Evolutionary Algorithms for Model-Based Clustering

EVOLUTIONARY ALGORITHMS FOR MODEL-BASED CLUSTERING

BY

REGINA S. KAMPO, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF MATHEMATICS & STATISTICS

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2021) McMaster University

(Mathematics & Statistics) Hamilton, Ontario, Canada

TITLE:	Evolutionary Algorithms for Model-Based Clustering
AUTHOR:	Regina S. Kampo M.Sc., (Statistics) McMaster Univeristy, Hamilton, Canada
SUPERVISORS:	Dr. Paul D. McNicholas (Supervisor), Dr. Sharon M. McNicholas (Co-supervisor)
NUMBER OF PAGES:	xxiii, 119

To my father, Victor Yao Kampo of blessed memory.

Abstract

Cluster analysis is used to detect underlying group structure in data. Model-based clustering is the process of performing cluster analysis which involves the fitting of finite mixture models. However, parameter estimation in mixture model-based approaches to clustering is notoriously difficult. To this end, this thesis focuses on the development of evolutionary computation as an alternative technique for parameter estimation in mixture models. An evolutionary algorithm is proposed and illustrated on the well-established Gaussian mixture model with missing values. Next, the family of Gaussian parsimonious clustering models is considered, and an evolutionary algorithm is developed to estimate the parameters. Next, an evolutionary algorithm is developed for latent Gaussian mixture models and to facilitate the flexible clustering of high-dimensional data. For all models and families of models considered in this thesis, the proposed algorithms used for model-fitting and parameter estimation are presented and the performance illustrated using real and simulated data sets to assess the clustering ability of all models. This thesis concludes with a discussion and suggestions for future work.

Acknowledgements

I am extremely grateful to Almighty God, for His grace and guidance from the beginning to the completion of this thesis. I would like to express my very great appreciation to my supervisors Dr. Paul McNicholas and Dr. Sharon McNicholas for their invaluable and constructive suggestions during the planning and development of this research work. Their willingness to give their time so generously has been very much appreciated. I would like to thank my supervisory committee member, Dr. Roman Viveros-Aguilera, for his insightful inputs and useful suggestions. I would also like to thank my external examiner, Dr. Jacques Julien and Dr. Giuseppe Melacini, who served as the chair for my defence.

I thank my fellow lab mates in the Computational Statistics Laboratory (McNicholas research group) for the stimulating discussions in our meetings and for all the fun we have had during our studies. My immense gratitude goes to the faculty and staff of the Department of Mathematics and Statistics for providing the resources I needed to complete this work. I also want to thank my professors, Dr. Joseph Beyene and Dr. Aaron Childs, as they greatly supported me during my graduate studies.

I am ever grateful to my dad (in loving memory), mum and siblings, for their prayers, encouragement, support and love in challenging moments. I am thankful for the support and unending love from my husband, Kenneth, throughout the years. Last but not the least, my sincere thanks go to my Westside Church family, friends and colleagues who have helped to make my stay in Canada wonderful through their encouragement, support and love.

Finally, I would like to acknowledge the financial support I received from the respective NSERC Discovery Grants and the Canada Research Chairs program towards my Ph.D.

Publications

The following articles based on the work in this thesis have been submitted or are in preparation.

- Kampo, R. S., McNicholas, S. M., and McNicholas, P. D. (2019), Clustering incomplete data using evolutionary algorithms. *Pattern Recognition Letters*. (revisions submitted September 2019).

- Kampo, R. S., McNicholas, S. M., and McNicholas, P. D., Evolutionary Algorithms for Latent Gaussian Mixture Models. *In preparation*.

- Kampo, R. S., McNicholas, S. M., and McNicholas, P. D., Evolutionary Algorithms for Gaussian Parsimonious Clustering Models. *In preparation*.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Clustering

1.1.1 Cluster Analysis

One of our most natural abilities is grouping and sorting objects into categories or groups. Classification can be described as a technique where group membership labels are assigned to unlabelled observations. It is natural to classify objects or individuals into groups to better understand the world around us and, as such, unsupervised classification, also known as clustering is the process of revealing underlying group structure in data. Cluster analysis is used for a wide variety of applications such as plant and animal ecology to reveal different types of species, market research to better understand the relationship between different groups of consumers or potential customers, crime analysis to identify areas where there are greater incidences of particular types of crime, etc.

In performing cluster analysis, there is no *a priori* knowledge of the class labels of observations or, at least the data is treated as such. However, at the other end of the classification spectrum is supervised classification, or discriminant analysis, where the class labels of some observations are known *a priori* and can be used to create a prediction rule for the classification of new observations.

The work in this thesis is mainly dedicated to developing novel approaches to parameter estimation in cluster analysis and, to establish clusters, or groups, in data such that the observations within groups are as similar as possible whilst observations in different groups are as dissimilar as possible. The methods developed in this thesis are tested on a variety of real data sets and may be applied to a wide range of real-world applications.

1.1.2 Clustering Methods

There are numerous techniques for performing cluster analysis in the literature. These techniques can be divided into two broad categories, distance based and model based clustering.

Clustering algorithms based on distance measures—e.g., k-means clustering (MacQueen et al., 1967), partition around medoids (PAM; Kaufman and Rousseeuw, 1990), hierarchical clustering (Ward Jr, 1963)—group objects together based on their distance from each other, such that objects in the same cluster are closer to each other, and objects in different clusters are farther away from each other, by computing the distance or the dissimilarity between the objects using some distance or dissimilarity measure. A major setback to such methods is that, for some types of

data, it may be a challenge, or impossible to define an appropriate distance metric. Also, as far as cluster structure is concerned, these approaches are quite rigid.

Parametric, model-based clustering techniques are commonly implemented using finite mixture models and they offer an alternative to distance based methods. According to Yeung et al. (2001):

> In the absence of a well-grounded statistical model, it seems difficult to define what is meant by a 'good' clustering algorithm or the 'right' number of clusters.

Finite mixture-models can be used for clustering by treating observations as a convex linear combination of probability densities. When clustering is based on finite mixture models, the component densities can be treated as similar to clusters and the problem is reduced to the assignment of observations to components — this technique is called model-based clustering (Fraley and Raftery, 2002; McNicholas, 2016). In recent years, a great deal of work has been done on model-based clustering (e.g., Gollini and Murphy, 2014; Tang et al., 2015; Melnykov, 2016; Wei et al., 2019; Tortora et al., 2020; Scott et al., 2020; Erola et al., 2020; Lee et al., 2020; Caruso et al., 2021; Chen and Zhang, 2021); however, relatively little attention has been paid to the various approaches to parameter estimation.

1.2 Evolutionary Computation

Evolutionary algorithms are used to find the global optimum in fitness landscapes, in particular where hardly much prior knowledge about the landscape is known (Pitzer

and Affenzeller, 2012). In an evolutionary algorithm, we aim at maximizing a predefined fitness function (Ashlock, 2010) by subjecting individuals to methods such as mutation, recombination, reproduction and selection. A fitness function can be defined as a function that can be used to assess the health of all individual members of a given population. As new members of a population mutate and reproduce from generation to generation, more fit members are chosen for further reproduction. The maximization of the fitness function to find the global maximum in the fitness landscape is analogous to maximizing the conditional expected value of the complete-data likelihood in an EM algorithm. Evolutionary algorithms have been successfully applied to cluster analysis (e.g., Hruschka et al., 2009; Andrews and McNicholas, 2013; McNicholas et al., 2020); but there is little work done on this topic in the literature. In this work, mixture-model based clustering methods are adapted to take into account elements of evolutionary computation.

1.3 Thesis Outline

1.3.1 Chapter 2

Background information is provided, including details on finite mixture models, missing data mechanisms, the EM algorithm and its variants, the Gaussian parsimonious clustering models, and mixtures of factor analyzers and extensions. Techniques for model-selection and performance assessment are also discussed.

1.3.2 Chapter 3

An evolutionary algorithm (EA) utilizing an evolutionary operation known as mutation is developed. This algorithm is applied for parameter estimation when handling data in the presence of missing values. The clustering ability of the method is illustrated on real and simulated data and compared to the well-known expectation-maximization (EM) algorithm.

1.3.3 Chapter 4

An EA with crossover followed by mutation is developed as a viable alternate to the EM algorithm for the parameter estimation in the family of Gaussian parsimonious clustering models (GPCMs). Model fitting and parameter estimation are discussed and excellent clustering performance is exhibited when the method is applied to several real data sets.

1.3.4 Chapter 5

Evolutionary computation is considered for estimating the parameters in the family of latent Gaussian mixture models, known as parsimonious Gaussian mixture models (PGMMs). The proposed technique is illustrated on both real and simulated data sets and its performance compared to the alternating expectation-conditional maximization (AECM) algorithm.

1.3.5 Chapter 6

A summary of the work demonstrated in this thesis is presented and possible research proposals for future work are also discussed.

1.4 Contributions of this Work

The impact of the work presented in this thesis on the body of evolutionary algorithms and model-based clustering literature is summarized as follows:

- An evolutionary algorithm is first used for model-based clustering in the presence of unobserved or missing values. It gives an approach that can be considered a hard model-based clustering with missing values under missing at random mechanism. The method is demonstrated using simulated and real data sets and performs favorably compared to both the EM algorithm and the mean imputation method.

- An EA has been developed as an alternative to the EM algorithm. This EA utilizes evolutionary operations known as crossover and mutation. The proposed method is the first used for parameter estimation in the family of the famous 14 GPCMs. Our proposed method is demonstrated on numerous real data sets and performed comparably to the well established EM algorithm.

- Maximum likelihood estimates for the parameters in a family of latent Gaussian mixture models, known as PGMMs, are typically found using an AECM algorithm. Rather than using an AECM algorithm, an evolutionary algorithm

is proposed. This EA makes use of the mutation operation. The proposed EA is illustrated on both real and simulated data sets and its performance is compared to the AECM algorithm.

Chapter 2

Background

2.1 Finite Mixture Models

Finite mixture models assume that a population can be modelled as a set of sub-populations, and each may be modelled by a statistical distribution. They present a natural approach for model-based clustering both for parameter estimation and to estimate group memberships. In general, a p-dimensional random vector $\mathbf{X}$ arises from a finite mixture model if for all $\mathbf{x} \in \mathbf{X}$, it has a density of the form

$$f(\mathbf{x}|\boldsymbol{\vartheta}) = \sum_{g=1}^{G} \pi_g f_g(\mathbf{x}|\boldsymbol{\theta}_g), \tag{2.1}$$

where $\pi_g > 0$ such that $\sum_{g=1}^{G} \pi_g = 1$ are the mixing proportions, $f_1(\mathbf{x} \mid \boldsymbol{\theta}_1), \ldots, f_g(\mathbf{x} \mid \boldsymbol{\theta}_g)$ are the component densities, $\boldsymbol{\vartheta} = (\boldsymbol{\pi}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_G)$ is the vector of parameters where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_G)$, and G denote the number of mixture components used to model the data. The use of finite mixture models began with the work of Pearson (1894)

who used a mixture of two normal distributions to model data on crabs sampled from the Bay of Naples. Extensive reviews of finite mixture models can be found in Everitt (1981), Titterington et al. (1985), McLachlan and Basford (1988), McLachlan and Peel (2000a) and McNicholas (2016).

When the density is expressed in the form of (2.1), each component density is typically assumed to follow the same statistical distribution, e.g., the multivariate Gaussian distribution. The multivariate Gaussian components has been widely used in the statistical literature for continuous multivariate data, due to their computational convenience. The earliest use of the finite mixture model for clustering can be found in Wolfe (1965), using a Gaussian mixture model. The Gaussian mixture density is defined as:

$$f(\mathbf{x} \mid \boldsymbol{\vartheta}) = \sum_{g=1}^{G} \pi_g \phi(\mathbf{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g), \tag{2.2}$$

where $\phi(\mathbf{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ is a p-dimensional multivariate normal density with mean $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$, $\boldsymbol{\vartheta} = (\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_G)$ denotes the parameters of the mixture model, with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_G)$, and π_g has the same meaning as before.

2.2 Mixture of Multivariate Gaussian Distributions

2.2.1 Model-Based Clustering

Model-based clustering is a fundamental statistical approach used to describe a clustering framework that uses statistical distributions, particularly mixture models.

Suppose that n p-dimensional data vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are observed and all are unlabelled or treated as such. The Gaussian model-based clustering likelihood can be expressed as

$$\mathcal{L}(\boldsymbol{\vartheta}) = \prod_{i=1}^{n} \sum_{g=1}^{G} \pi_g \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g), \tag{2.3}$$

where π_g can be thought of as the *a priori* probability that observation $\mathbf{x}_i$ belongs in component g (McLachlan and Peel, 2000a; McNicholas, 2016). To facilitate clustering, the notation $\mathbf{z}_i$ is introduced where, $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})$ denote the component (group) membership of observation i, such that $z_{ig} = 1$ if observation i belongs to the gth component and $z_{ig} = 0$ otherwise. Within the EM algorithm framework, the z_{ig} are replaced by their expected values:

$$\hat{z}_{ig} = \frac{\hat{\pi}_g \phi(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g)}{\sum_{h=1}^{G} \hat{\pi}_h \phi(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_h, \hat{\boldsymbol{\Sigma}}_h)}, \tag{2.4}$$

for $i = 1, \ldots, n$ and $g = 1, \ldots, G$. After the parameters have been estimated, the predicted group memberships are determined from the *a posteriori* probability that observation $\mathbf{x}_i$ belongs to component g—this is given by $\hat{z}_{ig}$ evaluated at the parameter estimates. The main difference between "soft" classification and "hard" classification boils down to how the $\hat{z}_{ig}$ is reported—when this value is reported exactly as computed, it is referred to as soft classification, and when rounded to 0 or 1, it is referred to as hard classification.

The most popular way to carry out hard classification is to report the maximum

a posteriori (MAP) classification, i.e., MAP$\{\hat{z}_{ig}\}$, where

$$\text{MAP}\{\hat{z}_{ig}\} = \begin{cases} 1 & \text{if } g = \text{argmax}_h\{\hat{z}_{ih}\}, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

The EM algorithm always allows values $\hat{z}_{ig} \in [0, 1]$ as the algorithm iterates, irrespective of whether $\hat{z}_{ig}$ or MAP$\{\hat{z}_{ig}\}$ is eventually returned. In this work, the evolutionary algorithm that is considered restricts $\hat{z}_{ig} \in \{0, 1\}$ at all times, which is the main feature that distinguishes it and the EM algorithm.

2.2.2 Gaussian Parsimonious Clustering Models

A p-dimensional random variable following a G-component Gaussian mixture model has a total of $G - 1 + Gp + Gp(p + 1)/2$ free parameter, with $G - 1$ coming from the mixing proportions, Gp from the means and $Gp(p + 1)/2$ from the covariance matrices. Celeux and Govaert (1995) introduced parsimony into Gaussian mixtures, by imposing constraints on the elements of the decomposed covariance structure $\boldsymbol{\Sigma}_g$. Banfield and Raftery (1993) proposed an eigen-decomposition of the component covariance matrices, i.e.,

$$\boldsymbol{\Sigma}_g = \lambda_g \boldsymbol{\Gamma}_g \boldsymbol{\Delta}_g \boldsymbol{\Gamma}'_g, \tag{2.5}$$

where $\boldsymbol{\Gamma}_g$ is an orthogonal matrix of eigenvectors of $\boldsymbol{\Sigma}_g$, $\boldsymbol{\Delta}_g$ is a diagonal matrix, such that $|\boldsymbol{\Delta}_g| = 1$, containing the normalized eigenvalues of $\boldsymbol{\Sigma}_g$ in decreasing order, and $\lambda_g = |\boldsymbol{\Sigma}_g|^{1/p}$ is the associated constant of proportionality. The scalar λ_g constitutes the volume in p-space, $\boldsymbol{\Delta}_g$ specifies the shape and $\boldsymbol{\Gamma}_g$ determines the orientation

(Banfield and Raftery, 1993; McNicholas, 2016). By imposing a combination of constraints on (2.5), Celeux and Govaert (1995) developed a family of 14 Gaussian parsimonious clustering models (GPCMs) and these models are presented in Table 2.1 (sourced from McNicholas, 2016).

Table 2.1: Nomenclature, descriptions, component covariance structure, and parameter counts of the component covariance matrices for each member of the GPCM family.

Model	$\lambda_g = \lambda$	$\boldsymbol{\Delta}_g = \boldsymbol{\Delta}$	$\boldsymbol{\Gamma}_g = \boldsymbol{\Gamma}$	$\boldsymbol{\Sigma}_g$	Number of Covariance Parameters
EII	Equal	Identity	Identity	$\lambda\mathbf{I}$	1
VII	Variable	Identity	Identity	$\lambda_g\mathbf{I}$	G
EEI	Equal	Equal	Identity	$\lambda\boldsymbol{\Delta}$	p
VEI	Variable	Equal	Identity	$\lambda_g\boldsymbol{\Delta}$	$p+G-1$
EVI	Equal	Variable	Identity	$\lambda\boldsymbol{\Delta}_g$	$Gp-G+1$
VVI	Variable	Variable	Identity	$\lambda_g\boldsymbol{\Delta}_g$	Gp
EEE	Equal	Equal	Equal	$\lambda\boldsymbol{\Gamma}\boldsymbol{\Delta}\boldsymbol{\Gamma}'$	$p(p+1)/2$
VEE	Variable	Equal	Equal	$\lambda_g\boldsymbol{\Gamma}\boldsymbol{\Delta}\boldsymbol{\Gamma}'$	$G+p-1+p(p-1)/2$
EVE	Equal	Variable	Equal	$\lambda\boldsymbol{\Gamma}\boldsymbol{\Delta}_g\boldsymbol{\Gamma}'$	$1+G(p-1)+p(p-1)/2$
EEV	Equal	Equal	Variable	$\lambda\boldsymbol{\Gamma}_g\boldsymbol{\Delta}\boldsymbol{\Gamma}_g'$	$p+Gp(p-1)/2$
VVE	Variable	Variable	Equal	$\lambda_g\boldsymbol{\Gamma}\boldsymbol{\Delta}_g\boldsymbol{\Gamma}'$	$Gp+p(p-1)/2$
VEV	Variable	Equal	Variable	$\lambda_g\boldsymbol{\Gamma}_g\boldsymbol{\Delta}\boldsymbol{\Gamma}_g'$	$G+p-1+Gp(p-1)/2$
EVV	Equal	Variable	Variable	$\lambda\boldsymbol{\Gamma}_g\boldsymbol{\Delta}_g\boldsymbol{\Gamma}_g'$	$1+G(p-1)+Gp(p-1)/2$
VVV	Variable	Variable	Variable	$\lambda_g\boldsymbol{\Gamma}_g\boldsymbol{\Delta}_g\boldsymbol{\Gamma}_g'$	$Gp(p+1)/2$

Parameter estimation for these 14 GPCM models is carried out using an EM algorithm (Dempster et al., 1977), and a detailed outline of the framework is given by Celeux and Govaert (1995). Browne and McNicholas (2014) developed an alternate approach for the EVE and the VVE models using fast majorization-minimization algorithms.

2.3 Mixture of Factor Analyzers Model

Although mixture models are widely used, however, they need to be adapted to cope with high-dimensional data sets. This is because as the data dimensionality p increases, the number of model parameters that must be estimated becomes large. The main contribution to the number of free parameters comes from the component covariance matrices.

