

https://dspace.org/

The effect of outliers in model-based clustering using the Expectation Maximization (EM) algorithm

Mpogeng, Reatile

2020
https://hdl.handle.net/10539/31067

Downloaded from a DSpace repository. https://dspace.org/

MSc Research Report



UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

The Effect of Outliers in Model-based Clustering using the Expectation Maximization (EM) Algorithm

By

Reatile Mpogeng
(1057086)

Supervisor
Nothabo Ndebele

A research report submitted to the Faculty of Science, University of the Witwatersrand, in partial fulfilment of the requirements for the degree of Master of Science

July 24, 2020

Contents

1 Introduction **1**

1.1 Introduction 1

1.2 Background 2

1.3 Statement of the Problem 3

1.4 Aims and Objectives 4

1.4.1 Aims 4

1.4.2 Objectives 4

1.5 Limitations and Assumptions 4

1.5.1 Limitations 4

1.5.2 Assumptions 5

2 Literature Review **6**

2.1 Introduction 6

2.2 Overview of clustering algorithms 7

2.3 Model-based clustering 9

2.3.1 The Expectation Maximization Algorithm 13

2.3.2 The use of a prior distribution 16

2.3.3 Performance Measures 18

2.3.3.1 Bayesian Information Criteria 18

2.3.3.2 Classification accuracy 19

2.4 Prior Work 20

2.5 Outliers 21

2.5.1 Outlier Identification 21

2.5.2 Outlier Detection algorithms 22

2.5.3 Dealing with outliers 24
2.6 Simulations 26

3 Methodology **28**
3.1 Data 28
3.1.1 Simulations 28
3.1.2 Real data 30
3.2 Algorithms 31
3.2.1 Visualisation of data 31
3.2.2 Outlier detection 31
3.2.3 Running the EM algorithm 32
3.2.3.1 Using a prior distribution 33
3.2.4 Performance measures 33
3.2.4.1 Testing the effect of outliers on the number of clusters 33
3.2.4.2 Testing the effect of outliers on the choice of models 33
3.2.4.3 Testing the effect of the outliers on the parameter estimation 34

4 Analysis **35**
4.1 Outlier detection 35
4.2 Results using the EM algorithm 37
4.3 Effect of outliers on number of clusters 42
4.4 Effect of outliers on the structure of the clusters 47
4.5 Effect of outliers on parameter estimates 49
4.6 Use of a Prior Distribution 53
4.6.1 Number of clusters using BIC 53
4.6.2 Strucuture of clusters 56
4.6.3 Parameter estimates 57
4.7 Further investigations into parameter estimation 59
4.8 Application to a real dataset 61
4.8.1 Mammography dataset 62
4.8.2 Lymphography dataset 65

5 Discussion **69**
5.1 Sensitivity of the algorithm to datapoints 69
5.2 Collapse of the algorithm 70
5.3 Difference in loglikelihood 73
5.4 Disruption of the algorithm by the prior distribution 75

6 Conclusion **77**

A Simulations **88**

B Parameter estimates **93**

C Depth-based outlier detection algorithm results **96**

D R codes **98**
D.1 Simulations 98
D.2 Model Fitting 99
D.3 Extraction of Models 103
D.4 Parameter estimates 106
D.5 Outlier detection 107

List of Figures

2.1 A plot of two clusters with two variables contaminated with 2 outliers. 7

2.2 Examples of clusters on which the a spherical cluster with equal volume (EII) and a diagonal cluster with equal shape and volume (EEI) model can be fit. 11

3.1 Simulation with 100 datapoints 30

4.1 Simulated datasets and the corresponding Mahalanobis distance outlier detection algorithm results. 36

4.2 Simulated datasets and the corresponding output from the depth-based outlier detection algorithm. 38

4.3 Results of running the EM algorithm on a simulated data set 40

4.4 Results from running the EM algorithm on the simulated dataset in Figure 4.3(a) without outliers. 41

4.5 BIC curves for the first 6 simulated Spherical and Homogeneous clusters with (a) 20, (b) 40, (c) 60 and (d) 80 data points respectively and 2 outliers in each case. 43

4.6 BIC curves of the simulated datasets with spherical and homogeneous clusters of sizes 20, 40, 60 and 80 respectively, without outliers. . . . 45

4.7 The loglikelihood from models fitted on datasets with 2 spherical homogeneous clusters of approximately equal sizes, with and without outliers. 48

4.8 Mean estimates for spherical homogeneous clusters with 2 outliers vs without the outliers. 50

4.9 Mean estimates for spherical homogeneous clusters with 2 outliers vs without the outliers continued. 51
4.10 BIC of spherical homogeneous clusters with outliers using prior distribution for data sizes with 20.40,60 and 80 data points. 54
4.11 BIC of spherical homogeneous clusters without outliers using prior distribution for data sizes with (a)20, (b)40, (c)60 and (d)80 data points. 55
4.12 The loglikelihood of spherical homogeneous clusters with and without outliers with a prior distribution. 56
4.13 Mean estimates of spherical homogeneous clusters with 2 outliers vs without outliers using the prior distribution. 58
4.14 Mean estimates as the number of deviations of the outliers away from the mean increases. 60
4.15 Varinace estimates as the number of standard deviations of the outliers away from the mean increases. 61
4.16 Probability plots for outliers using the Mahalanobis distance and the depth-based outlier detection algorithms respectivel 63
4.17 BIC curves of the Mammography dataset, with and without outliers, respectively. 64
4.18 BIC curves of the Mammography dataset, with and without outliers, respectively, using the prior distribution. 65
4.19 Probability plots for the Mahalanobis distance and the depth based algorithms for outlier detection 66
4.20 BIC curves for the Lymphography dataset 67
4.21 BIC curves for the Lymphography dataset with and without outliers, using the prior distribution 68

5.1 Classifications of spherical homogeneous clusters with outliers 71
5.2 Density of spherical homogeneous clusters with outliers. 72
5.3 Three Spherical inhomogeneous clusters vs BIC curves 74

A.1 Simulations of spherical clusters with 2 outliers 89
A.2 Simulations of spherical homogeneous clusters with 2 outliers continued. 90
A.3 Simulations of spherical clusters with 5 outliers 91

A.4 Simulations of spherical homogeneous clusters continued with 5 outliers continued 92

B.1 Mean estimates for spherical homogeneous clusters with 2 outliers vs without the outliers with the prior distribution. 94

B.2 Mean estimates of spherical homogeneous clusters with 2 outliers vs without the outliers with the prior distribution continued. 95

C.1 Classification plot for the Depth-based algorithm with threshold of 0.04 97

Dedication

This dissertation is dedicated to two of the strongest women in my life, who gave me an opportunity to pursue my studies, my grandmother, Eunice Mathume Tshehla and my mother, Mavis Malebese Mpogeng.

Acknowledgement

A special gratitude to my supervisor Nothabo Ndebele. She has been amazing in supporting and guiding me in conducting my research. Her timeous responses helped me complete my work in time.

I am also grateful to the Statistics lecturers from the School of Statistics and Actuarial Science at the University of the Witwatersrand, who equipped me with the primary knowledge that I used to do my dissertation.

Last but not least, I would like to thank the Nedbank Eyethu Community Trust for granting me the funds I needed to do my dissertation.

Declaration

I declare that this research project is my own unaided work. It has not been submitted before for any degree or examination to any other University.

Chapter 1

Introduction

1.1 Introduction

The value and use of data is becoming prominent in bettering society and businesses. In agriculture, enormous amounts of data are being collected through weather forecasting, remote sensing and geographic information systems. With the use of data mining techniques, this data can be used, for example, in the discovery of information in agriculture (Cebeci and Yildiz, 2015). Business intelligence and analytics are growing to become an imperative field of work for researchers, which extends the need for data-related solutions (Chen et al., 2012). Classification, clustering, regression, outlier detection and correlation, are some of the data mining techniques that can be useful in the understanding of societal and corporate problems through data.

Data mining tools are often used to create models that are aimed at representing a real phenomenon. The drawback of these data mining tools is that most of them are based on statistical theory that is based on sets of assumptions which are not necessarily practical. Of interest in this study is clustering algorithms.

1.2 Background

Clustering is one of the most important data mining tools. A clustering algorithm learns to group objects based solely on attributes of those objects and this is where the description of clustering as an unsupervised learning technique comes from. Information on the actual group/cluster in which an object belongs is unknown or undisclosed. Clustering has been defined in multiple ways that all revolve around finding the best grouping of unlabelled data. Some researchers define clustering as; the arrangement of a set of patterns into clusters based on some similarity (Jain et al., 1999), or, a separation of an unlabelled data set of a fixed size into a fixed and discrete set of naturally unseen data structures (Xu and Wunsch, 2005). A classic description of clustering is that observations in the same cluster must be homogeneous as much as possible, those in different clusters must be inhomogeneous as much as possible and the measure of similarity must be clear and have a practical meaning (Xu and Tian, 2015a).

There is a wide range of clustering algorithms categorized according to the underlying processes used to reveal clusters. For example, hierarchical, partitioning, fuzzy, model-based and density-based clustering algorithms (Xu and Tian, 2015a).

Clustering has many uses including exploratory data analysis and as a method for classification. Applications of clustering are found in data compression, vector quantization (Gersho and Gray, 2012), probability density estimation and maximization of entropy (Xu and Wunsch, 2005).

Nevertheless, some of the main challenges in clustering relate to high dimensional data, unconventional shapes and densities and when data has noise and/or outliers (Ertöz et al., 2003; Steinbach et al., 2004). Some clustering methods, for example, partition based and fuzzy algorithms tend to have low performance in data that has noise or outliers (Gosain and Dahiya, 2016). In Model-based Clustering (MBC), clusters with very few observations, or those with linearly related variables, may cause the algorithm to collapse (Fraley and Raftery, 1998). Central to the topic of this research project is the study of model-based clustering algorithms in the presence of outliers.

Data points are regarded as outliers based on a particular definition which varies according to the assumptions of the data structure as well as the methods used to identify the outliers (Ben-Gal, 2005). Outliers may arise as a result of mechanical faults, alterations to system behaviour, fraudulent behaviour as well as human error (Hodge and Austin, 2004). Outlier detection involves the identification of outliers in data sets. Modelling data with outliers can cause incorrect model choices and biased parameter estimators. Hence the importance of outlier detection algorithms to help identify these outliers and therefore deal with them before data is used for modelling and analysis (Williams et al., 2002).

There are several ways proposed for dealing with outliers when performing clustering exercises. A homogeneous Poisson process has been used to model noise and/or outliers in mixture models. The Poisson process aims to represent nonconforming data. Another way of dealing with outliers is iterated sampling (Fayyad and Smyth, 1997). This approach is best for data with isolated outliers, where points that have low probability are eliminated from the clusters. The process is iterated until all remaining points are of higher probability.

In the context of clustering, there are performance measures used to study the ability of clustering algorithms to recover clusters. Bayesian Information Criterion (BIC) is used to choose the best fitting model (Kass and Raftery, 1995) and classification accuracy counts the number of observations correctly allocated in a cluster (Meilă and Heckerman, 2001). The time it takes an algorithm to run and learn from data is also another measure used to study an algorithm's performance.

1.3 Statement of the Problem

Outliers may pose a problem in model-based clustering as they may cause biases in parameter estimates, identifying the shape of clusters and identifying the number of clusters. This could be explained by the definition of an outlier by Johnson et al. (2002) as an observation that looks incompatible with the rest of the data. This study is done

in order to understand the effect that outliers have when applying clustering techniques.

1.4 Aims and Objectives

1.4.1 Aims

This investigation aims to study model-based clustering (MBC) of data in the presence of outliers using the Expectation Maximization (EM) algorithm.

1.4.2 Objectives

This study will answer the following questions;

1. How do outliers affect the EM algorithm's ability to recover the number of clusters using the BIC criteria?
2. Using the loglikelihood that maximises the BIC, what is the effect of the outliers on the recovery of the structure of clusters?
3. How are the parameters estimated using the EM algorithm affected by the outliers?
4. How does the use of a prior distribution in the EM algorithm affect the effect of the outliers?

1.5 Limitations and Assumptions

1.5.1 Limitations

There are different ways of measuring the performance of MBC techniques in recovering clusters. The conclusions of the results obtained in this study are therefore based on the measurement of performance chosen. In this study, the Bayesian Information Criterion and the loglikelihood will provide the means to measure the performance of

the algorithm. Consequently, this may limit the use of the results of this study in cases where different performance measures are used.

There are several ways in which outliers can occur or be simulated. A simulation is a data generation process that is usually used to study the performance of algorithms (Maitra and Melnykov, 2010). This study is based on the definition of outliers to be points that lie beyond a predetermined contour of the respective clusters. Therefore different results might be obtained using a different definition of outliers.

1.5.2 Assumptions

Since the algorithms are based on statistical theory, there is a set of assumptions that has to be made;

1. Data comes from a mixture of models, usually of the same family.
2. A sample of the data can be taken in accurate proportions from each cluster/-component. Such accurate proportions refer to a sample of a desired size that can be taken from each cluster.
3. Objects clustered together as a result of the MBC algorithm are assumed to be independent and identically distributed.

Chapter 2

Literature Review

2.1 Introduction

In disciplines such as medicine and demography, amongst others, the ability to identify groups of data with similar characteristics can be useful. For instance, in Oncology, separating cancerous and non-cancerous cells can be very helpful in the diagnosis of cancer. Identifying the working class and unemployed citizens within a certain geographical area can help determine the profitability of luxurious outlets such as spas. Such identified groups of data sharing similar characteristics are regarded as clusters. An alternative definition of a cluster is that, a cluster is a region in which the occurrence of observations is high compared to other regions (Likas et al., 2003).

It is sometimes the case that a dataset with clusters contains some observations that appear to not belong to the dataset. These could be because of human error or new/unexpected occurrences. These data points can be regarded as anomalies or outliers. An example of such a scenario is illustrated in Figure (2.1) below. The figure illustrates two clusters in grey and black, and amongst these are two outliers in red.

One aspect of clusters which may be useful is the centre of the cluster. The centre of a cluster j, which we can denote as c_j, is the point that best suits to be in the middle of a cluster based on datapoints surrounding it (Gough, 2001). This may or may not

be an existing datapoint. Another interesting aspect of clusters is the overlap between clusters. This is referred to as the pairwise overlap between clusters (Melnykov et al., 2012).

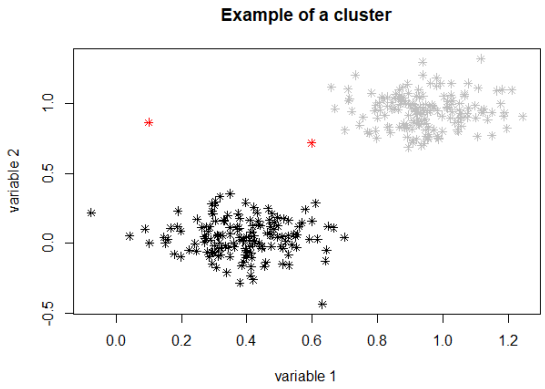


Figure (2.1) A plot of two clusters with two variables contaminated with 2 outliers.

The existence of clusters is hardly known beforehand and has to be identified. Identification of these clusters is called clustering. Clustering is the process of discovering whether there are any naturally inherent arrangements of data into homogeneous groups of objects. The homogeneity of objects is determined based on similarity measures. There are different types of similarity measures. Distance-based similarity measures include the Euclidean and the Manhattan distances which measure similarity of 2 objects according to the distance between the objects and the cluster centre (Schoenharl and Madey, 2008). Another similarity measure is the standard correlation coefficient which considers the shape of the clusters to capture similarity. This measure is often used in gene expression data (Eisen et al., 1998). Algorithms that carry out the process of identifying groups of objects within a dataset, are called clustering algorithms. In Melnykov et al. (2012), the pairwise overlap is defined to be a measure of the degree of interaction between clusters. The interaction or pairwise overlap between clusters may make it difficult to recover clusters within data.

2.2 Overview of clustering algorithms

Clustering algorithms are divided into various categories such as partition-based algorithms, hierachical algorithms, algorithms based on fuzzy theory and model-based algorithms (Xu and Tian, 2015b). In partition-based algorithms, the centres of datapoints within clusters are regarded as the cluster centres c_j where $j = 1, 2, ..., k$ in the case of a dataset with k clusters. The centres are iteratively updated so as to minimise a distance measure or increase similarity between clusters. Examples of such algorithms are the k-means algorithm (Cebeci and Yildiz, 2015) and k-medoids (Park and Jun, 2009). These algorithms run over relatively shorter periods of time compared to other clustering algorithms. Since these algorithms revolve around the centre of a cluster, they do not function well with non-spherical data and are relatively sensitive to outliers because outliers would not conform to the nature of the data and hence may affect the mean of clusters (Cebeci and Yildiz, 2015). A defining factor is that the number of clusters need to be prespecified and this leads the performance of the algorithm being reliant on methods used to determine the number of clusters.

The construction of hierarchical relationships among data can be used to form clusters (Johnson, 1967). This can happen in two ways. Hierarchical agglomerative clustering where each datapoint represents an individual cluster to start with, then pairs of clusters or data points that are closely alike are grouped to form one. The alternative is hierarchical divisive clustering where all datapoints start off as one cluster, and according to some dissimilarity measure they are separated. Balanced Iterative Reducing Clustering using Hierachies (BIRCH) (Zhang et al., 1997) and Robust Clustering algorithms for categorical attributes (ROCK) are examples of hierarchical clustering algorithms (Jain and Dubes, 1988). Hierarchical algorithms are advantageous over those based on partitions in that they function well for data structures with arbitrary shapes and attributes of arbitrary types. Unlike partition-based algorithms, the number of clusters need not be prespecified. The drawback of the algorithms is that the ranking of the objects tends to cause the algorithm to run over relatively longer periods of time.

Clustering algorithms based on fuzzy theory substitute the idea that an instance can either belong or not belong to a particular cluster, with the reasonable idea that the

relationship among objects can be quantified by values on the scale from 0 (no similarity) to 1 (highest similarity) inclusive. The Fuzzy K-Means algorithm is a variation of the k-means algorithm which uses fuzzy theory to identify clusters (Dunn, 1973). Algorithms based on fuzzy theory have relatively high accuracy in clustering (Xu and Tian, 2015a). Fuzzy algorithms require a prespecified number of clusters and are sensitive to initialization parameters and this may be a disadvantage.

Model-Based clustering (MBC) involves the selection of a best fitting probability model (with parameters) for each cluster. MBC models are well developed models that allow an adequate description of data where each model has its own defining characteristics. However, the resulting clusters are sensitive to the parametrization of the models and this can be a disadvantage of the algorithm. The algorithms are mostly developed for Gaussian clusters and this is a challenge since clusters can occur in different forms not conforming to the Gaussian nature. Nevertheless, a proposition was made for dealing with non-Gaussian clusters by reparametrizing the covariance matrices of the clusters (Banfield and Raftery, 1993). Sophisticated statistical packages have been developed for the special study of MBC algorithms such as the Mclust (Fraley and Raftery, 2006) and MixSim (Melnykov et al., 2012) packages in R Core.

2.3 Model-based clustering

As an improvement from heuristic methods, it was realised that cluster analysis can be done using probability models which are more robust and based on statistical theory (Bock, 1996). Identification of clusters is viewed as model fitting for groups of objects and the list of possible model structures is shown in Table (2.1). Figure (2.2) shows examples of clusters on which the EII and EEI models may be fit.

