + 1</td>
</tr>
<tr>
<td>CIIU</td>
<td>C</td>
<td>I</td>
<td>I</td>
<td>U</td>
<td>1 + G</td>
</tr>
<tr>
<td>UIIC</td>
<td>U</td>
<td>I</td>
<td>I</td>
<td>C</td>
<td>$(G - 1) + 1$</td>
</tr>
<tr>
<td>UIIU</td>
<td>U</td>
<td>I</td>
<td>I</td>
<td>U</td>
<td>$(G - 1) + G$</td>
</tr>
<tr>
<td>CICC</td>
<td>C</td>
<td>I</td>
<td>C</td>
<td>C</td>
<td>$p + 1$</td>
</tr>
<tr>
<td>CICU</td>
<td>C</td>
<td>I</td>
<td>C</td>
<td>U</td>
<td>$p + G$</td>
</tr>
<tr>
<td>UICC</td>
<td>U</td>
<td>I</td>
<td>C</td>
<td>C</td>
<td>$p + (G - 1) + 1$</td>
</tr>
<tr>
<td>UICU</td>
<td>U</td>
<td>I</td>
<td>C</td>
<td>U</td>
<td>$p + (G - 1) + G$</td>
</tr>
<tr>
<td>CIUC</td>
<td>C</td>
<td>I</td>
<td>U</td>
<td>C</td>
<td>$Gp - (G - 1) + 1$</td>
</tr>
<tr>
<td>CIUU</td>
<td>C</td>
<td>I</td>
<td>U</td>
<td>U</td>
<td>$Gp - (G - 1) + G$</td>
</tr>
<tr>
<td>UIUC</td>
<td>U</td>
<td>I</td>
<td>U</td>
<td>C</td>
<td>$Gp + 1$</td>
</tr>
<tr>
<td>UUUU</td>
<td>U</td>
<td>I</td>
<td>U</td>
<td>U</td>
<td>$Gp + G$</td>
</tr>
<tr>
<td>CCCC</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>$[p(p + 1)/2] + 1$</td>
</tr>
<tr>
<td>CCCU</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>U</td>
<td>$[p(p + 1)/2] + G$</td>
</tr>
<tr>
<td>UCCC</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>$[p(p + 1)/2] + (G - 1) + 1$</td>
</tr>
<tr>
<td>UCCU</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>U</td>
<td>$[p(p + 1)/2] + (G - 1) + G$</td>
</tr>
<tr>
<td>CUCC</td>
<td>C</td>
<td>U</td>
<td>C</td>
<td>C</td>
<td>$G[p(p + 1)/2] - (G-1)(p) + 1$</td>
</tr>
<tr>
<td>CUUC</td>
<td>C</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td>$G[p(p + 1)/2] - (G-1)(p) + G$</td>
</tr>
<tr>
<td>UUCU</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>U</td>
<td>$G[p(p + 1)/2] - (G-1)(p-1) + 1$</td>
</tr>
<tr>
<td>CUCU</td>
<td>C</td>
<td>U</td>
<td>U</td>
<td>C</td>
<td>$G[p(p + 1)/2] - (G-1)(p-1) + G$</td>
</tr>
<tr>
<td>CUUU</td>
<td>C</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>$G[p(p + 1)/2] - (G-1) + 1$</td>
</tr>
<tr>
<td>UUUU</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>$G[p(p + 1)/2] + 1$</td>
</tr>
<tr>
<td>UUUU</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>$G[p(p + 1)/2] + G$</td>
</tr>
</tbody>
</table>

$t$ takes the form

$$f(x \mid \vartheta) = \sum_{g=1}^{G} \pi_g f_t(x \mid \mu_g, \sigma_g^2, \nu_g),$$

where $\mu_g$ is the component mean and $\sigma_g^2$ is the component variance. We once again allow constraints on the variance and degrees of freedom, by permitting them to be equal across groups. The four resulting models are summarized in Table 3.2. Note that we now refer to the ‘$t$EIGEN family’ as the combination of all 28 univariate and multivariate models.
Table 3.2: Univariate model names and the number of variance and degrees of freedom parameters requiring estimation. ‘C’ denotes constrained, ‘U’ denotes unconstrained.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\sigma_g^2 = \sigma^2$</th>
<th>$\nu_g = \nu$</th>
<th>Free Variance and DF Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>univCC</td>
<td>C</td>
<td>C</td>
<td>1 + 1</td>
</tr>
<tr>
<td>univCU</td>
<td>C</td>
<td>U</td>
<td>1 + $G$</td>
</tr>
<tr>
<td>univUC</td>
<td>U</td>
<td>C</td>
<td>$G + 1$</td>
</tr>
<tr>
<td>univUU</td>
<td>U</td>
<td>U</td>
<td>$G + G$</td>
</tr>
</tbody>
</table>

3.2.1 Parameter Estimation

As in the majority of mixture modelling implementations, we use a variant of the EM algorithm in order to perform parameter estimation. The EM algorithm consists of two steps (expectation and maximization) performed iteratively until convergence. On the E-step, we compute the expected value of the complete-data log-likelihood and on the M-step the parameters are maximized according to the complete-data log likelihood. The expectation-conditional maximization (ECM) algorithm (Meng and Rubin, 1993) adjusts the M-step to allow several, more efficient conditional maximization (or CM) steps. The code herein makes use of multiple E-steps at each iteration, which is called a multi-cycle ECM algorithm.

In order to implement an EM algorithm, we need to be able to compute the expectation of the complete-data log-likelihood, conditional on the observed data, for our mixture models. First, we define a random variable $Z_{ig}$ such that $z_{ig} = 1$ if observation $i$ belongs to group $g$ and otherwise $z_{ig} = 0$. In a model-based classification (or semi-supervised) scenario, we observe the first $k$ observations with known group membership, and the remaining observations, giving $n$ total observations, with
unknown group membership. The likelihood can thus be written

$$L(\vartheta) = \prod_{i=1}^{k} \prod_{g=1}^{G} \left[ \pi_g f_i(x_i | \mu_g, \Sigma_g, \nu_g) \right]^{z_{ig}} \times \prod_{j=k+1}^{n} \sum_{h=1}^{G} \pi_h f_j(x_j | \mu_h, \Sigma_h, \nu_h).$$

(3.1)

In a model-based clustering scenario, none of the group memberships are known, and therefore the likelihood is simply the right hand side of Equation 3.1.

Specifics will not be given for the various tEIGEN models with regards to parameter estimation in this article. The reader is referred to Andrews and McNicholas (2012) for details on the tEIGEN models, and furthermore to Celeux and Govaert (1995) for the multivariate Gaussian equivalent (where the mathematics are largely analogous).

### 3.2.2 Initialization

EM algorithms for both clustering and classification require the initialization of either the unknown $z_{ig}$ cluster memberships (which uses maximum likelihood estimation to initialize the parameters) or of the model parameters (which uses expected value computations to initialize the $z_{ig}$). Our algorithm makes use of the former, with several built-in options for initialization: $k$-means, ‘hard’ random, ‘soft’ random, and uniform — as described by Andrews et al. (2011) and only available for classification. The difference between hard random and soft random is the nature of the randomly initialized $z_{ig}$: hard referring to 0’s and 1’s, while soft takes on random values between 0 and 1, inclusive. As a further alternative, the user can give specific cluster memberships as initialization, as described in Section 3.3. Note that the degrees of freedom must be initialized, and this is also user-specified; the default value is 50.
3.2.3 Convergence

Convergence in our algorithm is determined by Aitken’s acceleration (Aitken, 1926), which at iteration $t$ is given by

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

where $l(t-1)$ refers to the log-likelihood at iteration $t - 1$, and so on. Böhning et al. (1994) propose the usage of $a(t)$ to compute an asymptotic estimate of the log-likelihood at iteration $t + 1$ by

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

As suggested by Lindsay (1995), we use the stopping criterion

$$l_\infty^{(t+1)} - l^{(t+1)} < \epsilon,$$

for user-specified $\epsilon$, with $\epsilon = 0.1$ as the default.

3.2.4 Model Selection

Selecting the ‘best’ model is a challenge within model-based clustering or classification applications, but the current consensus (Fraley and Raftery, 2002; McNicholas and Murphy, 2008, 2010a; Andrews and McNicholas, 2011b) among researchers is the usage of the Bayesian information criterion (BIC, Schwarz, 1978), which is calculated as

$$\text{BIC} = 2l(x, \hat{\theta}) - r \log n,$$
where \( l(x, \hat{\theta}) \) is the maximized log-likelihood, \( \hat{\theta} \) is the MLE of \( \theta \), \( r \) is the number of free parameters in the model, and \( n \) is the total number of observations.

Another model selection technique permitted with the \texttt{teigen} package is the integrated completed likelihood (ICL Biernacki et al., 2000). The ICL makes use of the concept of uncertainty rising from the probabilistic nature of model-based clustering. Generally, and also in the case of the \texttt{teigen} package, the main result of a clustering/classification algorithm is a vector of group memberships. These are technically the maximum a posteriori (MAP) classifications, which for any estimate \( \hat{z}_{ig} \) are given by

\[
\text{MAP}\{\hat{z}_{ig}\} = \begin{cases} 
1 & \text{if } \max_g \{z_{ig}\} \text{ occurs in component } g, \\
0 & \text{otherwise}.
\end{cases}
\]

The approximate ICL is then calculated by

\[
\text{ICL} \approx \text{BIC} + \sum_{i=1}^{n} \sum_{g=1}^{G} \text{MAP}\{\hat{z}_{ig}\} \log \hat{z}_{ig},
\]

which is essentially the BIC penalized by the amount of classification uncertainty contained in the model.

### 3.3 Code specifics

The \texttt{teigen} package contains one main function, \texttt{teigen()}, which allows the user to perform model-based clustering or classification with some flexibility on the specifics. All code for the \texttt{teigen} was written in R (R Development Core Team, 2012). The \texttt{teigen()} function holds the following structure
`teigen(x, Gs = 1:9, models = "all", init = "kmeans", scale = TRUE, 
    dfstart = 50, clas = 0, known = NULL, training = NULL,
    gauss = FALSE, dfupdate = TRUE, eps = 0.1)`

and takes the following arguments:

- **x** is a numeric matrix, data frame, or vector (for univariate data) where the rows are observations and the columns are the variables.

- **Gs** is a number or vector indicating the number of groups to fit. Default is 1 through 9.

- **models** is a character vector giving the models to fit. Alternatively, "all" runs all tEIGEN models (default), "dfunconstrained" runs all the unconstrained degrees of freedom models, "dfconstrained" runs all the constrained degrees of freedom models, "mclust" approximates MCLUST models and restricts degrees of freedom estimation to approximately normal, "gaussian" is similar but includes two further mixture models than MCLUST.

- **init** is a list of initializing classification of the form that `init[[G]]` contains the initializing vector for all G considered. Alternatively, the user can use a character string indicating initialization method. Currently the user can choose from "kmeans" (default), hard random - "hard", soft random - "soft", and "uniform" can be used in classification only.

- **scale** is a logical indicating whether or not the function should scale the data. Default is TRUE and is the suggested method.

- **dfstart** is the initialized value for the degrees of freedom. The default is 50.
• **clas** is a value between 0-100 indicating the percentage of data taken to be known. Note that a vector of known classifications is needed for **clas** greater than 0. See next argument for an alternative. Default is 0 and performs clustering, otherwise the algorithm chooses the training index randomly (and will return it via **index**).

• **training** is an optional indexing vector for the observations whose classification is taken to be known.

• **known** is a vector of known classifications that can be numeric or character - these are optional for clustering but necessary for classification. Must be the same length as the number of rows in the data set. If using in a ‘true’ classification sense, give samples with unknown classification the value NA (see examples below).

• **gauss** is a logical indicating if the algorithm should approximate the gaussian distribution. If **models="mclust"** then **gauss=TRUE** is forced. When **gauss=TRUE**, degrees of freedom are set to 200 and not updated.

• **dfupdate** is a logical indicating whether or not the degrees of freedom should be estimated. If FALSE, the value from **dfstart** is used instead.

• **eps** is the value of tolerance for the convergence criterion, as discussed in Section 3.2.3

Output from **teigen()** comes in the form of a two-pronged list. The main contents are the results from the model chosen by the BIC, with an additional list
containing the results from the model chosen by the ICL:

- **x** is the data used for clustering/classification.

- **bic** is the BIC value corresponding to the selected model (i.e., the maximum BIC value computed).

- **classification** is a vector of group classifications as determined by the BIC.

- **allbic** is the matrix of BIC values according to model and G. A value of -Inf is returned when the model did not converge or was not run for other reasons.

- **bestmodel** is a character string giving best model (BIC) details.

- **G** is the value corresponding to the number of components chosen by the BIC.

- **df** is the estimated degrees of freedom for each group (BIC).

- **tab** is the classification table for BIC model (only available when known is given). When classification is used the known observations are left out of the table.

- **index** is the indexing vector giving observations taken to be known (only available when clas > 0 or training is given).

- **fuzzy** is the fuzzy clustering matrix for the model selected by the BIC.

- **logl** is the log-likelihood corresponding to the model with the best BIC.

- **iclresults** is a list containing all the previous outputs, except x and index, pertaining to the model chosen by the best ICL (all under the same name except allicl and icl are the equivalent of allbic and bic, respectively).
3.4 Examples

Herein, we present a number of clustering/classification examples to illustrate the teigen package. We make use of a number of data sets from the base R distribution, as well as a couple available in the gclus library (Hurley, 2004).