Factor analysis, first introduced by the psychologist, Spearman (1904), is a technique that is used to reduce a large number of variables into a fewer number of factors. This technique was later on explained in statistical terms by Bartlett (1953) and Lawley and Maxwell (1962). Consider independent p-dimensional random variables $\mathbf{X}_1, \ldots, \mathbf{X}_n$, the factor analysis model can be written

$$\mathbf{X}_i = \boldsymbol{\mu} + \boldsymbol{\Lambda}\mathbf{U}_i + \boldsymbol{\epsilon}_i, \tag{2.6}$$

for $i = 1, ..., n$, where $\boldsymbol{\Lambda}$ is a $p \times q$ matrix of factor loadings with $q < p$, $\mathbf{U}_i \sim \mathrm{N}(\mathbf{0}, \mathbf{I}_q)$ denotes the latent factors, and $\epsilon_i \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Psi})$, with $\boldsymbol{\Psi} = \mathrm{diag}(\psi_1, \ldots, \psi_p)$. The $\mathbf{U}_i$ and $\boldsymbol{\epsilon}_i$ are both independently distributed and independent of each another. Under this model, the marginal distribution of $\mathbf{X}_i$ is $\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{\Psi})$. Similar to the factor analysis model, the mixture of factor analyzers (MFA) model is expressed as:

$$\mathbf{X}_i = \boldsymbol{\mu}_g + \boldsymbol{\Lambda}_g\mathbf{U}_{ig} + \boldsymbol{\epsilon}_{ig}, \tag{2.7}$$

with probability π_g, for $i = 1, ..., n$ and $g = 1, ..., G$. The MFA model originally

proposed by Ghahramani and Hinton (1997) and Hinton et al. (1997), reduces the number of parameters to be estimated via restrictions on the component covariance matrices $\boldsymbol{\Sigma}_g$. The density of this MFA model is that of a Gaussian mixture model with component covariance structure $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g$. Thus the density of $\mathbf{X}_i$ from the MFA model is

$$\begin{aligned} f(\mathbf{x}_i) = \sum_{g=1}^{G} & \frac{\pi_g}{(2\pi)^{p/2}|\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g|^{1/2}} \\ & \times \ \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_g)'(\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g)^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_g)\right\}. \end{aligned} \tag{2.8}$$

A number of models have been developed that extend the MFA model as a result of additional constraints imposed on the component covariance parameters. McLachlan and Peel (2000b) presented a more general mixture of factor analyzers model ($\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g$), and Tipping and Bishop (1999) introduced a mixture of probabilistic principal component analyzers model ($\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \psi_g\mathbf{I}_p$). McNicholas and Murphy (2008) constructed a family of parsimonious Gaussian mixture models (PGMM), and this family of models is discussed in the subsection below.

2.3.1 Parsimonious Gaussian Mixture Models

McNicholas and Murphy (2008) considered the combinations of the constraints $\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}$, $\boldsymbol{\Psi}_g = \boldsymbol{\Psi}$, and the isotopic constraint $\boldsymbol{\Psi}_g = \psi_g\mathbf{I}_p$, providing a class of eight different PGMMs. These constraints tend to reduce the number of free parameters estimated, making PGMM models more suitable for clustering high-dimensional data. These PGMMs are presented in Table 2.2, where "C" means a constraint is imposed and

"U" means constraint not imposed (notations borrowed from McNicholas, 2016). The maximum likelihood estimates of the parameters for the members of the PGMM family are performed using the AECM algorithm.

Table 2.2: The nomenclature, component covariance structure, and parameter counts of the component covariance matrices for each member of the PGMM family.

Model ID	$\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}$	$\boldsymbol{\Psi}_g = \boldsymbol{\Psi}$	$\boldsymbol{\Psi}_g = \psi_g \mathbf{I}_p$	$\boldsymbol{\Sigma}_g$	Free Covariance Parameters
CCC	C	C	C	$\boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \psi\mathbf{I}_p$	$pq - q(q-1)/2 + 1$
CCU	C	C	U	$\boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{\Psi}$	$pq - q(q-1)/2 + p$
CUC	C	U	C	$\boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \psi_g\mathbf{I}_p$	$pq - q(q-1)/2 + G$
CUU	C	U	U	$\boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{\Psi}_g$	$pq - q(q-1)/2 + Gp$
UCC	U	C	C	$\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \psi\mathbf{I}_p$	$G[pq - q(q-1)/2] + 1$
UCU	U	C	U	$\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}$	$G[pq - q(q-1)/2] + p$
UUC	U	U	C	$\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \psi_g\mathbf{I}_p$	$G[pq - q(q-1)/2] + G$
UUU	U	U	U	$\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g$	$G[pq - q(q-1)/2] + Gp$

2.4 EM Algorithm and Extensions

2.4.1 The EM Algorithm

The EM algorithm (Dempster et al., 1977) is an iterative technique that computes the maximum likelihood (ML) estimates and has been the most common approach for the fitting of mixture models, and parameter estimation in model-based clustering. The EM algorithm is designed for application to data that is incomplete or treated as such. In clustering applications, the term incomplete data refers to the unobserved group membership labels and sometimes other latent variables and, complete data refers to both observed and missing data.

The EM algorithm is based on the complete data and it alternates between two steps, an expectation (E-) step and a maximization (M-) step. In each E-step, the expected value of the complete-data log-likelihood, namely the so-called $\mathcal{Q}$ function, is calculated conditional on the observed data and the current parameter estimates. In the M-step, the expected complete-data log-likelihood $\mathcal{Q}$ is maximized with respect to the model parameters. The EM algorithm iterates between the E-step and M-step until some stopping/convergence criterion is met. Ng et al. (2012) outlined a detailed description of the EM algorithm.

2.4.2 The AECM Algorithm

The AECM algorithm (Meng and Van Dyk, 1997) is a variant of the EM algorithm, or more precisely, it is an extension of the expectation-conditional maximization (ECM) algorithm (Meng and Rubin, 1993). The ECM algorithm replaces the M-step in the EM algorithm by a sequence of conditional maximization (CM-) steps. The AECM algorithm incorporates a series of CM-steps instead of a single M-step and also allows for different specification of the complete-data in each stage of the algorithm. Similar to the regular M-step, the CM-step will maximize the conditional expectation of its corresponding complete-data log-likelihood at each cycle. A detailed description as well as illustrative examples of the EM algorithm and its extensions can be found in McLachlan and Krishnan (2007).

2.4.3 Convergence

The EM algorithm and its extensions iteratively update the model parameters until certain predefined criteria are satisfied. A common approach is the lack of progress, where the algorithm is stopped depending on the difference in successive observed log-likelihood values. In this case the EM algorithm is stopped when

$$l^{(r+1)} - l^{(r)} < \epsilon, \tag{2.9}$$

where ϵ is some small value, and $l^{(r)}$ is the observed log-likelihood from iteration r.

An alternate method is the Aitken's acceleration-based criterion (Aitken, 1926), which is used to evaluate the asymptotic maximum of the log-likelihood at each iteration of the EM algorithm. At iteration r, it is expressed as

$$a^{(r)} = \frac{l^{(r+1)} - l^{(r)}}{l^{(r)} - l^{(r-1)}}, \tag{2.10}$$

where $l^{(r)}$ is the log-likelihood value evaluated at iteration r. The Aitken's accelerated estimate of the log-likelihood at iteration $r+1$ considered by Böhning et al. (1994) and Lindsay (1995) is

$$l_{\infty}^{(r+1)} = l^{(r)} + \frac{1}{1 - a^{(r)}}(l^{(r+1)} - l^{(r)}). \tag{2.11}$$

McNicholas et al. (2010) stopped the algorithm when

$$l_{\infty}^{(r+1)} - l^{(r)} < \epsilon, \tag{2.12}$$

provided this difference is positive.

2.5 Evolutionary Computation

Evolutionary computation (EC) is a paradigm consisting of computational intelligence technique strongly inspired by the theory of evolution from the perspective of biology. An EC initially starts with a population of individuals that are considered to be candidate solutions. Operations inspired from natural evolution, such as crossover and mutation are carried out on these individuals which then reproduce and replace the less fit members of the population. In mutation, parts of a solution are modified randomly to generate a new solution, e.g., in cluster analysis, one can select an observation at random and change the group to which it belongs. On the other hand, crossover involves the combination of two or more solutions, in some fashion, to give a new solution.

Evolutionary algorithms (EAs) are useful for overcoming difficult optimization problems. Similarly to how nature works, evolutionary algorithms use two basic ideas: reproduction and survival of the fittest. Survival of the fittest is a way of describing the process of natural selection which in this case refers to the individuals or solutions that are selected for the reproduction phase. Fitness is determined in the context of a fitness function and the less fit members from a generation are replaced by new, fitter, members. This evolutionary process—mutation, crossover, and survival of the fittest—is repeated until some stopping rule is satisfied. The fitness function used in a particular case depends on the goal of the EA. In fact,

if the optimization problem is multi-objective, then there can be multiple fitness functions or criteria. The general steps of an EA is presented below.

```
initialize population of solutions called members
calculate the fitness of all members
while optimal solution is not attained
select parents
apply genetic operators, crossover and mutation to the selected individuals
calculate fitness values of new individuals
select individuals for the next generation
end while
return the best individuals
```

A detailed survey on EAs for clustering is presented by Hruschka et al. (2009), where they remarked that a lot of work needs to be done on the theoretical foundations of EAs as they can be seen as heuristic-based approach to solving hard optimization problems. Also, Weicker and Weicker (2003) pointed out that a solid theoretical basis for these applications is still lacking whilst outlining a networked understanding of EAs.

The field of EAs is quite active and recent work includes contributions by Hasnat et al. (2017), Das et al. (2019), Lin et al. (2019), Tautenhain and Nascimento (2020), Hassan and Rashid (2021) and Luo et al. (2021). Comprehensive coverage of EAs is given in the monographs by Deb (2001) and Ashlock (2010). In the field of model-based clustering, there are existing EA approaches that focus on evolving the parameter space (Martınez and Vitria, 2000; Pernkopf and Bouchaffra, 2005) whilst

Andrews and McNicholas (2013) and McNicholas et al. (2020) propose approaches that focus instead on mutations among the cluster membership labels where the data is completely observed, i.e., the case of null proportion of missing data. Mutations among the cluster membership labels can be considered as having two theoretical advantages (Andrews and McNicholas, 2013). Firstly, the hard cluster membership space is finite unlike the parameter space and, secondly, the ability to make "educated" random mutations on the cluster memberships by using the expectations that would be used in an EM algorithm. Due to these advantages, the approach developed in this thesis focuses on mutation and crossover among the cluster membership labels. In Chapter 3, our proposed EA is built to perform model-based clustering when data are missing at random using a single fitness function, i.e., the (observed) log-likelihood, and mutations. In Chapter 4, the EA developed utilizes crossover followed by mutation and, in Chapter 5, the EA approach focuses on mutation only.

2.6 Missing Data Mechanism

The presence of unobserved or missing data poses a particularly significant difficulty in clustering because, in addition to the usual challenges, the subpopulation to which an observation with missing data belongs is unknown. Orchard et al. (1972) commented that the best way to treat missing data is not to have them and, while true, is often not practical. As Allison (2002) observed, anyone who works with data sooner or later runs into problems with missing data. The maximum likelihood and Bayesian methods are two popular imputation paradigms for analyzing data with

missing observations.

Little and Rubin (1987) and Rubin (1976) classified the missing data mechanism into three categories that remain in use today: (a) missing completely at random (MCAR), (b) missing at random (MAR), and (c) missing not at random (MNAR), a.k.a not missing at random (NMAR). In the missing data literature, data are often partitioned into two parts: the observed data ($\mathbf{X}^{\text{o}}$) and the missing data ($\mathbf{X}^{\text{m}}$). In this context, the mechanism of missing data can be elegantly described through the relationships between $\mathbf{X}^{\text{o}}$, $\mathbf{X}^{\text{m}}$, and the "cause" of the missingness. For MCAR, the cause of missingness is independent of both $\mathbf{X}^{\text{o}}$ and $\mathbf{X}^{\text{m}}$. MAR is a process in which the cause of missingness is not related to $\mathbf{X}^{\text{m}}$, but may depend on $\mathbf{X}^{\text{o}}$. In this case MCAR can be seen as a special case of MAR. The missing data mechanism is MNAR if the data missingness are related to $\mathbf{X}^{\text{m}}$ or some unobserved latent variables. In Chapter 3 of this thesis, the missing data mechanism is assumed to be missing at random (MAR), under which the missing data mechanisms are ignorable for methods using the maximum likelihood approach.

2.7 Model Selection and Performance Assessment

2.7.1 Bayesian Information Criterion

In a family of models, an appropriate model often has to be selected. For example, it is necessary to select an adequate number of components G, and/or the component covariance structure. The Bayesian information criterion (BIC; Schwarz et al., 1978)

is a common technique for model selection in model-based clustering and is given by

$$\text{BIC} = 2l(\hat{\boldsymbol{\vartheta}}) - \rho \log n, \tag{2.13}$$

where $l(\hat{\boldsymbol{\vartheta}})$ is the maximized log-likelihood, $\hat{\boldsymbol{\vartheta}}$ represents the maximum likelihood estimate of $\boldsymbol{\vartheta}$, n is the number of observations, and ρ denotes the number of free parameters to be estimated in the model. Fraley and Raftery (1998, 2002) provide evidence that the BIC performs well as a model selection criterion for mixture models. Alternatives for model selection are suggested, *inter alia*, by Biernacki et al. (2000), but none have been consistently better. Lopes and West (2004) show how the BIC can be used to select the number of latent factors for a factor analysis model. Throughout this thesis, the BIC is used as a model selection criterion to choose the number of groups, latent factors, and covariance structure where appropriate. Note also that we choose the model with the largest BIC value from among a set of competing models.

2.7.2 Adjusted Rand Index

In this thesis, clustering is performed on data sets for which the true fundamental groups are known *a priori*. This allows us to evaluate the clustering efficiency of the models developed here by comparing the known class labels with the estimated cluster members. However, each analysis is performed as a true clustering problem and the actual class members are entirely hidden from our algorithms and are not used to aid the clustering.

The Rand index (RI; Rand, 1971) is a method used for assessing class agreement

and is calculated as a cross-tabulation between the true class labels and the MAP classifications. In general, the Rand index can be calculated as

$$\frac{\text{number of pairwise agreements}}{\text{total number of pairs}},$$

where the total number of pairs is the sum of the number of pairwise agreements and the number of pairwise disagreements. A pairwise agreement occurs when two observations belonging to the same cluster are assigned the same label or, when two observations belonging to different clusters are actually assigned different labels by the mixture model. The RI takes on values between 0 and 1, where a value of 1 indicates perfect class agreement. The RI may be difficult to interpret sometimes for smaller values, because its expected value is greater than 0 under random classification.

The adjusted Rand index (ARI) introduced by Hubert and Arabie (1985), corrects the RI for the number of pairwise agreements that would be expected to occur if the observations were classified at random. Similar to the RI, an ARI value of 1 indicates perfect agreement between the true class labels and the MAP classifications. Under random classification, the expected ARI value is 0, whilst a negative value shows that the classification is worse than classifying randomly.

Chapter 3

Clustering Incomplete Data using an Evolutionary Algorithm

3.1 Introduction

With the increasing emphasis on data science, clustering (or unsupervised classification) has burgeoned into an important subfield of machine learning. In a clustering scenario, each observation comes from one of a number of subpopulations—a.k.a. groups, classes or clusters—with distinguishable features and the objective is to find to which subpopulation each observation belongs to. The presence of missing data poses a particularly significant difficulty in clustering because, in addition to the usual challenges, the subpopulation to which an observation with missing data belongs is unknown. The Gaussian component model with an EM algorithm is often used to tackle missing data in the unsupervised classification paradigm. However,

despite the great benefits associated with the EM algorithm, it is susceptible to becoming stuck at local maxima and also, the algorithm is very reliant on starting values. In this chapter, rather than using an EM algorithm, we develop an EA for clustering partially observed data. The EA facilitates a different search of the fitness landscape, i.e., the likelihood surface, when compared to the EM algorithm and so it is of interest to compare the two.

3.2 Gaussian Mixture Models with Missing Data

Consider a model-based clustering scenario, the maximum likelihood estimation of (2.3) when the random variables $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are not completely observed, i.e., when the pattern of missingness is arbitrary and MAR — thus we assume the missing data mechanism here to be MAR.

To set up the updates for the Gaussian mixture model with missing values, $\mathbf{X}_i$ is partitioned into the observed part $\mathbf{X}_i^{\text{o}}$ and the missing part $\mathbf{X}_i^{\text{m}}$ with dimensions $p_i^{\text{o}} \times 1$ and $p_i^{\text{m}} \times 1$, respectively, where $p_i^{\text{o}} + p_i^{\text{m}} = p$. Borrowing standard notation (e.g., Lin et al., 2006; Wei et al., 2019), two missing indicator matrices are also introduced, denoted by $\mathbf{O}_i$ $(p_i^{\text{o}} \times p)$ and $\mathbf{M}_i$ $(p_i^{\text{m}} \times p)$, which can be extracted from a p-dimensional identity matrix $\mathbf{I}_p$, corresponding to the respective row positions of $\mathbf{X}_i^{\text{o}}$ and $\mathbf{X}_i^{\text{m}}$ in $\mathbf{X}_i$ such that $\mathbf{X}_i^{\text{o}} = \mathbf{O}_i\mathbf{X}_i$ and $\mathbf{X}_i^{\text{m}} = \mathbf{M}_i\mathbf{X}_i$. It can be easily verified that $\mathbf{X}_i = \mathbf{O}_i'\mathbf{X}_i^{\text{o}} + \mathbf{M}_i'\mathbf{X}_i^{\text{m}}$ and $\mathbf{O}_i'\mathbf{O}_i + \mathbf{M}_i'\mathbf{M}_i = \mathbf{I}_p$.

To denote which component each data vector $\mathbf{x}_i$ belongs to, it is convenient to introduce $\mathbf{z}_1, \ldots, \mathbf{z}_n$, where $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})$ with $z_{ig} = 1$ if $\mathbf{x}_i$ belongs to the gth

component and $z_{ig} = 0$ otherwise. Parameter estimation can be performed using the EM algorithm, where the complete-data comprise the observed $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and the labels $\mathbf{z}_1, \ldots, \mathbf{z}_n$. The complete-data likelihood is

$$\mathcal{L}_c(\boldsymbol{\vartheta}) = \prod_{i=1}^{n}\prod_{g=1}^{G}[\pi_g \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)]^{z_{ig}}. \tag{3.1}$$

Following the EM algorithm outlined by Lin et al. (2006), the complete-data log-likelihood function is expressed as

$$l_c(\boldsymbol{\vartheta}) = \sum_{g=1}^{G}\sum_{i=1}^{n} z_{ig} \log \pi_g + \frac{1}{2}\sum_{g=1}^{G}\left(\log\left|\boldsymbol{\Sigma}_g^{-1}\right|\sum_{i=1}^{n} z_{ig} - \sum_{i=1}^{n} z_{ig}(\boldsymbol{\Delta}_{ig}^{\text{o}} + \boldsymbol{\Delta}_{ig}^{\text{m.o}})\right), \tag{3.2}$$

where $\boldsymbol{\Sigma}_g^{-1} = \mathbf{S}_{ig}^{\text{oo}} + \mathbf{S}_{ig}^{\text{mm.o}}$, $\mathbf{S}_{ig}^{\text{oo}} = \mathbf{O}_i'(\mathbf{O}_i\boldsymbol{\Sigma}_g\mathbf{O}_i')^{-1}\mathbf{O}_i$, $\boldsymbol{\Delta}_{ig}^{\text{o}} = (\mathbf{x}_i - \boldsymbol{\mu}_g)'\mathbf{S}_{ig}^{\text{oo}}(\mathbf{x}_i - \boldsymbol{\mu}_g)$,

$$\boldsymbol{\Delta}_{ig}^{\text{m.o}} = (\mathbf{x}_i - \boldsymbol{\mu}_g)'\mathbf{S}_{ig}^{\text{mm.o}}(\mathbf{x}_i - \boldsymbol{\mu}_g)\text{and}$$

$$\mathbf{S}_{ig}^{\text{mm.o}} = [\mathbf{M}_i(\mathbf{I}_p - \boldsymbol{\Sigma}_g)\mathbf{S}_{ig}^{\text{oo}}]'[\mathbf{M}_i(\mathbf{I}_p - \boldsymbol{\Sigma}_g)\mathbf{S}_{ig}^{\text{oo}}\boldsymbol{\Sigma}_g\mathbf{M}_i']^{-1} \times \mathbf{M}_i(\mathbf{I}_p - \boldsymbol{\Sigma}_g)\mathbf{S}_{ig}^{\text{oo}}.$$

In the E-step, the expected value of the complete-data log-likelihood is updated. In practice, this amounts to replacing the z_{ig} in (3.2) by their expected values

$$\hat{z}_{ig} = \frac{\hat{\pi}_g \phi_{p_i^{\text{o}}}(\mathbf{x}_i^{\text{o}} \mid \hat{\boldsymbol{\mu}}_{ig}^{\text{o}}, \hat{\boldsymbol{\Sigma}}_{ig}^{\text{oo}})}{\sum_{h=1}^{G}\hat{\pi}_h \phi_{p_i^{\text{o}}}(\mathbf{x}_i^{\text{o}} \mid \hat{\boldsymbol{\mu}}_{ih}^{\text{o}}, \hat{\boldsymbol{\Sigma}}_{ih}^{\text{oo}})}, \tag{3.3}$$

where $\hat{\boldsymbol{\mu}}_{ig}^{\text{o}} = \mathbf{O}_i\hat{\boldsymbol{\mu}}_g$ and $\hat{\boldsymbol{\Sigma}}_{ig}^{\text{oo}} = \mathbf{O}_i\hat{\boldsymbol{\Sigma}}_g\mathbf{O}_i'$. Note that, in the E-step, we are conditioning on the current parameter estimates, hence the use of hats on the parameters in (3.3).