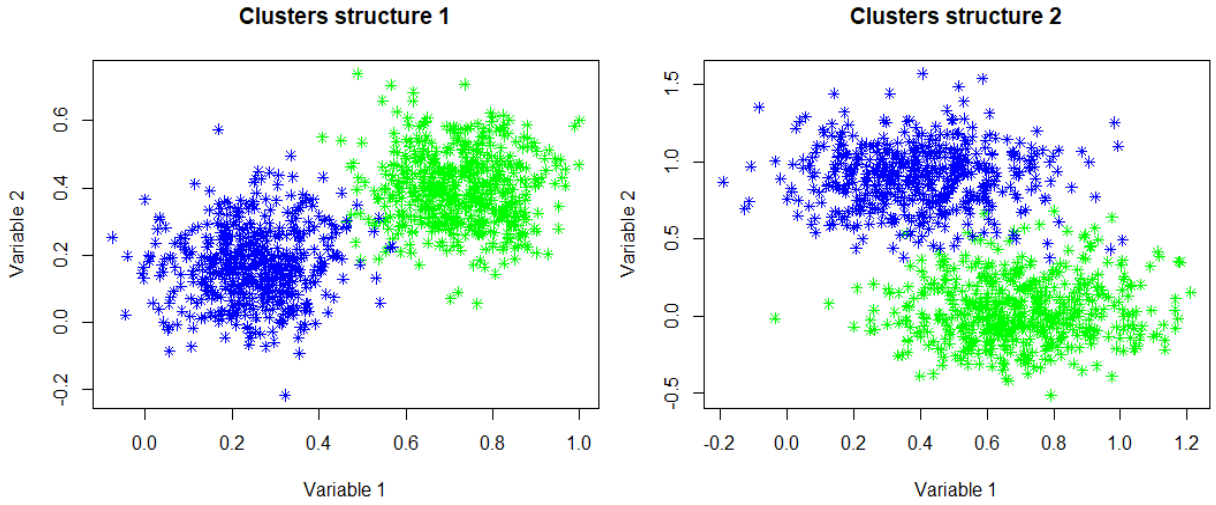
In model-based clustering the data is considered as observations from a mixture of models. A cluster is referred to as a component and each is characterized by a probability distribution. Finite mixture models are known to be combinations of these probability distributions. The prevalence of the approach of using finite mixture models in clustering is relatively recent (Cheeseman et al., 1996). Unlike heuristic clustering

algorithms, finite mixtures are not only useful for the purpose of assigning individual objects into clusters but also in describing an entire distribution (Benaglia et al., 2009). Alternatives to finite mixtures are Dirichlet process mixtures (Frühwirth-Schnatter and Malsiner-Walli, 2019). Unlike finite mixture models, Dirichlet process mixtures do not assume a mixture of a finite number of components. Although the number of parameters in Dirichlet process mixtures is infinite, it has been found to be possible to do an inference in these infinite mixture models using Markov chain Monte Carlo methodology (Escobar and West, 1995; Rasmussen et al., 2008; Rasmussen, 2000)

Table (2.1) Table of models in Model-based clustering

Model Structure	**Description**
EII	spherical, equal volume
VII	spherical, unequal volume
EEI	diagonal, equal volume and shape
VEI	diagonal, varying volume, equal shape
EVI	diagonal, equal volume, varying shape
VVI	diagonal, varying volume and shape
EEE	ellipsoidal, equal volume, shape, and orientation
EVE	ellipsoidal, equal volume and orientation
VEE	ellipsoidal, equal shape and orientation
VVE	ellipsoidal, equal orientation
EEV	ellipsoidal, equal volume and equal shape
VEV	ellipsoidal, equal shape
EVV	ellipsoidal, equal volume
VVV	ellipsoidal, varying volume, shape, and orientation

Non-statistical models that can be used are Self-Organizing Maps (SOM) which are neural network learning methods (Kangas et al., 1996). In SOMs, data is mapped to an n-dimensional grid of neurons. There is an input space and an output space. Patterns



(a) The EII model cluster.

(b) The EEI model cluster.

Figure (2.2) Examples of clusters on which the a spherical cluster with equal volume (EII) and a diagonal cluster with equal shape and volume (EEI) model can be fit.

that are close in the input space are mapped to units that are close in the output space, and patterns that are close in the output space are mapped to units that are close in the input space (Lampinen and Oja, 1992).

Models that are used in MBC are usually multivariate distributions (Yeung et al., 2001). Distributions such as the multivariate normal distribution are used when two or more variables are involved. The set of n variables $\boldsymbol{X} = (X_1, X_2, ..., X_n)$ where $-\infty < x_i < \infty$ and n is an integer such that $n > 1$, follow a multivariate normal distribution if the joint density function is expressed as (Tong, 2012);

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp^{\frac{(\boldsymbol{x}-\mu)' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\mu)}{2}}, \tag{2.1}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, ..., \mu_n)$ is the vector of means of each variable in $\boldsymbol{X}$ and $\boldsymbol{\Sigma}$ is the $n \times n$ covariance matrix such that $\boldsymbol{\Sigma_{i,j}}$ is the covariance between the i^{th} and j^{th} variables. In MBC, the aim is to find the best estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

A special case of the multivariate distribution is the bivariate normal distribution in which $n = 2$. The random variable $\boldsymbol{X} = (X_1, X_2)$ has a bivariate normal distribution if the density function has the form:

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp^{\frac{(\boldsymbol{x}-\mu)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\mu)}{2}}, \tag{2.2}$$

where

$$\boldsymbol{\mu} = (\mu_1, \mu_2)' \text{ and } \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}. \tag{2.3}$$

Consider a set of data;

$$\begin{bmatrix} x_1 & h_1 j \\ x_2 & h_2 j \\ . & . \\ . & . \\ . & . \\ x_n & h_n j \end{bmatrix}$$

where x_i is the i^{th} observation of the multivariate random variable $\boldsymbol{X}$ and h_{ij} is 1 if the i^{th} observation belongs to cluster j and 0 if not, for $i = 1, 2, ..., n$. Considering a maximum number of P clusters, the underlying assumption in model-based clustering is that the density of the observations x_i given the appropriate cluster j is $\prod_{j=1}^{P} f_j(\boldsymbol{x_i}|\boldsymbol{\theta_j})^{h_{ij}}$, which is the mixture of distributions $f_j, j = 1, 2, ..., P$, where the h_{ij}'s are discrete independent and identically distributed according to some multinomial distribution with a total number of P categories and probabilities $\tau_1, \tau_2, ..., \tau_j$ and $\boldsymbol{\theta_j}$ is the vector of the parameters of f_j. The corresponding loglikelihood function is

$$l(\boldsymbol{\theta_j}, \tau_j, h_{ij}|\boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j=1}^{P} h_{ij}[\log(\tau_j f_j(\boldsymbol{x_i}|\boldsymbol{\theta_j}))], \tag{2.4}$$

where f_j and $\boldsymbol{\theta_j}$ are the probability density and vector of parameters of the j^{th} cluster, respectively, and τ_j is the probability of an observation belonging to cluster j, under the constraint $\sum_{j=1}^{P} \tau_j = 1$. The density f_j is usually modelled using a multivariate normal distribution with mean $\boldsymbol{\mu_j}$ and covariance matrix $\sum_{\boldsymbol{j}}$. Data modelled using the

normal mixtures are characterised by ellipsoidal densities with increased density/concentration of objects nearer to the mean $\boldsymbol{\mu_j}$. The covariance matrix is such that;

$$\sum_j = \boldsymbol{\lambda_j D_j A_j D_j^T}, \tag{2.5}$$

where $\boldsymbol{D_j}$ is an orthogonal matrix of eigenvectors, $\boldsymbol{A_j}$ is a diagonal matrix whose elements are proportional to the eigenvalues and $\boldsymbol{\lambda_j}$ is an associated constant of proportionality. $\boldsymbol{D_j}$ regulates the orientation, $\boldsymbol{A_j}$ the shape and $\boldsymbol{\lambda_j}$ the volume of the j^{th} cluster, respectively (Banfield and Raftery, 1993). For instance, a model with $\sum_j = \boldsymbol{\lambda I}$, is one in which clusters are spherical and have the same size. This is the model EII described in Table (2.1). A model in which $\sum_j = \boldsymbol{\lambda_j I}$ has spherical clusters with different volumes. This model is known as the VII model described in Table (2.1). Another model has $\sum_j = \boldsymbol{\lambda_j A_j}$ in which clusters are assumed to have different shapes, sizes and orientation, but equal covariance. Furthermore, $\sum_j = \boldsymbol{\lambda_j D_j D_j^T}$ is one with components that have equal shape or volume. This model was used in (Murtagh and Raftery, 1984) to produce clusters in character recognition.

2.3.1 The Expectation Maximization Algorithm

In order to estimate the probabilities of membership to a cluster j as well as the mixture model parameters μ_j and Σ_j, the Expectation Maximization (EM) algorithm is used. This algorithm allows the iterative allocation of objects within clusters using mixture models.

$$L_{cl}(\boldsymbol{\theta_1, \theta_2, ..., \theta_P}, t_i, t_2, ..., t_n|\mathbf{y}) = \prod_{i=1}^{n} f_{t_i}(y_i|\boldsymbol{\theta_{t_i}}) \tag{2.6}$$

The EM algorithm requires an initialisation of the mean ($\boldsymbol{\mu_j}$) and covariance matrices ($\boldsymbol{\Sigma_j}$) of clusters. The model-based hierarchical clustering algorithm has been used to obtain the initial parameter values of models (Fraley and Raftery, 2000). The algorithm works by computing an approximate maximum for the classification likelihood in Equation (2.6), where $\mathbf{y}$ is a set of observations from a sample, y_i is i^{th} observation from the sample, n is the sample size and t_i an indicator of a unique classification for

each observation, that is, $t_i = j$ if y_i belongs to the j^{th} cluster. The algorithm works by successively merging pairs of clusters, including singletons, that result in the greatest increase in the likelihood function in Equation (2.6). This algorithm has important use in automated taxonomy generation and has further been generalised to other models and applications (Banfield and Raftery, 1993).

Provided the initialization of parameters of the probability models, the EM algorithm runs over two phases, the E-step and the M-step, that aim to maximize the likelihood in Equation (2.4). The E-step does not require knowledge of the family of distributions that describes the finite mixture models. In this step, the algorithm chooses a function that creates a lower bound for the likelihood in Equation (2.4). The M-step which is divided into two steps, the maximization of h_{ij} which does not depend on the family of mixture model, and the maximization related to f_j for $j = 1, 2, ..., P$. In this step, the algorithm produces another parameter set that also maximizes the likelihood. The estimates of the likelihood are monotonically increasing with every iteration. The algorithm also requires initialization of parameter h_{ij}. The theoretical value of h_{ij} is $h_{ij} = E[h_{ij}|x_i, \boldsymbol{\theta_1}, \boldsymbol{\theta_2}, ..., \boldsymbol{\theta_j}]$ where $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, ..., \boldsymbol{\theta_P}$ are the parameters for clusters $1, 2, ...$ and P, respectively, and the classification of x_i is based on the maximum value of h_{ij}.

The EM algorithms works as follows:

1. Determine the highest number of clusters to be considered.
2. The E-step: Initialize h_{ij} using the classification likelihood in Equation (2.6).
3. M-step: Given h_{ij}, calculate the estimates of the maximum likelihood parameters;
   $n_j \leftarrow \sum_{i=1}^{n} h_{ij}$,
   $\tau_j \leftarrow \frac{n_j}{n}$,
   $\mu_j \leftarrow \frac{\sum_{i=1}^{n} h_{ij} x_i}{n_j}$ and
   $\sum_j$ which depends on the structure of the data (Celeux and Govaert, 1995) where n_j is the size of the j^{th} cluster , τ_j is the estimated probability of belonging to the j^{th} cluster, μ_j is the mean of the j^{th} cluster and Σ_j is the covariance matrix of the j^{th} cluster.

4. Provided the estimates of the parameters from the M-step, calculate h_{ij}; $h_{ij} \leftarrow \frac{\tau_j f_j(x_i|\mu_j,\sum_j)}{\sum_{j=1}^{P} \tau_j f_j(x_i|\mu_j,\sum_j)}$.

5. Repeat the EM step for each of the clusters.

6. Continue iteratively until the loglikelihood in Equation (2.4) is at maximum and the algorithm converges. The convergence of the algorithm is determined by the iterative convergence tolerance which is a value predetermined by the user through experience and/or knowledge of the data at hand. This process then leads to relocation of observations in appropriate clusters.

In general terms, the Expectation Maximization algorithm can be thought of as an extension of the maximum likelihood estimation in that they are both optimization algorithms. The expectation maximization algorithm seeks for parameters θ_k, for $k = 1, 2, ..., P$ that maximize the likelihood in Equation (2.4). The difference is that the information on observations can be regarded as incomplete since information on clusters is unknown or undisclosed but is deduced using the algorithm. As a result, the likelihood in Equation (2.4) has multiple local maxima and has no closed form compared to the maximum likelihood estimation which will often have a global maximum and a closed mathematical form (Do and Batzoglou, 2008).

An alternative to the EM algorithm is the classification expectation maximization algorithm. However, the classification maximization approach maximizes the classification likelihood in Equation (2.6) and not the finite mixture likelihood in Equation (2.4) (Celeux and Govaert, 1995). Another technique used to obtain a maximization of the finite mixture model likelihood function is the Newton-Raphson method, however, the EM algorithm is more robust, simple and implementaion is convenient (Do and Batzoglou, 2008).

The EM algorithm, however, has limitations. One of the main issues with maximization algorithms is that they might be trapped by a local maximum instead of a global maximum. Since the algorithm uses thresholds to separate clusters, in cases where the clusters are unclear or overlap inaccurate thresholds can be chosen. Thirdly, the algorithm is reliant on an initialisation algorithm, therefore one must be careful with the

choice of the initial partitioning algorithm. The model-based hierarchical clustering technique has been used successfully (Fraley and Raftery, 2002) and has been built into the EM algorithm in statistical software such as R as the preferred initialization algorithm (Fraley and Raftery, 2006). Other limitations of the EM algorithm occur when the number of clusters is too high and/or when the clusters consist of too few observations (Yeung et al., 2001). Such scenarios cause the covariance matrices of clusters to be singular. A singular matrix is a square matrix without an inverse. As a result of the singularity of covariance matrices, the algorithm collapses. A proposition has been made that helps prevent the EM algorithm from breaking down, this is the Degenerate EM algorithm. Once a covariance matrix is detected to be singular, artificial perturbations are applied to it. This assists in keeping the likelihood function bounded by forcing the covariance matrices to be non-singular (Lin and Zhu, 2004).

2.3.2 The use of a prior distribution

A common issue with the Expectation Maximization algorithm is failure to converge where the likelihood becomes infinite. In general, most likelihood functions are unbounded and there often are paths in the parameter space where the likelihood tends to infinity (Titterington et al., 1985). The usual causes are singularity of the covariance matrices, which arises most often when the covariances of variables in clusters are allowed to vary or when the number of clusters is very high.

The EM algorithm is able to use conditional probabilities to handle cases were the algorithm collapses and to find better estimates when fitting models. This is done by incorporating prior knowledge of the parameters. Considering two events A and B, the conditional probability of event A given event B is defined as;

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{2.7}$$

where $P(A \cap B)$ is the probability of event A and B, and $P(B)$ is the probability of event B.

A simple variation of this equation is;

$$P(A \cap B) = P(A|B)P(B). \tag{2.8}$$

In the context of modelling, unknown parameters in Θ, can be modeled using a probability distribution $f_{\Theta}(\theta)$. This distribution is based on our prior knowledge of the parameters Θ hence it is known as the prior distribution. Observed data $\boldsymbol{X}$, have the probability distribution $f_{\boldsymbol{X}|\Theta}(\boldsymbol{x}|\theta)$. Therefore, using Equation (2.8) the joint distribution of the parameters in Θ and the random variable $\boldsymbol{X}$ is (Tiao and Zellner, 1964)

$$f_{\boldsymbol{x},\Theta}(\boldsymbol{x},\theta) = f_{\boldsymbol{X}|\Theta}(\boldsymbol{x}|\theta)f_{\Theta}(\theta). \tag{2.9}$$

Therefore, in the context of the parameter Θ, we can deduce that ;

$$f_{\Theta|X}(\theta|\boldsymbol{x}) = \frac{f_{\boldsymbol{x},\Theta}(\boldsymbol{x},\theta)}{f_X(\boldsymbol{x})} \tag{2.10}$$

and $f_{\Theta|X}(\theta|\boldsymbol{x})$ is known as the posterior distribution of $\boldsymbol{X}$. In cases when the prior and posterior distributions of the parameters have the same form, then the prior is referred to as the conjugate prior distribution (Kadane et al., 2006).

Therefore, the prior distribution is a distribution of parameters which together with knowledge of the distribution of data can be used for further inference about the true parameters (Gelman, 2006). The inference could be used in mixture models where the parameters of the mixtures are estimated based on their prior distribution. In the cases when the provided data is very small or insufficient to make estimates of the parameter of interest, then the prior distribution is important. In the opposite scenario, the results obtained with and without the use of the prior distribution will be very similar.

When faced with a clustering problem such that the cluster sizes are very small or the clusters are sparse, the EM algorithm fails to produce results for some models. A prior distribution can be used as a method that regularizes the EM algorithm from failure to fit some models. It is worth the effort to understand whether optimal results are not produced by the models whose results were not produced by the algorithm. In this

case, a prior distribution of the parameters of the mixture model is introduced in Fraley and Raftery (2007).

Considering a simple case of a univariate dataset with observations $y_1, y_2, ..., y_n$, the difference in parameter estimates when the EM algorithm is run using the prior distribution is illustrated in Table (2.2). The parameter z_{jk} is the conditional probability that observation j belongs to the k^{th} cluster. The mean of the k^{th} cluster is $\bar{y}_k$ and the number of observations from the k^{th} cluster is n_k. The parameters μ_p, κ_p, υ_p and s_p are the mean, shrinkage, degrees of freedom of the prior distribution and the scale, respectively. These are assumed to be the same for all clusters. The default values of the parameters may are as follows;

1. μ_p is the mean of the data.
2. $\kappa_p = 0.01$.
3. $\upsilon_p = d + 1$, where d is the dimension of the data.
4. $s_p = \frac{covariance(data)}{G^{\frac{2}{d}}}$ where G is the number of components in the data.

Derivations of these parameters can be found in Fraley and Raftery (2005).

Table (2.2) Parametrization with and without using the prior distribution in the EM algorithm.

Parameter	Without Prior	With Prior
μ_j	$\bar{y}_j$	$\frac{n_j\bar{y}_j+\kappa_p\mu_p}{\kappa_p+n_j}$
σ^2	$\frac{\sum_{k=1}^{G}\sum_{j=1}^{n} z_{jk}(y_i-\bar{y}_k)^2}{n}$	$\frac{s_p^2+\sum_{k=1}^{G}[\frac{\kappa_p n_k}{\kappa_p+n_k}(\bar{y}_k-\mu_p)^2+\sum_{j=1}^{n} z_{jk}(y_i-\bar{y}_k)^2]}{\upsilon_p+n+G+2}$
σ_j^2	$\frac{\sum_{j=1}^{n} z_{jk}(y_i-\bar{y}_k)^2}{n_k}$	$\frac{s_p^2+\frac{\kappa_p n_k}{\kappa_p+n_k}(\bar{y}_k-\mu_p)^2+\sum_{j=1}^{n} z_{jk}(y_i-\bar{y}_k)^2}{\upsilon_p+n_k+3}$

2.3.3 Performance Measures

The question that then rises is which of the models derived from the EM algorithm can be chosen as the best representation of the clusters? The Bayesian Information criteria is the most used measure of performance in model-based clustering. The classification/clustering accuracy is a popular measure used for any clustering algorithm.