3.4.1 Model-based clustering: the basics on Old Faithful

The old faithful data set is available in base R distributions as faithful. It is bivariate data measuring the time to eruption and length of eruption, both in minutes. In this first example, we use the entire tEIGEN family and also run an approximation of its Gaussian equivalent. For both runs, we utilize soft random starting values and, in the interest of reproducibility, we use the set.seed() function.

```r
data(faithful)
set.seed(139786)
teigen_faith <- teigen(faithful, Gs=1:4, init="soft")
teigen_faith$bestmodel

"The best model (BIC of -834.7) is UUCC with G=2"

set.seed(139786)
gaussian_faith <- teigen(faithful, Gs=1:4, init="soft", models="gaussian")
gaussian_faith$bestmodel

"The best model (BIC of -831.46) is CICC with G=3"

plot(faithful, col=c("blue","red","black",
                     "green")[teigen_faith$clas])
plot(faithful, col=c("blue","red","black",
                     "green",)[gaussian_faith$clas])
```

The resulting plots can be seen in Figure 3.1.
Figure 3.1: Plots of model-based clustering results on the old faithful data set for the $t$EIGEN family (left) and approximate Gaussian equivalent (right).

3.4.2 Model-based classification: the basics on the Iris data set

The famous iris data set is available in base R distributions as iris. It contains four measurements on 150 irides, hailing from three different species. We demonstrate the various ways of performing model-based classification with the teigen package. First, we use only the models from $t$EIGEN with unconstrained degrees of freedom, and show how this package can simulate a classification scenario by simply giving the percentage of data we wish to take as ‘unknown’. We take 50% of the data to have unknown membership and use the uniform initialization.

```r
data(iris)
set.seed(357678)
tclass_iris <- teigen(iris[,-5], init="uniform",
models="dfunconstrained", known=iris[,5], clas=50)
tclass_iris$bic
```

-806.9793
Note that the table that appears with the `$tab` call contains only those observations taken to have unknown cluster membership. This table plainly shows that we have misclassified only three observations. The index of the observations that are are taken to be known are stored in `$index`. Therefore, as an illustration, we can run the exact same analysis as before, but this time using a training index rather than percentage of data taken to be known.

```r
trainingset <- tclass_iris$index
tclass_iris2 <- teigen(iris[, -5], init = "uniform",
                     models = "dfunconstrained", known = iris[, 5],
                     training = trainingset)
tclass_iris2$bic
-806.9793
```

Of course, in a true classification scenario there exists a subset of the data where group membership is unknown. In this scenario, the known classification vector should have `NA` inputted for those values. The following gives an example of how this should be done.
irisknown <- iris[,5]
irisknown[-trainingset] <- NA
irisknown[1:5]

<NA>    setosa <NA>    setosa setosa

tclass_iris3 <- teigen(iris[,-5], init="uniform",
models="dfunconstrained", known=irisknown,
training=trainingset)

tclass_iris3$bic

-806.9793

tclass_iris3$tab

<table>
<thead>
<tr>
<th></th>
<th>newmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>known</td>
<td>1 2 3</td>
</tr>
<tr>
<td>setosa</td>
<td>0 0 0</td>
</tr>
<tr>
<td>unknown</td>
<td>27 21 27</td>
</tr>
<tr>
<td>versicolor</td>
<td>0 0 0</td>
</tr>
<tr>
<td>virginica</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>

We can see now that 27 of the unknown observations fall into mixture components 1 and 3 (the Setosa and Virginica components) while 21 fall into mixture component 2 (the Versicolor component). This is actually the same classification example as the previous one; with identical results except that we pretend the observations have unknown group membership.