It follows that, the expected value of the complete-data log-likelihood is

$$\mathcal{Q}(\boldsymbol{\vartheta}) = \sum_{g=1}^{G}\sum_{i=1}^{n}\hat{z}_{ig}\log\pi_g + \frac{1}{2}\sum_{g=1}^{G}\left(\log \mid \boldsymbol{\Sigma}_g^{-1} \mid \sum_{i=1}^{n}\hat{z}_{ig} - \operatorname{tr}\left(\boldsymbol{\Sigma}_g^{-1}\sum_{i=1}^{n}\boldsymbol{\Omega}_{ig}\right)\right), \quad (3.4)$$

where

$$\boldsymbol{\Omega}_{ig} = \hat{z}_{ig}\left[(\hat{\mathbf{x}}_{ig} - \boldsymbol{\mu}_g)(\hat{\mathbf{x}}_{ig} - \boldsymbol{\mu}_g)' + (\mathbf{I}_p - \hat{\boldsymbol{\Sigma}}_g\hat{\mathbf{S}}_{ig}^{\text{oo}})\hat{\boldsymbol{\Sigma}}_g\right]$$

and

$$\hat{\mathbf{x}}_{ig} = \hat{\boldsymbol{\mu}}_g + \hat{\boldsymbol{\Sigma}}_g\hat{\mathbf{S}}_{ig}^{\text{oo}}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_g).$$

In the M-step, the model parameters are updated, by maximizing $\mathcal{Q}(\boldsymbol{\vartheta})$. Specifically, $\mathcal{Q}(\boldsymbol{\vartheta})$ is maximized with respect to π_g, $\boldsymbol{\mu}_g$, and $\boldsymbol{\Sigma}_g$, producing the following updates

$$\hat{\pi}_g = \frac{1}{n}\sum_{i=1}^{n}\hat{z}_{ig}, \quad \hat{\boldsymbol{\mu}}_g = \frac{1}{n_g}\sum_{i=1}^{n}\hat{z}_{ig}\hat{\mathbf{x}}_{ig}, \quad \hat{\boldsymbol{\Sigma}}_g = \frac{1}{n_g}\sum_{i=1}^{n}\hat{\boldsymbol{\Omega}}_{ig},$$

where $n_g = \sum_{i=1}^{n}\hat{z}_{ig}$ and $\hat{\boldsymbol{\Omega}}_{ig}$ is $\boldsymbol{\Omega}_{ig}$ as above with $\boldsymbol{\mu}_g$ replaced by $\hat{\boldsymbol{\mu}}_g$. The EM algorithm alternates between the E-step and the M-step until some stopping criterion is satisfied. The EM algorithm is heavily dependent on starting values and is prone to stopping at local maxima (McLachlan and Krishnan, 2007). These issues arise because of the single path monotonic nature of the EM algorithm. When used for model-based clustering, the EM algorithm can be initialized in two ways. Either by defining the initial model parameters and computing the expected class membership indicators, or by specifying starting values for the $\hat{z}_{ig}$ and initializing the model parameters according to the updates. To initialize the EM algorithm herein, the

$k-$means function was used to generate the starting values for $\hat{z}_{ig}$.

3.3 Evolutionary Algorithm for Clustering with Missing Data

3.3.1 Model and Fitness Function

The underlying model considered here is the mixture of multivariate Gaussian distributions. Also, all the group labels are assumed to be unknown even in cases where they are actually known for the purpose of clustering. Whereas the EM algorithm deals with the expected value of the complete-data loglikelihood, the EA developed in this chapter is single-objective, i.e., built to optimize one fitness function, and the fitness function is based on the (observed) log-likelihood, i.e.,

$$l(\boldsymbol{\vartheta}) = \sum_{i=1}^{n} \log \left\{ \sum_{g=1}^{G} \pi_g \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right\}, \tag{3.5}$$

where $\boldsymbol{\vartheta} = (\pi_1, \ldots, \pi_G, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_g)$ represents the model parameters. The EA developed herein can be seen as an approach for hard model-based clustering with missing data. The term "hard" in this context simply means that the estimated group membership labels are forced to take the values $\tilde{z}_{ig} \in \{0, 1\}$ as compared to the soft labels $\hat{z}_{ig} \in [0, 1]$ used in the EM algorithm. In order to avoid any mix-ups with the expected values $\hat{z}_{ig}$ used in the EM algorithm, we use $\tilde{z}_{ig}$ to denote the estimate of z_{ig} used in the EA. As the EA iterates, the estimated value of z_{ig} continues to change.

3.3.2 Evolutionary Algorithm

Our EA uses several individual parents, each of which is cloned multiple times with the cloned offspring reproducing. Also, because each step of the algorithm requires calculations among either parents or children, two further sets of indices are needed: $j = 1, ..., J$ indexes children and $k = 1, ..., K$ indexes parents. The fitness function for the jth child is simply the log-likelihood (3.5) evaluated using the expectations $\hat{\mathbf{x}}_{ig}$ and the parameter estimates $\hat{\pi}_{1j}, \ldots, \hat{\pi}_{Gj}, \hat{\boldsymbol{\mu}}_{1j}, \ldots, \hat{\boldsymbol{\mu}}_{Gj}, \hat{\boldsymbol{\Sigma}}_{1j}, \ldots, \hat{\boldsymbol{\Sigma}}_{Gj}$.

The K fittest solutions at each generation survive. These K solutions are cloned (i.e., produce children) to the next generation which ensures that the maximum fitness from one generation to another is non-decreasing. To be specific, the initial K single parents and all the clones are arranged in a descending order of fitness, where we choose the topmost K to become the next generation of single parents. Based on $\tilde{z}_{igk}$, the updates $\hat{\pi}_{gk}$, $\hat{\boldsymbol{\mu}}_{gk}$ and $\hat{\boldsymbol{\Sigma}}_{gk}$ $(g = 1, \ldots, G)$ are computed as for the EM algorithm (Section 3.2). Then, we can compute the probability that observation $\mathbf{x}_i$ belongs to component g for parent k given these current parameter estimates:

$$\hat{z}_{igk} = \frac{\pi_{gk}\phi_{p_i^{\mathrm{o}}}(\mathbf{x}_i^{\mathrm{o}} \mid \boldsymbol{\mu}_{gk}^{\mathrm{o}}, \boldsymbol{\Sigma}_{gk}^{\mathrm{oo}})}{\sum_{h=1}^{G} \pi_{hk}\phi_{p_i^{\mathrm{o}}}(\mathbf{x}_i^{\mathrm{o}} \mid \boldsymbol{\mu}_{hk}^{\mathrm{o}}, \boldsymbol{\Sigma}_{hk}^{\mathrm{oo}})} \tag{3.6}$$

for $i = 1, \ldots, n$, $g = 1, \ldots, G$ and $k = 1, \ldots, K$. The $\tilde{z}_{igk}$ are mutated by randomly sampling each observation's cluster membership according to the probabilities (3.6). At each iteration, K parents are stored and, during the reproduction phase, we sample J new matrices of $\tilde{z}_{igk}$ according to the $\hat{z}_{igk}$ corresponding to each of the K parents. As our EA is an iterative procedure, the stopping criterion adopted here is

the lack of progress approach where the EA is stopped once stagnation occurs. In this particular instance, when the log-likelihoods from the top K solutions remain the same or fail to increase over three consecutive generations, our EA is terminated.

Pseudo-Code

The detailed procedure followed in our EA is summarized in the following pseudo-code (Algorithm 1). Note that the code used in this thesis was written in R (R Core Team, 2020).

Algorithm 1 EA for GMM with Missing Data

initialize $\tilde{z}_{igk}$ matrices using k-means
initialize: k sets of parameters based on these $\tilde{z}_{igk}$
`stag` $= 0$
while `stag` < 3 **do**
  mutate: calculate $\hat{z}_{igk}$; sample J children (clones) $\tilde{z}_{igk}$ accordingly
  update: all J sets of parameters
  fitness: calculate log-likelihood for each of J children (clones)
  survival: sort J children (clones) and K parents in descending fitness order, select top K as new parents
  if the log-likelihoods of the top K solutions are the same as previous cycle **then**
    `stag++`
  else
    `stag` $= 0$
  end if
end while
return $\tilde{z}_{igk}$ corresponding to the highest log-likelihood

Our EA is fitted for different values of J and K and the best combination of J and K is then selected via the BIC. The effectiveness of our EA for traversing the fitness (log-likelihood) surface is illustrated in Section 3.4.

3.4 Illustrations

To assess the performance of our EA, we compare it to the EM algorithm in Section 3.2 and with the straightforward mean imputation (MI) method — where the missing data is replaced with the sample mean of the associated variable — followed by an EM algorithm on the resulting (complete) data. In the applications here, the data sets are complete and, for illustration purposes, we consider different degrees of missingness by excluding observations using an MAR mechanism, provided that each observation has at least one observable attribute. All the data sets discussed here are scaled before the analysis.

For the data sets considered, we assumed that there is no prior knowledge of the labels or the number of components (i.e., they are treated as a genuine clustering example). As is common, the BIC is used for model selection; in our case, to select J, K, and G. Because the true group labels are available for each data, the true and the predicted classifications can be compared using the ARI. Steinley (2004) outlined detailed arguments justifying the use of ARI in this instance, contrary to alternatives such as the misclassification rate.

3.4.1 Simulation

Two simulation scenarios are considered. In each case, the data are generated via the `genRandomClust` function from the R package called `clusterGeneration` (Qiu and Joe, 2006) for varying dimensions. In all, 100 artificially missing data sets are created by deleting at random under various specified missing proportions r.

In Scenario I, we generated a data set with $p = 3$ variables, $n = 323$ observations and well separated clusters by setting `sepVal` $= 0.03$, `numNonNoisy` $= 3$ and the remaining settings were left at default — a pairs plot of this data is presented in Figure 3.1. Over the 100 replicates for each missing proportion r, we report the average ARI values for different values of the number of survivors K and the number of children J in our EA (Table 3.1).

Table 3.1: Average ARI values for the number of survivors K and the number of children J, with varying missingness proportions r, for our EA for the simulated data from Scenario I (100 replicates).

		J		
r	K	10	20	30
5%	1	0.9110	0.9100	0.9103
	2	0.9862	0.9859	0.9910
	4	0.9932	0.9938	0.9935
10%	1	0.9148	0.9110	0.9098
	2	0.9657	0.9702	0.9802
	4	0.9808	0.9811	0.9799
20%	1	0.9129	0.9132	0.9123
	2	0.9235	0.9236	0.9139
	4	0.9274	0.9271	0.9271
30%	1	0.8584	0.8564	0.8579
	2	0.8687	0.8683	0.8621
	4	0.8651	0.8632	0.8657

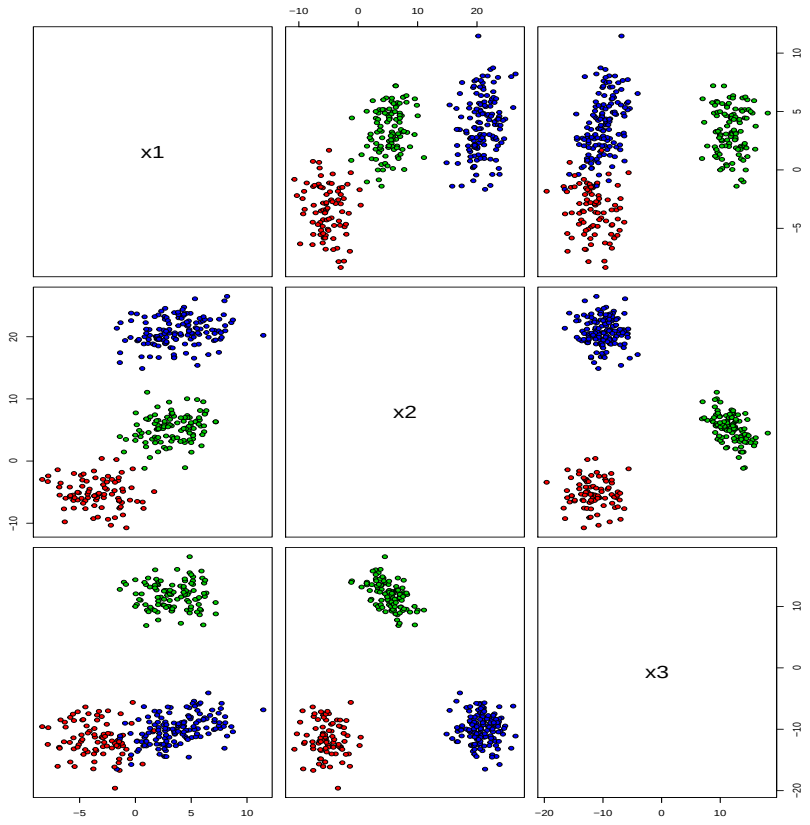

Figure 3.1: A pairs plot of the simulated data set in Scenario I, where colours reflect true classes.

It is unsurprising that, as the missingness proportion r increases, the classification performance, i.e., the average ARI, decreases (Table 3.1). For a fixed number of survivors K, increasing the number of children J has effectively no impact on the average ARI values—this is true for all considered values of r. However, increasing the number of survivors K results in notably improved ARI values for missing proportions $r \in \{5\%, 10\%\}$. There was a small improvement when K was increased at $r = 20\%$ and 30%. Based on these observations, we consider $K \geq 2$ survivors

hereafter.

The results from comparing our EA to the EM and MI approaches (Table 3.2) in Scenario I, show that both the EA and EM notably outperformed MI in all cases. Furthermore, the classification performance of the EA is better than the EM with higher average ARI values and lower associated standard deviations in all cases. The BIC selected an EA with $K = 2$ and $J = 20$, and the associated average BIC values are compared with those from the EM and MI approaches in Table 3.3.

Table 3.2: Average ARI values with standard deviations (sd) for EA ($K = 2$ and $J = 20$), EM and MI under varying missing proportions r for the simulated data in Scenario I (100 replicates).

r		EA	EM	MI
5%	mean	0.9859	0.9486	0.9139
	sd	0.0542	0.1458	0.0284
10%	mean	0.9702	0.9281	0.7915
	sd	0.0540	0.1524	0.1197
20%	mean	0.9236	0.9113	0.6607
	sd	0.0470	0.0852	0.0864
30%	mean	0.8683	0.8503	0.6053
	sd	0.0309	0.0877	0.0964

Table 3.3: Average BIC values for EA ($K = 2$ and $J = 20$), EM and MI for the simulated data from Scenario I.

r	EA	EM	MI
5%	−1166.62	−1212.46	−1494.87
10%	−1148.10	−1192.63	−1595.03
20%	−1093.40	−1106.37	−1502.74
30%	−1030.00	−1036.91	−1407.68

In Scenario II, we generate a data set with $p = 3$ variables, $n = 323$ observations and substantially overlapping clusters. This is a very difficult clustering problem (see

Figure 3.2) and classification results close to perfect are not expected. The results (Table 3.4) again show that the EA and EM approaches outperform the MI approach. This time, the EA and EM algorithms give comparable classification performance with the EA approach consistently obtaining slightly higher average ARI values in each case with very similar standard deviations. As usual, the EA is chosen based on the BIC. The chosen EA ($K = 2$ and $J = 20$) and the EM algorithm have very similar average BIC values in this scenario (Table 3.5).

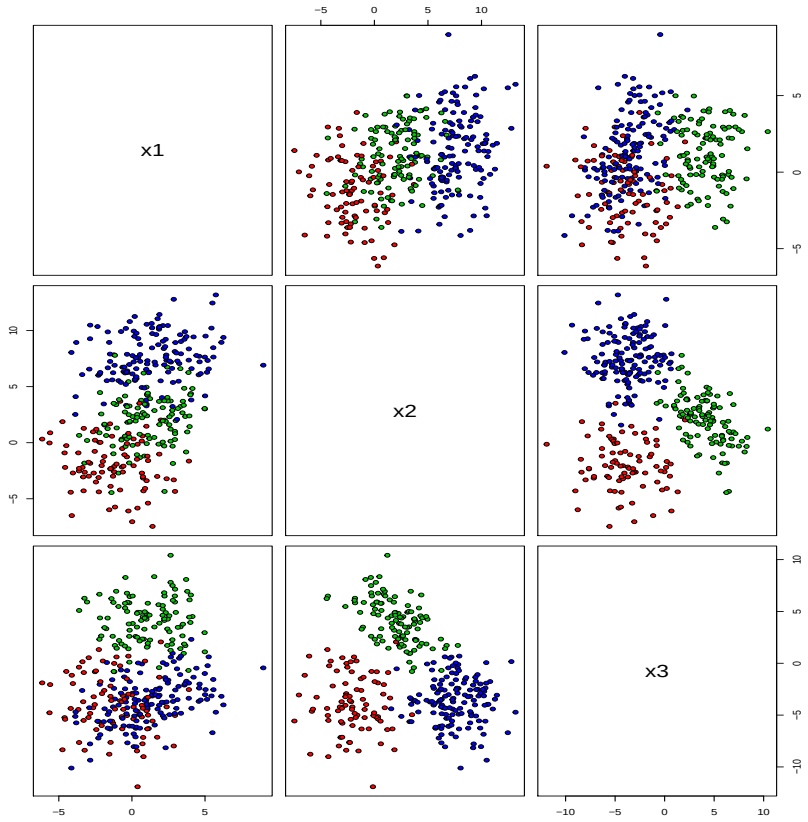

Figure 3.2: A pairs plot of the simulated data set in Scenario II, where colours reflect true classes.

Table 3.4: Average ARI values with standard deviations (sd) for EA ($K = 2$ and $J = 20$), EM and MI under varying missing proportions r for the simulated data from Scenario II (100 replicates).

r		EA	EM	MI
5%	mean	0.7991	0.7923	0.7633
	sd	0.0204	0.0189	0.0255
10%	mean	0.7577	0.7518	0.6792
	sd	0.0269	0.0253	0.0357
20%	mean	0.6754	0.6682	0.4558
	sd	0.0336	0.0352	0.1021
30%	mean	0.5900	0.5812	0.2629
	sd	0.0565	0.0646	0.0622

Table 3.5: Average BIC values for EA ($K = 2$ and $J = 20$), EM and MI for the simulated data from Scenario II.

r	EA	EM	MI
5%	−2490.49	−2489.95	−2586.26
10%	−2381.07	−2380.45	−2577.81
20%	−2155.33	−2154.67	−2509.16
30%	−1925.14	−1925.17	−2362.14

To investigate the run time of our proposed EA compared to EM and MI, we generated data sets with similar specifications as Scenario I, but with different numbers of variables, $p = \{5, 15, 30\}$. From Fig.3.3, we have the average run time given in seconds for the various methods. The EA has the longest run time, which is not surprising since it requires calculations among children (clones) at each iteration of the algorithm.