2.3.3.1 Bayesian Information Criteria

Bayes factor and posterior probabilities can be used to arrive at a criterion of measurement of an algorithm's performance (Kass and Raftery, 1995). Suppose there are K models G_1, G_2,..., G_K with prior probabilities $f(G_i)$, with $i = 1, 2, ..., K$. By Bayes theorem, the posterior probability of G_i given data S is proportional to the probability of observing the data S, given the model G_i;

$$f(G_i|S) \propto f(S|G_i)f(G_i). \tag{2.11}$$

In the event of unknown parameters, the law of total probability states

$$f(S|G_i) = \int f(S|\theta_i, G_i)f(\theta_i|G_i)\, d\theta. \tag{2.12}$$

The quantity $f(S|G_i)$ is called the integrated likelihood model of G_i. Bayes factor is the ratio of two integrated likelihoods, for example, of models G_1 and G_2. If the ratio $\frac{f(S|G_1)}{f(S|G_2)}$ is greater than 1, then G_1 fits the data better (Fraley and Raftery, 2000). However, the challenge with this method is that the computation of Equation (2.12) is not always easy. The Bayesian Information Criterion (BIC) is then used as an estimate of the factors;

$$2\log f(S|G_i) \approx 2\log f(S|\theta_i, G_i) - v_i \log(n) = BIC_i \tag{2.13}$$

where v_i is the number of unknown independent parameters that require estimation in G_i (Schwarz et al., 1978). The model producing the maximum BIC fits the data best. BIC is therefore not only to be used as a theoretical and systematic way to single out the best parametrization for the model but also for the choice of the number of components.

Alternatively, an approximation of the integrated likelihood, Equation (2.12), based on the classification likelihood, Equation (2.6), has been used and referred to as the Approximate Weight of Evidence (AWE) (Banfield and Raftery, 1993). This approach has however consistently underperformed in comparison with the BIC. Other criteria include an informational complexity criterion called ICOMP (Bozdogan, 1994) and an integrated classification likelihood (Biernacki et al., 2000). Unlike the BIC, these approaches have been primarily established for the choice of the number of clusters. However, they supposedly could also be extended to choosing the model (Fraley and Raftery, 2000). Moreover, the BIC is able to compare more than 2 clusters, which is not the case with significance tests and also allows compared models to not be nested. (Fraley and Raftery, 1998)

2.3.3.2 Classification accuracy

The BIC measures the performance of the algorithm based on the best fitting model. A more direct approach is the classification accuracy. This is the proportion of observations for which the most likely cluster obtained from the model is the true cluster. If majority of observations from a cluster are recovered by the model as belonging to the same cluster, then this cluster is mapped to the true cluster. The number of observations allocated into the correct cluster by the model is then counted and divided by the total number of observations (Meilă and Heckerman, 2001).

A limitation of this performance measure is that the true number of clusters as well as the observations' true clusters need to be known. This is not always the case in practice.

2.4 Prior Work

The Expectation Maximisation algorithm has been previously applied in many studies. For instance, in gene expression data, synthetic datasets were created from ovary data by preserving the mean vector and covariance matrix, but assuming a mixture model. Another dataset was a randomly resampled dataset from the ovary data. In both these datasets, the EM algorithm provided superior results and not only selected the correct

number of clusters but also the correct models for the data (Yeung et al., 2001).

The EM algorithm has also been applied to some datasets in R that have been thoroughly studied. One of which is the Faithful dataset which is a two-dimensional dataset consisting of waiting times between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA, (Azzalini and Bowman, 1990). The Bayesian information criterion is used to measure the performance of the algorithm. Majority of the BIC curves maximized at 2 clusters with spherical, ellipsoidal and diagonal clusters. However, one BIC curve maximized at 3 clusters with the ellipsoidal clusters that have equal volume, shape, and orientation (EEE) with the maximum BIC slightly higher than that of all the models (Fraley and Raftery, 2006).

Another dataset is the wreath dataset in R consisting of a simulation of 14 components from a normal mixture model in which the covariance matrices have equal size and shape but different orientations (Fraley et al., 2005). The BIC curves across different models all produced 14 components and the best fitting model was the ellipsoidal, equal volume and equal shape (EEV). These are accurate results as per parameters used in the simulation. It is worth noting that all models, including those suggesting spherical and diagonal shapes, produced the correct number of clusters even though the models themselves do not accurately describe the data in terms of shape, volume and orientation (Fraley and Raftery, 2006).

In the Faithful dataset, a simulaton of 500 noise data was added to a Poisson distribution. Previously, the majority of the models would peak at 2 clusters. However, after the addition of the noise, which can be regarded as observations that do not belong to the clusters, the majority of the models produce BIC curves that peaked at 3 clusters. Illustrations can be seen in Fraley and Raftery (2006).

2.5 Outliers

2.5.1 Outlier Identification

Understanding the effect of outliers on data requires a clear definition of what an outlier is. The accurate definition of an outlier depends on the data structure. However, some holistic definitions have been made. An early proposition was that an outlier is any observation whose removal from a sample changes the estimate of a parameter of interest by more than 10 percent (Hansen et al., 1983). Hawkins (1980) defines an outlier as an observation whose deviation from others makes an impression that it was produced by a different procedure. Another definition is that an outlier is an observation in the data which looks incompatible with the rest of the data set (Johnson et al., 2002). In literature, the natural method of identifying outliers is when the variable of interest has a particular mathematical distribution. In the case of a symmetric distribution such as the Gaussian distribution, it would be sensible to consider:

$$K = P(|a_1 - \mu| < s, |a_2 - \mu| < s, ..., |a_n - \mu| < s) = (P(|a - \mu| < s))^n, \quad (2.14)$$

where $a_1, a_2, ..., a_n$ is an independent sample of the random variable A, with a size of n from the symmetric distribution with mean μ and s is a distance from the most distant outlier. The value of s may be calculated using a distance metric of the researcher's choice. If K is different from 1 by say 0.05, then it would make sense to say that there is only 5 percent probability that one or more observations would lie farther than s. Considering the case where the distribution used is the standard normal distribution, then the candidate value of s would be such that $P(A < s) = (1 - \alpha)^{\frac{1}{n}}$, where $\alpha = 0.05$ in this case (Ghosh and Vogt, 2012).

Generating a sample from a Normal distribution with known mean and standard deviation, the z-score of the most extreme potential outlier is calculated. Subsequently, an estimate of the likelihood of the most extreme value being in the two tails is made and if this is low then the value will be declared as an outlier. The decision on what constitutes a low value is determined in hypothesis testing for whether object are outliers or not and is subjective. This approach was implemented by Benjamin Pierce (1809-1880), his son Charles Pierce (1925-2000) and several other researchers (Ghosh

and Vogt, 2012).

2.5.2 Outlier Detection algorithms

In practice, one is hardly presented with data in which the outliers are known. Therefore, it becomes of interest how outliers are practically identified in order to study their effect on the set of data and the applicable algorithms. Outlier detection has become valuable in the identification of rare events such as in intrusion detection, stock analysis and in medical diagnostics (Gao et al., 2011) as well as in marketing (Aggarwal and Philip, 2005).

Several techniques have been put in place for outlier detection, some of which are distance-based algorithms (Knorr et al., 2000). These are non-parametric algorithms which are usually local distance-based measures. An advantage of such algorithms is that they work well with large datasets (Bay and Schwabacher, 2003; DuMouchel and Schonlau, 1998). An extension on these was done in which n of the points that have the highest distance, $D_k(p)$, from their nearest k neighbours are considered to be outliers. $D_k(p)$ is the distance of the k^{th} nearest neighbourhood of a point p (Ramaswamy et al., 2000). The distance metric used in Ramaswamy et al. (2000) is the square of the Euclidean distance as it involves fewer and less expensive computations than the Euclidean distance itself. The computation of the k^{th} nearest neighbour is a 10 step complex algorithm. An example of such algorithms is the Mahalanobis distance measure, in which observations located relatively far from the centre of a distribution are regarded as outliers. This is a well-known statistical measure. The Mahalanobis distance (M_k) for each multivariate data point x_k, where $k = 1, 2, 3, ..., n$ is calculated as:

$$M_k = ((x_k - \bar{x}_n)^T V_n^{-1} (x_k - \bar{x}_n))^{\frac{1}{2}} \tag{2.15}$$

where

$$V_n = \frac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x}_n)(x_k - \bar{x}_n)^T. \tag{2.16}$$

Data points with the largest Mahalanobis distance are regarded as outliers. The drawback of such algorithms is that there might be masking effects in which the Mahalanobis distance of outliers is decreased. For instance, when a group of outliers attracts $\bar{x}_n$ and inflates V_n towards itself. Other drawbacks are swamping effects, in which the Mahalanobis distance of non-outlying data points is increased. This might happen for instance when a group of outlying data points attract $\bar{x}_n$ and inflate V_n away from the non-outlying data points (Ben-Gal, 2005).

Another method is a depth-based approach in which each observation is assigned a depth in the k-dimensional space. A depth function $D(x, F)$ provides the depth of an object x in the distribution function F. High depth implies that the object x is close to the centre of the distribution and low depth implies the object is further from the centre. The function $O(x, F) = 1 - D(x, F)$ provides the corresponding extent to which object x is an outlying. Objects whose level of outlying lies above a specified threshold λ are regarded as outliers. The choice of λ depends on the outlier function $O(x, F)$ and the size of the sample. One method used in Dang and Serfling (2010) is based on a contamination model for F;

$$F = (1 - \epsilon)G + \epsilon H \tag{2.17}$$

where G is a known ideal model distribution and H an unknown source of outliers or contaminants tending to have high levels of being outliers. The extreme values from G are false positives and those from H are true outliers. The desired threshold λ should be high enough to produce a small false positive rate;

$$(1 - \epsilon)P_G(O(\boldsymbol{X}, H) > \lambda) \approx P_G(O(\boldsymbol{X}, G) > \lambda) \tag{2.18}$$

however the threshold must also be low enough to identify the true outliers;

$$\epsilon P_H(O(\boldsymbol{X}, H) > \lambda) \approx \epsilon \tag{2.19}$$

A snag of this method is that it might become inefficient for high dimension data (Dang and Serfling, 2010).

Distribution based methods in which stochastic distributions are used to model data and the outliers are realised as a result of their relationship with the distribution chosen. However, in datasets with multidimensionality distribution-based algorithms tend to fail in outlier detection (Ramaswamy et al., 2000).

A density-based approach was introduced which used local outlier factors (LOF) that rely on the local density of the surrounding points (Breunig et al., 2000). Most outlier detection methods work on the basis that outlier detection is a binary task in which datapoints are either outliers or not, however the LOF works by assigning a degree of outlier-ness to each data point. For points that are within a cluster, the LOF is approximately 1. The LOF of points that lie on the outskirts of clusters or outside clusters is kept between bounds. These bounds are determined based on the minimum distances between a point and the nearest neighbouring points. The LOF method however has the disadvantage that since it is a density-based algorithm, its performance is dependent on the accuracy of the density estimate.

2.5.3 Dealing with outliers

In statistical approaches it is assumed that the parameters and the type of expected outliers are known. In practice this is not always the case (Barnett and Lewis, 1974). Using more empirical approaches such as distance to measure outliers, one needs to consider how far an object should be in order to be regarded as an outlier. There are also difficulties experienced in cases where data structures have both high density and sparse regions (Breunig et al., 2000). Another material question is, are all outliers important? A common phenomenon when dealing with outliers is that of influential points. Outliers that greatly affect a model are regarded as influential points. The measure of influence is usually based on how an outlier affects a regression line. Significant changes caused by outliers on the line result in the outliers being regarded as influential points (Stevens, 1984). An influential point could lead to clustering algorithms producing clusters that are more sparse than they should be. For instance, in model-based clustering, the distribution of clusters can be modelled to have longer tails than they actually have. That is, clusters would allow more coverage than they should. This then would be a problem if the resulting clusters are to be used further,

for example, for classification.

Mangiameli et al. (1996) studied the effect of imperfections such as outliers, dispersion and non-uniform cluster densities on cluster analysis. The test is based on 252 data sets using hierarchical clustering algorithms amongst others; agglomerative clustering and nearest neighbourhoods. The hierarchical algorithms were being compared to Self-Organising Maps (SOMs) which are more sophisticated algorithms for clustering. The difference in the results of the two approaches was immense. The SOMs perform consistently well whereas hierarchical approaches struggle to a significant degree. The difference in these performances could be a flag to indicate the possible effect of outliers on clustering.

There are techniques that have been proposed to deal with outliers. Shared Nearest Neighbour (SNN) is one of them. SNN works on the basis of finding how many nearest neighbours two points share. This then defines the measure of how similar the two points are. Core points are then identified, on which clusters are based (Ertöz et al., 2003). Outliers are removed since they are likely to not share nearest neighbours with other points. Density Based Spatial Clustering Application of Noise (DBSCAN) algorithms are sometimes used to deal with outliers, particularly for low dimensional data and have shown good results. The density associated with the point is measured using the count of points within a designated radius, *Eps*, from that particular point. Core points are then defined to be points with density that is above a specified threshold, *MinPts*. Outliers are then identified based on the idea of these points not being core points and/or not having core points within a specified radius (Sander et al., 1998). While this approach works in some cases, its challenge is with sparse data with clusters that have outliers. It may wrongfully identify points as outliers. These techniques are however mostly applicable in a discrete fashion. For instance, SNN clusters are recovered purely by nearest neighbourhoods. However, when using probability distributions to recover clusters such as in model-based clustering, the SNN might not be suitable since it is based on distance.

Unlike heuristic clustering algorithms, MBC algorithms are principled and theoretical methods based on well-known statistical distributions (Yeung et al., 2001). Challenges

such as outliers may be dealt with in a manner that can be supported by statistical theory. Iterated sampling and the Poisson process have been proposed for dealing with outliers in MBC algorithms. Iterated sampling is a technique that involves sampling from an entire dataset. A clustering model is built from the sampled data then the model is used for classification on the entire dataset using probabilities as generated by the model. Residuals of all points are computed. Points that fit in the model with low probability are then removed and considered as outliers. The recovery of clusters is then deployed on the remaining points (Fayyad and Smyth, 1997). Another approach to dealing with outliers in a model-based approach, is using a Poisson process to represent and model datapoints that appear to be outliers in a dataset (Banfield and Raftery, 1993). The resulting model is defined as;

$$L(\theta_1, \theta_2, ..., \theta_P, \tau_1, \tau_2, ..., \tau_P|\boldsymbol{x}) = \prod_{i=1}^{n}[\frac{\tau_0}{V} + \sum_{k=1}^{P} \tau_k f(\boldsymbol{x_i}|\boldsymbol{\theta_k})] \tag{2.20}$$

where $\tau_k \geq 0$ and $\sum_{k=0}^{P} \tau_k = 1$ and V is refered to as the hypervolume of the data region.

A simulation of 550 data points from a two-dimensional Gaussian distribution was done for the purpose of clustering in the paper(Banfield and Raftery, 1993). Another simulation was done to represent noise. Noise can be mislabeled datapoints (class noise) or errors in the values of explanatory variables (attribute noise) (Salgado et al., 2016). A model-based clustering algorithm was able to reproduce the exact cluster means; however the cluster sizes were smaller compared to the true clusters (Fraley and Raftery, 2002). However, the Poisson process can result in sampling bias. The sample taken from a Poisson process may not be the best representative of outliers of the variable of interest.

2.6 Simulations

In order to study clustering algorithms real datasets may be used. As an alternative, simulations may be used. A simulation or a simulator is defined as a device that attempts to recreate characteristics of the real world (Beaubien and Baker, 2004). Sim-

ulations are used in the study of evolutionary robotics (Jakobi et al., 1995), in dental education (Buchanan, 2001) and in the study of clustering algorithms (Bouchet and Kandrup, 1985; Bowers et al., 2006; Kwon et al., 2010). In model-based clustering, simulations are done to produce finite mixture models (Chen et al., 2004; Karlis and Xekalaki, 2003). In order to simulate finite mixture models, the following parameters need to be specified; the family of distribution to be simulated, the parameters of the distributions, the overlap between clusters, the heterogeneity or homogeniety, and the eccentricity of the clusters (Melnykov et al., 2012). The overlap between clusters is defined in terms of the sum of the two misclassification probabilities $\omega_{i|j}$ and $\omega_{j|i}$, where $\omega_{i|j}$ is the probability of mistakenly classifying an observation that belongs to the j^{th} cluster to the i^{th} cluster and $\omega_{j|i}$ is the probability of mistakenly classifying an object that belongs to the i^{th} to belong to the j^{th}cluster (Maitra and Melnykov, 2010). Homogeneity or heterogeity of clusters refers to the similarities in shape, volume, orientaion of clusters. The eccentricity describes the level of curvature of clusters. In the study of model-based clustering, finite mixture models are usually simulated from the normal family of distributions (Dasgupta, 1999; Maitra and Melnykov, 2010; Melnykov et al., 2012; Milligan, 1985). The mean vector μ_k of the normal mixture of k components is obtained as a sample of k independent realisations of the p-variate uniform hypercube with bounds specified by the user, where p is the dimension of the data. The covariance matrices are obtained as samples from the Wishart distribution with parameter p and $p + 1$ degrees of freedom (Maitra and Melnykov, 2010).

Outliers may be simulated to contaminate the finite mixtures. In Melnykov et al. (2012) the simulation of outliers in the statistical software R is stipulated. Outliers are simulated beyond a prespecified contour of a distribution. For instance, if the specified contour is set at 0.0001 then outliers would be simulated in the lower tails of density of f in Equation (2.1) such that for any outlier x, $F(x, \mu, \sigma) \leq 0.0001$. The number of times the iterations of outliers are simulated is also specified.

Chapter 3

Methodology

3.1 Data

The analysis will be carried out using two different groups of data sets. The first is simulated data on which the effect of outliers will be studied with the flexibility of tuning the data to study different structures of clusters with outliers. The second group of datasets consists of real-world data that will be used as a means of validating the results obtained from using simulated data.

3.1.1 Simulations

The simulations were done in the R statistical software version 3.5.2. The functions that were used to simulate data are the MixSim and simdataset functions from the MixSim (Melnykov et al., 2012) package.

The simulation was done as follows:

1. Set a seed of 11111
2. Set up a normal mixture model with the 2 spherical components using the parameters as follows.

- Set the dimensions of the clusters to 2. The dimensions of the clusters are chosen to avoid the complexity of higher dimensional data and to obtain graphical representations that are easier to study.
- Set the means of variable 1 and 2 in cluster 1 as $(0.50, 0.97)$ and for the second cluster $(0.77, 0.90)$. This is to create clusters that are not too far from each other so that there is a level of overlap. See sample in Figure (3.1).
- There is no covariance between the variables so as to avoid complexity within clusters. The variance are equal in each cluster and are 0.004.
- The clusters are set to be homogeneous. The homogeneity of clusters allows the study to focus mainly on the effect of outliers.
- A low level of overlap of 0.02 creates a mixture from which to easily simulate distinct clusters.
- Set the eccentricity to be 1. This produces the desired spherical cluster.