### 3.4.3 Model-based clustering: specifying initialization on Body

The body data set is available in the gclus library as body. It contains a variety of measurements on male and female participants. Herein, we analyze the data using model-based clustering with the entire tEIGEN family and illustrate how to perform user-specified initializations. We initialize using hierarchical clustering via
the \texttt{hclust} function in R. We also illustrate the usage of the ICL as model selection and show how to find model parameters.

```r
library(gclus)
data(body)
bodydist <- dist(scale(body[, -25]))
hbody <- hclust(bodydist)
initial <- list()
for(i in 1:5){
  initial[[i]] <- cutree(hbody, k=i)
}
teigen_body <- teigen(body[, -25], Gs=1:5, init=initial, known=body[, 25])
teigen_body$iclresults$bestmod

"The best model (ICL of -18661.52) is UCCU with G=2"

teigen_body$iclresults$tab

\begin{tabular}{ccc}
  known & 1 & 2 \\
  0 & 257 & 3 \\
  1 & 5 & 242 \\
\end{tabular}

nteigen_body$iclresults$parameters$df

20.88554 36.15527

The \textit{tEIGEN} family misclassifies only 8 of the 507 observations; misplacing 5 men and 3 women into the wrong gender specification. Running the Gaussian approximation (via adding \texttt{models="gaussian"} to the \texttt{teigen()} call in this example) results in 3 groups being selected by the BIC and many more misclassifications.

3.4.4 Model-based clustering: univariate clustering on Bank Notes

The bank notes data set is available in the \texttt{gclus} library as \texttt{bank}. It contains a number of measurements on both counterfeit and genuine Swiss bank notes. We select the \texttt{Diagonal} variable to perform univariate clustering on.
library(gclus)
data(bank)
attach(bank)
set.seed(20637)
teigen_bank <- teigen(Diagonal,init="hard",known=Status)
teigen_bank$bestmod

"The best model (BIC of -511.66) is univUC with G=2"

teigen_bank$tab

<table>
<thead>
<tr>
<th>bestzmap</th>
<th>known 1 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>96 4</td>
</tr>
<tr>
<td>1</td>
<td>0 100</td>
</tr>
</tbody>
</table>

The number of misclassified bank notes is only four. Interestingly, these four bank notes are all true notes that have been misclassified as counterfeit. Once again, running the Gaussian approximation in this example results in a poor solution where the number of groups is deemed to be only one.

3.5 Summary

The teigen R package was introduced and described in detail. It allows the user to fit a family of multivariate and univariate $t$-distribution mixture models to numerical data, with a number of built-in options and error-catches. Examples further illustrate to the reader how teigen should be utilized and show its effectiveness as a clustering technique.
Chapter 4

Mixtures of Modified $t$-Factor Analyzers for Model-based Clustering, Classification, and Discriminant Analysis

We shift gears now to a family of mixture models with a different parameterization of the covariance structure. While the $t$EIGEN family is quite useful in a variety of clustering scenarios, it lacks the ability to cluster data in high dimensions (except under the dubious assumption of independence among the variables). The modified factor analyzer family, on the other hand, is well-suited for high dimensionality due to the implicit variable reduction performed within the covariance estimation.

This article by Andrews and McNicholas (2011b) originally appears in the Journal of Statistical Planning and Inference and is reprinted in its entirety, in accordance with Elsevier’s author-usage policies regarding theses and dissertations.
4.1 Introduction

The application of mixture models to data for the purposes of clustering or classification has received renewed attention within the literature over the past half-decade or so (see Fraley and Raftery, 2006; McLachlan et al., 2007; McNicholas and Murphy, 2008; Karlis and Santourian, 2009; Scrucca, 2010; McNicholas, 2010; Andrews and McNicholas, 2011a, for examples). The prevalent approach sees the fitting of a family of mixture models, followed by the selection of the best model from amongst this family. The best model is typically selected via some criterion such as the Bayesian information criterion (BIC Schwarz, 1978), and a family of models usually arises from the imposition of constraints upon the component covariance structure. When mixture models are used for clustering, this is known as model-based clustering; when they are used for classification in a semi-supervised fashion, this is known as model-based classification; and when we estimate parameters based on a subset of the data, with the objective of classifying the reminder of the data, this is called model-based discriminant analysis (cf. McLachlan, 1992, Chapter 2, for extensive details).

The MCLUST family of models is now very well established within the literature (cf. Banfield and Raftery, 1993; Celeux and Govaert, 1995; Fraley and Raftery, 2002); the popularity of this approach is due in part to the availability of the mclust package (Fraley and Raftery, 2006) through the R software (R Development Core Team, 2012). Fraley and Raftery (2002) used MCLUST for both model-based clustering and discriminant analysis, while Dean et al. (2006) use MCLUST for model-
based classification. The underlying mixture model is a Gaussian mixture model with an eigen-decomposed component covariance structure,

\[ f(x \mid \vartheta) = \sum_{g=1}^{G} \pi_g \phi(x \mid \mu_g, \lambda_g D_g A_g D'_g), \]

where \( \pi_g \in [0, 1] \), such that \( \sum_{g=1}^{G} \pi_g = 1 \), are called mixing proportions, \( \phi(x \mid \mu_g, \lambda_g D_g A_g D'_g) \) is the density of a \( p \)-dimensional multivariate Gaussian random variable with mean \( \mu_g \) and eigen-decomposed covariance matrix \( \Sigma_g = \lambda_g D_g A_g D'_g \), and \( \vartheta \) denotes the model parameters. Note that, in this eigen-decomposition, \( \lambda_g \) is a constant, \( D_g \) is a matrix consisting of the eigenvectors of \( \Sigma_g \), and \( A_g \) is a diagonal matrix with entries proportional to the eigenvalues of \( \Sigma_g \). Imposing various combinations of the constraints \( A_g = A, D_g = D, \lambda_g = \lambda, A = I_p, \) and \( D = I_p \), where \( I_p \) is the \( p \times p \) identity matrix, leads to a family of mixture models and the best model is selected using the BIC.

Another family of Gaussian mixture models, based on the latent Gaussian mixture model, has more recently been developed. This family of models, known as parsimonious Gaussian mixture models (PGMMs; McNicholas and Murphy, 2008), is based on the mixtures of factor analyzers model (Ghahramani and Hinton, 1997; Tipping and Bishop, 1999a; McLachlan and Peel, 2000b),

\[ h(x \mid \vartheta) = \sum_{g=1}^{G} \pi_g \phi(x \mid \mu_g, \Lambda_g \Lambda'_g + \Psi_g), \]

where \( \Lambda_g \) is a \( p \times q \) matrix of factor loadings and \( \Psi_g \) is a diagonal \( p \times p \) matrix with positive diagonal entries (cf. Section 4.2.1). Imposing the various combinations of the constraints \( \Lambda_g = \Lambda, \Psi_g = \Psi, \) and \( \Psi_g = \psi_g I_p \) leads to a family of eight mixture
models. McNicholas and Murphy (2008) and McNicholas et al. (2010) use the PGMM approach for model-based clustering, while McNicholas (2010) use the PGMMs for model-based classification.

Both the MCLUST and PGMM approaches are based on the Gaussian mixture model. In fact, the Gaussian mixture model has traditionally received most of the attention within the mixture modelling literature. While this is changing year-on-year, exploitation of the most natural alternative — the mixtures of multivariate $t$-distributions — is still relatively rare. McLachlan and Peel (1998) introduced the the multivariate $t$-distribution as a heavy-tailed alternative to the multivariate Gaussian distribution for model-based clustering. The model density for mixtures of multivariate $t$-distributions is given by

$$f_t(x | \theta) = \sum_{g=1}^{G} \pi_g \zeta(x | \mu_g, \Sigma_g, \nu_g),$$

where $\pi_g$ are the mixing proportions and

$$\zeta(x | \mu_g, \Sigma_g, \nu_g) = \frac{\Gamma \left( \left[ \nu_g + p \right] / 2 \right) |\Sigma_g|^{-1/2}}{(\pi \nu_g)^{p/2} \Gamma \left( \nu_g/2 \right) \left[ 1 + \delta(x, \mu_g | \Sigma_g) / \nu_g \right]^{(\nu_g + p)/2}},$$

is the density of a $p$-dimensional multivariate $t$-distribution with mean $\mu_g$, covariance matrix $\Sigma_g$ and degrees of freedom $\nu_g$, where $\delta(x, \mu_g | \Sigma_g) = (x - \mu_g)' \Sigma_g^{-1} (x - \mu_g)$ is the squared Mahalanobis distance between $x$ and $\mu_g$. Almost a decade later, McLachlan et al. (2007) introduced a mixtures of multivariate $t$-factor analyzers model and, even more recently, Andrews and McNicholas (2011a) extended a number of the PGMMs to give a family of six mixtures of multivariate $t$-factor analyzers models, including models with constrained $\nu_g$. In addition, Andrews and McNicholas (2011b)
used mixtures of multivariate $t$-distributions for model-based classification.

In Section 4.2 we introduce modified factor analyzers and subsequently develop a family of mixtures of multivariate $t$-distributions that utilize this covariance structure. We discuss methodology for model-based clustering, classification, and discriminant analysis using this family of models. In total, 24 models are introduced, 18 of which have not appeared in the literature previously. This novel family of mixture models are applied to real data in sections 4.4.2 and 2.3.2 and we close, with discussion and suggestions for further work, in Section 4.5.

4.2 Methodology

4.2.1 Factor Analysis & Variations

Let $X_i$ be a $p$-dimensional random vector. Then, the factor analysis model (Spearman, 1904) can be written

$$X_i = \mu + \Lambda U_i + \epsilon_i,$$

(4.1)

where $\Lambda$ is a $p \times q$ matrix of factor loadings, $\epsilon_i \sim N(0, \Psi)$ is a vector of error terms, and $U_i$ is the vector of factors; here, $q \ll p$. The marginal distribution of $X_i$ arising from this model is $\mathcal{N}(\mu, \Lambda \Lambda' + \Psi)$; this covariance structure is the component covariance structure ($\Sigma_g = \Lambda_g \Lambda_g' + \Psi_g$) in the mixtures of factor analyzers and in the PGMM family of mixture models in general.

The probabilistic principal component analyzers (PPCA) model of Tipping and Bishop (1999b) is a notable variation of the factor analysis model. This model
emerges from the factor analysis model by imposing the isotropic constraint $\Psi = \psi I_p$; this suggests that the variance of the error term $\epsilon_i$ in Equation 4.1 is the same across all $p$ variables. In recent work by McNicholas and Murphy (2010b), another variation of the factor analysis model was introduced by setting $\Psi = \varpi \Delta$, where $\varpi$ is a constant and $\Delta = \text{diag}\{\delta_1, \delta_2, \ldots, \delta_p\}$ such that $|\Delta| = 1$. This is a more sophisticated alternative to the PPCA, which is in fact the special case where $\Delta = I_p$. The marginal distribution of $X_i$ arising from the modified factor analysis model is $N(\mu, \Lambda \Lambda' + \varpi \Delta)$.

McNicholas and Murphy (2010b) used mixtures of modified factor analyzers for model-based clustering. In this case, the model density is the same as that for the PGMMs but with component covariance structure $\Sigma_g = \Lambda_g \Lambda_g' + \varpi_g \Delta_g$. Now, there is the option to constrain this component covariance structure in a number of ways: $\Lambda_g = \Lambda$, $\varpi_g = \varpi$, $\Delta_g = \Delta$, and $\Delta_g = I_p$. Imposing, or not, each of these constraints led McNicholas and Murphy (2010b) to a family of twelve Gaussian mixture models which they called the expanded PGMM (EPGMM) family. The twelve modified factor analysis covariance structures used in the EPGMMs are given in Table 4.1, along with their factor analysis analogues, where applicable.

The covariance structures given in Table 4.1 will be used as component covariance structures for a family of mixtures of multivariate $t$-distributions herein. We will also explore the option of constraining $\nu_g = \nu$, which can lead to more stable estimation of $\nu_g$ (cf. Andrews and McNicholas, 2011a). The resulting family of twenty-four mixture models, which shall be known as mixtures of modified $t$-factor analyzers (MOD$t$), is developed in the following sections.
Table 4.1: The twelve modified factor analysis covariance structures, along with their factor analysis analogues, where applicable, and the number of covariance parameters.

<table>
<thead>
<tr>
<th>Factor Analysis Structure</th>
<th>Modified Structure</th>
<th>Covariance Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma_g = \Lambda \Lambda' + \psi I_p$</td>
<td>$\Sigma_g = \Lambda \Lambda' + \varpi I_p$</td>
<td>$pq - q(q - 1)/2 + 1$</td>
</tr>
<tr>
<td>$\Sigma_g = \Lambda \Lambda' + \psi_g I_p$</td>
<td>$\Sigma_g = \Lambda \Lambda' + \varpi_g I_p$</td>
<td>$pq - q(q - 1)/2 + G$</td>
</tr>
<tr>
<td>$\Sigma_g = \Lambda_g \Lambda'_g + \psi I_p$</td>
<td>$\Sigma_g = \Lambda_g \Lambda'_g + \varpi I_p$</td>
<td>$G[pq - q(q - 1)/2] + 1$</td>
</tr>
<tr>
<td>$\Sigma_g = \Lambda_g \Lambda'_g + \psi_g I_p$</td>
<td>$\Sigma_g = \Lambda_g \Lambda'_g + \varpi_g I_p$</td>
<td>$G[pq - q(q - 1)/2] + G$</td>
</tr>
<tr>
<td>$\Sigma_g = \Lambda \Lambda' + \Psi$</td>
<td>$\Sigma_g = \Lambda \Lambda' + \varpi \Delta$</td>
<td>$pq - q(q - 1)/2 + [G + (p - 1)]$</td>
</tr>
<tr>
<td>$\Sigma_g = \Lambda_g \Lambda'_g + \Psi$</td>
<td>$\Sigma_g = \Lambda_g \Lambda'_g + \varpi \Delta_g$</td>
<td>$G[pq - q(q - 1)/2] + [G + (p - 1)]$</td>
</tr>
<tr>
<td>$\Sigma_g = \Lambda \Lambda' + \Psi_g$</td>
<td>$\Sigma_g = \Lambda \Lambda' + \varpi \Delta_g$</td>
<td>$pq - q(q - 1)/2 + G + (p - 1)$</td>
</tr>
<tr>
<td>$\Sigma_g = \Lambda_g \Lambda'_g + \Psi_g$</td>
<td>$\Sigma_g = \Lambda_g \Lambda'_g + \varpi \Delta_g$</td>
<td>$G[pq - q(q - 1)/2] + [1 + G(p - 1)]$</td>
</tr>
</tbody>
</table>

4.2.2 Three Likelihoods

To facilitate clustering, classification, and discriminant analysis, we let $z_i$ denote the component membership of observation $i$, so that $z_{ig} = 1$ if observation $i$ belongs to group $g$ and $z_{ig} = 0$ otherwise. First, we focus on clustering and so suppose that we observe $n$ $p$-dimensional data vectors $x_1, \ldots, x_n$, all of which have unknown group memberships. Using the entirely unconstrained modified factor analysis covariance structure, the likelihood for the mixtures of modified $t$-factor analyzers model can be written

$$L(\theta \mid x, z) = \prod_{i=1}^n \sum_{g=1}^G \pi_g \zeta(x_i \mid \mu_g, \Lambda_g \Lambda'_g + \varpi_g \Delta_g).$$

Next, suppose that we are operating within the model-based classification paradigm; we have $n$ observations, of which $k$ have known group memberships. Suppose that we order these $n$ observations so that it is the first $k$ that have known group memberships.
Then, the likelihood can be written

\[ L(\theta \mid x, z) = \prod_{i=1}^{k} \prod_{g=1}^{G} \left[ \pi_g \zeta(x_i \mid \mu_g, \Lambda_g \Lambda_g' + \varpi_g \Delta_g) \right]^{z_{ig}} \prod_{j=k+1}^{n} \prod_{h=1}^{G} \pi_h \zeta(x_j \mid \mu_h, \Lambda_h \Lambda_h' + \varpi_h \Delta_h). \]

Clearly, model-based clustering is a special case of model-based classification which arises upon setting \( k = 0 \) within the latter paradigm. Finally, let us consider model-based discriminant analysis. As in the case of model-based classification, we have \( n \) observations, of which \( k \) have known group memberships. However, rather than using all \( n \) observations to estimate the unknown component memberships, we proceed as follows. As before, order the \( n \) observations so that it is the first \( k \) that have known group memberships. Then, form the likelihood

\[ L(\theta \mid x, z) = \prod_{i=1}^{k} \prod_{g=1}^{G} \left[ \pi_g \zeta(x_i \mid \mu_g, \Lambda_g \Lambda_g' + \varpi_g \Delta_g) \right]^{z_{ig}}, \]

from the \( k \) observations with known group memberships; using the maximum likelihood estimates arising from this likelihood, we can then compute the expected values