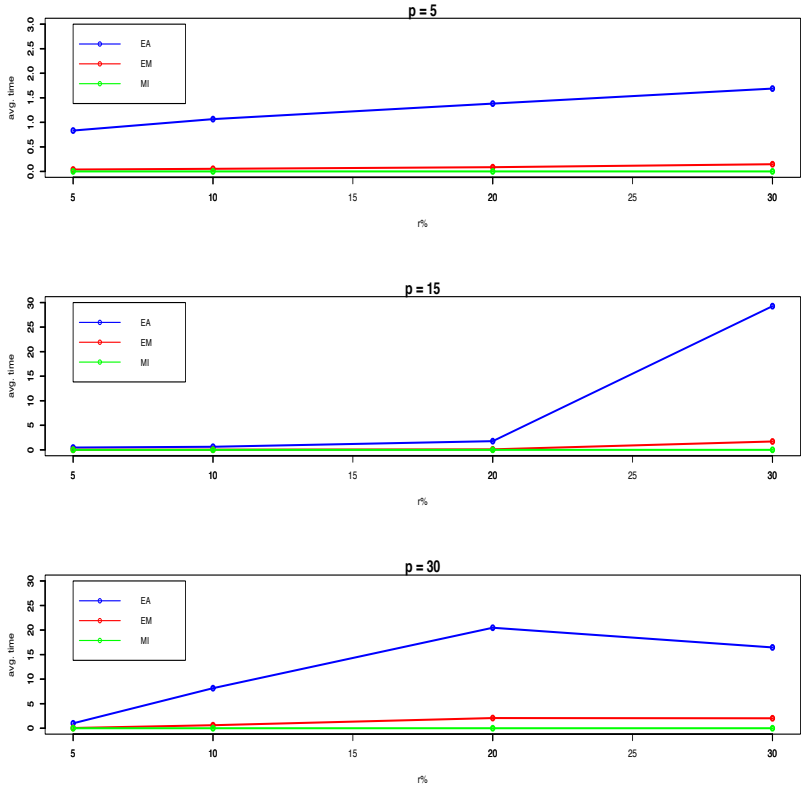

Figure 3.3: Plot of the average run time in seconds for the EA, EM and MI vs varying missingness proportions r, for simulated data sets with 5, 15 and 30 variables (100 replicates).

3.4.2 Real Data

Iris Data

Fisher (1936) discussed four measurements in centimeters on the attributes of the petal length, petal width, sepal length and sepal width from three species of irises (*Iris setosa*, *Iris virginica* and *Iris veriscolor*) originally collected by Anderson (1935). The data set consists of 50 samples from each of the three species and is available in

the R package `datasets`. The results from applying the EA, EM and MI approaches to the iris data are summarized in Table 3.6. For the EAs, we consider $K \in \{1, 2\}$ and $J \in \{10, 20\}$. For $K = 1$ and regardless of J, the EA yielded slightly inferior ARI values compared to $K = 2$—this is not surprising when one considers the simulation scenarios (Section 3.4.1). However, it is worth noting that $K = 1$ still resulted in superior average ARI values compared to the EM and the MI approaches. Overall, the MI approach is the worst performer. The EA chosen based on the BIC, i.e., with $K = 2$ and $J = 20$, outperforms the EM algorithm in all cases, with consistently higher average ARI values and lower associated standard deviations as well as superior average BIC values (see Tables 3.6 and 3.7).

Table 3.6: Average ARI values with standard deviations (sd) for EA ($K = 2$ and $J = 20$), EM and MI under varying missing proportions r for the iris data (100 replicates).

r		EA	EM	MI
5%	mean	0.8977	0.8849	0.5581
	sd	0.0500	0.0900	0.1257
10%	mean	0.8491	0.8187	0.6071
	sd	0.1248	0.1542	0.1512
20%	mean	0.8331	0.7984	0.5441
	sd	0.0925	0.1383	0.1402
30%	mean	0.7781	0.7046	0.3897
	sd	0.0966	0.1505	0.1182

Diabetes Data

The diabetes data studied by Reaven and Miller (1979) considers the relationship between chemical and overt diabetes in 145 (non-obese) adults. Three measurements are taken on the subject: degree of glucose intolerance, insulin response to oral

Table 3.7: Average BIC values for EA ($K = 2$ and $J = 20$), EM and MI for the Iris data.

r	EA	EM	MI
5%	−782.46	−795.00	−1029.07
10%	−744.98	−768.20	−1144.12
20%	−756.59	−739.25	−1246.76
30%	−720.98	−708.00	−1221.41

glucose and insulin resistance where patients are classified into one of three types. The data is freely available in the `mclust` package (Scrucca et al., 2017) in R. Our EA is applied to these data with varying missing proportions r for $K \in \{1, 2\}$ survivors and $J \in \{10, 20\}$ children. The BIC chooses an EA with $K = 2$ and $J = 20$. The MI approach again gives poor performance, while the EA and EM approaches give comparable performance (Tables 3.8 and 3.9). However, the EA consistently has slightly higher average ARI values and lower standard deviations compared to the EM algorithm.

Table 3.8: Average ARI values with standard deviations (sd) for EA ($K = 2$ and $J = 20$), EM and MI under varying missing proportions r for the diabetes data (100 replicates).

r		EA	EM	MI
5%	mean	0.6613	0.6279	0.5776
	sd	0.0259	0.0779	0.0588
10%	mean	0.6461	0.6138	0.4929
	sd	0.0481	0.0818	0.0800
20%	mean	0.6109	0.5951	0.4594
	sd	0.0648	0.0831	0.0673
30%	mean	0.5773	0.5552	0.3806
	sd	0.0803	0.0865	0.0671

Table 3.9: Average BIC values for EA ($K = 2$ and $J = 20$), EM and MI for the diabetes data.

r	EA	EM	MI
5%	−480.12	−486.77	−593.34
10%	−478.04	−484.27	−650.38
20%	−463.78	−467.61	−689.53
30%	−438.71	−439.50	−665.38

Female Voles Data

The female voles data are available in the `Flury` package (Flury, 2012) in R. The data contain six morphometric measurements, as well as age, for 86 female voles from two species: *Microtus californicus* and *Microtus ochrogaster*. Again, the three approaches were applied. For the EAs, we consider $K \in \{2, 4\}$ and $J \in \{10, 20, 30\}$ and the BIC selected the best combination of K and J to be 4 and 20, respectively. The results (Tables 3.10 and 3.11) again show that the MI approach gives the worst performance. The EA and EM approaches give similar performance, with the EA approach again always having a slightly higher average ARI value and smaller standard deviation.

Table 3.10: Average ARI values with standard deviations (sd) for EA ($K = 4$ and $J = 20$), EM and MI under varying missing proportions r for the female voles data (100 replicates).

r		EA	EM	MI
5%	mean	0.9418	0.9165	0.9027
	sd	0.0463	0.0613	0.0464
10%	mean	0.9209	0.9105	0.8761
	sd	0.0564	0.0591	0.0498
20%	mean	0.8914	0.8857	0.7684
	sd	0.0654	0.0726	0.1241
30%	mean	0.8358	0.8317	0.6416
	sd	0.0911	0.0932	0.1513

Table 3.11: Average BIC values for EA ($K = 4$ and $J = 20$), EM and MI for the female voles data.

r	EA	EM	MI
5%	−1342.46	−1345.60	−1365.60
10%	−1298.10	−1299.74	−1403.33
20%	−1201.79	−1200.32	−1433.75
30%	−1064.97	−1050.19	−1412.77

Banknote Data

The banknote data are freely available from the `mclust` package in R. They contain six measurements, all in millimeters (mm), on 100 genuine and 100 counterfeit Swiss 1000-franc bank notes. This is the easiest of the clustering problems we consider amongst the famous real data sets considered herein. The EA with varying missing proportions r for $K \in \{2, 4\}$ survivors and $J \in \{10, 20, 30\}$ children is applied to these data and similar results are obtained for the various EA scenarios; however, the EA with $K = 4$ and $J = 30$ is selected by the BIC. As before, the MI approach gives the worst performance, in terms of both average ARI and average BIC, for all r; however, its classification performance is closer to the EA and EM approaches for smaller values of r. On this data set, the EA and EM approaches give very similar performance overall (Tables 3.12 and 3.13).

Table 3.12: Average ARI with standard deviations (sd) for EA, EM and MI under varying missingness proportions r for the banknote data (100 replicates).

r		EA	EM	MI
5%	mean	0.9695	0.9701	0.9642
	sd	0.0144	0.0144	0.0182
10%	mean	0.9616	0.9609	0.9487
	sd	0.0178	0.0175	0.0334
20%	mean	0.9323	0.9291	0.8933
	sd	0.0226	0.0231	0.0559
30%	mean	0.8845	0.8776	0.7534
	sd	0.0331	0.0343	0.0967

Table 3.13: Average BIC values for EA ($K = 4$ and $J = 30$), EM and MI for the banknote data.

r	EA	EM	MI
5%	−2695.88	−2695.83	−2853.36
10%	−2590.28	−2590.38	−2888.84
20%	−2369.03	−2370.03	−2900.19
30%	−2142.42	−2143.46	−2844.96

Body Data

Heinz et al. (2003) report data on 24 body dimension measurements as well as age, weight, height, and gender for 260 women and 247 men. The total of 507 people involved in this study were active individuals i.e., exercised several hours a week, and in their twenties and thirties with a few older men and women. The body data is available in the R package gclus (Hurley, 2004).

Applying the EA approach introduced herein to these data with varying missing proportions, r, for $K \in \{1, 2\}$ survivors and $J \in \{10, 20\}$ children, the BIC chooses an EA with $K = 2$ and $J = 20$. Again, the MI approach gives poor performance in terms of the ARI values except for $r = 5\%$ with lower BIC values (Tables 3.14 and 3.15). On the other hand, the EA outperformed the EM with consistently higher ARI and BIC values with comparable standard deviations.

Table 3.14: Average ARI with standard deviations (sd) for EA, EM and MI under varying missingness proportions r for the body data (100 replicates).

r		EA	EM	MI
5%	mean	0.8262	0.8029	0.8262
	sd	0.0269	0.0269	0.0685
10%	mean	0.8197	0.8045	0.7668
	sd	0.0382	0.0271	0.1025
20%	mean	0.8113	0.8053	0.5214
	sd	0.0520	0.0417	0.3078
30%	mean	0.8131	0.8094	0.0978
	sd	0.0366	0.0323	0.0496

Table 3.15: Average BIC values for EA ($K = 2$ and $J = 20$), EM and MI for the body data.

r	EA	EM	MI
5%	−19347.42	−19356.84	−22713.78
10%	−18763.29	−18769.37	−24193.48
20%	−17526.48	−17534.03	−25782.64
30%	−16237.60	−16242.12	−26119.67

3.5 Discussion

An EA for model-based clustering with incomplete data has been developed and implemented in R. Two simulation studies and real analyses using five famous data sets revealed that the EA approach usually gives comparable or superior performance when compared to the EM algorithm. Note that Dasgupta and Raftery (1998) consider that BIC differences of more than 10 constitute "very strong evidence". Across all 20 real data scenarios considered in this chapter (five data sets for four values of r), there was a difference of more than 10 in the average BIC between the EA and EM approaches on just six occasions. In five of the six cases, the EA had the better average BIC value. Further, in simulation Scenario I, there was a difference of more than 10 between the average BIC values for the EA and EM approaches for three of the four r values considered. In all three cases, the EA had the better average BIC value. This being said, comparing the average BIC values is probably not quite as meaningful as comparing the average ARI values. The reason is twofold: the calculation of the log-likelihood for each EA uses hard $\tilde{z}_{ig}$ whereas the log-likelihood in the case of the EM approach is computed using soft $\hat{z}_{ig}$; and, when assessing an approach for clustering, the ultimate assessment is often taken to be classification

performance. Comparing the EA and EM approaches across the 20 real data scenarios based on average ARI values, we see that the average ARI value is higher for the EA approach in 19/20 cases. Hence the EA produces more accurate clustering results albeit being less efficient computationally. This work is most important as the first use of evolutionary algorithms for clustering with missing data. Because the proposed EA uses hard classifications, one could view it as an extension of k-means clustering to both non-spherical clusters and incomplete data. Herein, the missing data mechanism is assumed to be MAR and a departure from this assumption will be considered as future work. Also, each generation of our EA uses mutations only and modifying our EA by using a crossover step followed by a mutation step will also be considered. The EM algorithm is well known to be sensitive to starting values, and a detailed assessment of the sensitivity of our EA to starting values will be a topic of future work. In this work, we focused on the classical EM algorithm and, as future work, other algorithms to estimate the mixture parameters such as classification EM and stochastic EM by Celeux and Govaert (1992) will be explored. Finally, while the code used for this work is written in `R`, we plan to develop `C` code to improve computational efficiency.

Chapter 4

Evolutionary Algorithms for Gaussian Parsimonious clustering Models

4.1 Introduction

Cluster analysis finds sub-groups of similar observations within populations. In model based clustering, the component densities in finite mixture models can be treated as analogous to clusters and the problem reduces to assigning observations to components (McNicholas, 2016). The EM algorithm is popularly employed in estimating the parameters in model-based clustering scenarios. In this chapter, we develop an EA utilizing a crossover step followed by a mutation step to estimate the parameters for each member of the GPCM family (Section 2.2.2) and subsequently

determine group memberships of each observation. To assess the performance of our proposed EA and the EM algorithm, we applied both algorithms to several real data sets.

4.1.1 Parameter Estimation for GPCMs

Consider the model-based clustering paradigm, and let z_{ig} denote component membership. Parameter estimation for each member of the GPCM family is carried out using an EM algorithm, which is based on the complete-data. The complete-data log-likelihood is given as

$$l_c(\boldsymbol{\vartheta}) = \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}[\log \pi_g + \log \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)]. \tag{4.1}$$

The E-step of the EM algorithm involves computing the expected value of the complete-data log-likelihood. This amounts to replacing the z_{ig} in (4.1) by their expected values

$$\hat{z}_{ig} = \frac{\hat{\pi}_g \phi(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g)}{\sum_{h=1}^{G} \hat{\pi}_h \phi(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_h, \hat{\boldsymbol{\Sigma}}_h)},$$

for $i = 1, \ldots, n$ and $g = 1, \ldots, G$. The M-step entails maximizing the expected value of (4.1) with respect to the model parameters. For an in-depth information on the EM algorithm parameter estimates for the GPCMs, see Celeux and Govaert (1995) and Fraley and Raftery (1999). The estimates of $\hat{z}_{ig}$, $\hat{\boldsymbol{\mu}}_g$ and $\hat{\pi}_g$ are the same for each member of the GPCM family with

$$\hat{\boldsymbol{\mu}}_g = \frac{\sum_{i=1}^{n} \hat{z}_{ig}\mathbf{x}_i}{\sum_{i=1}^{n} \hat{z}_{ig}} \quad \text{and} \quad \hat{\pi}_g = \frac{n_g}{n} \quad \text{where} \quad n_g = \sum_{i=1}^{n} \hat{z}_{ig}.$$

4.2 Evolutionary Algorithms for GPCMs

4.2.1 Model and Fitness Function

The underlying model considered here is a mixture of multivariate Gaussian distributions. Again, this is a clustering technique, hence all component memberships are unknown or treated as such. The fitness function for the EA developed herein is based on the (observed) log-likelihood

$$l(\boldsymbol{\vartheta}) = \sum_{i=1}^{n} \log \left\{ \sum_{g=1}^{G} \pi_g \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right\}, \tag{4.2}$$

where $\boldsymbol{\vartheta} = (\pi_1, \ldots, \pi_G, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_g)$ denotes the model parameters.

The estimated value of the component membership labels, z_{ig} evolves as our EA progresses. We use $\tilde{z}_{ig}$ to denote the estimate of z_{ig} in our EA and $\hat{z}_{ig}$ for the expected values used in the EM algorithm. The estimated component membership labels of $\mathbf{x}_i$ in our EA is given by $\tilde{\mathbf{z}}_i = (\tilde{z}_{i1}, \ldots, \tilde{z}_{iG})$ which are restricted to values $\tilde{z}_{ig} \in \{0, 1\}$ as compared to the soft labels $\hat{\mathbf{z}}_i \in [0, 1]$ used in the EM algorithm. Hence the EA developed can be seen as an approach for hard model-based clustering. The fitness function is the log-likelihood (4.2) evaluated at the estimates

$$\tilde{\boldsymbol{\mu}}_g = \frac{\sum_{i=1}^{n} \tilde{z}_{ig} \mathbf{x}_i}{\sum_{i=1}^{n} \tilde{z}_{ig}}, \qquad \tilde{\pi}_g = \frac{n_g}{n} \quad \text{and} \quad \tilde{\boldsymbol{\Sigma}}_g \tag{4.3}$$

where $n_g = \sum_{i=1}^{n} \tilde{z}_{ig}$ and $\tilde{\boldsymbol{\Sigma}}_g$ for each member of the GPCM family as presented in Table 2.1 (Section 2.2.2).

4.2.2 Evolutionary Algorithm

For this EA, we used a number of single parents and each is cloned several times, with the cloned children reproducing. Each step of the algorithm requires calculations among either parents or children, hence two further sets of indices are introduced: $j = 1, \ldots, J$ indexes children and $k = 1, \ldots, K$ indexes parents. Our EA discussed here utilizes both crossover and mutation operations.

For each cloned child, we select two observations at random and swap their group membership labels, i.e., $\tilde{z}_i$ values. This swap is only necessary if they have different membership labels. The offspring from different single parents are created in such a way that they are never crossed. After the crossover, the children together with the parents are arranged in a descending order of fitness. The new generation of single parents are then selected by choosing the top K from the ordered list. This crossover process avoids stopping at local maxima of the fitness surface, i.e., the log-likelihood surface. However, each iteration also includes a mutation step, since simply replacing the membership labels of one point with the membership labels of another may not improve the clustering results. During the mutation stage, random mutations are generated within the $\tilde{z}_{ik}$. The K parents will produce J children each with J taken to be approximately n in number. These children will each have their $\tilde{z}_i k$ mutated in such a way as to alter the cluster they belong to. The crossover step followed by a mutation step are performed multiple times until our EA stagnates.

The following is pseudo-code (Algorithm 2) to outline the EA developed here. Note that the code used in this thesis was written in R and that the comments used in the pseudo-code below are in R comment style i.e., #.

It should be noted that, after the first crossover step, the parents are simply the best K elements in terms of fitness. Also, the estimates in (4.3) are computed in `update` whilst `fitness` entails computing the log-likelihood (4.2) with these estimates. There is a good general interpretation of this EA: the crossover step provides diversity, and the mutation step allows improved fitness (clustering), which crossover alone cannot provide.

Algorithm 2 EA for Gaussian Model-Based Clustering

initialize $\tilde{z}_{igk}$ matrices using k-means

initialize: k sets of parameters based on these $\tilde{z}_{igk}$

`stag` $= 0$

while `stag` $<$ `stagnation` **do**

First, crossover

crossover: select two unequal labels from $\tilde{z}_{igk}$ and swap them to get J children (clones)

update: all J sets of parameters

fitness: calculate log-likelihood for each of J children (clones)

survival: sort J children (clones) and K parents in descending fitness order, select top K as new parents

Now, mutate

mutate: change the $\tilde{z}_{igk}$ for each value of i such that its cluster membership changes

update: all J sets of parameters

fitness: calculate log-likelihood for each of J children (clones)

survival: sort J children (clones) and K parents in descending fitness order, select top K as new parents

if the log-likelihoods of the top K solutions the same as previous cycle **then**

`stag++`

else

`stag` $= 0$

end if

end while

return $\tilde{z}_{igk}$ corresponding to the highest log-likelihood

4.2.3 Computational Aspect

Initialization

To initialize z_{igk}, a k-means clustering algorithm was run and the resulting $\tilde{z}_{igk}$ was then taken as the starting group membership labels for all the models.