3. Set up a contour of 0.01 from the tails of the components from which outliers will be simulated. This is so that outliers are far away from the mean of the components so as least affect parameter estimates in the modelling process later.

4. Set the number of outlier iterations to 1000 to allow the simulating function enough attempts to simulate outliers.

5. Simulate 2 clusters, one from each normal mixture component, with 2 outliers in the dataset.

6. The total number of data points is 20 (excluding the 2 outliers) which are equally divided between clusters. Using the simdataset function, the datapoints are not exactly equal amounts in each cluster, however, the homogeneity between clusters keeps their sizes only a few data points out.

7. Save the simulated dataset.

8. Create a second dataset by removing the outliers from the clusters. Then save the dataset.

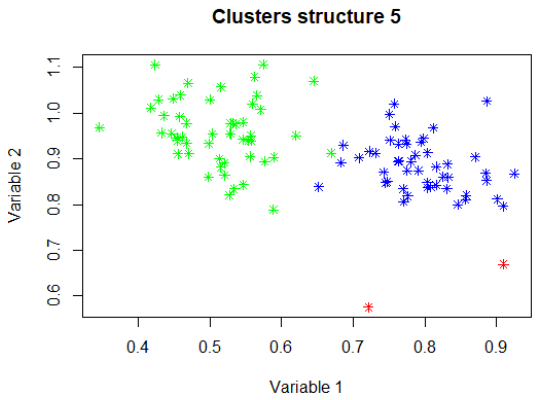


Figure (3.1) Simulation with 100 datapoints

Table (3.1) All simulated datasets

Group of simulations	Number of datasets
Set A	12
Set A without outliers	12
Set B	12
Set B without outliers	12

9. Repeat steps 5 to 8 including 5 outliers instead of 2.

Steps 5 to 9 were repeated for the following number of clusters; $(40, 60, 80, 100, 120, 140, 160, 180, 200, 220, 240)$ keeping all else constant. This was done in order to understand the effect of outliers on the EM algorithm over clusters with increasing data sizes and an increasing number of outliers. If we refer to the 12 simulations with 2 outliers as set A and the simulations with 5 outliers as B, then the total simulations will be as tabulated in Table (3.2).

3.1.2 Real data

The outcomes from studying the effect of outliers on the simulated data sets were validated using 2 real datasets. The first dataset used is a Mammography dataset which has 11183 records described using 6 continuous attributes of which will all be used in this

study. The data set is publicly available on the Open ML website and is used in Gao et al. (2011). The data had 2 classes describing the condition of the mammography test, with labels "Normal" and " Anomaly". The dataset has 10923 normal records and 260 outliers/anomalies. The Mammography dataset is relatively large compared to the sizes of the simulations used in this study. This data was used to study whether the smaller simulated datasets could provide insights into similarities and/or differences in the extremely large data size. A Lymphography dataset was used, which is smaller, with 142 normal records and 6 outliers described by 3 numeric variables and 16 categorical attributes of which will all be used in this study (Lazarevic and Kumar, 2005; Nguyen et al., 2010).

3.2 Algorithms

3.2.1 Visualisation of data

Firstly, a generation of plots was done to visualise the clusters. The clusters were distinguishable by colour of data points and the outliers were highlighted using a different colour. These plots were done for all spherical and homogenous clusters, with 2 outliers and with 5. These plots of the simulated clusters can be seen in Appendix A.

3.2.2 Outlier detection

It is not usually the case that outliers are known to exist before conducting analysis on data. Therefore, as a prelimainiary step, 2 outlier detection algorithms are run on the simulated datasets to verify if the simulated outliers will be identified by an independent outlier detection algorithm. The algorithms were also run on the 2 real datasets. In order to do this, the Mahalanobis distance algorithm as well as the depth-based algorithm for outlier detection were used. Both methods require specification of a threshold to be used in identifying outliers. The threshold used for the Mahalanobis distance was 0.99 which is the complement of the percentile used for simulations. In the depth-based algorithm, data points that have a depth lower than the threshold depth

are regarded as outliers. Therefore, a lower threshold would be able to recover outliers. After several runs, the threshold was set to be 0.06 at which the algorithm better identified outliers compared to thresholds lower than 0.06.

3.2.3 Running the EM algorithm

The EM algorithm was run over each of the 24 simulated datasets with outliers and 24 datasets without outliers as follows:

1. The EM algorithm was run on the first 12 simulated datasets with spherical homogeneous clusters and 2 outliers.
2. Thereafter, the outliers were removed from the datasets and the EM algorithm run on the uncontaminated datasets.
3. The EM algorithm was then run on the other 12 simulated datasets with 5 outliers.
4. The outliers were also removed from these datasets and the EM algorithm was run on the uncontaminated datasets.
5. The outputs from the running the EM algorithm in each of the 4 steps were the number of clusters produced, the BIC values, the fitted model and the likelihood values obtained. These would later be used for analysis.
6. The Mahalanobis distance algorithm and the depth-based outlier detection algorithms were used to detect outliers in the real datasets; the Mammography and Lymphography datasets.
7. The EM algorithm was then run on the real datasets with the outliers.
8. The outliers were removed from the real datasets and the EM algorithm was run on the uncontaminated datasets.

3.2.3.1 Using a prior distribution

In an attempt to remedy the algorithm from the effects of the outliers, the EM algorithm was rerun on the same datasets with the steps above however now incorporating a prior distribution. In order to incorporate the prior distribution, the hyperparameter in Table (2.2) were set with default values as defined in subsection (2.3.2) for to set up a simpler environment for the study;

1. Mean μ_p was set to be the mean of a whole simulated dataset. The mean of each dataset was calculated.
2. Shrinkage κ_p set at 0.01.
3. Degrees of freedom v_p were set at 3 because the data has 2 dimensions.
4. The scale of the data was set at 0 because there was no covariance between variables in the data.

3.2.4 Performance measures

3.2.4.1 Testing the effect of outliers on the number of clusters

The Bayesian Information Criterion was used to demonstrate the number of clusters that the EM algorithm produced. The BIC values were extracted from the EM algorithm models from subsection (3.2.3) and plotted against the number of clusters. The number of clusters could also be extracted from the model output.

3.2.4.2 Testing the effect of outliers on the choice of models

In order to study the models chosen by the EM algorithm to best suit the clusters with and without outliers, the model names are extracted from each run of the algorithm. The model for each dataset was extracted.

3.2.4.3 Testing the effect of the outliers on the parameter estimation

In order to study the parameter estimates, the following steps were followed:

1. The mean and variance parameter estimates from each fitted models were extracted.
2. The parameter estimates were compared against the true parameter values.

For a further study into the effect of the outliers on the parameter estimates the following steps were followed:

1. Simulate a dataset with 120 datapoints and 2 spherical homogeneous clusters each with 60 datapoints. All other parameters remain as used previously.
2. Simulate outliers at multiple standard deviations from the mean of the clusters. The standard deviation used were the set $(1, 1.25, 1.5, 1.75, 2)$ in the respective order.
3. Fit the EM algorithm to the datasets.
4. Extract and study the mean estimates and covariance matrices.

Chapter 4

Analysis

4.1 Outlier detection

Before considering the effect of the outliers on the clusters and the EM algorithm, it is important to understand whether an independent outlier detection algorithm would be able to identify outliers in the simulated dataset. It is also essential that the outlier generation algorithm does not function in the same way as the outlier detection algorithm. This is in order to avoid trivial results and also for the results to be unbiased to the outlier detection algorithms.

The algorithms used in this study are the Mahalanobis distance and the depth-based outlier detection algorithms. In Figure (4.1) and (4.2) are 3 of the simulated datasets, each with the resulting output from the Mahalanobis distance and the depth-based algorithm, respectively, in the form of a plot to the left. The first column of the figures shows the simulated data sets and the second column of figures indicates the resulting outputs from running the outlier detection algorithms, for each of the clusters in the first column.

As seen in Figure (4.1), in no case does the Mahalanobis algorithm create falsely positive outliers. However, there are cases in which false negatives are realised where outliers are recognised as normal data points. This could be because the threshold of

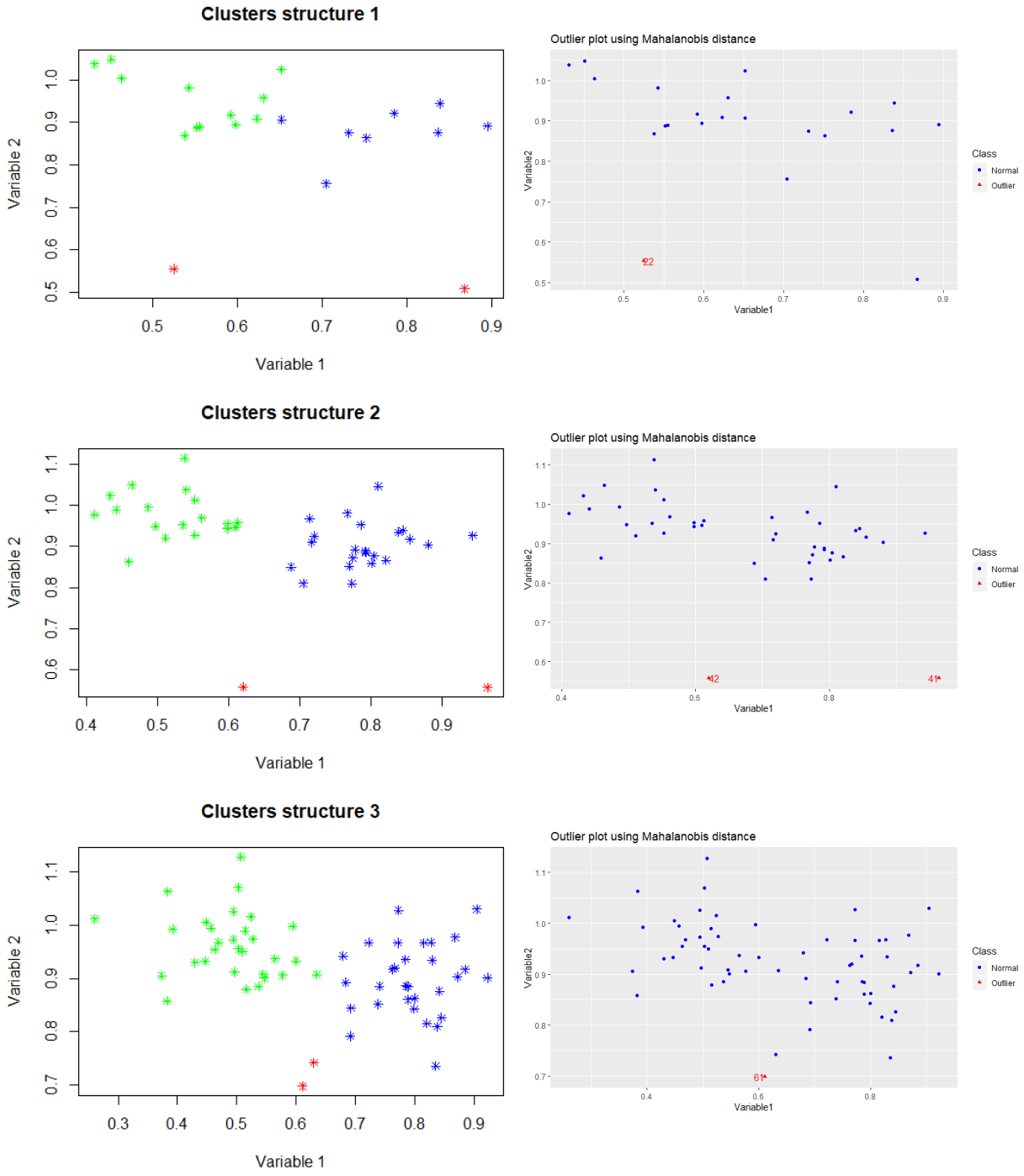


Figure (4.1) Simulated datasets and the corresponding Mahalanobis distance outlier detection algorithm results.

the Mahalanobis distance algorithm is based on the emperical distributions of mixture models instead of the theoretical distributions.

In Figure (4.2), the figures on the left are 3 simulated datasets with 2 clusters and 2 outliers in each dataset. The clusters are distinguished by colour, one in green and the other in blue. The outliers are the datapoints in red. To the right of each dataset is the corresponding output from the depth-based outlier detection algorithm identifying each data point as either an outlier or not an outlier. Points that are identified as outliers are in red and those that are not identified as outliers are in blue. The depth-based outlier detection algorithm produces results for all datasets starting at a threshold of 0.06. Attempts were made to recover outliers at lower thresholds such as 0.04, for which the unsuccessful results may be seen in Appendix C. The depth-based algorithm struggles more in identifying the outliers. This is illustrated in Figure (4.2), where some normal data points are regarded as outliers and outliers are unidentified even in cases where one should be in suspicion of them. An example would be the data set in Cluster Structure 1 where the outlier in the bottom right of the simulated dataset is identified as a normal point by the depth-based outlier detection algorithm.

When comparing Figure (4.1) and (4.2), the Mahalanobis distance algorithm produced slightly better results in that it left fewer outliers unidentified in all the 3 data sets.

Since the outlier detection algorithms are able to recover some of the outliers in the simulated datasets, the proceeding section studies how these outliers affect the expectation maximization algorithm.

4.2 Results using the EM algorithm

This section studies the results obtained from running the EM algorithm on a single simulated dataset and investigates how these are indicative of the effect of the outliers on the algorithm's ability to recover the number of components, the structure of the

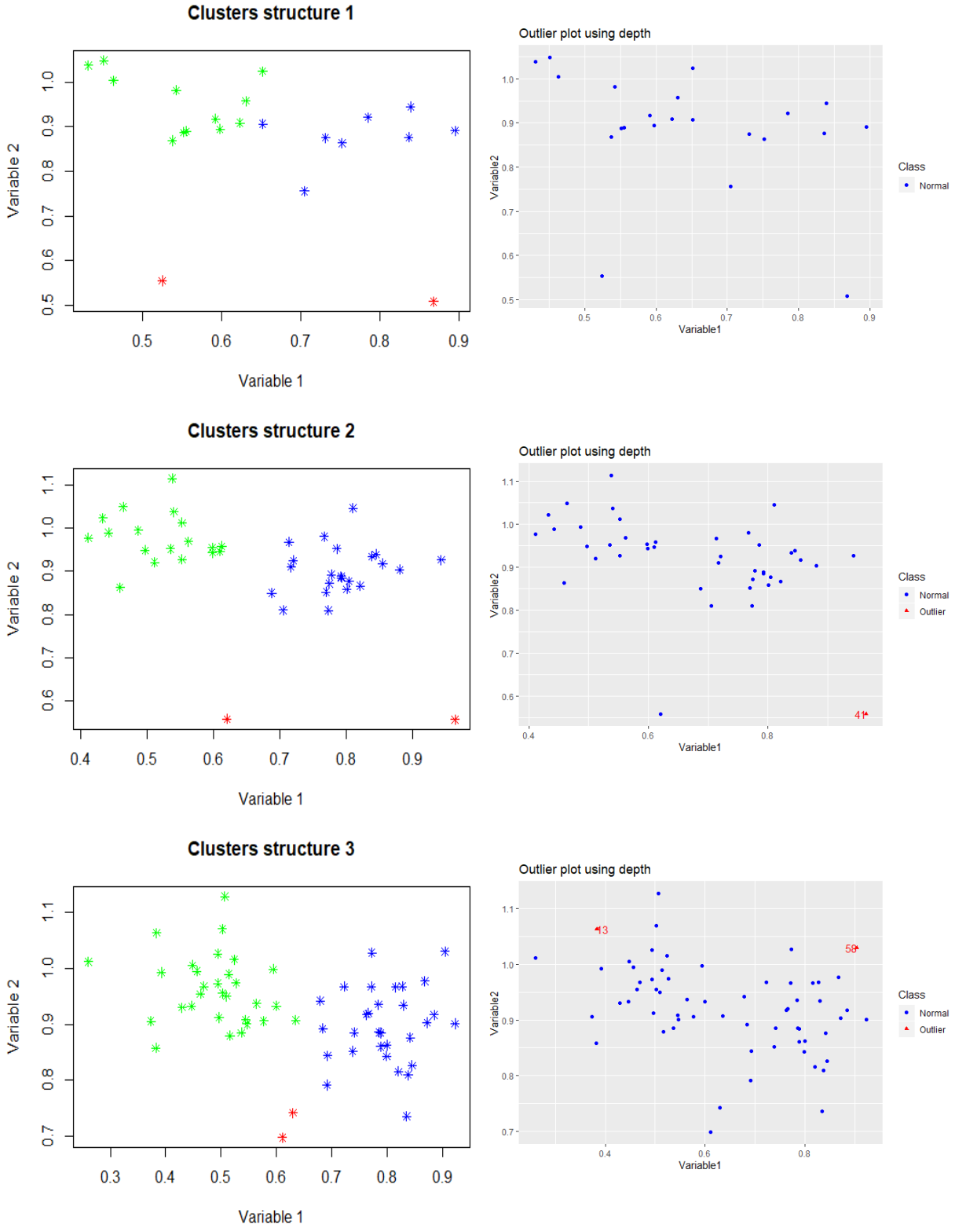


Figure (4.2) Simulated datasets and the corresponding output from the depth-based outlier detection algorithm.

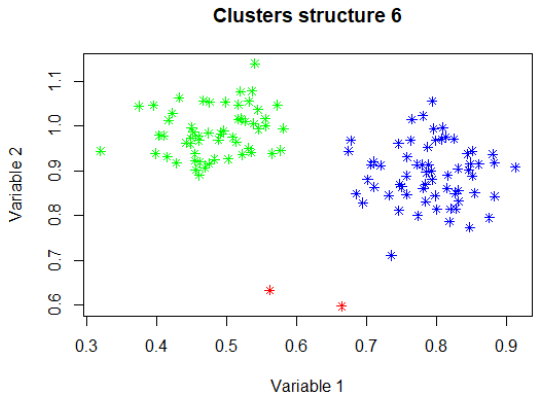
components and the parameter estimates of the statistical models fitted on the components.