\[ \hat{z}_{jg} = \frac{\hat{\pi}_g \zeta(x_j \mid \mu_g, \Lambda_g \Lambda_g' + \varpi_g \Delta_g)}{\sum_{h=1}^{G} \hat{\pi}_h \zeta(x_j \mid \mu_h, \Lambda_h \Lambda_h' + \varpi_h \Delta_h)}, \quad (4.2) \]

for \( j = k + 1, \ldots, n \). This expected value plays the role of a discriminant rule and the predicted group memberships are simply given by the maximum \textit{a posteriori} classifications

\[ \text{MAP}\{\hat{z}_{jg}\} = \begin{cases} 1 & \text{if } \max_g \{\hat{z}_{jg}\} \text{ occurs at component } g, \\ 0 & \text{otherwise,} \end{cases} \]

for \( j = k + 1, \ldots, n \).
4.2.3 Parameter Estimation

In the case of model-based discriminant analysis, only the $u_{ig}$ are missing and so a relatively straightforward variation of the expectation-maximization (EM) algorithm (Dempster et al., 1977) will suffice. The EM algorithm is an iterative technique for finding maximum likelihood estimates when data are incomplete. Note that, although the Dempster et al. (1977) reference is widely used for the EM algorithm, it has been pointed out by Titterington et al. (1985, Section 4.3.2), amongst others, that comparable approaches were previously used by Baum et al. (1970), Orchard and Woodbury (1972) and Sundberg (1974). The EM algorithm is based on the complete-data vector, which is the missing data plus the observed data. In the E-step the expected value of the complete-data log-likelihood is computed and then, in the M-step, the complete-data log-likelihood is maximized with respect to the model parameters. These two steps are repeated iteratively until convergence.

The expectation-conditional maximization (ECM) algorithm (Meng and Rubin, 1993) replaces each M-step with a number of conditional maximization steps that can be more computationally efficient. The alternating expectation-conditional maximization (AECM) algorithm (Meng and van Dyk, 1997) is an extension of the ECM algorithm in which the complete-data is permitted to change at each stage. Since, in the case of model-based clustering and classification, the MOD$t$ family will have two sources of missing data; namely, the latent $u_{ig}$ and the unknown group memberships represented by $z_{ig}$, the AECM algorithm will be used. Comprehensive details on the EM algorithm and extensions thereof are given by McLachlan and Krishnan (1997).
Most of the details of parameter estimation henceforth follow in an analogous fashion to that described by Andrews and McNicholas (2011a) and so only outline details are given in most places; the exceptions occurring in the places where the mathematics differs significantly. Consider the model-based clustering application of the MODt family. Following McLachlan and Peel (1998), the random variable $W_{ig} \sim \text{gamma}(\nu_g/2, \nu_g/2)$ is introduced to aid in parameter estimation; consult McLachlan and Peel (1998) for details. McLachlan et al. (2007) deduce that the distribution of observation $x_i$ in component $g$, conditional on the missing data, is given by

$$x_i | u_{ig}, w_{ig}, z_{ig} = 1 \sim \mathcal{N}(\mu_g + \Lambda_g u_{ig}, \Psi_g/w_{ig}),$$

and it follows that the complete-data log-likelihood can be written

$$l_c(\vartheta | x, u) = \sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} \log [\pi_g \gamma(w_{ig} | \nu_g/2, \nu_g/2) \phi(u_{ig} | 0, I_q/w_{ig}) \times \phi(x_i | \mu_g + \Lambda_g u_{ig}, \varpi_g \Delta_g/w_{ig})],$$

where $\gamma(\cdot)$ is the gamma density. To compute the expected value of the complete-data log-likelihood, we need the conditional expectations of terms like $Z_{ig}W_{ig}U_{ig}$ and $Z_{ig}W_{ig}U_{ig}U'_{ig}$. These expected values are given by McLachlan et al. (2007);

$$\mathbb{E}[Z_{ig}W_{ig}U_{ig} | x_i, w_{ig}] = w_{ig}\beta_g(x_i - \mu_g),$$

$$\mathbb{E}[Z_{ig}W_{ig}U_{ig}U'_{ig} | x_i, w_{ig}] = I_q - \beta_g \Lambda_g + w_{ig}\beta_g(x_i - \mu_g)(x_i - \mu_g)'\beta'_g,$$

where $\beta_g = \Lambda_g'(\Lambda_g \Lambda_g' + \Psi_g)^{-1}$.

At each E-step, the $z_{ig}$ and the $w_{ig}$ are replaced by their respective conditional expected values. That is, the $z_{ig}$ are replaced by $\hat{z}_{ig}$ (cf. Equation 4.2) for all


\[ \hat{w}_{ig} = \frac{\nu_g + p}{\nu_g + \delta(x_i, \mu_g \mid \Sigma_g)}, \]

where, again, \( \delta(x_i, \mu_g \mid \Sigma_g) \) is the squared Mahalanobis distance between \( x_i \) and \( \mu_g \).

On the first CM-step, the complete-data consist of the \( x_i \), the \( z_{ig} \), and the \( w_{ig} \). The mixing proportions \( \pi_g \) and component means \( \mu_g \) are updated by

\[ \hat{\pi}_g = \frac{n_g}{n} \quad \text{and} \quad \hat{\mu}_g = \frac{\sum_{i=1}^n \hat{z}_{ig}\hat{w}_{ig}x_i}{\sum_{i=1}^n \hat{z}_{ig}\hat{w}_{ig}}, \]

respectively, where \( n_g = \sum_{i=1}^n \hat{z}_{ig} \). The estimates for the \( \nu_g \) do not exist in closed form but, in the unconstrained case, estimates can be found by setting

\[ 1 - \varphi \left( \frac{\hat{\nu}_g^{\text{new}}}{2} \right) + \log \left( \frac{\hat{\nu}_g^{\text{new}}}{2} \right) + \varphi \left( \frac{\hat{\nu}_g^{\text{old}} + p}{2} \right) + \frac{1}{n_g} \sum_{i=1}^n \hat{z}_{ig}(\log \hat{w}_{ig} - \hat{w}_{ig}) - \log \left( \frac{\hat{\nu}_g^{\text{old}} + p}{2} \right) \]

equal to zero and solving for \( \hat{\nu}_g^{\text{new}} \), where \( \hat{\nu}_g^{\text{old}} \) is the previous estimate of \( \nu_g \) and \( \varphi(\cdot) \) is the digamma function.

On the second CM-step, the complete-data comprise the \( x_i \), the \( z_{ig} \), the \( w_{ig} \), and the \( u_{ig} \). We update \( \Lambda_g \), \( \varpi_g \), and \( \Delta_g \), but the form of the updates will depend upon the covariance structure used (see Table 4.1). Suppose that \( \Sigma_g = \Lambda_g \Lambda_g' + \varpi \Delta_g \), and recall that \( \Delta_g \) is defined so that \( |\Delta_g| = 1 \). This constraint, \( |\Delta_g| = 1 \), will have to be accounted for in the maximization of the expected value of the complete-data log-likelihood; as in McNicholas and Murphy (2010b), we will do this using the method of Lagrange multipliers (Lagrange, 1788). Using \( Q(\Lambda_g, \varpi, \Delta_g) \) to denote the expected value of the complete-data log-likelihood, we form the Lagrangian

\[ L(\Lambda_g, \varpi, \Delta_g, \lambda_g) = Q(\Lambda_g, \varpi, \Delta_g) - \lambda_g(|\Delta_g| - 1), \]
where \( \lambda_g \) is the Lagrange multiplier. Differentiating \( L(\Lambda_g, \varpi, \Delta_g, \lambda_g) \) with respect to \( \Lambda_g, \varpi^{-1}, \Delta_g^{-1}, \) and \( \lambda_g \), respectively, and solving the resulting score functions leads to the updates

\[
\hat{\Lambda}_g^{\text{new}} = S_g \hat{\beta}_g \Theta_g^{-1},
\]

\[
(\hat{\varpi})^{\text{new}} = \frac{1}{p} \sum_{g=1}^{G} \hat{\pi}_g \text{tr}\{\hat{\Delta}_g^{-1}(S_g - \hat{\Lambda}_g^{\text{new}} \hat{\beta}_g S_g)\},
\]

\[
\hat{\Delta}_g^{\text{new}} = \frac{\text{diag}\{S_g - \hat{\Lambda}_g^{\text{new}} \hat{\beta}_g S_g\}}{(\hat{\varpi})^{\text{new}} (1 + 2\lambda_g/n_g)},
\]

\[
\lambda_g = \frac{n_g}{2} \left[ \frac{1}{(\hat{\varpi})^{\text{new}}} \left( \prod_{j=1}^{p} \xi_{gj} \right)^{1/p} - 1 \right],
\]

where \( S_g = (1/n_g) \sum_{i=1}^{n} z_{ig} \hat{w}_{ig}(x_i - \hat{\mu}_g)(x_i - \hat{\mu}_g)' \), the term \( \xi_{gj} \) is given by the \( j \)th element along the diagonal of the matrix \( S_g - \hat{\Lambda}_g^{\text{new}} \hat{\beta}_g S_g \), and we write \( \Theta_g = I_p - \hat{\beta}_g \hat{\Lambda}_g + \hat{\beta}_g S_g \hat{\beta}_g' \).

Parameter estimates in all other cases are analogous to those given in McNicholas and Murphy (2010b), but with \( S_g \) as defined herein. Convergence of these EM algorithms is determined based on an asymptotic estimate of the log-likelihood that follows from Aitken’s acceleration (cf. Aitken, 1926; Böhning et al., 1994; Lindsay, 1995). See McNicholas et al. (2010) for details.

### 4.3 Computational Issues

#### 4.3.1 Model Selection & Performance Assessment

The best member from amongst the family of models introduced here will be determined using the BIC, which is given by \( \text{BIC} = 2l(x, \hat{\Phi}) - m \log n \), where
$m$ is the number of free parameters, $n$ is the sample size, $l(x, \hat{\Phi})$ is the maximized log-likelihood, and $\hat{\Phi}$ is the maximum likelihood estimate of $\Phi$. The use of the BIC for model selection in model-based clustering is well established (cf. Leroux, 1992; Kass and Raftery, 1995; Kass and Wasserman, 1995; Keribin, 1998, 2000).

In the data analyses of Section 4.4, we use the adjusted Rand index (Hubert and Arabie, 1985) to measure class agreement. The Rand index (Rand, 1971) is given by

$$(\text{number of pairwise agreements})/(\text{total number of pairs}).$$

A well known shortcoming of the Rand index is that its expected value is greater than 0 under random classification and so small values of the Rand index are difficult to interpret. The adjusted Rand index corrects the Rand index for chance by accounting for the fact that classification performed randomly would probably correctly classify some cases. The adjusted Rand index has an expected value of 0 under random classification and perfect classification would result in a value of 1.

### 4.3.2 The Woodbury Identity

Implementation of these models was carried out in the R software. Following McLachlan and Peel (2000a), McNicholas and Murphy (2008), and Andrews and McNicholas (2011a), we make use of the Woodbury identity (Woodbury, 1950) in our computations. This identity allows us to avoid inverting any non-diagonal $p \times p$ matrices. Given an $n \times n$ matrix $A$, an $n \times k$ matrix $U$, a $k \times k$ matrix $C$ and a $k \times n$ matrix $V$, the Woodbury identity states that $(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} +
\( V A^{-1} U )^{-1} V A^{-1} \). Now, setting \( U = \Lambda_g \), \( V = \Lambda'_g \), \( A = \varpi_g \Delta_g \), and \( C = I_q \) gives

\[
(\varpi_g \Delta_g + \Lambda_g \Lambda'_g)^{-1} = (\varpi_g \Delta_g)^{-1} - (\varpi_g \Delta_g)^{-1} \Lambda_g [I_q + \Lambda'_g (\varpi_g \Delta_g)^{-1} \Lambda_g ]^{-1} \Lambda'_g (\varpi_g \Delta_g)^{-1},
\]

(4.4)

and even though the left-hand-side of Equation 4.4 involves inversion of a \( p \times p \) matrix, the right hand side leaves only diagonal and \( q \times q \) matrices to be inverted. This presents a major computational advantage, especially when \( p \) is large and \( q \ll p \).

A useful identity for the determinant of the covariance matrix follows, namely

\[
|\Lambda_g \Lambda'_g + \varpi_g \Delta_g| = |\varpi_g \Delta_g| / |I_q - \Lambda'_g (\Lambda_g \Lambda'_g + \varpi_g \Delta_g)^{-1} \Lambda_g|.
\]

### 4.3.3 Initialization

The initialization of the parameters is carried out as described by McNicholas and Murphy (2008). The initialization of the group memberships \( \hat{z}_{ig} \) will depend on the application. In model-based clustering applications, we initialize the \( \hat{z}_{ig} \) using the model-based agglomerative hierarchical routine contained in the \texttt{mclust} package, but for model-based classification applications we follow Andrews and McNicholas (2011b) in initializing the \( \hat{z}_{ig} \) uniformly for observations with unknown group memberships, so that \( \hat{z}_{ig} = 1/G \) for \( j = k + 1, \ldots, n \).
4.4 Data Analyses

4.4.1 Three Real Data Sets

Forina et al. (1986) recorded 28 chemical and physical properties of three types of wine from the Piedmont region of Italy. There are 178 samples of the three varieties: Barolo, Grignolino, and Barbera. A subset of 13 of these variables (Table 4.2) is available in the gclus package (Hurley, 2004).

Table 4.2: Thirteen chemical and physical properties of Italian wines available in the gclus library.

<table>
<thead>
<tr>
<th>Chemical and Physical Properties</th>
<th>Alcohol</th>
<th>Proline</th>
<th>OD$<em>{280}$/OD$</em>{315}$ of diluted wines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malic acid</td>
<td>Ash</td>
<td>Total phenols</td>
<td>Alcalinity of ash</td>
</tr>
<tr>
<td>Hue</td>
<td></td>
<td>Nonflavonoid phenols</td>
<td>Magnesium</td>
</tr>
<tr>
<td>Flavonoids</td>
<td>Color intensity</td>
<td>Proanthocyanins</td>
<td></td>
</tr>
</tbody>
</table>

The crabs data consist of biological measurements on 200 leptograpus crabs collected in Fremantle, Western Australia; 50 male and 50 female, for each of two species; 100 orange and 100 blue. These data, which first appeared in Campbell and Mahon (1974), were sourced from the MASS library for R (cf. Venables and Ripley, 1999). There are five variables, corresponding to the five measurements that were taken on each crab; these measurements are given in Table 4.3.

Table 4.3: Details of the five measurements that were taken on the leptograpus crabs.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>FL</td>
<td>Frontal lobe size in millimeters.</td>
</tr>
<tr>
<td>RW</td>
<td>Rear width in millimeters.</td>
</tr>
<tr>
<td>CL</td>
<td>Carapace length in millimeters.</td>
</tr>
<tr>
<td>CW</td>
<td>Carapace width in millimeters.</td>
</tr>
<tr>
<td>BD</td>
<td>Body depth in millimeters.</td>
</tr>
</tbody>
</table>
The Iris data (Anderson, 1935; Fisher, 1936) are famous throughout the literature and have become a staple in clustering applications. The data contain four measurements — sepal length, sepal width, petal length, and petal width (in centimeters) — on 150 samples of Iris setosa, Iris versicolor, and Iris virginica flowers.

### 4.4.2 Model-Based Classification & Discriminant Analysis

We analyze each of the data sets mentioned in Section 4.4.1. To simulate a classification scenario, 25 subsets of 50% were chosen as the training set randomly in each case and then analyzed using model-based classification and discriminant analysis techniques. For all the discriminant analyses run herein, note that the maximum number of components per each known group was restricted to four. The classification performances for all methods are given in Table 4.4.

Table 4.4: Adjusted Rand indices for model-based classification (MBC) and discriminant analysis (DA) techniques applied to all three real data sets for 25 runs with 50% known.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Wine</th>
<th>Iris</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODt MBC</td>
<td>0.96</td>
<td>0.92</td>
</tr>
<tr>
<td>EPGMM MBC</td>
<td>0.95</td>
<td>0.93</td>
</tr>
<tr>
<td>MCLUST MBC</td>
<td>0.89</td>
<td>0.90</td>
</tr>
<tr>
<td>MODt DA</td>
<td>0.71</td>
<td>0.89</td>
</tr>
<tr>
<td>EPGMM DA</td>
<td>0.88</td>
<td>0.89</td>
</tr>
<tr>
<td>MCLUST DA</td>
<td>0.79</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Top classification performance is achieved by a model-based classification technique for both of the data sets considered. The modified $t$-factor structure is selected in a number of runs for the Iris data set, and results in an improvement in the adjusted Rand from 0.89 (under the 16 model $t$-factor analyzers family) to 0.93.
Both modified factor analyzer families perform similarly in model-based classification on the two data sets considered, achieving greater classification performance than that of the MCLUST family of models.

It is important to note the superiority of the model-based classification technique over the discriminant analysis method across the board. For each family, model-based classification with that family outperforms the discriminant analysis from the same family. This lends further credence to the assertion that model-based classification is a better alternative.

### 4.4.3 Model-Based Clustering

The results from the application of all model-based clustering techniques to both data sets are given in Table 4.5.

<table>
<thead>
<tr>
<th></th>
<th>Wine</th>
<th>Iris</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ARI G q Cov</td>
<td>ARI G q Cov</td>
</tr>
<tr>
<td>MOD$t$</td>
<td>0.96 3 4 $\Lambda\Lambda'$ + $\varpi_g\Delta_g$ 16.6</td>
<td>0.87 3 2 $\Lambda_g\Lambda_g'$ + $\varpi\Delta$ 31.3</td>
</tr>
<tr>
<td>EPGMM</td>
<td>0.80 4 4 $\Lambda\Lambda'$ + $\varpi_g\Delta_g$ –</td>
<td>0.87 3 2 $\Lambda_g\Lambda_g'$ + $\varpi\Delta$ –</td>
</tr>
<tr>
<td>MCLUST</td>
<td>0.90 3 – $\lambda_gA_g$ –</td>
<td>0.57 2 – $\lambda_gD_gA_gD_g'$ –</td>
</tr>
</tbody>
</table>