Stopping Criterion

Our stopping criterion is simple lack of progress. Specifically, the EA is stopped once stagnation occurs, i.e., we terminate the EA when the log-likelihoods from top K solutions fail to increase over a number of consecutive generations.

4.3 Illustrations

In this section, our EA is applied to several real data sets that are commonly used for illustration in the mixture model-based clustering literature. We compare the performance of our EA — which we refer to here as crossover followed by mutation EA (CMEA)— and the EM algorithm. The EM algorithm is implemented using the gpcm function in the mixture package (Pocuca et al., 2021) from R with the argument, start set to 0 for the $k-$means function to be used for initialization. This is consistent with the initialization approach adopted in our EAs thus making it appropriate to compare the two approaches. To be complete, we also considered the EA developed in Chapter 3 — which we refer to as mutation only EA (MEA) to avoid confusion between the two EAs.

All the illustrations in this section are performed as a true cluster analysis, nevertheless the true labels are always known making it possible for the predicted classifications to be compared. We carried out this comparison using the ARI. We used the BIC to choose the model type (EII, VII,...,VVV) and the number of components G, for the GPCMs as well as the number of parents, clones and the stagnation values. All the analysis are performed using the R programming software and the data sets considered are scaled prior to analysis using the scale function in R.

4.3.1 Banknote Data

The first real data considered is the famous banknote data set previously analyzed. The CMEA approach introduced herein is applied to these data, for the GPCM family using k-means start, with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and stagnation $\in \{2, 4\}$. The best model selected was the EEV with BIC $= -2781.27$ and the number of components, $G = 2$, for stagnation $= 2$, $J = 10$ and $K = 1$. Table 4.1 represents the cross-tabulation of the MAP classifications from the CMEA versus the true class. A near perfect classification performance was achieved, with just one misclassification, i.e., ARI $= 0.980$. Over all 8 runs, i.e., combinations of K, J and stagnation for the EEV, identical results were obtained. The MEA is also applied to these data with the same parameters as above and the results are identical to that of the CMEA — the accompanying classification performance is given in Table 4.2 with ARI $= 0.980$.

All fourteen GPCMs were fitted using the mixture package with $k-$means start and the best model selected was the EEV with BIC $= -2781.26$ for $G = 2$ resulting in one misclassified observation (Table 4.3; ARI=0.980). The results from both EAs

and the EM are similar.

Table 4.1: Cross-tabulation of the predicted classifications (1,2) from CMEA versus true class (Counterfeit or Genuine) for the banknote data.

	Cluster	
	1	2
Counterfeit	100	0
Genuine	1	99

Table 4.2: Cross-tabulation of the predicted classifications (1,2) from MEA versus true class (Counterfeit or Genuine) for the banknote data.

	Cluster	
	1	2
Counterfeit	100	0
Genuine	1	99

Table 4.3: Cross-tabulation of the predicted classifications (1,2) from EM versus true class (Counterfeit or Genuine) for the banknote data.

	Cluster	
	1	2
Counterfeit	100	0
Genuine	1	99

4.3.2 Coffee Data

Streuli (1973) reports on the chemical composition of coffee samples and is available from the R package pgmm. The Coffee data comprises 43 samples from 29 countries.

Each sample is either of the Arabica or Robusta variety. We excluded total chlorogenic acid from the analysis since it is the sum of neochlorogenic, isochlorogenic and chlorogenic acid values, hence twelve out of the thirteen chemical constituents are considered.

Our CMEA is applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and stagnation $\in \{2, 4\}$. The best model, i.e., the model with the highest BIC $= -1334.22$, was the VEI model with $G = 2$, for stagnation $= 2$, $J = 10$ and $K = 1$. Over all 8 runs — combinations of K, J and stagnation — identical and perfect classification performance was obtained with no misclassifications (Table 4.4; ARI $= 1.00$). Similar results were obtained for the MEA but are not reported here.

Analysis of the coffee data using the `mixture` package with k-means start selected the VEI model with BIC $= -1334.22$ for $G = 2$ resulting in no misclassified observations (Table 4.5; ARI $= 1.00$). Again, these results are similar to the results obtained from the EAs.

Table 4.4: Cross-tabulation of the predicted classifications (1,2) from CMEA versus true class (Arabica or Robusta) for the coffee data.

	Cluster	
	1	2
Arabica	36	0
Robusta	0	7

Table 4.5: Cross-tabulation of the predicted classifications (1,2) from EM versus true class (Arabica or Robusta) for the coffee data.

	Cluster	
	1	2
Arabica	36	0
Robusta	0	7

4.3.3 Australian Institute of Sports (AIS) Data

The Australian institute of sports (AIS) data was sourced from the R package alr3 (Weisberg, 2014, 2018). The data reports on physical measurements and blood measurements from high performance athletes at the AIS, for 202 athletes (100 females; 102 males) on 11 quantitative variables.

The CMEA approach was applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and stagnation $\in \{2, 4\}$. The best model selected was the EEV model with BIC $= -2479.26$ for $G = 2$, stagnation $= 2$, $J = 10$ and $K = 1$. Over all 8 runs — combinations of K, J and stagnation — similar and near perfect classification performance was achieved with four misclassifications (Table 4.6; ARI $= 0.922$). Again, similar results were obtained for the MEA but are not reported here.

Table 4.6: Cross-tabulation of the predicted classifications (1,2) from our CMEA versus true class (male or female) for the AIS data.

	Cluster	
	1	2
Male	98	4
Female	0	100

Analysis of the AIS data using the `mixture` package with k-means start selected

the EEV model with BIC = −2479.18 for $G = 2$ components. This resulted in four misclassified observations (Table 4.7; ARI = 0.922). These results are similar to the results obtained from the EAs.

Table 4.7: Cross-tabulation of the predicted classifications (1,2) from EM versus true class (male or female) for the AIS data.

	Cluster	
	1	2
Male	98	4
Female	0	100

4.3.4 Female Voles Data

The real data analyzed here is the female voles data that was previously analyzed. Both CMEA and MEA are applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and $\mathsf{stagnation} \in \{2, 4\}$. The EAs selected the EEE covariance structure with BIC = −1316.72 for $G = 2$ components for $\mathsf{stagnation} = 2$, $J = 10$ and $K = 1$. Over all 8 runs — combinations of K, J and $\mathsf{stagnation}$ — similar and near perfect classification performance was achieved with two misclassified observations (Table 4.8; ARI = 0.908).

Fitting all fourteen GPCMs with `mixture` package with k-means start selected the EEE covariance structure with BIC = −1316.70 for $G = 2$ components. This resulted in two misclassified observations (Table 4.9; ARI = 0.908). Both the EAs and the EM selected the same model with the same number of components. Also, the ARIs from the cross-tabulation of both methods are similar.

Table 4.8: Cross-tabulation of the predicted classifications (1,2) from CMEA versus true class (Californicus or Ochrogaster) for the female voles data.

	Cluster	
	1	2
Californicus	41	0
Ochrogaster	2	43

Table 4.9: Cross-tabulation of the predicted classifications (1,2) from EM versus true class (Californicus or Ochrogaster) for the female voles data.

	Cluster	
	1	2
Californicus	41	0
Ochrogaster	2	43

4.3.5 Seeds Data

The Seeds data set (Charytanowicz et al., 2010) publicly available via the UCI Machine Learning Repository is considered here. The data set contains measurements on kernels from three varieties of wheat: Kama, Rosa, and Canadian. Each of the variety consists of 70 elements. We consider seven measurements for each variety of wheat and attempt to cluster the scaled data according to wheat variety.

Both CMEA and MEA are applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and $\mathsf{stagnation} \in \{2, 4\}$. The CMEA selected the EEV covariance structure with BIC = 139.17 for $G = 3$ components for $\mathsf{stagnation} = 2$, $J = 30$, and $K = 1$. Over all 8 runs — combinations of K, J and $\mathsf{stagnation}$ — identical and good classification performance was obtained with thirty-one misclassified observations (Table 4.10; ARI = 0.630). The MEA approach produced similar results to the CMEA approach but

are not reported here.

Table 4.10: Cross-tabulation of the predicted classifications (1,2,3) from CMEA versus true class (Kama, Rosa or Canadian) for the seeds data.

	Cluster		
	1	2	3
Kama	54	1	15
Rosa	15	55	0
Canadian	0	0	70

Analysis of the seeds data using the `mixture` package with k-means start selected the EEV model with BIC = 139.209 for $G = 3$ components. This resulted in thirty-one misclassified observations (Table 4.11; ARI = 0.630). These results are similar to the results obtained from the EAs.

Table 4.11: Cross-tabulation of the predicted classifications (1,2,3) from EM versus true class (Kama, Rosa or Canadian) for the seeds data.

	Cluster		
	1	2	3
Kama	54	1	15
Rosa	15	55	0
Canadian	0	0	70

4.3.6 Iris Data

We also consider the famous (Fisher's and Anderson's) iris data set that was previously analyzed. Both CMEA and MEA are applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and $\mathsf{stagnation} \in \{2, 4\}$. The CMEA selected the VEV covariance structure with BIC = -789.45 for $G = 3$ components for $\mathsf{stagnation} = 4$, $J = 10$, and

$K = 1$. Over all 8 runs — combinations of K, J and stagnation — identical and good classification performance was obtained with four misclassified observations (Table 4.12; ARI = 0.922) except for the combination stagnation = 2, $J = 10$, and $K = 1$ where we had five missclassified observations. The MEA selected the VEV covariance structure with BIC = -789.45 for $G = 3$ components for stagnation = 2, $J = 10$, and $K = 1$. Over all 8 runs — combinations of K, J and stagnation — identical and good classification performance was obtained with four misclassified observations similar to the results obtained by CMEA.

Table 4.12: Cross-tabulation of the predicted classifications (1,2,3) from CMEA versus true class (Setosa, Versicolor or Virginica) for the iris data.

	Cluster		
	1	2	3
Setosa	50	0	0
Versicolor	0	46	4
Virginica	0	0	50

Performing the analysis of the iris data using the `mixture` package with k-means start selected the VEV model with BIC = -789.37 for $G = 3$ components. This resulted in four misclassified observations (Table 4.13; ARI = 0.922). Again, these results are similar to the results obtained from the EAs.

Table 4.13: Cross-tabulation of the predicted classifications (1,2,3) from EM versus true class (Setosa, Versicolor or Virginica) for the iris data.

	Cluster		
	1	2	3
Setosa	50	0	0
Versicolor	0	46	4
Virginica	0	0	50

4.3.7 Italian Wine Data

The Italian wine data set (Forina et al., 1986) was sourced from the pgmm package in R and contains twenty-seven chemical measurements on 178 samples of three varieties of red wine: Barolo, Grignolino and Barbera. We attempt to cluster the scaled data according to wine varieties.

The CMEA approach was applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and stagnation $\in \{2, 4\}$. The best model selected was the VVI with BIC $= -12103.88$ for $G = 3$, stagnation $= 2$, $J = 10$ and $K = 1$. Over all 8 runs — combinations of K, J and stagnation for the VVI model — identical and good classification performance was obtained with six misclassifications (Table 4.14; ARI = 0.895). Again, similar results were obtained for the MEA but not reported here.

Analysis of the Italian wine data with twenty-seven variables using the `mixture` package with k-means start selected the VVI model with BIC $= -12103.74$ for $G = 3$ components. This resulted in six misclassified observations (Table 4.15; ARI = 0.895). These results are similar to the results obtained from the EAs.

Table 4.14: Cross-tabulation of the predicted classifications (1,2,3) from CMEA versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with twenty-seven variables.

	Cluster		
	1	2	3
Barolo	58	1	0
Grignolino	4	66	1
Barbera	0	0	48

Table 4.15: Cross-tabulation of the predicted classifications (1,2,3) from EM versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with twenty-seven variables.

	Cluster		
	1	2	3
Barolo	58	1	0
Grignolino	4	66	1
Barbera	0	0	48

4.3.8 Crabs Data

The Crabs data reported by Campbell and Mahon (1974) are available in the MASS library (Ripley et al., 2020; Venables and Ripley, 2002) for R. It comprises 200 rows describing five morphological measurements on two species of crab (blue and orange) and, further separated into two genders.

Both CMEA and MEA are applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and stagnation $\in \{2, 4\}$. The CMEA selected the EVV covariance structure with BIC = 39.22 for $G = 2$ components for stagnation $= 2$, $J = 10$ and $K = 4$. A cross-tabulation of the predicted classifications (1,2) versus true class (Male or Female) resulted in an ARI value of 0.756 (Table 4.16). Similar results were obtained for the

MEA but are not reported here.

Table 4.16: Cross-tabulation of the predicted classifications (1,2) from CMEA versus true class (Female or Male) for the crabs data.

	Cluster	
	1	2
Female	90	10
Male	3	97

Fitting all fourteen GPCMs with `mixture` package with k-means start selected the EVV covariance structure with BIC = 39.34 for $G = 2$ components. A cross-tabulation of the predicted classifications (1,2) versus true class (Male or Female) resulted in an ARI value of 0.736 (Table 4.17). Both the CMEA and the EM selected the same model with the same number of groups. However, CMEA has a higher ARI value compared to the ARI value from the EM.

Table 4.17: Cross-tabulation of the predicted classifications (1,2) from CMEA versus true class (Female or Male) for the crabs data.

	Cluster	
	1	2
Female	89	11
Male	3	97

4.3.9 Olive Oil Data

Forina and Tiscornia (1982) and Forina et al. (1983) report the percentage composition of eight fatty acids found by lipid fraction of 572 Italian olive oils. The data

originate from three regions namely, Southern Italy, Sardinia, and Northern Italy. However, there are several different areas in each region. Southern Italy is made up of Calabria, Sicily, South and North Apulia; Sardinia is divided into Costal Sardinia and Inland Sardinia; and Northern Italy comprises East Liguria, West Liguria and Umbria. These data are publicly available within the pgmm package (McNicholas et al., 2018) for R.

The EAs and the EM are applied to the Olive Oil data to classify them into the appropriate area. Both CMEA and MEA selected the EVV model with BIC = -5817.73 for $G = 3$ components for stagnation $= 4$, $J = 10$ and $K = 1$. A cross-tabulation of the predicted classifications (1,2,3) versus true class (Southern Italy, Sardinia, Northern Italy) resulted in an ARI value of 0.524 (Table 4.18).

Table 4.18: Cross-tabulation of the predicted classifications (1,2,3) from CMEA versus true class (Southern Italy, Sardinia, Northern Italy) for the olive data.

	Cluster		
	1	2	3
Southern Italy	222	0	101
Sardinia	0	98	0
Northern Italy	0	151	0

Again, fitting all fourteen GPCMs, the EM selected the EVV model with BIC = -5817.45 for $G = 3$ components and this resulted in an ARI value of 0.525 from the cross-tabulation of the predicted classification versus true class (Table 4.19). The EAs and the EM selected the same model with similar BIC values as well as the ARI values.

Table 4.19: Cross-tabulation of the predicted classifications (1,2,3) from CMEA versus true class (Southern Italy, Sardinia, Northern Italy) for the olive data.

	Cluster		
	1	2	3
Southern Italy	223	0	100
Sardinia	0	98	0
Northern Italy	0	151	0

4.3.10 Wisconsin Breast Cancer Data

The Wisconsin breast cancer (WBC) data are available from the UCI Machine Learning Repository (Dua and Graff, 2019) and contains 30 quantitative features computed from digitized images of 567 fine needle aspirates of breast masses. Of the 569 samples, 357 are benign and 212 are malignant.

The EA approaches introduced herein are applied to these data for the GPCM family using k-means start, with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and $\mathsf{stagnation} \in \{2, 4\}$. The best model selected was the VEE with BIC $= -6452.22$ and the number of components, $G = 2$, for $\mathsf{stagnation} = 2$, $J = 10$ and $K = 1$. Table 4.20 represents the cross-tabulation of the MAP classifications from the CMEA versus the true class. This resulted in 149 misclassified points with an ARI $= 0.213$. Over all 8 runs, i.e., combinations of K, J and $\mathsf{stagnation}$ for the VEE, identical results were obtained.

Analysing the WBC data using the `mixture` package with k-means start selected the VEE model with BIC $= -6467.60$ for $G = 2$ components. This resulted in 150 misclassified observations (Table 4.21; ARI $= 0.212$). The WBC data is a difficult data to classify and this is evident from the very low ARI values from both the CMEA and EM. The ARI value from the CMEA is similar to the EM; however, the

Table 4.20: Cross-tabulation of the predicted classifications (1,2) from CMEA versus true class (Benign or Malignant) for the Wisconsin breast cancer data.

	Cluster	
	1	2
Benign	319	38
Malignant	111	101

BIC values are not the same.

Table 4.21: Cross-tabulation of the predicted classifications (1,2) from EM versus true class (Benign or Malignant) for the Wisconsin breast cancer data.

	Cluster	
	1	2
Benign	309	48
Malignant	102	110

4.3.11 Thyroid Data

The next real data considered is the thyroid data reported by Coomans et al. (1983) and are publicly available in the mclust (Fraley and Raftery, 1999) library for R. The data set comprises five laboratory tests performed on a sample of 215 patients to correctly classify a patient's thyroid state as euthyroidism, hypothyroidism or hyperthyroidism.

The CMEA approach was applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and stagnation $\in \{2, 4\}$. The best model selected was VVV with BIC $= -1209.57$ for $G = 3$, stagnation $= 2$, $J = 30$ and $K = 1$. Over all 8 runs — combinations of K, J and stagnation — similar and very good classification performance was achieved

with nine misclassifications (Table 4.22; ARI = 0.863). Again, similar results were obtained for the MEA but are not reported here.

Table 4.22: Cross-tabulation of the predicted classifications (1,2,3) from our CMEA versus true class (Euthyroidism, Hypothyroidism or Hyperthyroidism) for the thyroid data.

	Cluster		
	1	2	3
Euthyroidism	145	3	2
Hypothyroidism	0	35	0
Hyperthyroidism	4	0	26

Analysis of the thyroid data using the `mixture` package with k-means start selected the VVV model with BIC = -1209.44 for $G = 3$ components. This resulted in nine misclassified observations (Table 4.23; ARI = 0.863). These results are similar to the results obtained from the EAs.

Table 4.23: Cross-tabulation of the predicted classifications (1,2,3) from EM versus true class (Euthyroidism, Hypothyroidism or Hyperthyroidism) for the thyroid data.

	Cluster		
	1	2	3
Euthyroidism	145	3	2
Hypothyroidism	0	35	0
Hyperthyroidism	4	0	26

4.3.12 US Crime Data

The US crime data are available from the MASS package in R. The data was collected by criminologists interested in the effect of punishment regimes on crime rates in 47

states in the US. The data consists of 15 features and is classified between Southern and non-Southern states.

Both CMEA and MEA are applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and stagnation $\in \{2, 4\}$. The CMEA selected the VII covariance structure with BIC $= -1929.30$ for $G = 2$ components for stagnation $= 2$, $J = 10$ and $K = 1$. Over all 8 runs — combinations of K, J and stagnation — identical and good classification performance was obtained. A cross-tabulation of the predicted classifications versus true class resulted in an ARI value of 0.603 (Table 4.24). Similar results were obtained for the MEA but are not reported here.

Table 4.24: Cross-tabulation of the predicted classifications (1,2) from CMEA versus true class (Non-Southern State or Southern State) for the US crime data.

	Cluster	
	1	2
Non-Southern State	31	0
Southern State	5	11

Fitting all fourteen GPCMs using the `mixture` package with k-means start, the EM selected the VII model with BIC $= -1929.22$ for $G = 2$ components and this resulted in an ARI value of 0.603 from the cross-tabulation of the predicted classification versus true class (Table 4.25). Again, these results are similar to the results obtained from the EAs.

Table 4.25: Cross-tabulation of the predicted classifications (1,2) from EM versus true class (Non-Southern State or Southern State) for the US crime data.