Figure 4.3(a) shows a plot of a simulation from a mixture of 2 spherical and homogeneous normal distributions. This simulation consists of one cluster of size 56 in a green colour and another of size 64 in blue colour, the two outliers are in red. The cluster sizes of the two homogeneous clusters are approximately equal. The mean vectors of the distributions are $(0.50, 0.97)$ for distribution from which the green datapoints are simulated and $(0.79, 0.90)$ for the blue dataset. The covariance matrices are identical and equal to:

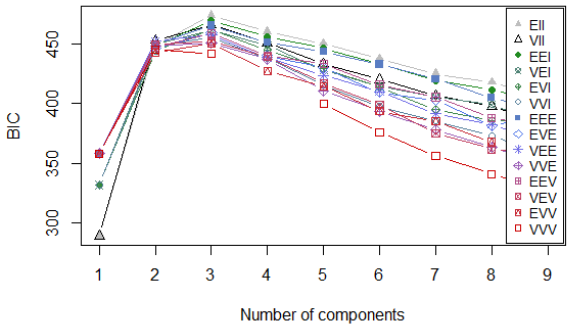
$$A = \begin{pmatrix} 0.004 & 0 \\ 0 & 0.004 \end{pmatrix}.$$

The BIC plots obtained from running the EM algorithm on this set of clusters is shown in Figure (4.3)(b). As seen in the BIC plot, multiple models have been fitted to the simulated dataset, these are indicated by the multiple curves distinguished by different symbols as indicated by the legend. According to practice, the optimal number of clusters in the data is the number of clusters to which the addition of another cluster does not result in better modeling of the data (Bholowalia and Kumar, 2014). In this case, this implies choosing the number of clusters after which the BIC value does not change significantly. However, the choice of the number of clusters is not always at the maximum value of the BIC. Indeed, extracting the optimal number of clusters from running the EM algorithm, the algorithm recovered 3 clusters. The classification plot in Figure 4.3(c) evidently shows that the EM algorithm has grouped the outliers into a cluster of their own, hence there are 3 clusters recovered from the data. However, some models such as the VVV (ellipsoidal, varying volume, shape, and orientation) also called the unconstrained model, have their maximum BIC value at 2 clusters. This then means that assuming ellipsoidal clusters with varying volume shape and orientation, the mixture contains only two clusters.

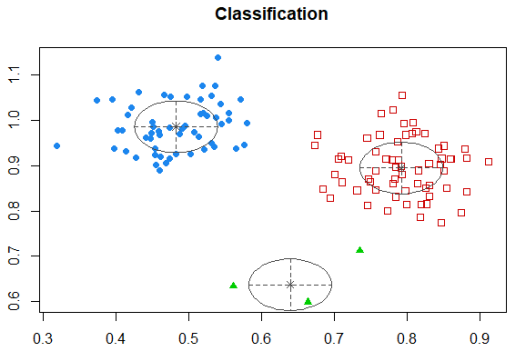
Considering the same data set without outliers, it is clear all models produce BIC curves that maximise at 2 clusters as seen in Figure (4.4). It is also clear from the classification plot in Figure 4.4(b) that there are only two clusters and all data points are correctly classified.



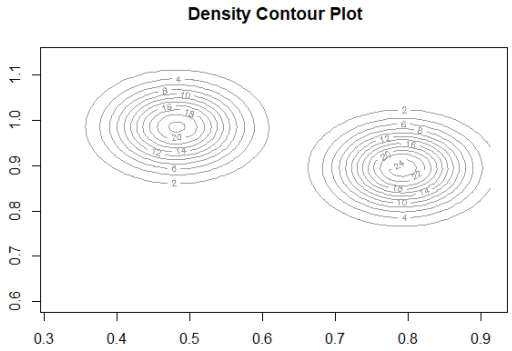
(a) The simulated dataset



(b) The BIC plot

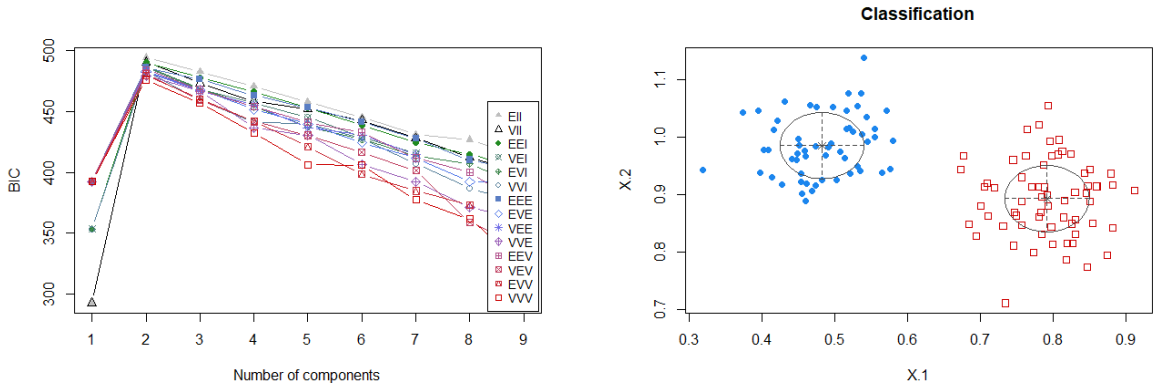


(c) The classification plot



(d) Density contour plot

Figure (4.3) Results of running the EM algorithm on a simulated data set



(a) The BIC plot (b) Classification plot

Figure (4.4) Results from running the EM algorithm on the simulated dataset in Figure 4.3(a) without outliers.

Now considering the structure of the clusters, according to the EM algorithm, the model that produces the loglikelihood that maximises the BIC as indicated in Equation (2.13), is the best fitting model. One may physically identify this model as the model with the highest BIC value after which no significant changes are seen. However, this is not always clear from graphical representations. The EM algorithm produces the EII (spherical, equal volume) model as the best model with a loglikelihood of 258.45. This is precisely the structure of the true mixture of distributions. In addition, the density contour plot in Figure 4.3(d) demonstrates spherical clusters. On the other hand, the loglikelihood for the same dataset without outliers, is 261.4609 which resulted in the same chosen model, EII.

The mean parameter estimates of the 3 simulated clusters are $(0.48, 0.99)$, $(0.79, 0.90)$ for the two mixtures and $(0.64, 0.64)$ for the cluster with outliers. Comparing these estimates with the true parameter values from the true mixture which are; $(0.50, 0.97)$ and $(0.79, 0.90)$ respectively. It is clear that the EM algorithm has produced fairly accurate parameter estimates. The estimated covariance matrices are all identical including that of the component with outliers. This is surprising because as seen in the classification plot, the data with outliers seems to be more sparse than the data without outliers and hence one would expect higher variance values. This covariance matrix is

estimated as:

$$\hat{A} = \begin{pmatrix} 0.0033 & 0 \\ 0 & 0.0033 \end{pmatrix}$$

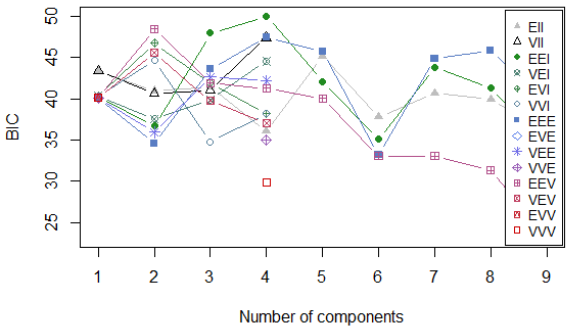
The same dataset without outliers as used in Figure (4.3) has produced mean parameter estimates of $(0.48, 0.99)$ and $(0.79, 0.89)$ in respect of the true parameter values. These are relatively the same estimates obtained from running the EM algorithm on data that has outliers. The corresponding covariance matrices for the two components are the same and are estimated to also be equal to $\hat{A}$.

The results from running the EM algorithm on this data set therefore indicate that, in the presence of a mixture with two components, the algorithm identifies a mixture containing 3 components when there are outliers, and only 2 when there are no outliers. Regardless of this, it is also evident that the choice of models and the parameter estimates are the same when the algorithm is run on a data set with outliers, and when the outliers are removed.

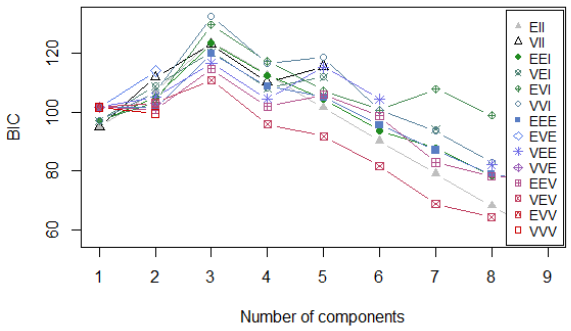
The next sections study the findings from this section in detail. For each simulated dataset the BIC plots and the fitted models as well as the parameter estimates of the probability models are collected and analysed.

4.3 Effect of outliers on number of clusters

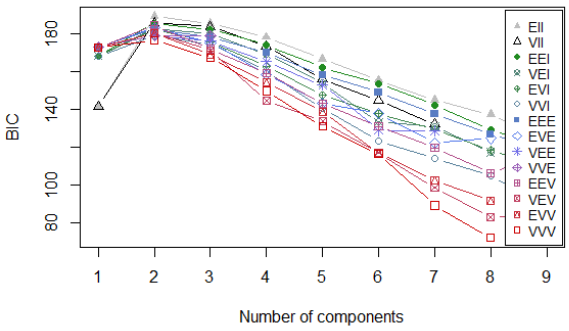
This section studies the effect of the outliers on the ability of the EM algorithm to identify the number of clusters using the BIC. Figure (4.5), shows the BIC curves obtained from running the EM algorithm on spherical and homogeneous clusters with a total number of datapoints ; 20, 40, 60 and 80 per cluster, respectively. These simulations can be seen in Appendix A. It is important to note that since the clusters are homogeneous, increasing the total number of datapoints implies that the cluster sizes are also increasing. Figure (4.6) shows the BIC curves obtained from running the EM algorithm on the same datasets without outliers. The model output of the number of clusters will be extracted and studied.



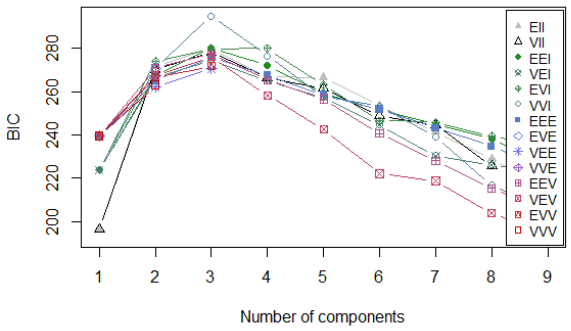
(a) BIC curves for dataset with 20 points



(b) BIC curves for dataset with 40 points



(c) BIC curves for dataset with 60 points



(d) BIC curves for dataset with 80 points

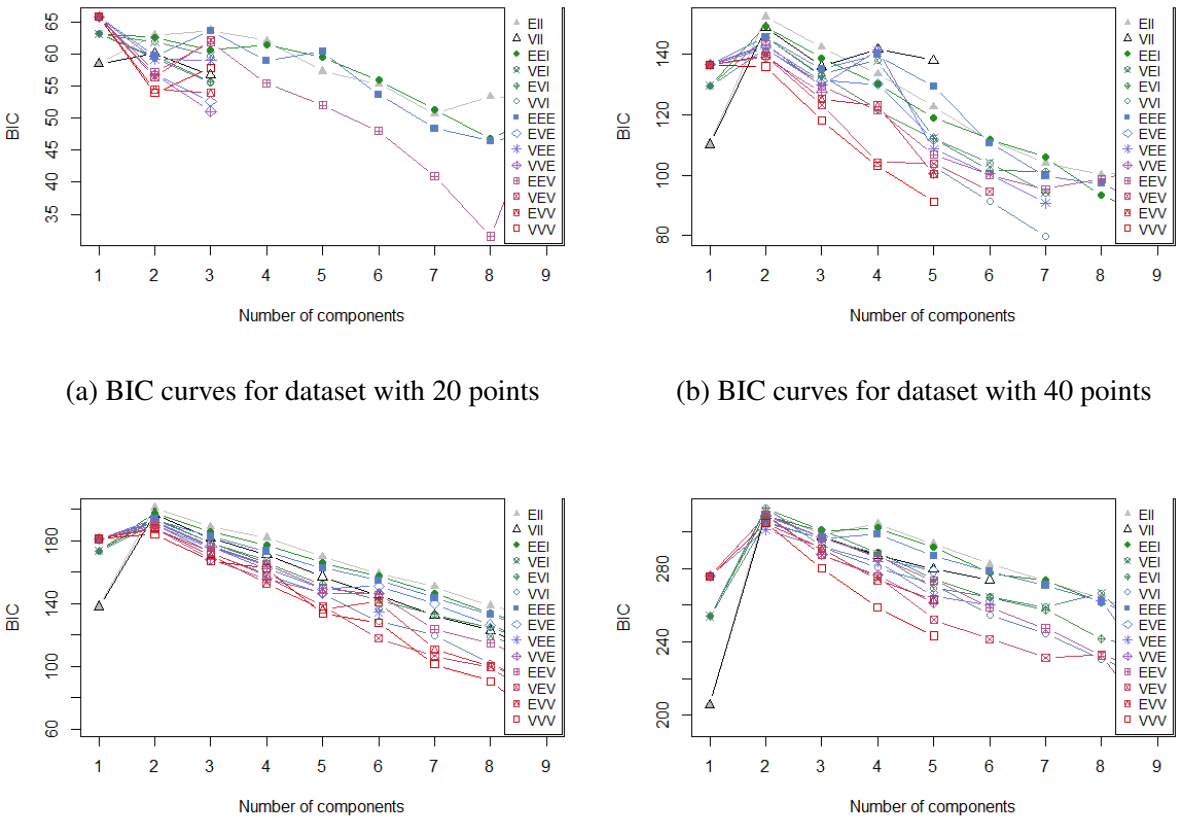
Figure (4.5) BIC curves for the first 6 simulated Spherical and Homogeneous clusters with (a) 20, (b) 40, (c) 60 and (d) 80 data points respectively and 2 outliers in each case.

In Figure (4.5), where there are 20 datapoints, the BIC curves produced by the models are very different from each other. For instance, at 2 clusters, some models are obtaining their maximum BIC values whereas some are at their minimum. However, this starts to change as the number of data points increases. The BIC curves start behaving in a similar manner as the number of data points in each cluster size increases, where the peaks of the different models coincide. As the cluster size increases, the choice in the number of clusters across the models alternates between 2 and 3.

Where the number of data points is 20, models such as the VII and VEE no longer produced BIC results for more than 4 clusters. The VII model still produced no results when the data size was doubled to 40. The VII model has spherical clusters with unequal volumes. Recalling that the simulated clusters are homogeneous however the cluster sizes are only approximately the same, then the VII model would be the best fit. Therefore, the VII model would not produce any results for high numbers of clusters since there are not many clusters in the datasets.

In Figure (4.6) where no outliers are present in the datasets, the BIC curves of the models in the dataset with 20 datapoints have a common behaviour. The majority of the BIC curves start off at a maximum of 1 cluster implying that there is possibly 1 cluster in the data as there is no significant change in BIC curves of the models such as EEI and EEV. The BIC curves decrease as the number of clusters increases. As the cluster size increases, the behaviour the BIC curves change in a similar manner. As seen in all the BIC curves with clusters that had 40 data points and above, the BIC curves peak only at 2 clusters as the first decisive maxima. Therefore, identifying the correct number of clusters in the datasets.

In the small dataset (with 20 datapoints), we can see that majority of the models fail to produce BIC outputs for more than 3 clusters. For instance, the EVE, VVE and the VII models produce no values after 3 clusters. Even where the datasets have 40 datapoints, models such as the VII and VVV still do not produce BIC values for clusters greater than 5. This is as a result of the EM algorithm collapsing. It would make sense for the algorithm to fail to produce results where the datasets are small such as in the case of 20 and 40 datapoints. The EM algorithm collapses when the covariance matrix is



(a) BIC curves for dataset with 20 points

(b) BIC curves for dataset with 40 points

(c) BIC curves for dataset with 60 points

(d) BIC curves for dataset with 80 points

Figure (4.6) BIC curves of the simulated datasets with spherical and homogeneous clusters of sizes 20, 40, 60 and 80 respectively, without outliers.

singular. An example of causes of the singularity is if a cluster has only 2 data points. Since the data is 2 dimensional the minimum number of data points within a cluster is 3. Therefore the algorithm is likely to collapse when producing too many clusters with only a few datapoints in each, for instance a cluster with 2 data points.

Now considering the observed effects of the outliers on the EM algorithm, in Figure (4.5) one of the effects is seen through the BIC on the instability of the BIC curves. The peaks of the BIC curves alternate between 2 and 3 as the cluster size increases. A significantly different observation is made when the EM algorithm is run on the

same dataset excluding the outliers. The BIC curves are seen in Figure (4.6) where the curves peak at 2 clusters consistently as the data size increases.

A realisation of the collapse of the algorithm is made where no BIC results are produced beyond a certain point for some models especially for the smaller cluster sizes. For example, in Figure (4.5a) a majority of the models collapsed and do not produce output for numbers of clusters above 4. The same applies to numbers of clusters beyond 3 and 5 in Figure (4.6a) and Figure (4.6b). It is worth noting that in the presence of outliers the point at which majority of the models fail to produce BIC is higher than that for clusters without outliers in all cases.

Another realisation is that, there is a set of 4 models that are able to produce BIC results after all other models have collapsed. These can be seen when the algorithm is run with outliers (Figure (4.5)) and without outliers (Figure (4.6)). These are the EEV (ellipsoidal, equal volume and equal shape), EEE (ellipsoidal, equal volume, shape, and orientation), EEI (diagonal, equal volume and shape) and EII (spherical, equal volume). Therefore, the outliers do not seem to affect the ability of the EM algorithm to fit these models.

It is also worth noting the homogeneity in the BIC curves as the number of clusters increases. In both cases with and without outliers, the BIC curves across all models tend to be very similar, both in the choice of the number of clusters and the changes across the different numbers of clusters. That is, models assuming different cluster structures are behaving the same in terms of the optimal number of clusters and the change in behaviour of the BIC as the number of clusters increase. Therefore, the BIC may not necessarily be informative in determining the best fitting model. In the following section, the likelihood function is used to recover the best fitting models for the data.

4.4 Effect of outliers on the structure of the clusters

In this section, the effect of the outliers on the ability of the EM algorithm to identify the cluster structure is studied. As highlighted in section (4.3), the algorithm choses the best fitting model to be the model whose loglikelihood maximises the BIC in each dataset. Table (4.1) shows the size of the datasets and the best fitting models according to the EM algorithm when run on spherical homogeneous clusters. Each dataset includes models fit in the existence and absence of outliers. This analysis uses the first 10 simulated datasets in Appendix A, with 2 outliers.

Table (4.1) Models fit by the EM algorithm on two spherical and homogeneous clusters with 2 outliers and without the outliers.

Dataset size	20	40	60	80	100	120	140	160	180	200	220	240
With outliers	EEV	EVI	EII	EVI	EII	EII	EII	EII	VII	EII	EII	EII
Without outliers	XXX	EVI	EII	EVI	EII	EII	EII	EII	VII	EII	EEI	EII

With outliers, the EEV model has the best fit on the dataset with 20 datapoints. The EEV model assumes ellipsoidal clusters with equal volume and shape. In the dataset with 40 and 80 datapoints, the EVI model has the best fit on the clusters. The EVI describes a dataset with diagonal clusters with equal volume and varying shape. As the cluster size increases, the EII model has the best fit for the clusters. The EII is the closest representation of the true clusters since it describes spherical clusters with equal volume. However, the VII model best fit the dataset with 180 datapoints. This model also assumes spherical clusters; however it assumes clusters with different volumes.

In the absence of outliers, the XXX model, which assumes a single component onto ellipsoidal normal clusters, is fit on the dataset with 20 datapoints. As the data size increases, the EM algorithm fits the same models as with the datasets with outliers. The EVI model is the best fit for the lower sized clusters with 40 and 80 datapoints. Then the EII model is considered best fit in datasets with 100 datapoints and above. However, similar to the datasets with outliers, the VII model is fitted on the dataset

with 180 datapoints.