The clustering analyses given here show that the MOD$t$ family is capable of outperforming the EPGMM family (in the case of the wine data set) as well as equalling the EPGMM family when appropriate (with the iris data set). For both data sets, a model with constrained degrees of freedom is chosen by the BIC from the MOD$t$ family; the respective degrees of freedom parameters are given as 16.6 for the wine data and 31.3 for the iris data.
4.5 Discussion

A novel family of mixture models, based on mixtures of modified $t$-factors, was introduced. This technique, known as MOD$t$, was used for model-based clustering, classification, and discriminant analysis. In addition, a Gaussian model-based technique, the EPGMMs, was used for model-based classification and discriminant analysis for the first time.

Future work will investigate on efficient implementation of these models, both in serial and in parallel, after the fashion of McNicholas et al. (2010). This will mean that the methodology described herein can be disseminated to the wider scientific community. Work will also be conducted on restricting the estimated degrees of freedom to one of a few values (cf. Besag et al., 1995).

Acknowledgements

This work was supported by a Discovery Grant (McNicholas) and a Postgraduate Doctoral Scholarship (Andrews) from the Natural Sciences & Engineering Research Council of Canada; by a grant from the Canada Foundation for Innovation — Leaders Opportunity Fund; and from the Ontario Ministry for Research & Innovation — Small Infrastructure Fund.
Part II

Variable Selection
Chapter 5

Variable Selection for Clustering and Classification

Moving away from the development of families, we now consider another issue that can be found in the field of clustering/classification: the fact that, in the vast majority of data sets, not all variables are active in cluster formation. As a consequence of this, removing uninformative or noisy variables from the data set can, in many cases, drastically improve clustering performance.

The article that follows is currently under review following revisions at the *Journal of Classification*, and is reprinted in its entirety.

5.1 Introduction

Variable selection is an important feature of many types of statistical analyses, including clustering and classification. The use of variable selection techniques can facilitate both model fitting and the interpretation of the results. With the continued growth in data dimensionality, the importance of efficient variable selection techniques is increasing. In this paper, we put forward a flexible variable selection technique that can be used in unsupervised, semi-supervised, or fully supervised classification contexts. We focus largely on the unsupervised format (i.e., clustering) and
specifically the usage of model-based techniques to guide the algorithm. However, the algorithm’s applicability under a classification format will be briefly illustrated in Section 5.4.

Model-based clustering is often used on high-dimensional data sets, such as those found in the field of bioinformatics. Though clustering on very large data sets is possible, it is difficult to execute for a number of reasons. The most obvious reason is time: as the dimensionality of the data increases, the number of parameters requiring estimation increases, often in a quadratic fashion. Of more importance, however, is that the human brain (accustomed to three-dimensions visually, plus a few other senses) is not prepared to understand dimensionality that can run well into the thousands. Thus, to facilitate interpretation of high-dimensional data sets, determining which variables are most active in cluster formation is important. A final consideration is the cost in creating high-dimensional data sets, which can be enormous; knowing which variables are important for differentiating between groups-of-interest can save both time and money.

Although algorithm efficiency is certainly a worthwhile reason in its own right, variable selection techniques can drastically improve clustering performance. This is achieved through eliminating noisy variables that can cloud the clustering algorithm’s ability to distinguish groups. Unfortunately, variable selection techniques do not necessarily improve clustering performance; it is, therefore, important that an inferior reduced-variable solution is not chosen over a solution on the full variable set (cf. Section 5.5).

In Section 5.2, we conduct a short review of comparable variable selection
techniques and other relevant background material. Then we discuss our methodology (Section 5.3), before running simulations (Section 5.4) and real-data examples (Section 5.5). Finally, we conclude with a summary and suggestions for future work (Section 5.6).

5.2 Background

A number of dimensionality reduction techniques are available to researchers interested in clustering data sets. For the purposes of microarray data sets, the select-genes procedure from the EMMIX-GENE software (McLachlan et al., 2002) fits multi-component mixture models to each variable and then calculates the likelihood ratio test statistic between these and the one-component model. Unfortunately, fitting mixture models to each individual variable is time consuming.

Another variable selection technique is given by Raftery and Dean (2006), whereby multiple models from the MCLUST family are compared using approximate Bayes factors (Kass and Raftery, 1995). This variable selection technique is readily available via the clustvarsel package (Dean and Raftery, 2006) in R. However, because some of the MCLUST models require the estimation of a number of parameters that is quadratic in the data-dimensionality, the clustvarsel package can be very slow in high-dimensions. Furthermore, the application of clustvarsel can sometimes lead to inferior results when compared to the use of mclust alone (cf. McNicholas and Murphy, 2008). A related approach, denoted selvarclust, is taken by Maugis et al. (2009), where the assumptions on the role of variables are relaxed with the potential
benefit of avoiding the over-penalization of independent variables.

In addition to these procedures, a number of implicit and explicit variable selection procedures are built into model-based clustering algorithms. Implicit variable selection procedures include approaches such as mixtures of factor analyzers (cf. Ghahramani and Hinton, 1997; Tipping and Bishop, 1999a; McLachlan and Peel, 2000b; McNicholas and Murphy, 2008, 2010c; Andrews and McNicholas, 2011a,b). An explicit dimensionality reduction approach is taken in some recent work by Scrucca (2010) and Bouveyron and Brunet (2012); the latter also gives a summary of other work in the area of dimensionality reduction with respect to clustering.

For the purposes of variable selection within a clustering context, the desire is to find variables that show differentiation between the \textit{a priori} unknown groups and eliminate variables that do not. The variable selection method introduced herein seeks precisely this, and is flexible enough to implement using a variety of clustering/classification techniques.

5.3 Methodology

5.3.1 Introduction

Variable selection for clustering and classification (VSCC) is intended to find the variables that simultaneously minimize the ‘within-group’ variance and maximize the ‘between-group’ variance. The combination of these two criteria will give variables that best show separation between the desired groups. Note that the within-group
variance for each variable \( j = 1, \ldots, p \) can be written as

\[
W_j = \frac{\sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} (x_{ij} - \mu_{gj})^2}{n},
\]

where \( x_{ij} \) is observation \( i \) on variable \( j \), \( \mu_{gj} \) is the mean of variable \( j \) in group \( g \), \( n \) is the number of observations, and \( z_{ig} \) is a group membership indicator variable defined so that

\[
z_{ig} = \begin{cases} 
1 & \text{if observation } x_i = (x_{i1}, \ldots, x_{ip}) \text{ belongs to cluster } g, \\
0 & \text{otherwise.}
\end{cases}
\]

The leftover variance within variable \( j \) not accounted for by \( W_j \), or \( \sigma^2_j - W_j \) in the common notation, is then a measure of the variance between groups. In general, calculation of this value will be necessary. However, if the data have been standardized to have equal variance across variables, then any variable minimizing the within-group variance is also maximizing the leftover variance.

The VSCC method utilized in this article will be applied to data that have been standardized to have mean 0 and variance 1 and, as such, calculation of the \( W_j \) is sufficient. In addition to the variance calculations, our method uses the correlation between variables; we let \( \rho_{ij} \) denote the correlation between variables \( i \) and \( j \). The actual implementation of VSCC procedures depends on the form of the data; in our analyses, we consider examples where no memberships are known (clustering) as well as where some observations have known membership (classification). Specifics regarding the implementation of the VSCC algorithm under each format can be found in Sections 5.3.5 and 5.3.6. In the sections that immediately follow, we motivate and describe the VSCC algorithm.
5.3.2 A Motivating Example

VSCC will proceed in a step-wise fashion after calculating the within-group variances. The first variable selected is the variable with the minimum $W_j$. One way to select from the remaining variables is by using simple, user-specified thresholding. For example, by sorting the $W_j$ in ascending order we could consider each variable in a step-wise manner and select those variables with $W_j$ less than some value $w$ where all $|\rho_{jr}|$ are also less than some value $c$, $\forall r \in V$; here, $V$ is the set of previously selected variables. While this approach could be useful, it requires the user to adjust the algorithm to maximize its effectiveness.

An additional concern behind this type of selection criterion can be shown via a simple example. Consider a three-dimensional data set where the first variable minimizes $W_j$ and so is already selected. Suppose that the remaining two variables can be summarized as follows.

- Variable 2: $W_2 = 0.6$ and $|\rho_{12}| = 0.75$.
- Variable 3: $W_3 = 0.2$ and $|\rho_{13}| = 0.75$.

Suppose we simply use the thresholds described previously. In the current example, if the correlation threshold was set at $c = 0.70$ then both variables would be considered equally ‘bad’ and neither would be selected. However, if the correlation threshold was set at $c = 0.80$, and assuming both that $|\rho_{23}| < 0.8$ and $w > 0.6$, then both variables would be selected. Under the argument that within-group variance is our primary concern and correlation is a secondary concern, we propose that, in this scenario, retaining Variable 3 and eliminating Variable 2 would be desirable. To this end, we
need to go beyond simple, user-specified thresholding.

5.3.3 The VSCC Method

We have illustrated a desire for a sliding correlation threshold that is more forgiving for small values of $W_j$ and more stringent for larger values. Thus, we seek to define a relationship between the within-group variance and between-variable correlation that properly expresses this goal. As a first attempt, we consider a linear relationship between the two quantities. Let $V$ represent the space of currently selected variables, then we select variable $j$ if for all $r \in V$,

$$|\rho_{jr}| < 1 - W_j.$$ 

Other potential relationships will be discussed shortly, but utilizing this relationship we can write the VSCC algorithm as follows:

1. Calculate within-group variances $W_j$.
2. Sort $W_j$ in ascending order, denote this sorted list $W_s$.
3. $W_1$ minimizes $W_s$ and is automatically selected and placed into the set of selected variables $V$. Set count $k = 2$.
4. If $|\rho_{kr}| < 1 - W_k$, for all $r \in V$, variable $s = k$ is placed into $V$.
5. If $k < p$, set $k = k + 1$ and return to Step 4. Else end algorithm.

The linear relationship defined in Step 4 of the VSCC algorithm might be too strong a criterion. For instance, a variable with within-group variation of 0.25
and correlation of 0.76 with one of the previously selected variables would be rejected. Given the interval that correlation values (and the $W_j$ when the data are standardized) will lie on, a simple fix is to consider relationships of order greater than one (Table 5.1); a visualization of these criteria is given in Figure 5.1.

Table 5.1: List of variance-correlation relationships considered for implementation into Step 4 of the VSCC algorithm.

| Relationship | $|\rho_{kr}| < 1 - W_k$ |
|-------------|------------------------|
| Linear      | $|\rho_{kr}| < 1 - W_k^2$ |
| Quadratic   | $|\rho_{kr}| < 1 - W_k^3$ |
| Cubic       | $|\rho_{kr}| < 1 - W_k^4$ |
| Quartic     | $|\rho_{kr}| < 1 - W_k^5$ |

![Figure 5.1](image)

Figure 5.1: Graphical representation of the selection criteria considered for implementation into Step 4 of the VSCC algorithm.

Many different relationships could be defined between the within-group variance and the between-variable correlation. However, many of these relationships would be intuitively silly. For example, any relationship that will allow a variable
with $W_j = 1$ to be selected should not be allowed. Also, any relationship that results in impossible values (those outside of the interval $[0,1]$) need not be considered. Obviously, piecewise relationships could be considered that solve some of these issues for more complicated relationships. However, we consider that the relationships in Table 5.1 constitute a relatively thorough, common-sense handling of the issue of variance-correlation relationships.

5.3.4 Subset Selection

Using multiple criteria for selecting variables will naturally lead to multiple subsets of variables — as many as five solutions under the current relationship structure. Under a clustering framework, one must define a method for choosing between these subsets without specific knowledge of which subset produces the best classifier. One could develop a Bayes factors framework similar to that used by clustvarsel (Raftery and Dean, 2006) to compare variable subsets. However, this approach is complicated by the fact that subsets will not necessarily differ by only one variable, as would happen in a truly step-wise approach. Instead, we introduce a novel approach to selecting variable subsets that relies on one of the major strengths of model-based clustering/classification: measuring the uncertainty of the classification.

The uncertainty for each observation is found simply through the fuzzy classification matrix; i.e., the $n \times G$ matrix containing the $\hat{z}_{ig}$. Each $\hat{z}_{ig}$ element of this matrix is a measure of the strength of evidence indicating observation $i$ belongs to group $g$. For well-defined clusters, the $\hat{z}_{ig}$ will all be approximately equal to 0, with one entry per row $i$ being approximately equal to 1. We take the uncertainty to
be the sum of all the $\hat{z}_{ig}$ entries, except the $\max_g \{\hat{z}_{ig}\}$ entries. This can be expressed as $\sum_{i=1}^{n} \sum_{g=1}^{G} \hat{z}_{ig} - \max_{g} \{\hat{z}_{ig}\}$ or equivalently as $n - \sum_{i=1}^{n} \max_{g} \{\hat{z}_{ig}\}$.

Selecting the variable subset that minimizes the uncertainty in the classification suggests we will be selecting the variables which produce the strongest group structure, so there is some intuitive appeal. In some ways, this is a large departure from information-based criteria, but the uncertainty is used in the calculation of the integrated completed likelihood by Biernacki et al. (2000), which marries the uncertainty and the Bayesian information criterion (BIC; Schwarz, 1978). The efficacy of using the uncertainty as a relationship selection criterion will be shown in Sections 5.4 and 5.5.