	Cluster	
	1	2
Non-Southern State	31	0
Southern State	5	11

4.3.13 Cervical Cancer Behavior Risk Data

We considered the Cervical cancer behavior risk (CCBR) data set reported by Sobar et al. (2016) and publicly available in the UCI Machine Learning Repository. The data set contains 19 attributes from 72 women regarding the risk of cervical cancer. Twenty-one of these women have cervical cancer whilst fifty-one of them do not have cervical cancer.

Both CMEA and MEA are applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and stagnation $\in \{2, 4\}$. The CMEA selected the VEI covariance structure with BIC $= -3733.43$ for $G = 2$ components for stagnation $= 2$, $J = 10$, and $K = 1$. Table 4.26 represents the cross-tabulation of the MAP classifications from the CMEA versus the true class. This resulted in nineteen misclassified points with an ARI $= 0.214$. Over all 8 runs, i.e., combinations of K, J and stagnation for the VEI, identical results were obtained. Similar results were obtained for the MEA.

Performing the analysis of the CCRB data using the `mixture` package with k-means start selected the VEI model with BIC $= -3733.39$ for $G = 2$ components. This resulted in nineteen misclassified observations (Table 4.27; ARI $= 0.214$). Again, these results are similar to the results obtained from the EAs.

Table 4.26: Cross-tabulation of the predicted classifications (1,2) from CMEA versus true class (Cervical cancer absent or Cervical cancer present) for the CCBR data.

	Cluster	
	1	2
Cervical cancer absent	34	17
Cervical cancer present	2	19

Table 4.27: Cross-tabulation of the predicted classifications (1,2) from EM versus true class (Cervical cancer absent or Cervical cancer present) for the CCBR data.

	Cluster	
	1	2
Cervical cancer absent	34	17
Cervical cancer present	2	19

4.3.14 Diabetes Data

Reaven and Miller (1979) examined the relationship among blood chemistry measures of glucose tolerance and insulin in 145 non-obese adults. For each subject six variables were measured where patients were classified as subclinical (chemical) diabetics, overt diabetics and normal after further analysis. The data is freely available in the heplots package (Fox et al., 2021) in R.

The CMEA approach was applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and stagnation $\in \{2, 4\}$. The best model selected was VEV with BIC $= -1134.39$ for $G = 3$, stagnation $= 4$, $J = 30$ and $K = 1$. Over all 8 runs — combinations of K, J and stagnation — identical and good classification performance was obtained with twenty-one misclassifications (Table 4.28; ARI $= 0.641$). Again, similar results were obtained for the MEA but are not reported here.

Table 4.28: Cross-tabulation of the predicted classifications (1,2,3) from our CMEA versus true class (Chemical Diabetic, Normal or Overt Diabetic) for the diabetes data.

	Cluster		
	1	2	3
Chemical Diabetic	25	10	1
Normal	5	71	0
Overt Diabetic	5	0	28

Analysing the diabetes data using the `mixture` package with k-means start selected the VEV model with BIC $= -1136.78$ for $G = 3$ components. This resulted in twenty misclassified observations (Table 4.29; ARI $= 0.666$). The ARI value from the EM is slightly higher than the ARI from the CMEA whereas the BIC from the CMEA is slightly larger than the BIC from the EM.

Table 4.29: Cross-tabulation of the predicted classifications (1,2,3) from EM versus true class (Chemical Diabetic, Normal or Overt Diabetic) for the diabetes data.

	Cluster		
	1	2	3
Chemical Diabetic	24	12	0
Normal	2	74	0
Overt Diabetic	6	0	27

4.3.15 Body Data

Finally, we also consider the body data set previously analyzed. Our CMEA is applied to these data with $K \in \{1, 4\}$, $J \in \{10, 30\}$ and $\mathsf{stagnation} \in \{2, 4\}$. The best model, i.e., the model with the highest BIC $= -18816.38$, was VEE model with

$G = 2$, for stagnation $= 2$, $J = 10$ and $K = 1$. Over all 8 runs — combinations of K, J and stagnation — similar and near perfect classification performance was achieved with nine misclassified observations (Table 4.30; ARI = 0.930). Similar results were obtained for the MEA but not reported here.

Table 4.30: Cross-tabulation of the predicted classifications (1,2) from CMEA versus true class (Female or Male) for the body data.

	Cluster	
	1	2
Female	254	6
Male	3	244

Fitting all fourteen GPCMs using `mixture` package with k-means start selected the VEE covariance structure with BIC $= -18815.98$ for $G = 2$ components. This lead to nine misclassified observations (Table 4.31; ARI = 0.930). Both the EAs and the EM selected the same model with the same number of components. Also, the ARIs from the cross-tabulation of both methods are identical.

Table 4.31: Cross-tabulation of the predicted classifications (1,2) from EM versus true class (Female or Male) for the body data.

	Cluster	
	1	2
Female	254	6
Mela	3	244

In Table 4.32, we give a summary of the BIC values for the models selected for each of the data sets as well as the ARI values from the cross-tabulation of the predicted values versus the true classes. We also report on the run time given in

seconds for each of the methods applied to the data sets but should only be taken as a rough guide because the code has yet to be optimized. The general observation is that the time used by the EM is consistently less for all the data sets compared to the CMEA and the MEA. On the other hand, the CMEA is the most expensive among the three approaches. Out of the fifteen real data sets considered here, the ARI values for eleven are the same for all the methods.

4.4 Discussion

In this chapter, an EA is introduced for parameter estimation in the family of mixture models, i.e., the GPCM family. This is the first use of an EA in clustering for the family of GPCMs within the literature. In fact, the closest approach considers only the VVV covariance structure (McNicholas et al., 2020). Each iteration of our EA uses a crossover step followed by a mutation step.

In general, when assessing an approach for clustering, the classification performance is often considered to be the ultimate assessment. Analysis of fifteen famous real data sets in R revealed that the EA approach usually gives identical classification performance to the EM algorithm — i.e., all the ARI values for eleven out of the fifteen real data sets are the same for both the EA and the EM. Across all fifteen real data scenarios, an excellent or good classification performance has been attained except for two data sets. Also, there has not been a BIC difference of more than 2.5 between the EA and the EM except for one data where the BIC difference is more than 15. Note that a BIC difference of more than 10 is considered "a strong

evidence".

It is not surprising that the EA is computationally expensive. The reason is two fold: the EAs involves calculations among parents and children at each iteration of the algorithm and this becomes increasingly time consuming; and, the EA code is written in R and all analysis were carried out in R, however, one of the main drawbacks is the fact that R is a much slower programming language compared to other languages.

By our illustrations using real data sets, we have shown that the EAs, both CMEA and MEA are good alternatives to the EM algorithm for classifying data in cluster analysis. As future work, we are keen on considering alternatives to the R programming language such as C or Python for implementing the EAs to ease the computational burden.

Table 4.32: A summary of the ARI, BIC and the time in seconds for the data sets considered for the CMEA, MEA and EM.

Data	CMEA			MEA			EM		
	ARI	BIC	Time	ARI	BIC	Time	ARI	BIC	Time
Banknote	0.980	−2781.27	0.181	0.980	−2781.27	0.097	0.980	−2781.26	0.001
Coffee	1.000	−1334.22	0.001	1.000	−1334.22	0.000	1.000	−1334.22	0.000
AIS	0.922	−2479.26	0.666	0.922	−2479.26	0.170	0.922	−2479.18	0.014
Female voles	0.908	−1316.72	0.145	0.908	−1316.72	0.115	0.908	−1316.70	0.001
Seeds	0.630	139.17	3.535	0.630	139.17	0.073	0.630	139.21	0.003
Iris	0.922	−789.45	0.813	0.922	−789.45	0.018	0.922	−789.37	0.001
Italian Wine	0.895	−12103.88	0.800	0.895	−12103.88	0.046	0.895	−12103.74	0.001
Crabs	0.756	39.23	11.281	0.756	39.15	3.164	0.738	39.34	0.046
Olive Oil	0.524	−5817.73	17.172	0.524	−5817.73	0.300	0.525	−5817.45	0.017
WBC	0.213	−6452.22	340.231	0.213	−6452.22	19.765	0.212	−6467.60	0.198
Thyroid	0.863	−1209.57	3.973	0.863	−1209.57	0.054	0.863	−1209.44	0.004
US crime	0.603	−1929.30	0.156	0.603	−1929.30	0.007	0.603	−1929.22	0.005
CCBR	0.214	−3733.43	0.047	0.214	−3733.43	0.032	0.214	−3733.43	0.001
Diabetes	0.641	−1134.39	0.987	0.641	−1134.39	0.272	0.666	−1136.78	0.012
Body	0.930	−18816.38	461.146	0.930	−18816.38	2.155	0.930	−18815.98	0.051

Chapter 5

An Evolutionary Algorithm for Latent Gaussian Mixture Models

5.1 Introduction

Mixture models must be adapted to handle high-dimensional data, given that vast amounts of data can be collected and stored with ease using modern technology. Some methods for clustering such data are based on the MFA model. McNicholas and Murphy (2008) developed a class of eight different PGMMs by extending the MFA model (see Section 2.3). However, maximum likelihood estimates for the parameters in the family of latent Gaussian mixture models, known as PGMMs, are typically found using an AECM algorithm. In this chapter, rather than using an AECM algorithm, we develop an EA to estimate these parameters and classify the data. The EA developed in this chapter is similar to the one developed in Chapter 3.

Again, it is of interest to compare the performance of our proposed algorithm to the AECM algorithm through illustrations using both real and simulated data sets.

5.2 Parameter Estimation for PGMMs

The maximum likelihood estimates of the parameters for members of the PGMM family are carried out using the AECM algorithm. This algorithm is a variant of the EM algorithm and uses different specifications of missing data at each stage.

In the first stage of the AECM algorithm, the unobserved group membership labels are $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$ when estimating π_g and $\boldsymbol{\mu}_g$. Thus, the complete-data log-likelihood is expressed as

$$l_1 = \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\log[\pi_g \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g)]. \tag{5.1}$$

The expected complete-data log-likelihood is of the form

$$\begin{aligned}\mathcal{Q}_1(\boldsymbol{\mu}_g, \pi_g) = &\sum_{g=1}^{G} n_g \log\pi_g - \frac{np}{2}\log 2\pi - \sum_{g=1}^{G}\frac{n_g}{2}\log|\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g| \\ &- \sum_{g=1}^{G}\frac{n_g}{2}\mathrm{tr}\left\{\mathbf{S}_g(\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g)^{-1}\right\},\end{aligned} \tag{5.2}$$

where $n_g = \sum_{i=1}^{n}\hat{z}_{ig}$ and $\mathbf{S}_g = (1/n_g)\sum_{i=1}^{n}\hat{z}_{ig}(\mathbf{x}_i - \boldsymbol{\mu}_g)(\mathbf{x}_i - \boldsymbol{\mu}_g)'$. Maximizing $\mathcal{Q}_1$ with respect to π_g and $\boldsymbol{\mu}_g$ yields $\hat{\pi}_g$ and $\hat{\boldsymbol{\mu}}_g$, respectively.

At the second stage of the AECM algorithm, the group membership labels $\mathbf{z}$ and the unobserved latent factors $\mathbf{u}$ are taken to be the missing data when estimating

$\boldsymbol{\Lambda}_g$ and $\boldsymbol{\Psi}_g$. The complete-data log-likelihood is given as

$$\begin{aligned} l_2 = C + \sum_{g=1}^{G} \Bigg[& -\frac{n_g}{2}\log|\boldsymbol{\Psi}_g| - \frac{n_g}{2}\text{tr}\{\boldsymbol{\Psi}_g^{-1}\mathbf{S}_g\} \\ & + \sum_{i=1}^{n} z_{ig}(\mathbf{x}_i - \boldsymbol{\mu}_g)'\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Lambda}_g\mathbf{u}_i - \frac{1}{2}\text{tr}\left\{\boldsymbol{\Lambda}_g'\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Lambda}_g\sum_{i=1}^{n} z_{ig}\mathbf{u}_i\mathbf{u}_i'\right\}\Bigg], \end{aligned} \tag{5.3}$$

where C is a constant with respect to $\boldsymbol{\Lambda}_g$ and $\boldsymbol{\Psi}_g$. The expected value of the complete-data log-likelihood evaluated with $\boldsymbol{\mu}_g = \hat{\boldsymbol{\mu}}_g$ and $\pi_g = \hat{\pi}_g$ can be written

$$\begin{aligned} \mathcal{Q}_2(\boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g) = C + \sum_{g=1}^{G} n_g \Bigg[& \frac{1}{2}\log|\boldsymbol{\Psi}_g^{-1}| - \frac{1}{2}\text{tr}\{\boldsymbol{\Psi}_g^{-1}\mathbf{S}_g\} + \text{tr}\{\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Lambda}_g\hat{\boldsymbol{\beta}}_g\mathbf{S}_g\} \\ & - \frac{1}{2}\text{tr}\{\boldsymbol{\Lambda}_g'\boldsymbol{\Psi}_g^{-1}\boldsymbol{\Lambda}_g\boldsymbol{\Theta}_g\}\Bigg], \end{aligned} \tag{5.4}$$

where C is a constant, $\hat{\boldsymbol{\beta}}_g = \hat{\boldsymbol{\Lambda}}_g'(\hat{\boldsymbol{\Lambda}}_g\hat{\boldsymbol{\Lambda}}_g' + \hat{\boldsymbol{\Psi}}_g)^{-1}$ and $\boldsymbol{\Theta}_g = \mathbf{I}_q - \hat{\boldsymbol{\beta}}_g\hat{\boldsymbol{\Lambda}}_g + \hat{\boldsymbol{\beta}}_g\mathbf{S}_g\hat{\boldsymbol{\beta}}_g'$.

The estimates of $\hat{z}_{ig}$, $\hat{\boldsymbol{\mu}}_g$ and $\hat{\pi}_g$ which are calculated in the first stage of the AECM algorithm are the same for each member of the PGMM family, thus:

$$\hat{z}_{ig} = \frac{\hat{\pi}_g\phi(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Lambda}}_g, \hat{\boldsymbol{\Psi}}_g)}{\sum_{h=1}^{G}\hat{\pi}_h\phi(\mathbf{x}_i \mid \hat{\boldsymbol{\mu}}_h, \hat{\boldsymbol{\Lambda}}_h, \hat{\boldsymbol{\Psi}}_h)}, \qquad \hat{\boldsymbol{\mu}}_g = \frac{\sum_{i=1}^{n}\hat{z}_{ig}\mathbf{x}_i}{\sum_{i=1}^{n}\hat{z}_{ig}} \quad \text{and} \quad \hat{\pi}_g = \frac{n_g}{n}.$$

The resulting estimates, when we impose constraints (Table 2.2) on $\boldsymbol{\Lambda}_g$ and $\boldsymbol{\Psi}_g$ matrices, can be easily derived from the expression $\mathcal{Q}_2(\boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g)$. Details of the parameter estimates for each member of the PGMM family are given by McNicholas and Murphy (2008). McLachlan and Peel (2000b) give extensive details of fitting the

AECM algorithm in the case where no constraints are imposed. The AECM algorithm updates the parameters iteratively until it convergences. The estimates of the *a posteriori* probability of group membership for each observation is the subsequent $\hat{z}_{ig}$ values and can be used to cluster observations into groups.

5.3 An Evolutionary Algorithm for Latent Gaussian Mixture Models

5.3.1 Model and Fitness Function

A mixture of factor analyzers model was selected as the basic model. For the purpose of clustering, all component memberships are unknown or treated as such.

The EA developed herein is built to optimize one fitness function, and the fitness function is the (observed) log-likelihood

$$l(\boldsymbol{\vartheta}) = \sum_{i=1}^{n} \log \left\{ \sum_{g=1}^{G} \pi_g \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g) \right\}, \tag{5.5}$$

where $\boldsymbol{\vartheta}$ denotes the model parameters. We use $\tilde{z}_{ig}$ to denote the estimate of z_{ig} used in the EA to avoid confusion with the expected values $\hat{z}_{ig}$ used in the AECM algorithm.

5.3.2 Evolutionary Algorithm

For this EA various single parents are considered and each is cloned multiple times with the cloned offspring reproducing as described here. Each step of the algorithm requires calculations among either children or parents, hence two further sets of indices are introduced: $j = 1, ..., J$ indexes children (clones) and $k = 1, ..., K$ indices parents. The fitness function for the jth child is just the log-likelihood (5.5) evaluated at the estimates $\tilde{\pi}_g$, $\tilde{\boldsymbol{\mu}}_g$, $\tilde{\boldsymbol{\Lambda}}_g$ and $\tilde{\boldsymbol{\Psi}}_g$.

Based on $\tilde{z}_{igk}$, the updates π_{gk}, $\boldsymbol{\mu}_{gk}$, $\boldsymbol{\Lambda}_{gk}$ and $\boldsymbol{\Psi}_{gk}$, for $g = 1, ..., G$ are computed as for the AECM algorithm (Section 5.2) for the family of the PGMM models. Thus, the estimated component membership of $\mathbf{x}_i$ for parent k given these current parameter estimates is

$$\hat{z}_{igk} = \frac{\pi_{gk}\phi(\mathbf{x}_i \mid \boldsymbol{\mu}_{gk}, \boldsymbol{\Lambda}_{gk}, \boldsymbol{\Psi}_{gk})}{\sum_{h=1}^{G} \pi_{hk}\phi(\mathbf{x}_i \mid \boldsymbol{\mu}_{hk}, \boldsymbol{\Lambda}_{hk}, \boldsymbol{\Psi}_{hk})}, \tag{5.6}$$

for $i = 1, ..., n$, $g = 1, ..., G$ and $k = 1, ..., K$. For each child, i.e., each clone of a single parent, the $\tilde{z}_{igk}$ are mutated by randomly sampling each observation's cluster membership according to the probabilities in (5.6).

At each iteration, K parents are stored, and during the reproduction phase, we sample J new matrices of $\tilde{z}_{igj}$ according to the $\tilde{z}_{igk}$ corresponding to each of the K parents. The top K fittest solutions at each generation are cloned (i.e., produce children) to the next generation which ensures that the maximum fitness from one generation to another is non-decreasing. That is, after each instance of mutation has been carried out on each cloned child, all the children and the original K single parents are put into one list in decreasing order of fitness, where the top K are

selected to become the new generation of single parents. The following pseudo-code (Algorithm 3) details the procedure followed in our EA.

Pseudo-Code

The following pseudo-code (Algorithm 3) details the procedure followed in our EA.

Algorithm 3 EA for Latent Gaussian Mixture Models

initialize $\tilde{z}_{igk}$ matrices using k-means and k-medoids

initialize: k sets of parameters based on these $\tilde{z}_{igk}$

`stag` $= 0$

while `stag` $<$ `stagnation` **do**

mutate: calculate $\hat{z}_{igk}$; sample J children (clones) $\tilde{z}_{igk}$ accordingly

update: all J sets of parameters

fitness: calculate log-likelihood for each of J children (clones)

survival: sort J children (clones) and K parents in descending fitness order, select top K as new parents

if the log-likelihoods of the top K solutions the same as previous cycle **then**

`stag++`

else

`stag` $= 0$

end if

end while

return $\tilde{z}_{igk}$ corresponding to the highest log-likelihood

5.3.3 Computational Aspect

Initialization

To initialize z_{igk}, both k-means and k-medoids clustering algorithms were run and the resulting $\tilde{z}_{igk}$ were then taken as the starting group membership labels for all the models— note that we fixed the number of parents $K = 2$ for this EA. The initial values for the elements of $\mathbf{\Lambda}_{gk}$ and $\mathbf{\Psi}_{gk}$ were generated from the eigen-decomposition of $\mathbf{S}_{gk}$ using Householder reduction and the QL method (details are given in Press et al., 2007).

Stopping Criterion

Our EA is stopped once stagnation occurs, i.e., we terminate the EA when the log-likelihoods from top K solutions fail to increase over a number of consecutive generations.