Now considering the effect of the outliers on the ability of the EM algorithm to recover the cluster structure. In Table (4.1), there is little difference between the models fitted on the datasets with outliers and those without outliers. The algorithm performs fairly well in identifying the shape on the clusters (EII (spherical, equal volume)) even when the outliers are present. Therefore, it looks like the effect of the outliers on the algorithm's ability to read the structure of the clusters is minimal. The same models are fitted on the clusters in the presence and absence of outliers. Therefore, the EM algorithm seems to be unaffected by the outliers when fitting models to the clusters.

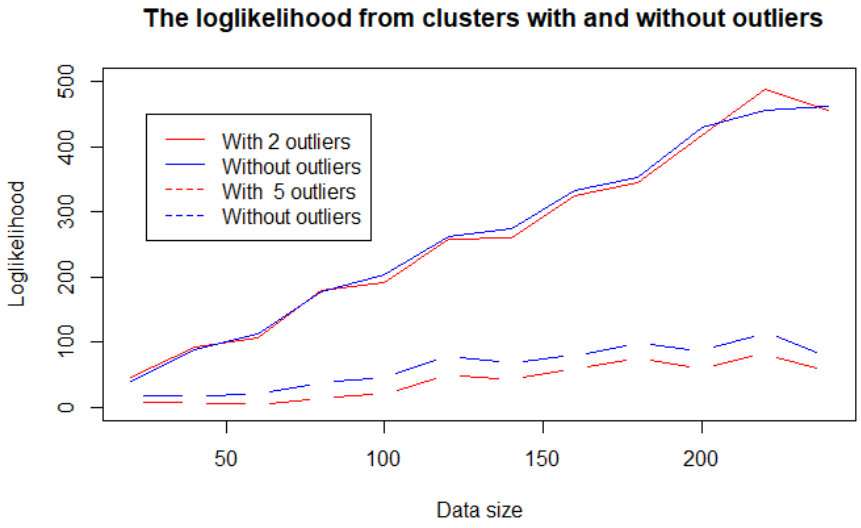


Figure (4.7) The loglikelihood from models fitted on datasets with 2 spherical homogeneous clusters of approximately equal sizes, with and without outliers.

Another important observation is the difference in the loglikelihood functions when there are outliers and when there are no outliers. The loglikelihoods are shown in Figure (4.7). The figure shows lines joining the loglikelihood obtained as a result of each model in Table (4.1) including the datasets with 220 and 240 datapoints, that is the first 12 simulated datasets in Chapter 3. The red lines indicate the loglikelihood with outliers (2 outliers for the straight line and 5 outliers for the segmented line) and likewise,

the blue lines are for the same clusters without outliers. The algorithm seems to reach higher loglikelihood values when there are no outliers in the data compared to when they are there. This is seen by the blue lines lying above the red lines. This difference seems to increase when the number of outliers is increased. This can be seen in the segmented lines. Therefore, this could imply that the existence of outliers reduces the likelihood function of the finite mixture model. This is a reasonable effect since outliers are likely to result in clusters being more spread out. Therefore the density reduces for all values and thus the likelihood is also reduced.

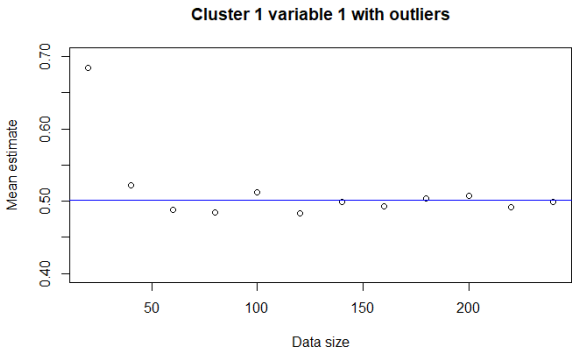
Thus far, the effect of the outliers on the algorithm seems to be more visible on the ability of the algorithm to recover the number of clusters, than on the algorithms ability to recover the structure of the clusters.

4.5 Effect of outliers on parameter estimates

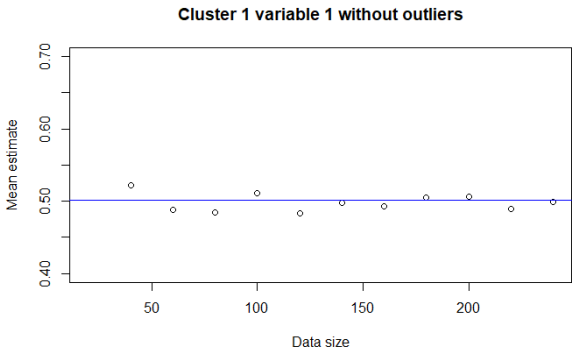
The EM algorithm recovers components of the mixture model with estimates of the parameters of each distribution. In this section, the effect of outliers on these parameter estimates is studied. In Figure (4.8), the mean estimates of variables in the recovered clusters are plotted together with the true parameter values indicated with a horizontal blue line.

As seen in the the first two plots in Figure (4.8), the estimates seem to be slightly affected by the presence of the outliers when there are fewer outliers. It is only for the first two datasets where the parameter estimates vary significantly when there are outliers.

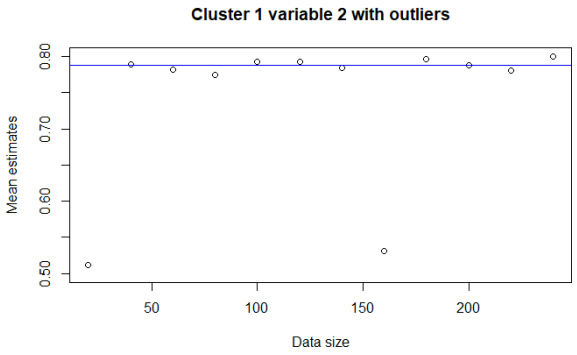
It might be worth noting that the outliers' contribution in the data is between 0.8 (in the largest dataset) and 9 percent (in the smallest dataset). Even though this is the case, the bad parameter estimates seem to not depend on the proportion of the outliers in the data. One would expect that in datasets where the outliers have a high proportion, then the effect of these outliers would be more visible than when the outliers take up a small



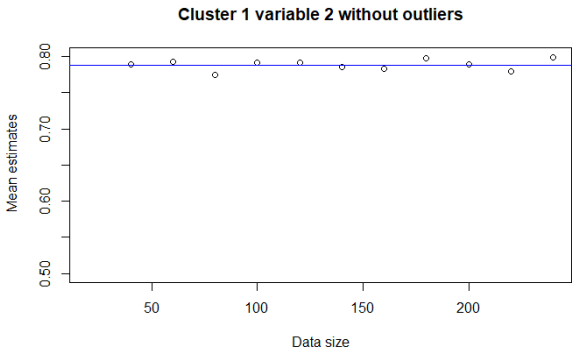
(a) Mean estimates with outliers



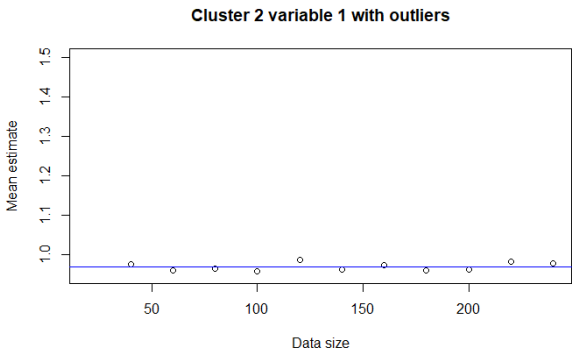
(b) Mean estimates without outliers



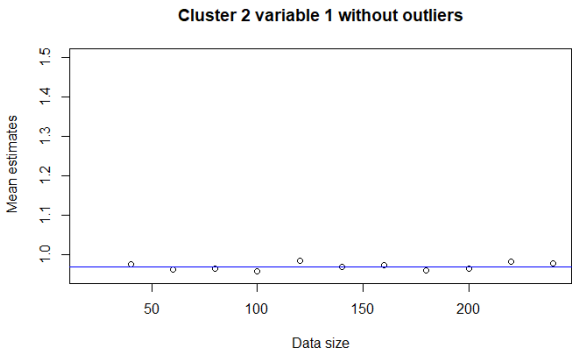
(c) Mean estimates with outliers



(d) Mean estimates without outliers

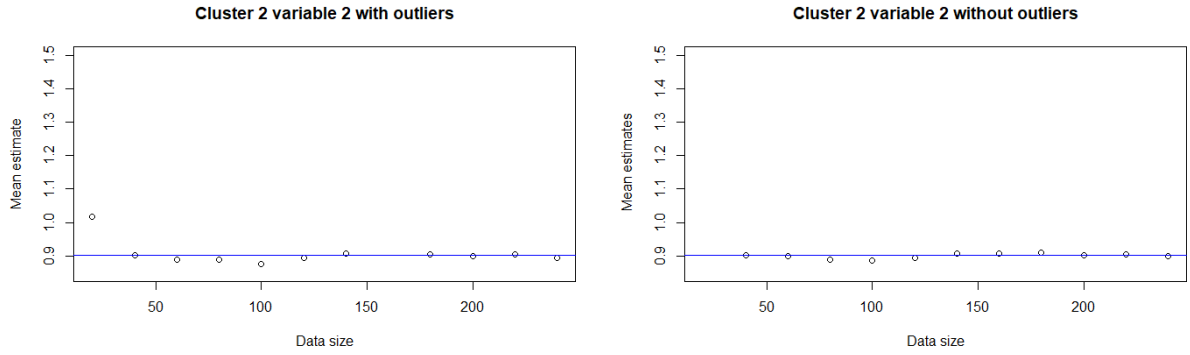


(e) Mean estimates with outliers



(f) Mean estimates without outliers

Figure (4.8) Mean estimates for spherical homogeneous clusters with 2 outliers vs without the outliers.



(a) Mean estimates with outliers (b) Mean estimates without outliers

Figure (4.9) Mean estimates for spherical homogeneous clusters with 2 outliers vs without the outliers continued.

propotion of the dataset. However, this does not appear to be the case, bad estimates occur in the larger clusters, even when better estimates occur at smaller cluster sizes.

Table (4.2) Variance estimates of variable 1 in clusters recovered by the EM algorithm on the data sets with 20, 40, 60, 80,100,120,140 and 160 datapoints with 2 outliers.

Data size	20	40	60	80	100	120	140	160
Cluster 1	0.0128	0.0038	0.0055	0.0067	0.0046	0.0033	0.0049	0.0039
Cluster 2	0.0128	0.0037	0.0055	0.0016	0.0046	0.0033	0.0049	0.0039
Cluster 3	0.0128	0.0296	-	0.0268	-	0.0033	-	0.0039
Cluster 4	0.0128	-	-	-	-	-	-	-

In Table (4.2), the variance estimates of variable 1 in the datasets is tabulated for increasing data sizes with 2 outliers. Even though there were only 2 clusters in the data, parameters estimates for cases were the algorithm recovered more than 2 clusters are also included in the table. Table (4.3) shows the variance estimates for the same datasets without the outliers. The outputs of variable 2 are showing similar characteris-

tics as variable 1 as described below and are included in Appendix B. Interestingly, the algorithm identified that there is no covariance between the variables and hence only the variance estimates are tabulated. Bearing in mind that the true variance of the variables in the mixture models is 0.004, we study the estimates from the EM algorithm.

Table (4.3) Variance estimates of variable 1 in models fit on the data sets with 20, 40, 60, 80,100,120,140 and 160 datapoints without outliers.

Data size	20	40	60	80	100	120	140	160
Cluster 1	0.0170	0.0034	0.0046	0.0067	0.0039	0.0033	0.0043	0.0038
Cluster 2	-	0.0033	0.0046	0.0016	0.0039	0.0033	0.0043	0.0038

As seen in the two tables, only a maximum of 2 clusters is recovered by the algorithm when there are no outliers. These are the clusters recovered by the EM algorithm from the data. This attests to the results in section (4.3) on the effect of the outliers on the number of clusters where only two clusters were identified. The variance estimates in Table (4.3) where the datasets exclude outliers are closer to the true variance compared to those in Table (4.2). However, the difference in the variance estimates in the two tables is very small.

In majority of the datasets, the variance estimates are constant for all clusters. Therefore, the algorithm seems to be able to recover the homogeneity of the cluster. This is true even when the number of clusters is overestimated. For example, in the first dataset in Table (4.2), even though the number of clusters is 4, all 4 clusters are homogeneous although the variance is overestimated. However the EVI model produced different variance estimates in that the variances were not constant within all clusters. This is understandable since the model fits data that has diagonal clusters of equal volume and varying shape.

4.6 Use of a Prior Distribution

The prior distribution has been used in sparse clusters were the EM algorithm could collapse due to singularity. Outliers can cause sparse regions within clusters. In this section, the results from running the EM algorithm with the use of a prior distribution are obtained. The effect of outliers under this condition is therefore studied in detail.

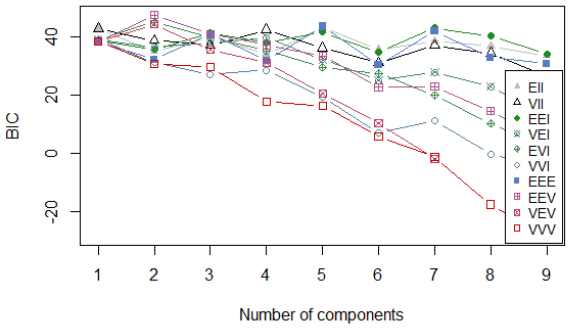
Similar to the previous sections, the BIC will be used to study the choice of the number of clusters when using the EM algorithm but now with the use of prior distribution. The model with the loglikelihood that maximises the BIC will be considered as the best fitting model. This section will conclude by investigating the parameter estimates obtained by running the EM algorithm using the prior distribution.

4.6.1 Number of clusters using BIC

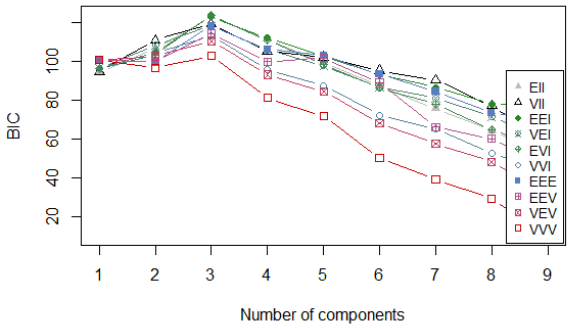
The same datasets used in the previous sections are studied in this section and the BIC plots from running the EM algorithm on these datasets using the prior distributions are shown in Figure (4.10) with outliers and in Figure (4.11) without outliers.

In Figure (4.10) we see more interesting results in the smallest/sparse clusters. The variation in the behaviour of the BIC is significantly reduced compared to when the prior distribution was not used. The existence of multiple peaks on the models is also removed, so that the curves are smoother and behaviour across the models is fairly uniform. The majority of the models identify 3 clusters from the datasets, from sparse to more dense clusters.

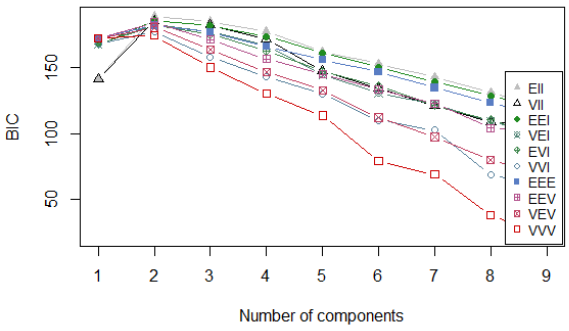
When the outliers are removed from the dataset, the resulting BIC plots are interesting. With the use of the prior distribution, it is evident that some BIC curves maximise at 3 clusters. This is contradicting what was observed when the prior distribution was not used. The BIC curves would consistently peak at 2 clusters across the datasets. This could be a result of the EM algorithm overfitting the dataset. As seen in Table (2.2), parameter estimates from the EM algorithm when using the prior distribution are more



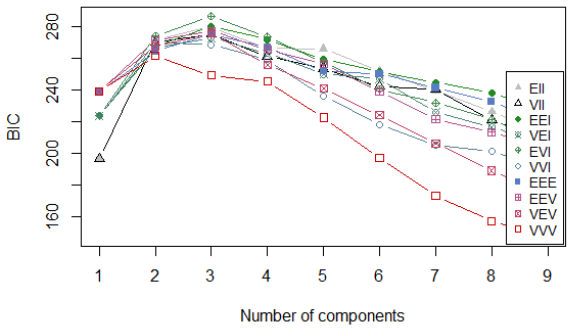
(a) BIC curves for dataset with 20 datapoints



(b) BIC curves for dataset with 40 datapoints

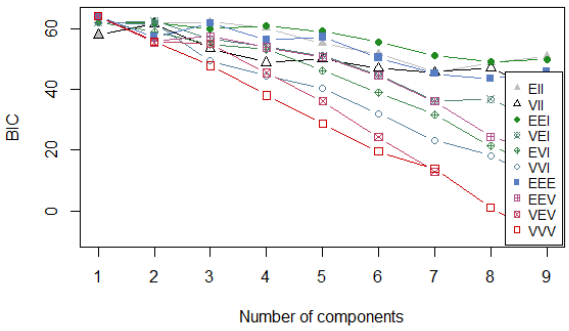


(c) BIC curves for dataset with 60 datapoints

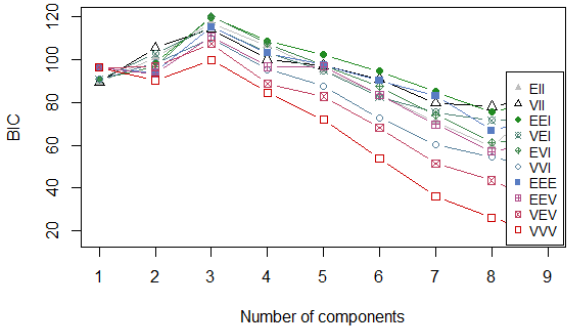


(d) BIC curves for dataset with 80 datapoints

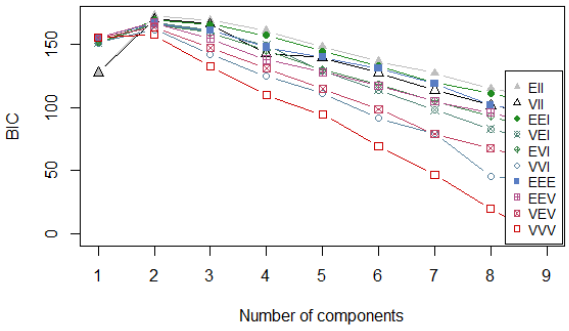
Figure (4.10) BIC of spherical homogeneous clusters with outliers using prior distribution for data sizes with 20.40,60 and 80 data points.



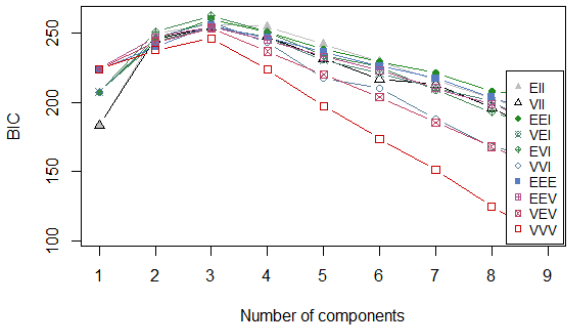
(a) BIC curves for dataset with 20 datapoints



(b) BIC curves for dataset with 40 datapoints



(c) BIC curves for dataset with 60 datapoints



(d) BIC curves for dataset with 80 datapoints

Figure (4.11) BIC of spherical homogeneous clusters without outliers using prior distribution for data sizes with (a)20, (b)40, (c)60 and (d)80 data points.

detailed compared to when the prior distribution is not used.