The authors note that one concern about using the uncertainty calculations is that, by definition, the uncertainty for any $G = 1$ solution will be 0. It is considered a strength within the clustering field that model-based clustering using the BIC can consider $G = 1$ as a solution and inform the user that there are, in fact, no groups in the data. Unfortunately, given a $G = 1$ solution, the VSCC algorithm cannot be computed as described in this paper. As such, it is an implicit assumption by even running VSCC that $G > 1$ groups exist. Tying into this assumption, it is therefore somewhat reasonable to ignore any variable subsets that produce $G = 1$ as a solution. We recognize this is an unfortunate consequence of utilizing model uncertainty as a subset selection device, and we leave this matter as the subject of future research.

Philosophically, it makes sense to approach variable selection under a “do no harm” mentality. In this vein, note that because we can calculate the uncertainty from the original (non-feature reduced) data set, this solution can be considered as
part of the variable selection process. In other words, under VSCC if the full data set results in the minimum uncertainty, we can select the full data set rather than a reduced set; note that for illustrative reasons we will ignore this ability during the simulations in Section 5.4.

5.3.5 Clustering

In a clustering scenario, the values of indicator variables $z_{ig}$ are unknown for all $i = 1, \ldots, n$. The VSCC method needs these variables to compute the $W_j$. However, an initial run of clustering can be used to give ‘good’ estimates of the component memberships $\hat{z}_{ig}$; any clustering approach could be used (e.g., agglomerative hierarchical, model-based, $k$-means).

Herein, we focus on the use of a model-based clustering procedure to initialize VSCC, specifically the *mclust* algorithm (Fraley and Raftery, 2003). Note that the applications in Section 5.5 will incorporate ‘hard’ (0’s and 1’s in the context of $\hat{z}_{ig}$) initializations from a model-based clustering technique, though ‘soft’ (‘fuzzy’ or probabilistic) classifications could be easily incorporated into the procedure instead.

Under the clustering format, the algorithm runs as follows:

1. Perform *mclust* under default settings.

2. Use the resultant (hard) $\hat{z}_{ig}$ to initialize VSCC.

3. Perform *mclust* on the (up to) five variable subsets (from the five relationships in Section 5.3.3) given by VSCC, selecting the best model via the BIC in each case.
4. Select the best variable subset according to the total model uncertainty, and report the results from mclust on that subset.

5.3.6 Classification

In a classification scenario, a subset of the $z_{ig}$ is known and can be utilized by the VSCC method to compute the $W_j$ in two potential ways. The first option is to use only the known $z_{ig}$ to calculate the $W_j$. The other option is to calculate the $W_j$ in a more semi-supervised format, where an original classification algorithm is run to find good estimates of the unknown $\hat{z}_{ig}$. The algorithms for both options follow.

**Supervised Algorithm**

1. Use only the known $z_{ig}$ to initialize VSCC.

2. Perform model-based classification on the (up to) five variable subsets given by VSCC, selecting the best model via the BIC in each case.

3. Select the best variable subset according to the total model uncertainty, and report the results from model-based classification on that subset.

**Semi-Supervised Algorithm**

1. Perform model-based classification to estimate the unknown $\hat{z}_{ig}$.

2. Use both the known $z_{ig}$ and the (hard) estimated $\hat{z}_{ig}$ to initialize VSCC.

3. Perform model-based classification on the (up to) five variable subsets given by VSCC, selecting the best model via the BIC in each case.
4. Select the best variable subset according to the total model uncertainty, and report the results from model-based classification on that subset.

5.3.7 Clustering/Classification Performance

The performance of a clustering algorithm, with respect to known groups present in the data, can be measured in a number of ways. Misclassification rates are often used, but this measure cannot be meaningfully interpreted unless we know the correct number of groups or the clustering algorithm chooses the correct number of groups, which is not always the case. An alternative is to use the Rand index (Rand, 1971), which is calculated as the number of pairwise agreements (between the estimated and known groups) divided by the number of pairs. Because the Rand index is tricky to interpret for low values, the adjusted Rand index was introduced by Hubert and Arabie (1985). It essentially accounts for the fact that two random groupings will have some pairwise agreements, thus making the adjusted Rand index equal to 0 for random clustering and 1 for perfect clustering.

5.4 Simulations

To determine the performance of VSCC under a variety of scenarios, we introduce several simulation studies. The clusterGeneration package (Qiu and Joe, 2006) from R is used to simulate data sets.
5.4.1 Increased Dimension

Herein we investigate how VSCC reacts to increased dimension. The `genRandomClust` function from `clusterGeneration` is used to generate data sets with 4 groups, with between 100 to 150 observations per group and where `sepVal = 0.7` (well separated groups). We generate under dimension size 45, 90, 120, and 150, each containing 33% noisy variables. 250 replicates of each data set are generated and analyzed using VSCC and `mclust` (under default settings). Summary results are given in Table 5.2.

Table 5.2: Summary of results from `mclust` and VSCC on the increased dimension simulations (250 runs per dimension size).

<table>
<thead>
<tr>
<th></th>
<th>$d = 45$</th>
<th>$d = 90$</th>
<th>$d = 120$</th>
<th>$d = 150$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mclust Mean Adj Rand</strong></td>
<td>0.79</td>
<td>0.36</td>
<td>0.30</td>
<td>0.23</td>
</tr>
<tr>
<td><strong>mclust SD Adj Rand</strong></td>
<td>0.15</td>
<td>0.14</td>
<td>0.17</td>
<td>0.17</td>
</tr>
<tr>
<td><strong>mclust Avg Runtime (sec)</strong></td>
<td>5.16</td>
<td>14.74</td>
<td>72.01</td>
<td>157.82</td>
</tr>
<tr>
<td><strong>VSCC Mean Adj Rand</strong></td>
<td>0.99</td>
<td>0.84</td>
<td>0.76</td>
<td>0.57</td>
</tr>
<tr>
<td><strong>VSCC SD Adj Rand</strong></td>
<td>0.03</td>
<td>0.20</td>
<td>0.30</td>
<td>0.33</td>
</tr>
<tr>
<td><strong>VSCC Avg Runtime (sec)</strong></td>
<td>25.95</td>
<td>42.89</td>
<td>160.04</td>
<td>265.62</td>
</tr>
</tbody>
</table>

VSCC performs stronger than `mclust` alone on all dimension sizes considered, according to mean adjusted Rand index. We do, however, note an increase in the standard deviation of the adjusted Rand as the dimension size increases. This is due, in large part, to an increased number of $G = 1$ solutions given by `mclust` for an initialization (which is counted as an adjusted Rand index of 0 for both `mclust` and VSCC). Because `mclust` performance is on average closer to 0, these $G = 1$ examples affect its standard deviation to a lesser extent.

Note also the increase in runtime for both procedures. Keep in mind that
VSCC runs \texttt{mclust} once on the full data set, and then up to five times on reduced-variable data sets. Thus, a large savings in computation time could be achieved by at least initializing VSCC using a faster clustering technique (\textit{k}-means, for example). To illustrate this point, if the initializations were given to VSCC ‘free-of-charge’ for the \( d = 150 \) simulations, VSCC’s average runtime would be merely 98.8 seconds.

\subsection*{5.4.2 Increased Number of Groups}

In this simulation, we investigate how VSCC reacts to different numbers of groups present in the data. Once again, the \texttt{genRandomClust} function is used to generate 250 replicates of each data set. In this study, we simulate under mostly default conditions — which includes \texttt{sepVal}=0.01, or not well separated groups — with 10 noisy and 10 non-noisy variables in each data set. Importantly, we generate data sets for each of \( G = 2, 4, 6, 8, 15, 20 \), which are then analyzed using VSCC and \texttt{mclust} (under default settings except for \( G = 15 \) and \( G = 20 \) data sets, where \texttt{mclust} is forced to consider \( G = 10, \ldots, 20 \)). Summary results are given in Table 5.3. We also provide Table 5.4 for more in-depth details on the \( G = 2 \) simulation to illustrate the specific variance-correlation relationships as well as the performance of choosing relationships via the total model uncertainty.

As the number of groups increase, the general trend for \texttt{mclust} is a reduction in clustering performance from 0.88 (\( G = 2 \)) to 0.73 (\( G = 20 \)), coupled with an increase in average runtime (2.47 to 197.31 seconds, respectively). Interestingly though, VSCC’s performance remains remarkably consistent, and arguably improves (ignoring \( G = 2 \)) as the number of groups increases to 20 (due to a reduction in
Table 5.3: Summary of results from \texttt{mclust} and VSCC on the varied number of groups simulations (250 runs per group structure).

<table>
<thead>
<tr>
<th></th>
<th>(G = 2)</th>
<th>(G = 4)</th>
<th>(G = 6)</th>
<th>(G = 8)</th>
<th>(G = 15)</th>
<th>(G = 20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{mclust} Mean Adj Rand</td>
<td>0.88</td>
<td>0.79</td>
<td>0.80</td>
<td>0.75</td>
<td>0.74</td>
<td>0.73</td>
</tr>
<tr>
<td>\texttt{mclust} SD Adj Rand</td>
<td>0.08</td>
<td>0.17</td>
<td>0.11</td>
<td>0.13</td>
<td>0.06</td>
<td>0.05</td>
</tr>
<tr>
<td>\texttt{mclust} Avg Runtime (sec)</td>
<td>2.47</td>
<td>6.89</td>
<td>10.76</td>
<td>16.59</td>
<td>97.51</td>
<td>197.31</td>
</tr>
<tr>
<td>VSCC Mean Adj Rand</td>
<td>0.89</td>
<td>0.82</td>
<td>0.85</td>
<td>0.83</td>
<td>0.84</td>
<td>0.82</td>
</tr>
<tr>
<td>VSCC SD Adj Rand</td>
<td>0.06</td>
<td>0.17</td>
<td>0.08</td>
<td>0.08</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>VSCC Avg Runtime (sec)</td>
<td>8.95</td>
<td>18.21</td>
<td>22.87</td>
<td>31.82</td>
<td>171.39</td>
<td>302.77</td>
</tr>
</tbody>
</table>

However, VSCC does suffer a similar fate in runtime due to its reliance on \texttt{mclust}.

Several things stand out in the results presented in Table 5.4. For one, the linear relationship (which is also the most stringent of those considered) performs terribly under this simulation. Fortunately, the rest of the relationships put up solid performances and, in fact, very similar performances in general, as three of the four select six variables most often. Interestingly, no one relationship on its own would outperform \texttt{mclust} on the full data set via either mean adjusted Rand index or standard deviation; by choosing the best relationship via the total model uncertainty.

Table 5.4: Detailed results from \texttt{mclust} and VSCC on the 250 \(G = 2\) simulations (10 noisy variables and 10 non-noisy variables). Mode # Vars includes the number of occurrences in parentheses.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Mean AR</th>
<th>SD AR</th>
<th>Mode # Vars</th>
<th>Mean Unc</th>
<th>SD Unc</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{mclust} Linear</td>
<td>0.88</td>
<td>0.08</td>
<td>20 (250)</td>
<td>5.88</td>
<td>3.81</td>
</tr>
<tr>
<td>VSCC Linear</td>
<td>0.40</td>
<td>0.30</td>
<td>2 (194)</td>
<td>14.45</td>
<td>13.27</td>
</tr>
<tr>
<td>VSCC Quadratic</td>
<td>0.81</td>
<td>0.09</td>
<td>3 (75)</td>
<td>9.67</td>
<td>5.26</td>
</tr>
<tr>
<td>VSCC Cubic</td>
<td>0.86</td>
<td>0.08</td>
<td>6 (74)</td>
<td>7.32</td>
<td>4.45</td>
</tr>
<tr>
<td>VSCC Quartic</td>
<td>0.87</td>
<td>0.09</td>
<td>6 (82)</td>
<td>6.67</td>
<td>4.29</td>
</tr>
<tr>
<td>VSCC Quintic</td>
<td>0.88</td>
<td>0.09</td>
<td>6 (71)</td>
<td>6.27</td>
<td>4.21</td>
</tr>
<tr>
<td>VSCC (min unc)</td>
<td>0.89</td>
<td>0.06</td>
<td>6 (75)</td>
<td>5.73</td>
<td>2.86</td>
</tr>
</tbody>
</table>
however, the VSCC algorithm does narrowly beat out the full data set in both categories. This lends support to the use of uncertainty as a selection method. This is not the only simulation where we see results such as this, but it is not universally true across all simulations. We note further support for using the uncertainty while discussing the real data applications in our concluding paragraphs.

5.4.3 Classification Example

To briefly demonstrate the feasibility of VSCC in a classification scenario, we apply the method under the supervised algorithm (cf. Section 5.3.6) to the $G = 15$ simulated data from the previous section. For each data set, we randomly select 50% of the data to have known membership and analyze using model-based classification with the MCLUST family of models, then compare these results to using VSCC (with the same model-based classification on the chosen variables). This is performed on the 250 data sets, and a summary of classification performance can be deduced through Figure 5.4.3. Note that the adjusted Rand index reported only considers the observations with ‘unknown’ cluster membership.

The mean adjusted Rand index for analysis on the full data set is 0.81 with a 0.05 standard deviation, while the VSCC reduced data set achieves a mean of 0.85 with a 0.02 standard deviation. Recall the $G = 15$ simulated data set contains 10 non-noisy and 10 noisy variables. VSCC (under the supervised algorithm with 50% known) picks out the 10 meaningful variables 226 times, or on over 90% of the data sets considered.
Figure 5.2: Histograms of classification performance on the $G = 15$ simulated data set by model-based classification and model-based classification with a reduced feature set selected by VSCC.

5.5 Applications

5.5.1 Introduction

The VSCC method will now be applied to real data sets under a clustering framework. An introduction to each of the four data sets is given at the beginning of each subsection. To facilitate interpretation, VSCC will be compared to the popular variable selection method introduced by Raftery and Dean (2006) and available as the clustvarsel package in R, as well as the selvarclust technique introduced by Maugis et al. (2009) and available as a command-line addition to the MIXMOD software (Biernacki et al., 2006). Recall from Section 5.3.5, we utilize VSCC under an mclust framework, meaning we use mclust to initialize the $\hat{z}_{ig}$ and to cluster the feature-reduced data sets. In VSCC, we will utilize mclust under the default settings: mclust considers $G = 1, \ldots, 9$. Hence, for fairness of comparison, clustvarsel and
selvarclust will be set to consider $G = 1, \ldots, 9$ as well. Note that all methods will be run on standardized variable (mean 0, variance 1) versions of the data discussed. Finally, selvarclust will be restricted to the covariance parameterizations available in mclust, again for fairness of comparison. Even so, due to a difference in initializations used by the MIXMOD software versus mclust, the results presented from selvarclust are not as directly comparable to the other two variable selection methods.

5.5.2 Leptograpsus Crabs

The Leptograpsus crabs data set can be found in the MASS library in R. It contains five length measurements on two different colour forms of the crabs, further separated into the two genders.

Table 5.5: Table of results from mclust, VSCC, clustvarsel, and selvarclust on the crabs data set. The relationship chosen for VSCC by the total model uncertainty is given in parentheses.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Adj Rand</th>
<th>Time (sec)</th>
<th>G</th>
<th>#Vars</th>
<th>Unc</th>
</tr>
</thead>
<tbody>
<tr>
<td>mclust</td>
<td>0.31</td>
<td>3.94</td>
<td>4</td>
<td>5</td>
<td>14.71</td>
</tr>
<tr>
<td>VSCC (Quintic)</td>
<td>0.76</td>
<td>12.49</td>
<td>5</td>
<td>4</td>
<td>10.96</td>
</tr>
<tr>
<td>clustvarsel</td>
<td>0.76</td>
<td>63.01</td>
<td>5</td>
<td>4</td>
<td>10.96</td>
</tr>
<tr>
<td>selvarclust</td>
<td>0.50</td>
<td>256.69</td>
<td>5</td>
<td>4</td>
<td>12.79</td>
</tr>
</tbody>
</table>

The results for the mclust initialization, VSCC, clustvarsel, and selvarclust on the crabs data set are given in Table 5.5. On this data set, clustvarsel (via approximate Bayes factors) and VSCC (via total uncertainty) agree on the solution that eliminates one variable and increases the adjusted Rand index from 0.31 to 0.76. In fact, the selvarclust algorithm selects the same variables, but