Model Selection and Performance

We propose using the BIC to choose the model type (CCC, CCU,...,UUU), the number of latent factors q and the number of components G, for the PGMMs as well as the number of J and the stagnation values. The effectiveness of the PGMMs in detecting group structures in data is illustrated by matching the predicted component membership labels of the observations with the true labels. Also, clustering performance of various methods is quantified using the ARI.

5.4 Illustrations

The purpose of these illustrations is to assess the performance of our EA by comparing it to the AECM algorithm in Section 5.2 using the function pgmmEM via the R package pgmm (McNicholas et al., 2018) and with model-based clustering using the mclust (model-based clustering) software (Fraley and Raftery, 1999, 2002) for R. For the data sets considered, we assumed that there is no prior knowledge of the labels or the number of components — i.e., they are treated as a genuine clustering example. However, in each case the true labels are known and thus it is possible to compare the predicted classifications. This comparison is carried out using the ARI. The data sets are scaled prior to analysis using the scale function in R.

5.4.1 Italian Wine Data

The Italian wine data set previously analyzed is considered here. We also analyzed the more common thirteen variable subset publicly available in the UCI Machine Learning data repository and as part of the gclus library (Hurley, 2004) for R.

The EA approach introduced herein is demonstrated on the analysis of these Italian wine data sets with stagnation $\in \{2, 3, 4, 5\}$ and $J \in \{10, 20, 30, 40, 50\}$. All eight PGMMs were fitted to the data sets for $G = 1, 2, 3$ and $q = 1, 2, 3$.

Twenty-Seven variables

The BIC for each model was computed and the model with the highest BIC value (-11575.78) was selected; this was the CUU model with $G = 3$ and $q = 3$ for stagnation $= 5$ and $J = 30$. For $J = 30$ and for all stagnation values, excellent classification performance was attained with 2 misclassifications (Table 5.1; ARI=0.967). However, for all stagnation values and $J \in \{10, 20, 40, 50\}$, similar and near perfect classification was achieved, with just one misclassification (Table 5.2; ARI = 0.983).

Table 5.1: Cross-tabulation of the predicted classifications (1,2,3) from our EAs versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with twenty-seven variables. The best model is CUU with $G = 3$ and $q = 3$ for stagnation $\in \{2,3,4,5\}$ and $J = 30$.

	Cluster		
	1	2	3
Barolo	59	0	0
Grignolino	0	69	2
Barbera	0	0	48

Table 5.2: Cross-tabulation of the predicted classifications (1,2,3) from our EAs versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with twenty-seven variables. The best model is CUU with $G = 3$ and $q = 3$ for stagnation $\in \{2,3,4,5\}$ and $J \in \{10,20,40,50\}$

	Cluster		
	1	2	3
Barolo	59	0	0
Grignolino	0	70	1
Barbera	0	0	48

The parsimonious Gaussian mixture models is applied to the wine data using

the pgmm software for R. Considering a random start, the best model selected was CUU with BIC = −11577.09 with $G = 3$ and $q = 3$. This resulted in 4 misclassified observations (Table 5.3). Using k-means starts, the best model based on the largest BIC (−11753.09) was CCU with $G = 2$ and $q = 3$ and the resulting classification performance is presented in Table 5.4. The classification performance of pgmm with random and k-means starts resulted in ARI values 0.931 and 0.439 respectively (Table 5.6) which is inferior to the EAs.

Table 5.3: Cross-tabulation of the predicted classifications (1,2,3) from pgmm with random starts versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with twenty-seven variables.

	Cluster		
	1	2	3
Barolo	59	0	0
Grignolino	2	62	2
Barbera	0	0	48

Table 5.4: Cross-tabulation of the predicted classifications (1,2,3) from pgmm with k-means starts versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with twenty-seven variables.

	Cluster	
	1	2
Barolo	59	0
Grignolino	63	8
Barbera	0	48

Model-based clustering was also performed on the wine data using the mclust software for R. The model with the highest BIC value (−11949.50) was the three

component mixture with the VVE (variable shape and different sized ellipses but with equal orientation) covariance structure. The classification performance is shown in Table 5.5 with ARI = 0.931. It is clear that, the performance of mclust on this data is inferior to our EA.

Table 5.5: Cross-tabulation of the predicted classifications (1,2,3) from the best model found using mclust versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with twenty-seven variables.

	Cluster		
	1	2	3
Barolo	58	1	0
Grignolino	1	70	0
Barbera	0	2	46

Table 5.6: Rand index, ARI and BIC for the models that were applied to the Italian wine data with twenty-seven variables.

Model	Rand index	ARI	BIC
EA	0.985	0.967	−11575.78
PGMM (random starts)	0.969	0.931	−11577.09
PGMM (k-means starts)	0.708	0.439	−11753.09
MCLUST	0.969	0.931	−11949.50

Thirteen variables

The EA developed is applied to the subset of the wine data with thirteen variables, and the best model i.e., the model with the largest BIC (-5318.89) was selected to be CUU with $G = 3$ and $q = 3$ for stagnation $= 5$ and $J \in \{40, 50\}$. Over all 20 runs, i.e., combinations of stagnation and J, identical classification performance was obtained. A cross-tabulation of the predicted group membership labels from the EAs against the true wine type is given in Table 5.7, resulting in 4 misclassifications with ARI = 0.933.

Table 5.7: Cross-tabulation of the predicted classifications (1,2,3) from our EAs versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with thirteen variables.

	Cluster		
	1	2	3
Barolo	58	1	0
Grignolino	0	68	3
Barbera	0	0	48

All eight PGMMs were fitted using the pgmm package in R with a random start and the best model selected was CUU with BIC $= -5318.04$ for $G = 3$ and $q = 2$ resulting in 4 misclassified observations as seen in Table 5.8. Using k-means starting values, again, the CUU model was selected with BIC $= -5318.10$ for $G = 3$ and $q = 2$ which results in 5 missclassifications (Table 5.9). Whilst the result with a random start is similar to our EAs in terms of the ARI (Table 5.11), the result with k-means starting values is inferior.

Analysis of the subset of the wine data using the mclust software yielded 3 groups

with a VVE covariance structure and a BIC value of -5403.77. This model gives slightly inferior classification performance (Table 5.10; ARI = 0.933) compared to our EA.

Table 5.8: Cross-tabulation of the predicted classifications (1,2,3) from pgmm with random starts versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with thirteen variables.

	Cluster		
	1	2	3
Barolo	58	1	0
Grignolino	0	68	3
Barbera	0	0	48

Table 5.9: Cross-tabulation of the predicted classifications (1,2,3) from pgmm with k-means starts versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with thirteen variables.

	Cluster		
	1	2	3
Barolo	58	1	0
Grignolino	0	67	4
Barbera	0	0	48

Table 5.10: Cross-tabulation of the predicted classifications (1,2,3) from the best model found using mclust versus true class (Barolo, Grignolino or Barbera) for the Italian wine data with thirteen variables.

	Cluster		
	1	2	3
Barolo	56	3	0
Grignolino	0	70	1
Barbera	0	0	48

Table 5.11: Rand index, ARI and BIC for the models that were applied to the Italian wine data with thirteen variables.

Model	Rand index	ARI	BIC
EA	0.970	0.933	−5318.89
PGMM (random starts)	0.970	0.933	−5318.04
PGMM (k-means starts)	0.963	0.917	−5318.10
MCLUST	0.969	0.930	−5403.77

5.4.2 Body Data

We consider the body data set that was previously analyzed. Applying the EA approach introduced herein with stagnation $\in \{2, 3, 4, 5\}$ and $J \in \{10, 20, 30, 40, 50\}$. All eight PGMMs are fitted with the number of components fixed at $G = 2$ and $q = 1, ..., 8$. The model with the highest BIC $= -18614.10$ was the CCU model with $G = 2$ and $q = 8$ for stagnation $= 5$ and $J = 10$. The associated classification performance is very good (ARI = 0.930; Table 5.12).

Table 5.12: Cross-tabulation of the predicted classifications (1,2) from our EAs versus true class (female or male) for the body data.

	Cluster	
	1	2
Female	256	4
Male	5	242

Fitting all eight PGMMs to the body data for $G = 2$ and $q = 1, ..., 8$ using the pgmm software with random starting values selected the CCU model for $G = 2$ and $q = 8$ with BIC $= -18622.89$. The associated classification performance is very good (ARI = 0.938; Tables 5.13 and 5.16). The k-means starts selected the CCU model

for $G = 2$ and $q = 7$ with BIC $= -18642.46$ as the best model and the resulting classification performance yielded ARI $= 0.930$ (Table 5.14). The ARI value from our EA is slightly inferior to the ARI from pgmm with random start but similar to the pgmm with k-means start (Table 5.16).

Table 5.13: Cross-tabulation of the predicted classifications (1,2) from pgmm with random starts versus true class (female or male) for the body data.

	Cluster	
	1	2
Female	257	3
Male	5	242

Table 5.14: Cross-tabulation of the predicted classifications (1,2) from pgmm with k-means starts versus true class (female or male) for the body data.

	Cluster	
	1	2
Female	256	4
Male	5	242

The mclust software selected an ellipsoidal, equal shape and orientation (VEE) model with 4 components and a BIC $= -18700.06$. From Table 5.15, the associated classification resulted in an ARI value of 0.616 which is inferior compared to the ARI from our EAs and pgmm with both k-means start and random start.

Table 5.15: Cross-tabulation of the predicted classifications (1,2,3,4) from the best model found using mclust versus true class (female or male) for the body data.

	Cluster			
	1	2	3	4
Female	174	70	1	2
Male	0	5	208	47

Table 5.16: Rand index, ARI and BIC for the models that were applied to the body data.

Model	Rand index	ARI	BIC
EA	0.965	0.930	−18614.10
PGMM (random starts)	0.969	0.938	−18622.89
PGMM (k-means starts)	0.965	0.930	−18642.46
MCLUST	0.808	0.616	−18700.06

5.4.3 Coffee Data

The coffee data set previously analyzed is considered here. The EA approach introduced herein is applied to these data, with stagnation $\in \{2, 3, 4, 5\}$ and $J \in \{10, 20, 30, 40, 50\}$. All eight PGMMs are fitted with $G = 1, 2, 3$ and $q = 1, 2, 3$. The model with the highest BIC (−1306.54) was the CCU model with $G = 2$ and $q = 1$ for stagnation $= 5$ regardless of the number of J. Thus over all 20 runs, identical and perfect classification performance was obtained with no misclassifications (Table 5.17; ARI $= 1.00$).

Analysis of the data using the pgmm software with random starts selected the best model to be CCU with BIC $= -1326.02$ for $G = 3$ and $q = 1$; whilst a k-means starts selected the CCU model as well with BIC $= -1306.13$ for $G = 2$ and $q = 1$. A cross-tabulation of the class from the best pgmm model with random and k-means

Table 5.17: Cross-tabulation of the predicted classifications (1,2) from our EAs versus true class (Arabica or Robusta) for the Coffee data.

	Cluster	
	1	2
Arabica	36	0
Robusta	0	7

starts are presented in Tables 5.18 and 5.19 respectively. The k-means starts result is inferior to our EAs in terms of the ARI values (Table 5.21; ARI = 0.383).

The mclust software yielded three groups with VEI covariance structure with the highest BIC value -1297.94. The VEI covariance structure signifies clusters that are shaped equally, with varying volume and aligned with the axis. Table 5.20 shows the classification from this analysis with ARI = 0.383. It is clear that the mclust did poorly compared to the EAs in separating coffee into the different varieties.

Table 5.18: Cross-tabulation of the predicted classifications (1,2,3) from pgmm with random starts versus true class (Arabica or Robusta) for the Coffee data.

	Cluster		
	1	2	3
Arabica	22	14	0
Robusta	0	0	7

Table 5.19: Cross-tabulation of the predicted classifications (1,2) from pgmm with k-means starts versus true class (Arabica or Robusta) for the Coffee data.

	Cluster	
	1	2
Arabica	36	0
Robusta	0	7

Table 5.20: Cross-tabulation of the predicted classifications (1,2,3) from the best model found using mclust versus true class (Arabica or Robusta) for the Coffee data.

	Cluster		
	1	2	3
Arabica	22	14	0
Robusta	0	0	7

Table 5.21: Rand index, ARI and BIC for the models that were applied to the Coffee data.

Model	Rand index	ARI	BIC
EA	1.000	1.000	−1306.54
PGMM (random starts)	0.659	0.383	−1326.02
PGMM (k-means starts)	1.000	1.000	−1306.13
MCLUST	0.659	0.383	−1297.94

5.4.4 Australian Open Men Data

The Australian Open Men's data are publicly available from the UCI Machine Learning Repository. The data contains the match statistics of men at the Australian open tennis tournament. There are 98 matches in total of which 34 quantitative match statistics were recorded and the results of each match is referenced on whether player 1 wins or loses.

Again, the EA developed is applied to these data with stagnation $\in \{2,3,4,5\}$ and $J \in \{10, 20, 30, 40, 50\}$. Fitting all eight PGMMs with $G = 2$ and $q = 1, 2, 3$; the UUU model with $G = 2$ and $q = 1$ was selected to be the best model with identical BIC value of -6702.66 over all 20 runs of the combinations of stagnation and J. Table 5.22 presents the cross tabulation of the MAP classifications from the EA versus the true results of player 1 in the tournament — near perfect classification performance was achieved with just 1 misclassification (ARI = 0.959; Table 5.25).

Table 5.22: Cross-tabulation of the predicted classifications (1,2) from our EAs versus true class (Player 1 loses or Player 1 wins) for the Australian Open Men data.

	Cluster	
	1	2
Player 1 loses	48	0
Player 1 wins	1	49

Contrarily, analysis of these data using the pgmm software with random starts selected the CUU model with BIC = -4747.64 for $G = 2$ and $q = 3$ resulting in 2 misclassified players (Table 5.23; ARI = 0.919). However, k-means starts selected the UUU model with BIC = -7718.26 for $G = 2$ and $q = 1$, resulting in a negative ARI value (Table 5.24; ARI = -0.010) which is not surprising from the cross tabulation of the predicted versus the true classes — the negative ARI value can be interpreted as classifications that are worse than would be expected under random classifications.

Analysing the data using model-based clustering, mclust, selected the XXX — i.e., ellipsoidal multivariate normal — model with 1 component with BIC = 5831.55 and ARI = 0 (Table 5.25) — thus the mclust performed poorly in classifying the players in the tournament.

Table 5.23: Cross-tabulation of the predicted classifications (1,2) from pgmm with random starts versus true class (Player 1 loses or Player 1 wins) for the Australian Open Men data.

	Cluster	
	1	2
Player 1 loses	50	0
Player 1 wins	2	46

Table 5.24: Cross-tabulation of the predicted classifications (1,2) from pgmm with k-means starts versus true class (Player 1 loses or Player 1 wins) for the Australian Open Men data.

	Cluster	
	1	2
Player 1 loses	23	25
Player 1 wins	23	27

Table 5.25: Rand index, ARI and BIC for the models that were applied to the Australian Open Men data.

Model	Rand index	ARI	BIC
EA	0.980	0.959	−6702.66
PGMM (random starts)	0.960	0.919	−4747.64
PGMM (k-means starts)	0.495	−0.010	−7718.26
MCLUST	0.495	0.000	5831.55

5.4.5 US Crime Data

The US crime data set previously analyzed is considered here. The EA approach is applied to these data with stagnation $\in \{2, 3, 4, 5\}$ and $J \in \{10, 20, 30, 40, 50\}$. All eight PGMMs are fitted with $G = 1, 2, 3$ and $q = 1, 2, 3$. The model with the highest BIC (-1625.24) was the CUU model with $G = 2$ and $q = 2$ for stagnation $= 5$ regardless of the number of J. A cross tabulation of the MAP classifications from the EA versus the true states is given in Table 5.26; the model correctly classifies all but 3 states resulting in an ARI value of 0.754.

Table 5.26: Cross-tabulation of the predicted classifications (1,2) from our EAs versus true class (Non-Southern State or Southern State) for the US crime data.

	Cluster	
	1	2
Non-Southern State	14	2
Southern State	1	30

All eight PGMMs are fitted to the US crime data for $G = 1, 2, 3$ and $q = 1, 2, 3$ using the pgmm software. Using random starting values, the best model selected is the CUU with BIC $= -1610.34$ for $G = 3$ and $q = 3$. Whilst k-means starts selected the CUU model with BIC $= -1619.55$ for $G = 2$ and $q = 2$. A cross tabulation of the MAP classifications from the best pgmm model with random and k-means starting values are presented in Tables 5.27 and 5.28 respectively. The ARI values from both random and k-means starts are inferior to our EA (Table 5.29).

The mclust software resulted in an ellipsoidal multivariate normal (XXX) model

with 1 component and a BIC value of −1708.02. From Table 5.29, the mclust performed poorly in separating the states in the US crime data.

Table 5.27: Cross-tabulation of the predicted classifications (1,2,3) from pgmm with random starts versus true class (Non-Southern State or Southern State) for the US crime data.

	Cluster		
	1	2	3
Non-Southern State	7	23	1
Southern State	1	2	13

Table 5.28: Cross-tabulation of the predicted classifications (1,2) from pgmm with k-means starts versus true class (Non-Southern State or Southern State) for the US crime data.

	Cluster	
	1	2
Non-Southern State	13	3
Southern State	1	30

Table 5.29: Rand index, ARI and BIC for the models that were applied to the US crime data.

Model	Rand index	ARI	BIC
EA	0.878	0.754	−1625.24
PGMM (random starts)	0.724	0.459	−1610.34
PGMM (k-means starts)	0.841	0.678	−1619.55
MCLUST	0.541	0.000	−1708.02

5.4.6 Australian Institute of Sports Data

The AIS data previously analyzed is considered here. The EA approach developed is applied to these data with stagnation $\in \{2, 3, 4, 5\}$, $J \in \{10, 20, 30, 40, 50\}$, $G = 2$, and the number of factors $q = 1, 2, 3, 4, 5$. Fitting all eight PGMMs, the best model for the range of parameters specified is a UUU model with $G = 2$ components and $q = 4$ factors. For all combinations of stagnation and J, we obtained a similar BIC value of -2788.72 for this model. Thus, over all 20 runs, identical and a very good classification performance was obtained with 4 missclasified athletes resulting in an ARI value = 0.922 (Tables 5.30 and 5.34).

Table 5.30: Cross-tabulation of the predicted classifications (1,2) from our EAs versus true class (male or female) for the AIS data.

	Cluster	
	1	2
Male	99	3
Female	1	99

Fitting all eight PGMMs using the pgmm package in R with random starting values, the best model (BIC = -2465.92) for the range of factors and components is the UCU model with $G = 2$ and $q = 4$ resulting in a poor ARI value = 0.304 (Table 5.31). On the other hand, k-means starting values selected the UUU model (BIC = -2394.34) with $G = 2$ and $q = 4$ and an ARI value = 0.811 (Table 5.32). Whilst the results from the k-means starting values is better than the random starting values from the pgmm, both are inferior to the ARI from the EA approach.

Analysis of this data using mclust software yielded 5 groups with EVE covariance

structure and a BIC value of −2392.88. This model gives a slightly better classification performance (Tables 5.33 and 5.34; ARI = 0.477) compared to pgmm with random starting values and an inferior ARI value compared to our EA approach.

Table 5.31: Cross-tabulation of the predicted classifications (1,2) from pgmm with random starts versus true class (male or female) for the AIS data.

	Cluster	
	1	2
Male	86	16
Female	29	71

Table 5.32: Cross-tabulation of the predicted classifications (1,2) from pgmm with k-means starts versus true class (male or female) for the AIS data.

	Cluster	
	1	2
Male	92	10
Female	0	100

Table 5.33: Cross-tabulation of the predicted classifications (1,2) from the best model found using mclust versus true class (male or female) for the AIS data.

	Cluster				
	1	2	3	4	5
Male	1	9	16	3	73
Female	65	5	9	21	0

Table 5.34: Rand index, ARI and BIC for the models that were applied to the AIS data.