Previously, when the EM algorithm was run without using a prior distribution, majority of the models would fail to produce results for higher numbers of cluster in both cases when outliers were present and absent. However, with the use of the prior distribution, that the algorithm hardly collapses and majority of the models are still able to be fit on the datasets.

4.6.2 Strucuture of clusters

Similar to the previous section, the effect of the outliers on the EM algorithm's ability to recover the cluster structure with the use of the prior distribution is studied. The study is done by extracting the names of the fitted models and analysing the log-likelihood from fitting the model with and without outliers. The number of outliers considered is 2.

Table (4.4) Models fit by the EM algorithm with the use of the prior distribution in the presence of 2 outliers and absence of outliers for clusters of different sizes.

Data size	20	40	60	80	100	120	140	160	180	200	220	240
With outliers	EEV	EVI	EII	EVI	EII	EII	EII	EII	VII	EII	EII	EEI
Without outliers	XXX	EVI	EII	EVI	EII	EII	EII	EII	VII	EII	EEI	EII

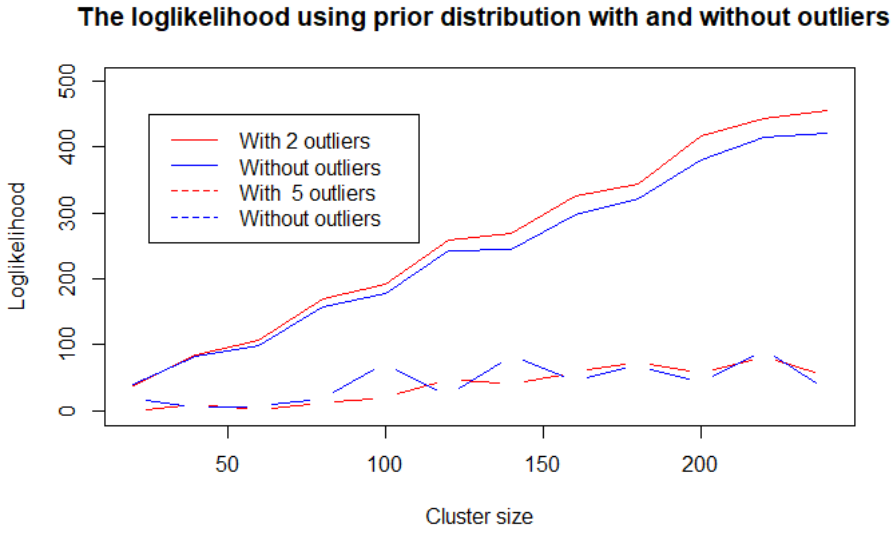


Figure (4.12) The loglikelihood of spherical homogeneous clusters with and without outliers with a prior distribution.

Table (4.4) shows the models fit on the datasets using the EM algorithm with a prior distribution for increasing sample sizes starting from 20 to 240. Comparing the models

fit when the outliers are present and when they are not, it seems there is not much of a difference. The only difference is with the smallest data size with 20 datapoints, where the EEV model was fit in the presence of outliers and XXX in the absence of outliers. The same observation was made when the EM algorithm was fit without the use of a prior distribution.

Figure (4.12) shows the loglikelihood from running the EM algorithm with the use of a prior distribution on the same datasets as in Section (4.4) where the prior distribution was not used. Similar to the previous runs, the effect of the outliers on the loglikelihood is still observed. The difference in the loglikelihood from which the models are produced is evident. However, it seems like the difference occurs in the opposite direction compared to when the prior distribution was not used. Previously, the loglikelihoods when outliers were absent seemed to be higher than when they were present, however, the opposite seems to true when using the prior distribution. The loglikelihood is higher when the outliers are present in the clusters. This is an interesting result. In Section (4.6.1), the results from the EM algorithm with the prior distribution indicated that the algorithm seems to be identifying 3 clusters instead of 2 and this could be explained by overfitting.

4.6.3 Parameter estimates

As with the previous case, the EM algorithm mean estimates using the prior distribution are shown in Figure (4.13). These results are obtained from the same dataset as with the case when the prior distribution was not used. The mean estimates seem to deviate the most from the true value when the outliers are removed. This is in contrast to the previous observation when the prior distribution was not used, where the mean estimates were better when the outliers were removed. For instance, overall the parameter estimates in Figure (4.9) are much closer to the true mean compared to parameter estimates in Figure (4.13) where estimates in data sets with 140, 160 and 240 are further away from the true value. This can be seen in the second figure in Figure (4.13).

The variance estimates from running the EM algorithm with and without outliers using the prior distribution are tabulated in Table (4.5) and (4.6) respectively.

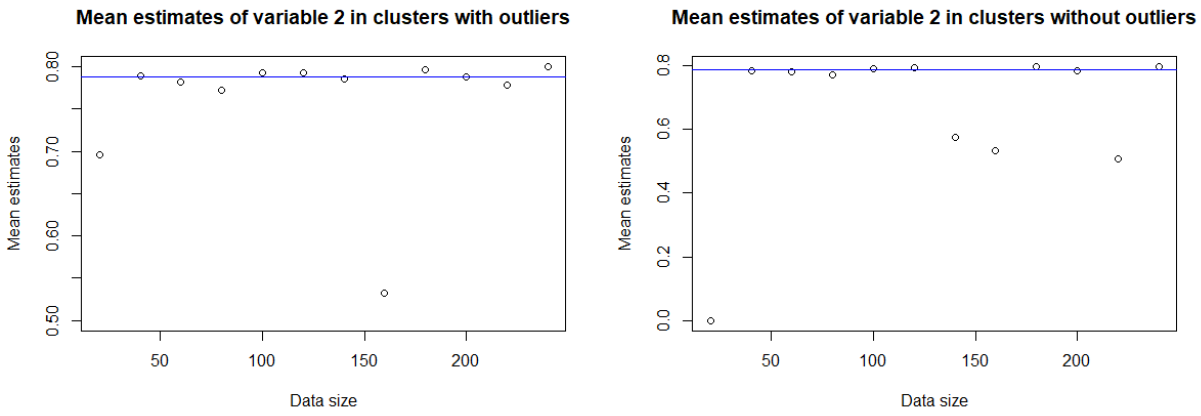


Figure (4.13) Mean estimates of spherical homogeneous clusters with 2 outliers vs without outliers using the prior distribution.

Table (4.5) Variance estimates of spherical and homogeneous clusters with outliers using the EM algorithm with prior distribution.

Cluster size	20	40	60	80	100	120	140	160
Cluster 1	0.0138	0.0034	0.0051	0.0050	0.0044	0.0031	0.0041	0.0038
Cluster 2	0.0146	0.0039	0.0051	0.0024	0.0044	0.0031	0.0041	0.0038
Cluster 3	-	0.0107	-	0.01001	-	0.0031	0.0041	0.0038

The estimates obtained in the presence and absence of outliers are only slightly different, by about 0.0001. However, in the presence of outliers the variance estimates seem to be lower with the use of the prior distribution. This can be seen for data sizes 40 to 160 in Table (4.5) in comparison to Table (4.2). There is no obvious result when comparing these tables to those obtained without using the prior distribution. The highest deviation from the true value when the prior distribution is not used is 0.0027 and and 0.0011 when the prior distribution is used in the presence of outliers. This is for data sizes above 40. For the data size with 20 data points the variance estimate is highly underestimated in both cases with the use and without the use of the prior distribution. A similar trend is observed in the presence of outliers where, the variance estimates are lower with the use of the prior in the absence of outliers compared to estimates

obtained without the use of the prior distribution.

Table (4.6) Variance estimates of spherical and homogeneous clusters without outliers using the EM algorithm with prior distribution.

Cluster size	20	40	60	80	100	120	140	160
Cluster 1	0.0128	0.0037	0.0051	0.0051	0.0043	0.0029	0.0041	0.0037
Cluster 2	-	0.0045	0.0051	0.0027	0.0043	0.0029	0.0041	0.0037
Cluster 3	-	0.0102	-	0.0101	-	0.0029	0.0041	0.0037

4.7 Further investigations into parameter estimation

The previous section studied the effect of outliers on the ability of the EM algorithm to estimate the parameters of fitted models. This section considers a further investigation of the effect of outliers on the EM algorithm where the distance of the outliers from the centre of the cluster is varied. The dataset used is the 6th simulated dataset which has 120 datapoints without outliers. This dataset was chosen as the clusters are distinct enough so that the only variation is in the location of the outliers in relation to the mean. Five datasets are created from this dataset by simulating ouliers from different standard deviations from the mean. The deviations are in increasing order $(1, 1.25, 1.5, 1.75, 2)$.

Figure (4.14) shows the mean estimates for the 2 variables in both clusters against the number of standard deviations that the outliers are away from the mean. A closer look at the cluster mean estimates of variable 1 shows that, mean estimates of variable 1 are moving further away from the true parameter as the outliers move further from the mean of the cluster. However, the mean estimates of variable 2 approach the true parameter values as the outliers' deviation from the mean increases. The remaining mean estimates may be seen in Appendix B. It is important to note that the scale is very small in these plots and the estimates are within 0.01 of the true value.

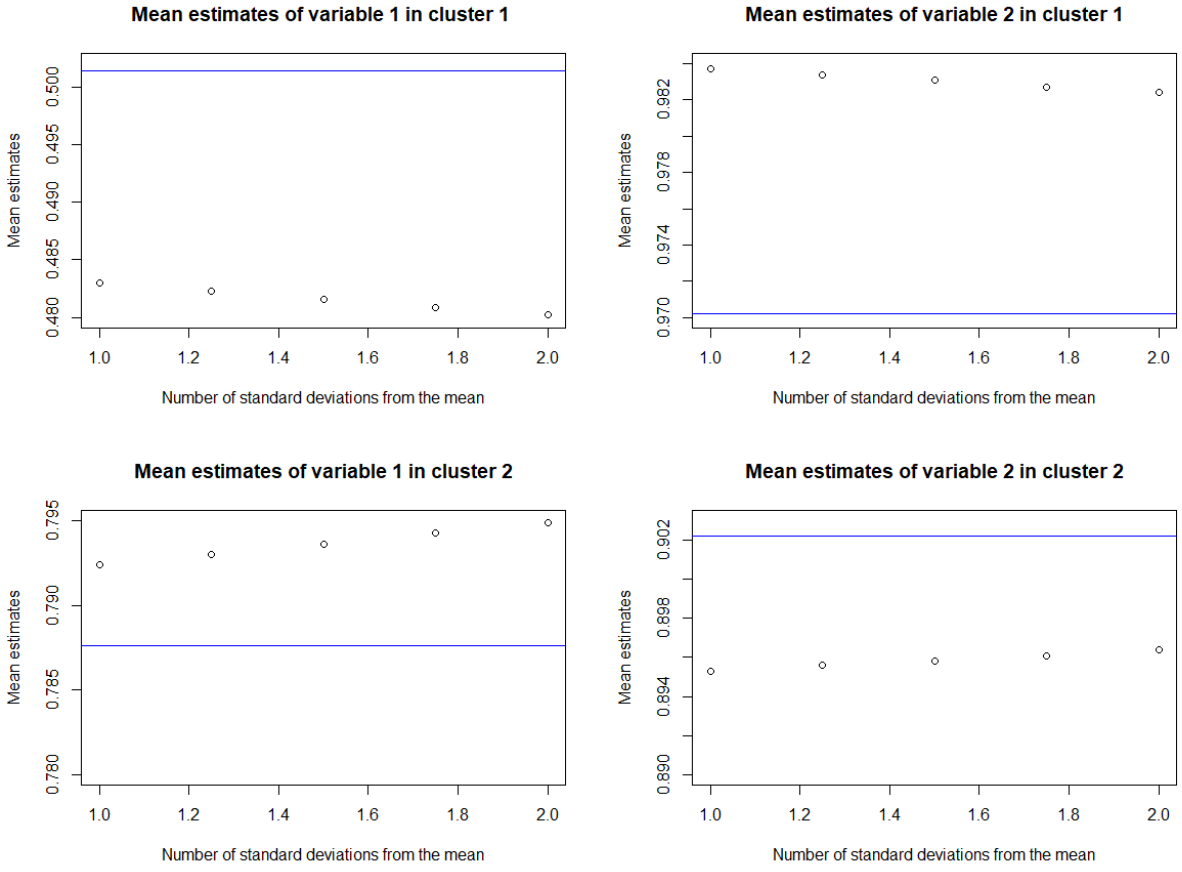


Figure (4.14) Mean estimates as the number of deviations of the outliers away from the mean increases.

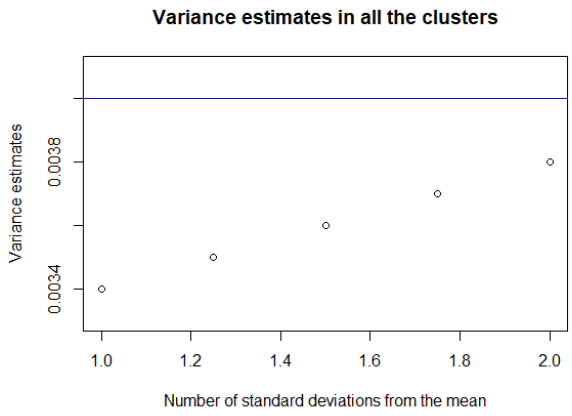


Figure (4.15) Varinace estimates as the number of standard deviations of the outliers away from the mean increases.

Figure (4.15) shows the variance estimates of the 5 datasets in which the outliers are at different standard deviations from the mean. The variance estimates are increasing as the outliers move further from the mean of the clusters. An interesting observation is that the variance estimates are increasing towards the true variance. Therefore, the EM algorithm has underestimated the variance since variance estimates obtained when outliers are further from the mean are closer to the true variance compared to when the outliers are closer to the mean of the variables. Indeed, when there are no outliers in the data, the variance estimate is 0.0033 which is lower than the true value of 0.0040. It is important to also note that the scale in the plot is very small and all the estimates are within 0.001 of the true value.

4.8 Application to a real dataset

In this section, the findings from the previous sections are studied on real datasets. Firstly, the outlier detection algorithms are run to get a fair understanding of the existence of outliers. Subsequently, the effect of outliers on the ability of the EM algorithm to recover the number and structure of clusters is then studied. Due to the dimensionality of the real datasets being outside the scope of this study, the parameter estimates

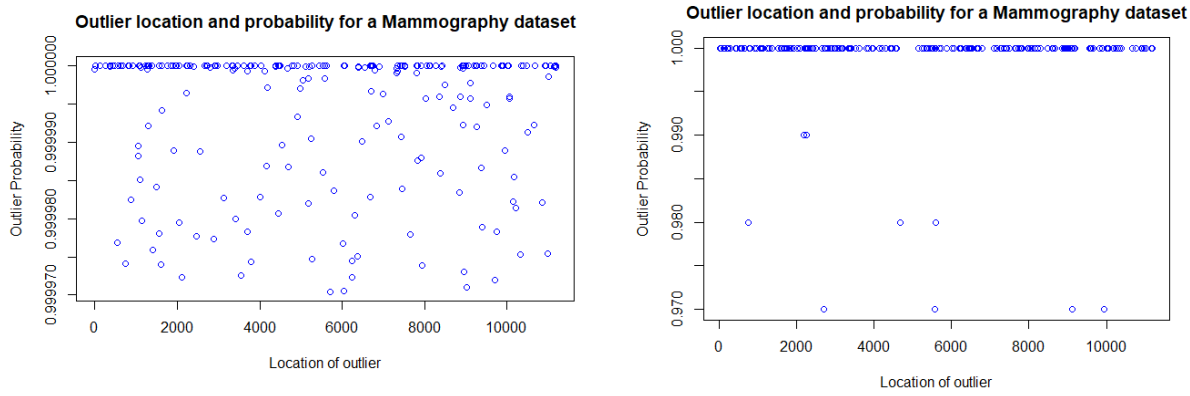
will be excluded. Then the prior distribution is also incorporated to study the results in a real dataset.

4.8.1 Mammography dataset

A Mammography dataset obtained and used in Gao et al. (2011) is used in this study. The data consists of 10923 normal records and 260 records that are anomalies with 6 continuous attributes. The Mahalanobis distance outlier detection algorithm identified 267 data points as outliers at a cut off of 0.99997. All the outliers were recovered. Several runs of the algorithm were made to get to the threshold of 0.99997 at which all the outliers are recovered. The depth-based outlier detection algorithm identified 274 outliers at threshold of 0.028. This was the threshold at which all the outliers were recovered. However, the Mahalanobis and depth-based outlier detection algorithms identified 7 and 14 normal datapoints as outliers, respectively.

The plots in Figure (4.16) show the outlier location and probability of all observations identified to be outliers by the Mahalanobis distance algorithm and the depth-based algorithm. The probability is defined as the probability of the record being an outlier. As seen in Figure (4.16), the depth-based outlier detection algorithm, unlike the Mahalanobis distance algorithm, seemed to be more robust in determining the probability of an outlier. This is because only a few outliers have a probability of being an outlier that is less than 1. The Mahalanobis distance shows less certainty about the records' status as being outliers or not. However, most of the probabilities are very close to 1 as the scale in the figures is very large.

Figure (4.17) shows the BIC curves from running the EM algorithm on the Mammography dataset with and without outliers, respectively. From the figure, the behaviour of the BIC curves when the outliers are included and removed seems to be similar in respect of the number of clusters. The EEV model consistently produces the highest BIC value, followed by the EEI and the EEE model whose BIC values lie close to each other. Lastly, the EII model produces the lowest BIC values. This shows that the algorithm is in favour of the EEV model with diagonal clusters. The EEV model shows

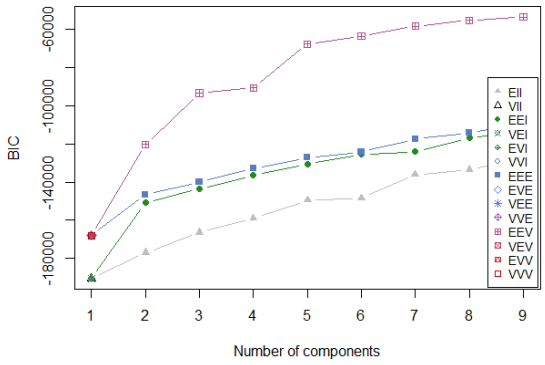


(a) Mahalanobis distance outlier detection output (b) Depth-based outlier detection output

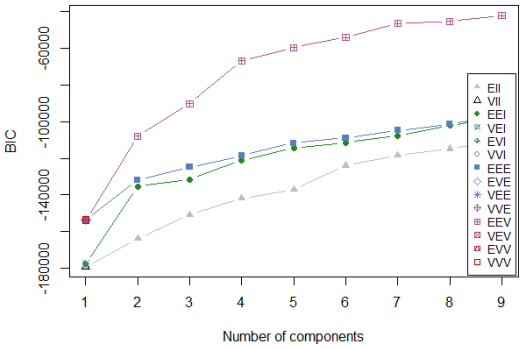
Figure (4.16) Probability plots for outliers using the Mahalanobis distance and the depth-based outlier detection algorithms respectivel

that, in the presence of outliers, there are definitely not 4 clusters, compared to when there are no outliers. This is seen in the decrease in the BIC value at 4 clusters in the BIC curves where outliers are present. Another difference is that, the BIC values are higher, when the outliers are absent compared to when they are present. This can be seen in the BIC curves in relation to the y-axis in both figures.