the results are very different due to the initializations used. We can, for all intents and purposes, consider the performance of all techniques equivalent. The main item of interest here is that VSCC accomplishes this task five times faster than clustvarsel and over 20 times faster than selvarclust.

Perhaps a more important aspect of this application is that VSCC manages to increase clustering performance off of a poor initialization (0.31 adjusted Rand, with the correct number of groups, to 0.76 adjusted Rand with group over-estimation). One argument against an approach such as VSCC could be that one might need ‘quite good’ initializations for the technique to be useful; the crabs data set, however, shows that this is not necessarily the case. Note that mclust actually chooses the correct number of groups, meaning that the adjusted Rand index is not artificially deflated by choice of large $G$; it is simply a poor clustering performance.

5.5.3 Italian Wine

The Italian wine data set is readily available in the gclus library in R and contains 13 chemical measurements on 178 samples of wine originating from three different varieties (Barolo, Grignolino, and Barbera).

Table 5.6: Table of results from mclust, VSCC, clustvarsel, and selvarclust on the wine data set. The relationship chosen for VSCC by the uncertainty is given in parentheses.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Adj Rand</th>
<th>Time (sec)</th>
<th>G</th>
<th>#Vars</th>
<th>Unc</th>
</tr>
</thead>
<tbody>
<tr>
<td>mclust</td>
<td>0.48</td>
<td>1.91</td>
<td>8</td>
<td>13</td>
<td>5.85</td>
</tr>
<tr>
<td>VSCC (Quartic)</td>
<td>0.90</td>
<td>10.25</td>
<td>3</td>
<td>9</td>
<td>0.90</td>
</tr>
<tr>
<td>clustvarsel</td>
<td>0.78</td>
<td>113.95</td>
<td>3</td>
<td>5</td>
<td>2.23</td>
</tr>
<tr>
<td>selvarclust</td>
<td>0.54</td>
<td>2220.42</td>
<td>7</td>
<td>8</td>
<td>6.35</td>
</tr>
</tbody>
</table>
From the results (Table 5.6), we can see that VSCC outperforms the clustering done by \texttt{mclust} alone as well as those done by \texttt{clustvarsel} and \texttt{selvarclust}. Running \texttt{mclust} on the variables selected by \texttt{selvarclust} results in the same performance as listed for \texttt{selvarclust}. On top of outperforming \texttt{clustvarsel} and \texttt{selvarclust} in clustering performance, VSCC runs over 10 and 200 times faster, respectively.

### 5.5.4 Swiss bank notes data

The Swiss bank notes data set is also available in the \texttt{gclus} library in \texttt{R} and contains six measurements on 200 monetary bills, of which some are legal tender and others are counterfeit notes.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Adj Rand</th>
<th>Time (sec)</th>
<th>G</th>
<th>#Vars</th>
<th>Unc</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{mclust}</td>
<td>0.68</td>
<td>2.34</td>
<td>4</td>
<td>6</td>
<td>6.16</td>
</tr>
<tr>
<td>VSCC (Quadratic)</td>
<td>0.85</td>
<td>8.52</td>
<td>3</td>
<td>4</td>
<td>0.17</td>
</tr>
<tr>
<td>\texttt{clustvarsel}</td>
<td>0.67</td>
<td>66.18</td>
<td>4</td>
<td>5</td>
<td>6.10</td>
</tr>
<tr>
<td>\texttt{selvarclust}</td>
<td>0.25</td>
<td>357.51</td>
<td>8</td>
<td>3</td>
<td>15.71</td>
</tr>
</tbody>
</table>

The results (Table 5.7) show that VSCC again results in the best clustering performance with an adjusted Rand index of 0.85, compared to 0.68 and 0.67 for \texttt{mclust} and \texttt{clustvarsel}, respectively. Running \texttt{mclust} on the variables chosen by \texttt{selvarclust} results in a 0.69 adjusted Rand index, leaving it roughly on par with the \texttt{mclust} and \texttt{clustvarsel} results. VSCC utilizes fewer variables (4 versus 5) and runs 8 times faster than \texttt{clustvarsel}. 
5.5.5 Coffee data

The coffee data set given by Streuli (1973) contains 13 chemical measurements on 43 samples of coffee hailing from one of two species: Arabica or Robusto.

Table 5.8: Table of results from mclust, VSCC, clustvarsel, and selvarclust on the coffee data set. The relationship chosen for VSCC by the uncertainty is given in parentheses.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Adj Rand</th>
<th>Time (sec)</th>
<th>G</th>
<th>#Vars</th>
<th>Unc</th>
</tr>
</thead>
<tbody>
<tr>
<td>mclust</td>
<td>1.00</td>
<td>0.16</td>
<td>2</td>
<td>13</td>
<td>0.00</td>
</tr>
<tr>
<td>VSCC (Quadratic)</td>
<td>1.00</td>
<td>1.19</td>
<td>2</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td>clustvarsel</td>
<td>0.41</td>
<td>2.79</td>
<td>3</td>
<td>6</td>
<td>0.42</td>
</tr>
<tr>
<td>selvarclust</td>
<td>0.37</td>
<td>404.67</td>
<td>4</td>
<td>7</td>
<td>0.23</td>
</tr>
</tbody>
</table>

These results (Table 5.8) serve as an example where a variable selection method can negatively affect clustering performance. While mclust performs perfect classification on the full data set, clustvarsel and selvarclust (including under mclust analysis of the selected variables) select too many groups. Perhaps more importantly, VSCC gives perfect classification while reducing the number of variables from 13 to 2: caffeine and fat content. A visualization of the clusters on these two variables is given in Figure 5.3.

5.6 Discussion and Future Work

A novel variable selection technique (VSCC) based on within-group variance was introduced and utilized under a model-based clustering framework. The strengths in the technique lie in the speed at which it can be run, as well as its intuitive appeal. It was shown to outperform or equal clustvarsel and selvarclust in clustering
performance, and significantly outperform them in speed, on four commonly used data sets. The authors wish to note that the VSCC relationship chosen by the total uncertainty was, in every real data set considered, the relationship that resulted in the highest adjusted Rand index. This, along with several of the simulation studies, lends support for the use of total uncertainty as a subset selection criteria.

The inner workings of VSCC are flexible enough to incorporate clustering/classification algorithms other than the model-based techniques covered in this article. One hurdle to overcome in this respect is an effective subset selection criterion — as non-model-based methods will not contain uncertainty measures — and this will be a subject of future research. In addition, the development of VSCC software for the R computing environment is intended pending code optimization and further testing.
Acknowledgements

This work was supported by a Postgraduate Doctoral Scholarship (Andrews) and a Discovery Grant (McNicholas) from the Natural Sciences & Engineering Research Council of Canada; by an Early Researcher Award from the Ontario Ministry of Research & Innovation (McNicholas); and by the University Research Chair in Computational Statistics at the University of Guelph (McNicholas).
Part III

Parameter Estimation
Chapter 6

Using Evolutionary Algorithms for Model-based Clustering

While variable selection is a useful tool for clustering and classification, there exists a much graver threat than noisy variables to the process of using mixture models for clustering: parameter estimation. Discussion of why typical parameter estimation procedures are problematic will be presented, and a new procedure is introduced to address some of the main obstacles.

The following article, in its entirety, is currently under peer review.

6.1 Introduction

Finite mixture models use a convex combination of probability densities to model data which arise from two or more subpopulations. This occurs when a vector $\mathbf{x}$ is a realization from the mixture distribution $f(\mathbf{x} | \vartheta) = \sum_{g=1}^{G} \pi_g p_g(\mathbf{x} | \theta_g)$, where $\pi_g > 0$ are mixing proportions so that $\sum_{g=1}^{G} \pi_g = 1$, $p_g(\mathbf{x} | \theta_g)$ is the density of component $g$ with parameters $\theta_g$, and $\vartheta = (\pi_1, \ldots, \pi_g, \theta_1, \ldots, \theta_G)$ denotes the parameters. Note that $f(\mathbf{x} | \vartheta)$ is called a $G$-component parametric finite mixture density; although common, the dependence upon parameters is not essential. Each component
density $p_g(\cdot)$ is commonly assumed to follow the same statistical distribution, most often the multivariate Gaussian distribution.

In a clustering scenario, data are assumed to arise from different groups with distinguishable characteristics. Thus, using finite mixture models to cluster data, known as model-based clustering, is a natural approach. Model-based clustering and finite mixture models in general have proven popular topics among researchers in statistics and the other computational sciences over the past couple of decades. Some notable work includes overviews (McLachlan and Peel, 2000a; Fraley and Raftery, 2002; Melnykov and Maitra, 2010; McNicholas, 2011), the development of families of mixture models (Celeux and Govaert, 1995; Fraley and Raftery, 1998; Bouveyron et al., 2007; McNicholas and Murphy, 2008), and extensions to non-Gaussian mixture distributions (McLachlan and Peel, 1998; Lin, 2010; Karlis and Santourian, 2009; Andrews and McNicholas, 2011b, 2012; Vrbik and McNicholas, 2012). Most of the recent literature makes use of the expectation-maximization (EM) algorithm (Dempster et al., 1977) or a close relative for parameter estimation. Alternative approaches that can be viewed as forerunners of the work introduced herein — including Celeux and Govaert (1992), Martinez and Vitria (2000), and Pernkopf and Bouchaffra (2005) — will be summarized in Sections 6.2.2 and 6.2.3. The remainder of this paper is laid out as follows. In Section 6.2, background material on relevant parameter estimation techniques is presented. Next, our evolutionary algorithms are outlined (Section 6.3) and illustrated on real data (Section 6.4). We conclude, in Section 6.5, with discussion and suggestions for future work.
6.2 Background

6.2.1 Model-based Clustering Likelihood

Herein, we assume a mixture of multivariate Gaussian distributions and so our model density is given by

\[ f(x \mid \vartheta) = \sum_{g=1}^{G} \pi_g \phi(x \mid \mu_g, \Sigma_g) \]

with mean vectors \( \mu_1, \ldots, \mu_G \) and covariance matrices \( \Sigma_1, \ldots, \Sigma_G \). In a clustering scenario, we assume no prior knowledge of component membership and so the likelihood of observing \( x_1, \ldots, x_n \) i.i.d. is

\[ \prod_{i=1}^{n} \sum_{g=1}^{G} \pi_g \phi(x_i \mid \mu_g, \Sigma_g) \]

The component membership labels can be treated as missing data; these are denoted \( Z_{ig} \), where \( z_{ig} = 1 \) if observation \( i \) belongs to group \( g \) and \( z_{ig} = 0 \) otherwise. The \( Z_{ig} \) can be collectively considered an \( n \times G \) matrix where each row contains \( G - 1 \) ‘zeros’ and only 1 ‘one’. Henceforth, we use the notation \( \tilde{z}_{ig} \) to represent the expected value, conditional on the observed data points, of \( z_{ig} \).

6.2.2 The EM Algorithm and Extensions Thereof

The expectation-maximization (EM) algorithm (Dempster et al., 1977) is most often utilized for mixture model parameter estimation. The EM algorithm is an iterative procedure for parameter estimation in the presence of missing data that alternates between E(xpectation)-steps and M(aximization)-steps. In the E-step, the expected value of the complete-data log-likelihood (which is the likelihood of the observed plus missing data) is computed. The M-step follows by maximizing the complete-data log-likelihood with respect to the unknown model parameters. In mixture modelling applications, the complete-data comprise the observed \( x_1, \ldots, x_n \)
and the missing component labels $z_1, \ldots, z_n$. The reader is referred to McLachlan and Krishnan (2008) for an extensive review of the EM algorithm and its various relatives.

EM algorithms require initialization in order to run. For model-based clustering, initialization requires either specifying starting values for the expected values $\tilde{z}_{ig}$ and using maximum likelihood estimation to solve for the model parameters, or specifying starting model parameters and calculating $\tilde{z}_{ig}$ via its expected value. EM algorithms and variants that are used in mixture modelling applications are usually single-path-monotonic and deterministic. This means that given any initialization, the algorithm will follow the same single, monotone-increasing path to convergence. While the deterministic tendency of the algorithm can be seen in a positive light because it implies consistency, the monotonicity is a double-edged sword. Monotone increasing is not a bad feature in its own right, since as the algorithm progresses parameter estimates can only improve (according to the log-likelihood), but combining monotonicity with a singular search path on a complicated surface — and a singularity riddled surface, at that (cf., Titterington et al., 1985) — results in a high susceptibility to reaching local maxima instead of global maxima.

Common work-arounds to this problem involve using multiple random starts or giving ‘good’ starting values; e.g., by initializing the $\tilde{z}_{ig}$ via another clustering algorithm (like $k$-means). Biernacki et al. (2003) and Biernacki (2004) give practical suggestions for initializing, whilst deterministic annealing (Ueda and Nakano, 1998) provides an additional safeguard by effectively ‘flattening’ the likelihood surface during the early stages of the EM algorithm. Susceptibility to local maxima, however,
remains a major problem.

Celeux and Govaert (1992) introduce the classification EM (CEM) and stochastic EM (SEM) algorithms specifically for use in model-based clustering applications. The CEM algorithm adjusts the regular E-step by forcing hard $\hat{z}_{ig}$ (0’s and 1’s) after the expected value calculation; i.e., for each $i$, $\max_{g=1,...,G}\{\hat{z}_{ig}\} = 1$ and the other $\hat{z}_{ig}$ are set to 0 (see Section 6.3.4 for further discussion). The SEM algorithm adjusts the regular E-step by randomly sampling a hard $\hat{z}_{ig}$ matrix given the probabilities found through the standard expected value calculation. We will compare the CEM and a slightly modified version of the SEM to both the EM algorithm and our evolutionary approaches in Section 6.4.

6.2.3 Evolutionary Algorithms

Evolutionary algorithms are often used to overcome difficult optimization problems. Much has been published over the past decade or so on evolutionary algorithms by authorities in the field, including work by Bäck (1996), Deb (2001), and Ashlock (2004); only a brief synopsis is given here. Borrowing from nature, evolutionary algorithms (EAs) arise from just a couple basic rules: reproduction and survival of the fittest. Survival of the fittest is a general term for the solutions that are selected for the reproduction step. In order for the algorithm to progress in a desired direction, a fitness calculation needs to be performed. For the purposes of this paper, we will focus only on single-objective evolutionary algorithms; which means that the optimization problem is focused on a single fitness criteria. A couple of common methods require definition vis-à-vis reproduction: mutation and cross-
breeding (or ‘recombination’). Mutation involves randomly changing aspects of a solution in order to give a new solution; in the context of model-based clustering, consider randomly selecting an observation and changing its cluster membership. Recombination happens when two or more solutions are combined in some fashion. Obviously this is a broad definition, but because our approach will not make use of recombination the specifics are not essential.