Model	Rand index	ARI	BIC
EA	0.961	0.922	−2788.72
PGMM (random starts)	0.652	0.304	−2465.92
PGMM (k-means starts)	0.905	0.811	−2394.38
MCLUST	0.739	0.477	−2392.88

5.4.7 Simulated Data

We demonstrated our approach on a simulated data set. The data is generated via the genRandomClust function from the R package clusterGeneration (Qiu and Joe, 2006). The data comprises $p = 20$ variables, $n = 332$ observations and substantially overlapping clusters; the remaining settings were left at default. This is a very difficult clustering problem and one should not expect perfect classification results. Each algorithm was initialized at random 30 times on this data for all the approaches considered. The EA consistently selected the CUU model for $G = 3$ and $q = 1$ with stagnation $= 5$ and $J = 40$. Using k-means starts and random starts for pgmm, both selected the CUU model for $G = 3$ and $q = 1$. The mclust selected a three component EVI model. These results are summarized in Table 5.35 with EA having the highest average ARI value. Overall, the EA resulted in a better classification performance compared to the pgmm with both random and k-means starting values and model-based clustering using mclust.

Table 5.35: Average rand index, ARI and BIC for the models that were applied to the simulated data.

Model	Rand index	ARI	BIC
EA	0.983	0.962	−18452.30
PGMM (random starts)	0.960	0.912	−18493.67
PGMM (k-means starts)	0.956	0.903	−18493.74
MCLUST	0.969	0.932	−18435.74

5.5 Discussion

An EA has been developed for latent Gaussian mixture models known as PGMMs. Each iteration of our EA uses a mutation step; no comparable approach in the literature has been taken for the family of PGMMs. The closest approach uses mutation for Gaussian mixture models (Andrews and McNicholas, 2013). The clustering philosophy associated with our approach is that of "hard" clustering, i.e., the estimated group membership labels are restricted to values $\tilde{z}_{ig} \in \{0,1\}$ as compared to the soft labels $\hat{z}_{ig} \in [0,1]$ used in the AECM algorithm.

We used the BIC in selecting the best model for each method in this analysis. In five of the data sets where the EA and pgmm with k-means starting values selected the same models, the EA consistently gave better BIC values except for one data set. However, comparing the BIC across the models selected by each method is not quite as meaningful since the methods select different models for most of the data sets hence the classification performance of each model is compared adopting the ARI values.

The application of the EAs for the family of PGMMs to the data sets considered shows that the results give excellent clustering performance compared to the other

approaches employed in our analysis. It can be seen that, fitting the parsimonious Gaussian mixture models using pgmm software is heavily dependent on starting values since both k-means and random starting values produced non-identical results in all eight data sets used in this work. However, this is not the case with the EA approach. The clusters found using the EA consistently showed greater ability to capture the group structure of the data than the other techniques, i.e., the EA gives superior clustering performance to pgmm and mclust.

There are several possible extensions of this work: as future work, it will be of interest to introduce a crossover step followed by a mutation step in the EA proposed herein (e.g., McNicholas et al., 2020). Also, a similar EA will be developed for situations where clusters may be non-Gaussian using non-Gaussian mixture models (e.g., Browne and McNicholas, 2015; Franczak et al., 2015).

Chapter 6

Conclusions

6.1 Summary

Mixture model-based clustering continues to grow in prominence in the literature, and hence different techniques for parameter estimation in clustering problems need to be explored. The work developed in this thesis has focused on the development and implementation of evolutionary algorithms in the field of model-based clustering.

In Chapter 3, an EA approach was developed for Gaussian mixture model-based clustering with incomplete data, i.e., when data is missing at random. This is the first use of an EA for clustering with missing data. Our EA utilizes an evolutionary operator known as mutation at each iteration. The EA which iterates until stagnation gives an approach that can be considered an alternative to the EM algorithm. Our proposed approach performed favourably or equivalently on both real and simulated data sets when compared to the EM algorithm.

In Chapter 4, we developed an EA called CMEA for parameter estimation in the family of Gaussian parsimonious clustering models. The approach uses crossover followed by a mutation step at each iteration. The EA uses hard classifications and could be viewed as an extension of k-means, and as an alternative to the EM algorithm. Several data analyses were carried out to illustrate our EA, and in most cases it was found to have the same performance as the EM algorithm. The relative performance of our EA and the EM algorithm is quite remarkable and shows that soft clustering is not necessarily preferable to hard clustering.

Finally, an EA was developed for latent Gaussian mixture models known as PGMMs in Chapter 5. Each iteration of the EA consists of a mutation step and can be considered as an alternative to the AECM algorithm. This approach was illustrated on several real data sets and a simulated data set. The EA gave superior performance in five cases, and identical performance in three cases when compared to the current state-of-the-art, i.e., the AECM algorithm.

6.2 Further Work

6.2.1 Non-Gaussian Distributions

The approach developed herein is based on the Gaussian mixture model, nevertheless it is of interest to depart from Gaussianity in future work. For example, this work could be extended to situations where clusters may be non-Gaussian using non-Gaussian mixture models. The most natural departure from Gaussian mixture models is the mixture of multivariate t-distributions. In general, the evolutionary

computation approach for parameter estimation will be applied to the members of the *t*EIGEN family introduced by Andrews and McNicholas (2012). The method could also be extended to the multivariate skew-t mixture models (Vrbik and McNicholas, 2012; Murray et al., 2014).

6.2.2 Missing Not At Random (MNAR)

In Chapter 3, we developed an evolutionary algorithm to cluster data with missing values under the MAR assumption, which are often referred to as ignorable missingness mechanism because the parameters that govern the missingness are separable from the parameters that govern the data. Although the MAR assumption is often reasonable, there are situations where this assumption is not achievable. Hence, it becomes necessary to model the missingness mechanism that may contain information about the parameters of the complete-data population. Therefore, future work focusing on the MNAR missing data mechanism would be beneficial.

6.2.3 Improvement to the Computational Efficiency

We have demonstrated in this thesis that evolutionary computations are effective in parameter estimation and clustering. However, there are computational challenges that must be addressed in the implementation of the evolutionary algorithms. Notably, during each generation of the algorithm, calculations among both clones (children) and parents are required which incurs additional computational overhead. This problem can be at least somewhat remedied by code optimization. R software has been used to write the code for implementing the proposed algorithms in this

thesis. However, developing analogous C code to form the basis for an R package is a possible solution to help improve computational efficiency. There is also work to be done on the alternative to the R programming language such as Python for implementing these algorithms.

Bibliography

Aitken, A. C. (1926). A series formula for the roots of algebraic and transcendental equations. *Proceedings of the Royal Society of Edinburgh*, 45(1):14–22. 17

Allison, P. D. (2002). Missing data: Quantitative applications in the social sciences. *British Journal of Mathematical and Statistical Psychology*, 55(1):193–196. 20

Anderson, E. (1935). The irises of the Gaspe peninsula. *Bulletin of the American Iris Society*, 59:2–5. 37

Andrews, J. L. and McNicholas, P. D. (2012). Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions. *Statistics and Computing*, 22(5):1021–1029. 105

Andrews, J. L. and McNicholas, P. D. (2013). Using evolutionary algorithms for model-based clustering. *Pattern Recognition Letters*, 34(9):987–992. 4, 20, 101

Ashlock, D. (2010). *Evolutionary Computation for Modeling and Optimization*. Springer, New York. 4, 19

Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, pages 803–821. 11, 12

Bartlett, M. S. (1953). Factor analysis in psychology as a statistician sees it. In *Uppsala Symposium on Psychological Factor Analysis*, number 3 in Nordisk Psykologi's Monograph Series, pages 23–43. Copenhagen: Ejnar Mundsgaards. 13

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

Böhning, D., Dietz, E., Schaub, R., Schlattmann, P., and Lindsay, B. G. (1994). The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family. *Annals of the Institute of Statistical Mathematics*, 46(2):373–388. 17

Browne, R. P. and McNicholas, P. D. (2014). Estimating common principal components in high dimensions. *Advances in Data Analysis and Classification*, 8(2):217–226. 12

Browne, R. P. and McNicholas, P. D. (2015). A mixture of generalized hyperbolic distributions. *Canadian Journal of Statistics*, 43(2):176–198. 102

Campbell, N. and Mahon, R. (1974). A multivariate study of variation in two species of rock crab of the genus leptograpsus. *Australian Journal of Zoology*, 22(3):417–425. 62

Caruso, G., Gattone, S., Fortuna, F., and Di Battista, T. (2021). Cluster analysis for mixed data: An application to credit risk evaluation. *Socio-Economic Planning Sciences*, 73:100850. 3

Celeux, G. and Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics & Data analysis*, 14(3):315–332. 45

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern recognition*, 28(5):781–793. 11, 12, 47

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P. A., Łukasik, S., and Żak, S. (2010). Complete gradient clustering algorithm for features analysis of x-ray images. In *Information Technologies in Biomedicine*, pages 15–24. Springer Berlin Heidelberg. 58

Chen, X. and Zhang, A. Y. (2021). Optimal clustering in anisotropic Gaussian mixture models. *arXiv preprint arXiv:2101.05402*. 3

Coomans, D., Broeckaert, I., Jonckheer, M., and Massart, D. (1983). Comparison of multivariate discrimination techniques for clinical data—application to the thyroid functional state. *Methods of Information in Medicine*, 22(02):93–101. 66

Das, A. K., Das, A. K., and Sarkar, A. (2019). An evolutionary algorithm-based text categorization technique. In *Computational Intelligence in Data Mining*, pages 851–861. Springer. 19

Dasgupta, A. and Raftery, A. E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93(441):294–302. 44

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester New York: John Wiley & Sons. 19

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38. 12, 15

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml`. 65

Erola, P., Björkegren, J. L., and Michoel, T. (2020). Model-based clustering of multi-tissue gene expression data. *Bioinformatics*, 36(6):1807–1813. 3

Everitt, B. S. (1981). *Finite Mixture Distributions*. Wiley Online Library. 9

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188. 37

Flury, B. (2012). *Flury: Data sets from flury, 1997*. R package version 0.1-3. 40

Forina, M., Armanino, C., Castino, M., and Ubigli, M. (1986). Multivariate data analysis as a discriminating method of the origin of wines. *Vitis*, 25(3):189–201. 61

Forina, M., Armanino, C., Lanteri, S., and Tiscornia, E. (1983). Classification of olive oils from their fatty acid composition. In *H. Martens and H. Russwurm Jr (Eds.), Food Research and Data Analysis*, pages 189–214. London: Applied Science Publishers. 63

Forina, M. and Tiscornia, E. (1982). Pattern recognition methods in the prediction of Italian olive oil origin by their fatty acid content. *Annali di Chimica*, 72:143–155. 63

Fox, J., Friendly, M., and Monette, G. (2021). *heplots: Visualizing Tests in Multivariate Linear Models*. R package version 1.3-8. 70

Fraley, C. and Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588. 22

Fraley, C. and Raftery, A. E. (1999). Mclust: Software for model-based cluster analysis. *Journal of Classification*, 16(2):297–306. 47, 66, 83

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631. 3, 22, 83

Franczak, B. C., Tortora, C., Browne, R. P., and McNicholas, P. D. (2015). Unsupervised learning via mixtures of skewed distributions with hypercube contours. *Pattern Recognition Letters*, 58(1):69–76. 102

Ghahramani, Z. and Hinton, G. E. (1997). The EM algorithm for factor analyzers. Technical Report CRG-TR-96-1, University Of Toronto, Toronto. 14

Gollini, I. and Murphy, T. B. (2014). Mixture of latent trait analyzers for model-based clustering of categorical data. *Statistics and Computing*, 24(4):569–588. 3

Hasnat, M. A., Velcin, J., Bonnevay, S., and Jacques, J. (2017). Evolutionary clustering for categorical data using parametric links among multinomial mixture models. *Econometrics and Statistics*, 3:141–159. 19

Hassan, B. A. and Rashid, T. A. (2021). A multidisciplinary ensemble algorithm for clustering heterogeneous datasets. *Neural Computing and Applications*, pages 1–24. 19

Heinz, G., Peterson, L. J., Johnson, R. W., and Kerk, C. J. (2003). Exploring relationships in body dimensions. *Journal of Statistics Education*, 11(2). 43

Hinton, G. E., Dayan, P., and Revow, M. (1997). Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks*, 8(1):65–74. 14

Hruschka, E. R., Campello, R. J., Freitas, A. A., et al. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2):133–155. 4, 19

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218. 23

Hurley, C. B. (2004). Clustering visualizations of multidimensional data. *Journal of Computational and Graphical Statistics*, 13(4):788–806. 43, 83

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, New York. 2

Lawley, D. N. and Maxwell, A. E. (1962). Factor analysis as a statistical method. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 12(3):209–229. 13

Lee, K. H., Xue, L., and Hunter, D. R. (2020). Model-based clustering of time-evolving networks through temporal exponential-family random graph models. *Journal of Multivariate Analysis*, 175:104540. 3

Lin, Q., Liu, S., Wong, K.-C., Gong, M., Coello, C. A. C., Chen, J., and Zhang, J. (2019). A clustering-based evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 23(3):391–405. 19

Lin, T. I., Lee, J. C., and Ho, H. J. (2006). On fast supervised learning for normal mixture models with missing information. *Pattern Recognition*, 39(6):1177–1187. 25, 26

Lindsay, B. (1995). Mixture models: Theory, geometry, and applications. In *NSF-CBMS Regional Conference Series in Probability and Statistics*, volume 5. California: Institute of Mathematical Statistics: Hayward. 17

Little, R. J. and Rubin, D. B. (1987). *Statistical Analysis with Missing Data*. Wiley, New York. 21

Lopes, H. F. and West, M. (2004). Bayesian model assessment in factor analysis. *Statistica Sinica*, 14:41–67. 22

Luo, N., Lin, W., Huang, P., and Chen, J. (2021). An evolutionary algorithm with clustering-based assisted selection strategy for multimodal multiobjective optimization. *Complexity*, 2021. 19

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. Oakland, CA, USA. 2

Martınez, A. M. and Vitria, J. (2000). Learning mixture models using a genetic version of the EM algorithm. *Pattern Recognition Letters*, 21(8):759–769. 19

McLachlan, G. and Krishnan, T. (2007). *The EM Algorithm and Extensions*. New York: John Wiley & Sons. 16, 27

McLachlan, G. J. and Basford, K. E. (1988). *Mixture Models: Inference and Applications to Clustering*. New York: Marcel Dekker. 9

McLachlan, G. J. and Peel, D. (2000a). *Finite Mixture Models*. New York: John Wiley & Sons. 9, 10

McLachlan, G. J. and Peel, D. (2000b). Mixtures of factor analyzers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 599–606. 14, 78

McNicholas, P., ElSherbiny, A., McDaid, A., and Murphy, T. (2018). *pgmm: Parsimonious Gaussian Mixture Models*. R package version 1.2.3. 64, 83

McNicholas, P. D. (2016). *Mixture Model-Based Classification*. Boca Raton: Chapman & Hall/CRC Press. 3, 9, 10, 12, 15, 46

McNicholas, P. D. and Murphy, T. B. (2008). Parsimonious Gaussian mixture models. *Statistics and Computing*, 18(3):285–296. 14, 76, 78

McNicholas, P. D., Murphy, T. B., McDaid, A. F., and Frost, D. (2010). Serial and parallel implementations of model-based clustering via parsimonious Gaussian mixture models. *Computational Statistics & Data Analysis*, 54(3):711–723. 17

McNicholas, S. M., McNicholas, P. D., and Ashlock, D. A. (2020). An evolutionary algorithm with crossover and mutation for model-based clustering. *Journal of Classification*, pages 1–16. 4, 20, 73, 102

Melnykov, V. (2016). Model-based biclustering of clickstream data. *Computational Statistics & Data Analysis*, 93:31–45. 3

Meng, X.-L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278. 16

Meng, X.-L. and Van Dyk, D. (1997). The EM algorithm—an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):511–567. 16

Murray, P. M., Browne, R. P., and McNicholas, P. D. (2014). Mixtures of skew-t factor analyzers. *Computational Statistics & Data Analysis*, 77:326–335. 105

Ng, S. K., Krishnan, T., and McLachlan, G. J. (2012). The EM algorithm. In *Handbook of Computational Statistics*, pages 139–172. Springer. 16

Orchard, T., Woodbury, M. A., et al. (1972). A missing information principle: Theory and applications. In *Proceedings of the 6th Berkeley Symposium on Mathematical*

Statistics and Probability, volume 1, pages 697–715. University of California Press Berkeley, CA. 20

Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110. 8

Pernkopf, F. and Bouchaffra, D. (2005). Genetic-based EM algorithm for learning Gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1344–1348. 19

Pitzer, E. and Affenzeller, M. (2012). A comprehensive survey on fitness landscape analysis. In *Recent Advances in Intelligent Engineering Systems*, pages 161–191. Springer. 3

Pocuca, N., Browne, R. P., and McNicholas, P. D. (2021). *mixture: Mixture models for clustering and classification*. R package version 2.0.4. 52

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press. 82

Qiu, W. and Joe, H. (2006). Generation of random clusters with specified degree of separation. *Journal of Classification*, 23(2):315–334. 32, 100

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 30

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850. 22

Reaven, G. and Miller, R. (1979). An attempt to define the nature of chemical diabetes using a multidimensional analysis. *Diabetologia*, 16(1):17–24. 38, 70

Ripley, B., Venables, B., Bates, D., Hornik, K., Gebhardt, A., Firth, D., and Ripley, M. (2020). *MASS*. R package version 7.3-52. 62

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, pages 581–592. 21

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464. 21

Scott, M. A., Mohan, K., and Gauthier, J.-A. (2020). Model-based clustering and analysis of life history data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*. 3

Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2017). mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):205–233. 39

Sobar, Machmud, R., and Wijaya, A. (2016). Behavior determinant based cervical cancer early detection with machine learning algorithm. *Advanced Science Letters*, 22(10):3120–3123. 69

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101. 13

Steinley, D. (2004). Properties of the Hubert-Arable adjusted Rand index. *Psychological methods*, 9(3):386. 31

Streuli, H. (1973). Der heutige stand der kaffeechemie. In *In Association Scientifique International du Cafe, Sixth International Colloquium on Coffee Chemistry*, pages 61–72. 54

Tang, Y., Browne, R. P., and McNicholas, P. D. (2015). Model based clustering of high-dimensional binary data. *Computational Statistics & Data Analysis*, 87:84–101. 3

Tautenhain, C. P. and Nascimento, M. C. (2020). An ensemble based on a bi-objective evolutionary spectral algorithm for graph clustering. *Expert Systems with Applications*, 141:112911. 19

Tipping, T. E. and Bishop, C. M. (1999). Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482. 14

Titterington, D. M., Smith, A. F., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. New York: John Wiley & Sons. 9

Tortora, C., McNicholas, P. D., and Palumbo, F. (2020). A probabilistic distance clustering algorithm using Gaussian and student-t multivariate density distributions. *SN Computer Science*, 1(2):65. 3

Venables, W. and Ripley, B. (2002). *Modern Applied Statistics with S*. New York: Springer Verlag. 62

Vrbik, I. and McNicholas, P. (2012). Analytic calculations for the EM algorithm for multivariate skew-t mixture models. *Statistics & Probability Letters*, 82(6):1169–1174. 105

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244. 2

Wei, Y., Tang, Y., and McNicholas, P. D. (2019). Mixtures of generalized hyperbolic distributions and mixtures of skew-t distributions for model-based clustering with incomplete data. *Computational Statistics & Data Analysis*, 130:18–41. 3, 25

Weicker, K. and Weicker, N. (2003). Basic principles for understanding evolutionary algorithms. *Fundamenta Informaticae*, 55(3-4):387–403. 19

Weisberg, S. (2014). *Applied Linear Regression*. Hoboken, NJ: Wiley. 56

Weisberg, S. (2018). *alr3: Companion to Applied Linear Regression*. R package version 2.0.8. 56

Wolfe, J. H. (1965). A computer program for the maximum-likelihood analysis of types. Technical report, U.S. Naval Personal Research Activity, San Diego. 9

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., and Ruzzo, W. L. (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987. 3