Existing approaches utilizing EAs for parameter estimation tend to focus on evolving the distribution parameter space. Specifically, Martinez and Vitria (2000) model a mixture of multivariate Gaussian distributions via an EA with mutation and recombination for a specified number of groups, while Pernkopf and Bouchafr (2005) extend this algorithm to search the number of groups as well. Both of these approaches focus the algorithm on mutations and recombinations of $\pi_1, \ldots, \pi_G$, $\mu_1, \ldots, \mu_G$, and $\Sigma_1, \ldots, \Sigma_G$. Our approach focuses instead on mutations among the cluster memberships, which has two theoretical advantages. Firstly, unlike searching the parameter space, the hard cluster membership space is finite; although it is enormous for seemingly trivial examples (for instance, with $G = 2$ there are $2^{n-1}$ possible unique clusterings and with just $n = 20$ samples this equates to 524,288 classifications). Secondly, as we shall see, using expected value computations will allow us to make ‘educated’ random mutations on the cluster memberships. We introduce several related evolutionary algorithms, built to perform model-based clustering, based on a single fitness function and mutations (cf. Section 6.3). For a more thorough introduction to evolutionary algorithms, the reader is encouraged to consult Ashlock (2004).
6.3 Methodology

6.3.1 Introduction

An EA is broadly defined and applicable to many optimization problems. In this section, we provide details for two new EAs, which are compared and contrasted with the EM algorithm in Section 6.4. Note that all EAs used herein are single-objective; i.e., built to optimize one fitness function. Since each step of the algorithm will require calculations among either parents or offspring, we must introduce two further subscripts into our mixture notation: subscript $j = 1, \ldots, J$ indexes offspring, while $k = 1, \ldots, K$ indexes parents.

6.3.2 Fitness Function

The fitness function for all of our EAs will be the log-likelihood. This is both intuitively appealing and facilitates easy comparison with the EM algorithm. The log-likelihood for mixture model $j$ can be calculated by

$$
\sum_{g=1}^{G} \left\{ n_{gj} \log(\pi_{gj}) - \frac{n_{gj}p}{2} \log 2\pi - \frac{n_{gj}}{2} \log(|\Sigma_{gj}|) + \sum_{i=1}^{n} (x_i - \mu_{gj})' \Sigma_{gj}^{-1} (x_i - \mu_{gj}) \right\},
$$

where $p$ is the dimension of the data, $n_{gj}$ are the number of observations in group $g$ of mixture model $j$ (similarly $\sum_{i=1}^{n} \hat{z}_{igj}$), and the other parameters are as defined previously. Note that the $\mu_{gj}$ and $\Sigma_{gj}$ can be estimated by maximum likelihood.
6.3.3 Survival

We will define survival in our algorithms as the \( K \) solutions with the best associated log-likelihoods from each generation. Herein, \( K \) will take values 1 or 4, depending on the algorithm. In addition, those \( K \) solutions will be cloned to the next generation, ensuring that the top \( K \) solutions for each generation is monotonic increasing. Note that larger values of \( K \) were considered, but the increase in performance was deemed too small to justify the additional computational overhead.

6.3.4 Reproduction

Random mutations will be generated within the \( \hat{z}_{igk} \) at each reproduction stage according to two different approaches. In the basic approach, denoted MEA (mixture evolutionary algorithm), each of the \( K \) parents will produce \( J/K \) offspring, and \( J \) will take a value approximately equal to \( n \). Each of these offspring will have a \( \hat{z}_{igk} \) mutated for one value of \( i \) such that its cluster membership changes. Note that mutating multiple observations to produce an offspring was considered, but the resulting algorithms consistently under-performed when compared to the single-mutation algorithm in terms of both efficiency and accuracy.

The second algorithm we propose makes use of expected value of the \( z_{igk} \). Given the maximum likelihood estimates resulting from the initial \( \hat{z}_{igk} \), we compute (as in the EM algorithm) the expected values

\[
\tilde{z}_{igk} = \frac{\pi_{gk} \phi(x_i \mid \mu_{gk}, \Sigma_{gk})}{\sum_{h=1}^{G} \pi_{hk} \phi(x_i \mid \mu_{hk}, \Sigma_{hk})},
\]

where \( i = 1, \ldots, n, g = 1, \ldots, G, \) and \( k = 1, \ldots, K \). Note that this is the probability
that observation \( i \) belongs to group \( g \) given the current parameter estimates. We can mutate the \( \hat{z}_{igk} \) by randomly sampling each observation’s cluster membership according to these probabilities. In this scenario, we store \( K \) parents at each iteration and during the reproduction phase we sample \( J/K \) new \( \hat{z}_{igj} \) matrices according to each of the \( K \) parent’s \( \tilde{z}_{igk} \). We denote this algorithm EvolvEM, due to it being essentially a cross-breed of both an evolutionary algorithm and an EM algorithm. Note that a monotonic increasing version of the SEM algorithm (Celeux and Govaert, 1992) arises as a special case of our approach if both the number of survivors \( K \) and the number of offspring \( J \) are set equal to 1. We denote this algorithm the monotonic stochastic expectation-maximization (mSEM) algorithm, and will compare its performance with the other algorithms in Section 6.4.

### 6.3.5 Convergence

Since we are restricting our \( \hat{z}_{igj} \) to 0’s and 1’s, potential log-likelihoods also fall into a large, finite search space. As such, our stopping criterion will be simple lack of progress. Specifically, the EA will stop when log-likelihoods from the top \( K \) solutions have failed to increase over the past four generations; i.e., the algorithm stops when the top \( K \) solutions have not changed in the past four generations.

### 6.3.6 Pseudo-Code

Here we present the pseudo-code for the EvolvEM algorithm. The more generic MEA algorithm follows a similar outline, but lacks the ‘Expectation’ step and tweaks the ‘Mutate’ step as described in Section 6.3.4.
Initialize random $\hat{z}_{igk}$ matrices

Maximize $k$ sets of parameters (using ML estimation)

Stagnation = 0

While Stagnation < 4

- **Expectation**: calculate $\hat{z}_{igk}$’s
- **Mutate**: sample $J/K$ offspring from each $\hat{z}_{igk}$
- **Maximize** all $J$ sets of parameters (using ML estimation)
- **Fitness**: calculate log-likelihood for all $J$ offspring
- **Survival**: sort $J$ offspring and $K$ parents in descending fitness order, select top $K$ as new parents

IF top $K$ solutions the same as previous cycle: Stagnation = Stagnation+1

ELSE: Stagnation = 0

Return $\hat{z}_{igk}$ corresponding to the highest log-likelihood

6.4 Applications

6.4.1 Introduction

Each data set is introduced and clustered via the EM, mSEM, CEM, MEA, and EvolvEM algorithms in Sections 6.4.2–6.4.4. Results for all of our analyses are discussed in Section 6.5.
6.4.2 Iris

The Iris data set (Anderson, 1935; Fisher, 1936) has been a longtime staple in statistical and clustering analyses. It contains four measurements on 150 samples of Iris setosa, Iris versicolor, and Iris virginica flowers. When clustering is applied to these data, the intent is usually to recover the three species; however, it is not a simple chore given the close proximity of the versicolor and virginica species (see Figure 6.1).

![Figure 6.1: Plot of sepal length versus sepal width for the Iris data.](image)

Results from all algorithms considered are given in Table 6.1. The EvolvEM algorithm, with $K = 4$ survivors per generation, gives the top performance across all categories of the Iris data. Particularly interesting is the small standard deviation in the final log-likelihood, as this suggests that the algorithm is relatively consistent in its final destination despite differing routes along the likelihood surface. Note that the EM-style algorithms that are most comparable to the evolutionary algorithms
introduced are the mSEM and CEM algorithms, as they give hard $\hat{z}_{ig}$ (0’s and 1’s) at each iteration. This is important to note because the EM algorithm has a distinct advantage over the EAs since it can more flexibly model the data by weighting each observation into all groups according to its expected value, resulting in better log-likelihood values. Yet, it is still the case that three out of four of the EA algorithms consistently outperformed the EM on the Iris data set.

Table 6.1: Log-likelihood summaries from 1000 randomly initialized runs of each algorithm on the Iris data set, fixing the number of groups to three.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean</th>
<th>Max</th>
<th>Min</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>mSEM</td>
<td>−312.02</td>
<td>−288.69</td>
<td>−438.84</td>
<td>29.87</td>
</tr>
<tr>
<td>EM</td>
<td>−302.67</td>
<td>−288.31</td>
<td>−422.20</td>
<td>18.42</td>
</tr>
<tr>
<td>CEM</td>
<td>−356.59</td>
<td>−288.47</td>
<td>−482.55</td>
<td>53.24</td>
</tr>
<tr>
<td>EvolvEM(1 survivor)</td>
<td>−301.16</td>
<td>−285.31</td>
<td>−418.53</td>
<td>17.84</td>
</tr>
<tr>
<td>EvolvEM(4 survivors)</td>
<td>−295.66</td>
<td>−283.09</td>
<td>−319.70</td>
<td>3.85</td>
</tr>
<tr>
<td>MEA (1 survivors)</td>
<td>−298.77</td>
<td>−288.63</td>
<td>−418.86</td>
<td>7.93</td>
</tr>
<tr>
<td>MEA (4 survivors)</td>
<td>−298.02</td>
<td>−288.63</td>
<td>−384.18</td>
<td>4.04</td>
</tr>
</tbody>
</table>

6.4.3 Italian Wines

The Italian wine data set (Forina et al., 1986) was sourced from the pgmm package (McNicholas et al., 2011) in R (R Development Core Team, 2012) and contains 27 chemical measurements on 178 samples of 3 varieties of red wine: Barolo, Barbera, and Grignolino. The top performers on this data set are the two MEA algorithms (Table 6.2). When we see that increasing the number of survivors has little effect on the performance of the one-survivor MEA, we can safely choose the one-survivor MEA as the top performer. Note that although the standard deviation of the MEAs is far larger than those of the other algorithms, the minimum log-likelihood values
obtained by the MEAs are either better or roughly equivalent to the mean values for
the EM algorithms. This suggests that the worst-case scenario for using MEA on
the Italian wine data set is a solution approximately as good as the EM, and better
than the mSEM or CEM algorithms. Both EvolvEM algorithms also outperform
the EM, mSEM, and CEM algorithms with better mean, maximum, and minimum
log-likelihoods.

Table 6.2: Log-likelihood summaries from 1000 randomly initialized runs of each
algorithm on the Italian wine data set, fixing the number of groups to three.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean</th>
<th>Max</th>
<th>Min</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>mSEM</td>
<td>−4512.88</td>
<td>−4385.61</td>
<td>−4641.22</td>
<td>41.77</td>
</tr>
<tr>
<td>EM</td>
<td>−4433.50</td>
<td>−4200.88</td>
<td>−4568.62</td>
<td>46.51</td>
</tr>
<tr>
<td>CEM</td>
<td>−4585.30</td>
<td>−4450.44</td>
<td>−4715.30</td>
<td>43.52</td>
</tr>
<tr>
<td>EvolvEM (1 survivor)</td>
<td>−4316.84</td>
<td>−4010.82</td>
<td>−4430.67</td>
<td>48.00</td>
</tr>
<tr>
<td>EvolvEM (4 survivor)</td>
<td>−4289.36</td>
<td>−4069.35</td>
<td>−4406.33</td>
<td>47.26</td>
</tr>
<tr>
<td>MEA (1 survivor)</td>
<td>−3959.16</td>
<td>−3653.97</td>
<td>−4374.47</td>
<td>120.88</td>
</tr>
<tr>
<td>MEA (4 survivor)</td>
<td>−3968.90</td>
<td>−3614.09</td>
<td>−4469.55</td>
<td>123.14</td>
</tr>
</tbody>
</table>

6.4.4 Leptograpsus Crabs

The Leptograpsus crabs data set (Venables and Ripley, 1999) can be found
in the MASS package in R. It contains five physical measurements on 200 crabs equally
split by gender (male and female) and colour (blue and orange). One defining charac-
teristic of this data set is how highly correlated the five measurements are (Figure 6.2).

Results from applying the clustering algorithms to the crabs data set are given in Ta-
ble 6.3. The top performer is the EvolvEM algorithm with $K = 4$ survivors per
generation, achieving the largest mean, maximum, and minimum likelihoods, as well
as the smallest standard deviation (albeit by a small margin).
6.5 Discussion and Future Work

The tip of the iceberg regarding evolutionary algorithms and their applicability to model-based clustering was further exposed by focusing an EA on the component memberships. Two algorithms were introduced and then compared with existing parameter estimation approaches, including the most common method: the expectation-maximization algorithm. Both evolutionary algorithms outperformed the mSEM and CEM algorithms on every data set according to the log-likelihood. In addition, at least one of the EAs (either EvolvEM or MEA) outperformed even the
regular EM algorithm on each data set. This is quite surprising given the advantage the EM algorithm has with soft cluster membership. It does, however, fit the general consensus that the EM algorithm is highly susceptible to local maxima and shows that the evolutionary approach is much more robust in this respect. One drawback of the evolutionary algorithms introduced in this paper is the additional computational overhead required; which will be at least partially mitigated pending code optimization. Regardless, the increased accuracy of the log-likelihood calculations provides ample argument for adopting the EA approach.

There are many avenues for future work using evolutionary algorithms in the field of mixture modelling. Two natural extensions of this work are incorporating statistical distributions other than the multivariate Gaussian distribution and/or incorporating mixture model families arising from reparameterizations of the covariance matrix.
Part IV

Conclusion
Chapter 7

Summary and Future Work

Five papers were introduced that serve as an illustration of the majority of the research completed during the course of my doctoral studies. An additional two papers (Andrews et al., 2011; Andrews and McNicholas, 2011a) were completed shortly after starting my doctoral studies and serve as forerunners to the main work being presented here; for the sake of succinctness, they have been omitted from this thesis.

The work presented in this thesis represents progress being made in the field of model-based clustering; which can be seen as a field with strong ties to mainstream statistics and machine learning communities. In Part I, two families and one software package were introduced. Obviously, the development of open-source software for the second family is of some priority — though the additional computational complexity makes this family less attractive on the consumer’s end, and further research is needed in this regard.

In Part II, a novel approach to variable selection was presented. Though partially heuristic in nature, its performance in comparison with other techniques is hard to ignore. In order to encourage usage, further research is needed to incorporate non-model-based techniques: such as $k$-means and hierarchical methods.
Finally, Part III saw the introduction to a new parameter estimation algorithm for model-based clustering. Though evolutionary algorithms represent an additional challenge in computational time, the approach was illustrated to be effective in avoiding some of the pitfalls of the typical expectation-maximization approach. Future research will include investigation of best-practice techniques for tuning parts of the algorithm. Parameter estimation (or rather, optimization of the parameters with respect to the log-likelihood) is at the forefront of issues regarding mixture model-based methods. Many other matters, such as model-selection via the BIC, could often be seen as a result of less-than-optimal parameter estimation. Consider the case of reaching a poor local maximum when estimating a $G = 2$ model followed by reaching the global maximum for a $G = 3$ model: comparing BIC values in this case could be, in fact, quite meaningless.

This work has motivated the general problem of clustering on top of putting forward several new options and techniques for solving the problem. I was once asked after a seminar whether those of us performing research in unsupervised learning believe that there will be some form of “grand unified theory” of clustering — wherein one algorithm will optimally solve all possible data grouping scenarios. I do not believe this to be the case; nor the intention of the majority of researchers in the field. Modern research will continue to put forth viable methods that improve upon, in one aspect or another, the methods that have been previously adopted. Due to the applied nature of clustering problems, there is always the potential that a new data set, or new type of data, will present major difficulties to the current status quo.
Bibliography