The overall difference in the BIC curves in respect of the choice of the number of clusters appears to be minimal. This could be because this is a very large dataset, hence results obtained from the simulations in the previous sections may not be visible. The dataset is separated into two, the normal objects and anomalies, however in both dataset (with and without outliers), the algorithm fit the EEV model to 9 clusters in both datasets which is well beyond the 1 cluster that exists in the data. It is important to note that the number of clusters recovered by the algorithm in not consistent with steps in Fraley and Raftery (1998) that identify the model parameters and the number of clusters by the first decisive maximum in the BIC curves. One may say that the first decisive maxima in Figure (4.17) are at 4 in (a) and 4 in (b).



(a) BIC curves with outlier

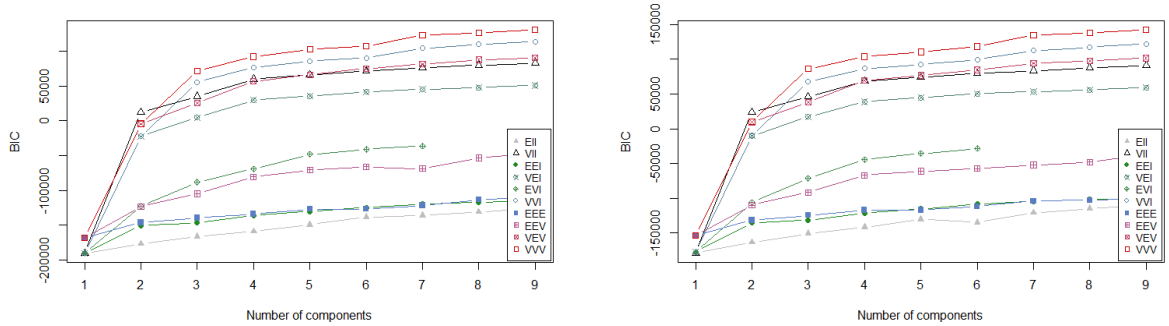


(b) BIC curves without outliers

Figure (4.17) BIC curves of the Mammography dataset, with and without outliers, respectively.

The EM algorithm fit the EEV (ellipsoidal, equal volume and equal shape) model on both datasets, where there are outliers and when the outliers are removed. The loglikelihoods were -25392.47 from the data containing outliers, and -19348.14 from the data without outliers. Therefore, similar to what was observed with the simulated datasets, the likelihood is higher when the outliers are removed from the data.

Now, running the EM algorithm with the use of a prior distribution. The BIC curves are shown in Figure (4.18), these are now smoother as it was the case when the algorithm was run with a prior distribution on the simulated datasets. However, the effect of the outliers is still not seen because the curves look identical. This could be because the data is very large. Removing the 274 outliers out of 11183 observations might not make a difference to the EM algorithm as the outliers make up 2.3 percent of the dataset.



(a) BIC curves with outliers.

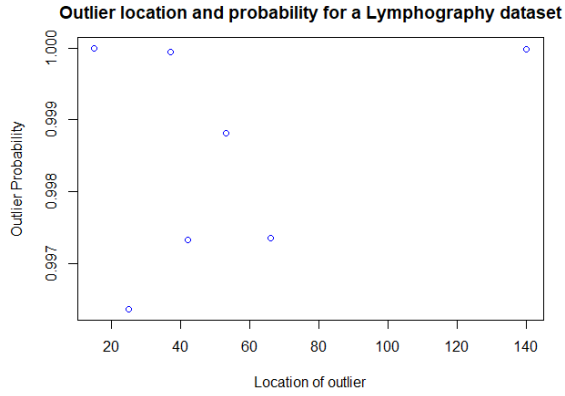
(b) BIC curves without outliers.

Figure (4.18) BIC curves of the Mammography dataset, with and without outliers, respectively, using the prior distribution.

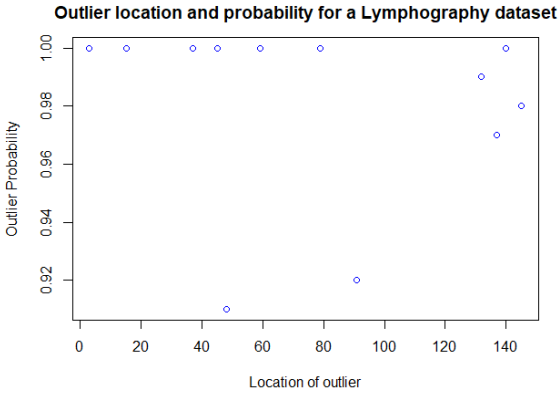
4.8.2 Lymphography dataset

Now considering a smaller dataset, which is more similar to the simulations used in terms of the size. This is data based on patients' radiological examination results. The data consists of 142 normal observations and 6 outliers, described by 3 numeric variables and 16 categorical attributes (Lazarevic and Kumar, 2005; Nguyen et al., 2010).

As seen in the probability plots in Figure (4.19), the depth-based algorithm produces outliers with the highest probabilities of being outliers as compared to the Mahalanobis distance.



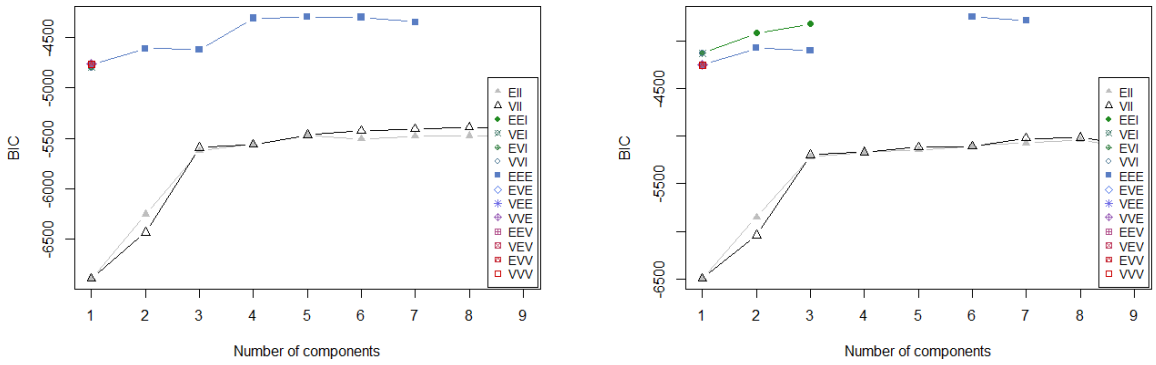
(a) Mahalanobis distance outlier detection output.



(b) Depth-based outlier detection output.

Figure (4.19) Probability plots for the Mahalanobis distance and the depth based algorithms for outlier detection

The BIC curves from running the EM algorithm on the Lymphography dataset with and without outliers, respectively, is shown in Figure (4.20). Some models have collapsed when the outliers are removed and this collapse occurred at lower cluster sizes in comparison to when outliers are included in the data. This collapse is seen in the BIC plot were no outliers are present, there is a gap between 3 and 6 outliers in the EEE model. Unlike the Mammography dataset, the BIC curves do not behave the same. For instance, the EEI model is fit when outliers are excluded and it is not fitted in the presence of outliers. The number of clusters recovered in the presence of outliers is 5 and those recovered in the absence of outliers is 6. It is important to note that in Fraley and Raftery (1998), the decision on the optimal number of clusters is based on the first decisive maximum. According the Fraley and Rafety, the optimal number of clusters is 3.

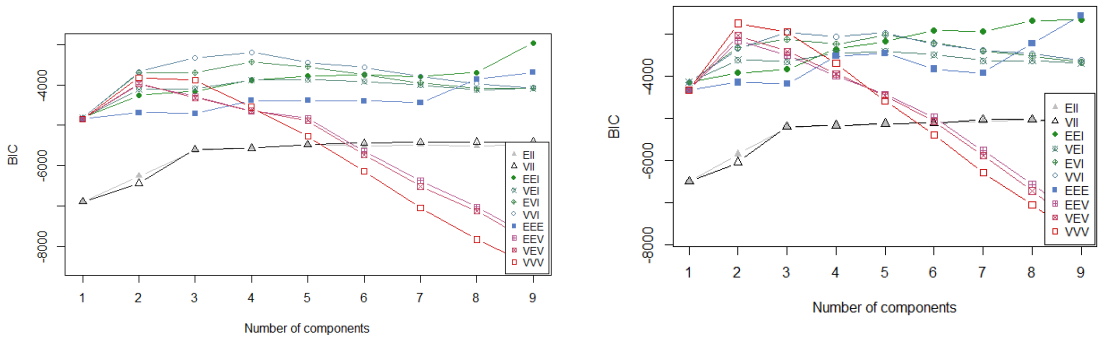


(a) BIC curves of dataset with outliers.

(b) BIC curves of dataset without outliers.

Figure (4.20) BIC curves for the Lymphography dataset

Incorporating the prior distribution within the EM algorithm, the results of the BIC curves are shown in Figure (4.21) for the Lymphography dataset with and without the outliers. A significantly larger number of models now produce BIC results hence appear in the BIC plot compared to when the prior distribution was not used. The number of clusters recovered by the algorithm in the presence and absence of outliers is 9. Similarly, using the decision method in Lazarevic and Kumar (2005); Nguyen et al. (2010), the first decisive local maximum occurs at 2 and 4 clusters respectively. The results for the number of clusters recovered maybe seen in the R code in Appendix D. However, the EEI model was best the fit in the presence of outliers and the EEE model was the best fit in the absence of outliers.



(a) BIC curves of dataset with outliers.

(b) BIC curves of dataset without outliers.

Figure (4.21) BIC curves for the Lymphography dataset with and without outliers, using the prior distribution

Chapter 5

Discussion

In this section, the results obtained from the analysis are discussed. The section is divided into subsections that specifically investigate the findings from the analysis.

5.1 Sensitivity of the algorithm to datapoints

This section discusses into detail how the contamination of the dataset with 2 outliers can lead the algorithm into recovering more clusters than there are in the datasets. This observation is interesting because without the outliers, none of the BIC curves maximise at 3 clusters, as seen in the analysis. This might be because of the location of the outliers relative to the clusters. Figure (5.1) shows the classification plots for the 6 datasets studied in the analysis. The plots indicate the grouping of each observation into a cluster using a blue, red and green colour. The circular figure indicates the cluster and intersection of the diagonal within these circular objects indicate the cluster centre. It appears to be the case that 3 clusters are recovered when the outliers are relatively far from the clusters. An example is in classification plot Figure (5.1)(b),(d) and (f). This is sensible, however, does not seem to hold all the time. In Figure (5.1)(e), there is an outlier that is fairly far from the clusters as in Figure (5.1)(b),(d) and (f) but the algorithm recovered 2 clusters. A closer look at the classification plots could suggest that, due to the fact that the clusters in Figure (5.1)(e) are more spread out compared to those in the Figure (5.1)(b),(d) and (f), it is easier for the outliers to be included within

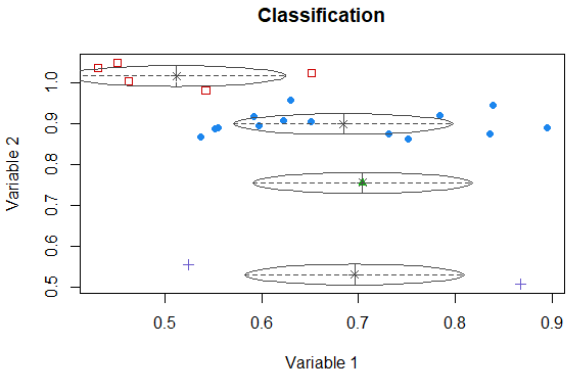
the existing cluster. In other words, the variance in Figure (5.1)(e) allows for outliers to fall in a region within the mixture model, where they would be regarded to fall in the tail of the distribution and not outside the distribution.

The density of the clusters seems to also explain the reason the EM algorithm recovered 3 clusters by grouping the outliers into a cluster of their own. Figure (5.2) shows the density plots of the same datasets studied in the analysis and whose classification plots are in Figure (5.1). The density contour plots shows that, in all clusters in which the outliers were grouped within their own cluster, the density at the centre of the cluster was very low compared to the case where the EM algorithm recovered 2 clusters and the density at the centre was high. It therefore seems to be the case that, the density towards the centre of the cluster affects the choice of the number of clusters chosen by the algorithm in that, the higher the density at the centre, the more an outlier will not be regarded as an observation belonging to that cluster.

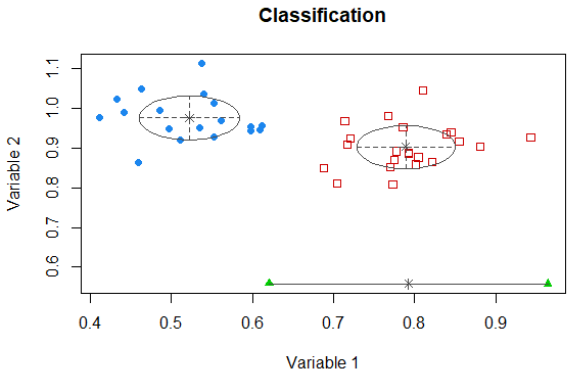
Perhaps one of the reasons why the outliers would be included within existing clusters even though it is clear that they do not belong is because of singularity of the covariance matrix. Since the data is 2-dimensional, then a minimum of 3 observations would be required to avoid the covariance matrix being ill-conditioned thus causing the algorithm to collapse. The collapse is discussed further in detail in the next section.

5.2 Collapse of the algorithm

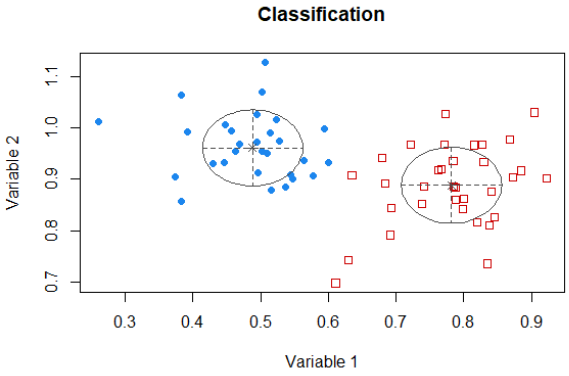
Another aspect of interest is that, looking at the manner in which the algorithm collapsed, it seems the point at which the majority of the models collapse might be informative. The collapse of the EM algorithm does not seem to occur before or at the exact number of clusters that actually exist in the dataset, which is 2. Therefore, this could tell us that there is no need to make suspicions or hypothesis that the number of clusters in the data set is beyond this point of the collapse. Figure (5.3) shows a completely different simulation of 3 spherical inhomogeneous clusters with 3 outliers. To the right of the simulations is the BIC curve obtained from running the EM algorithm on each dataset. Looking at the manner of collapse in the BIC curves, a similar observation is



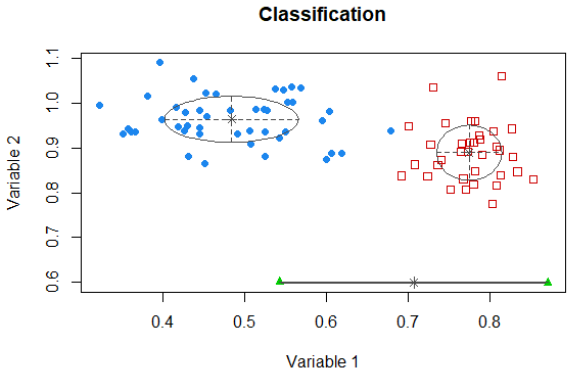
(a) Classification plot 1.



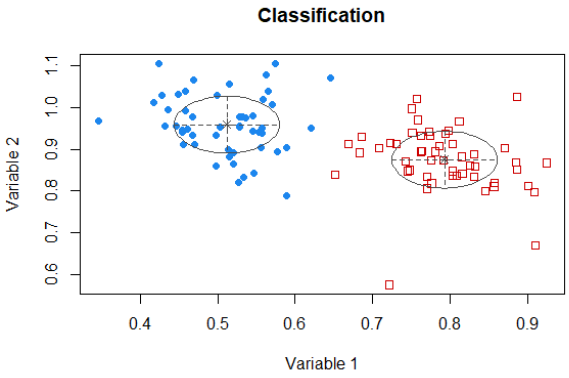
(b) Classification plot 2.



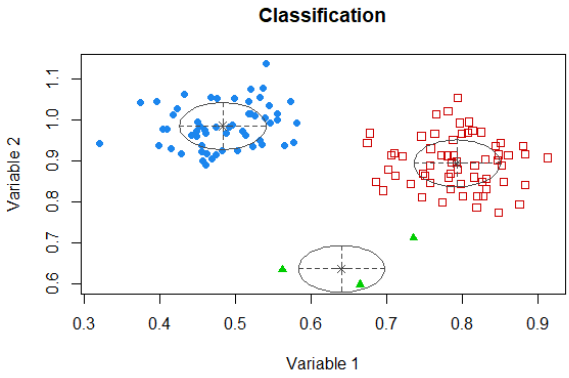
(c) Classification plot 3.



(d) Classification plot 4.

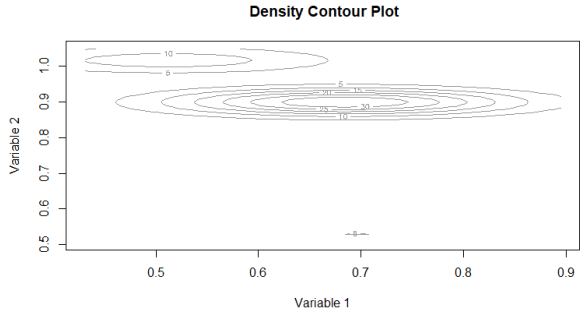


(e) Classification plot 5.

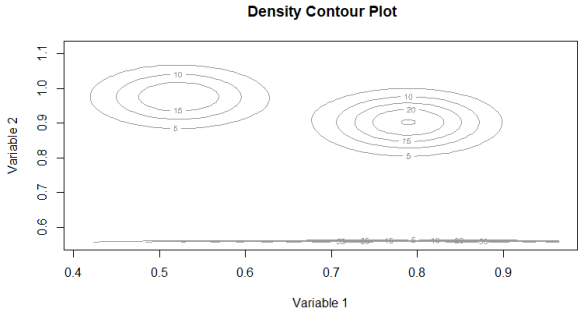


(f) Classification plot 6.

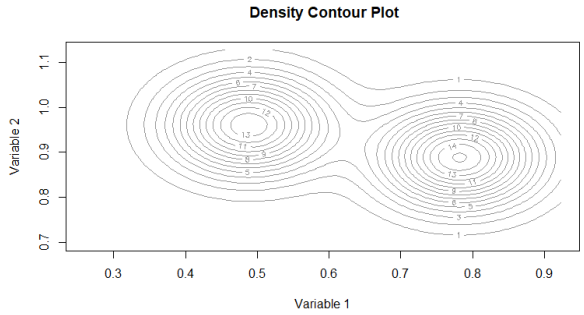
Figure (5.1) Classifications of spherical homogeneous clusters with outliers



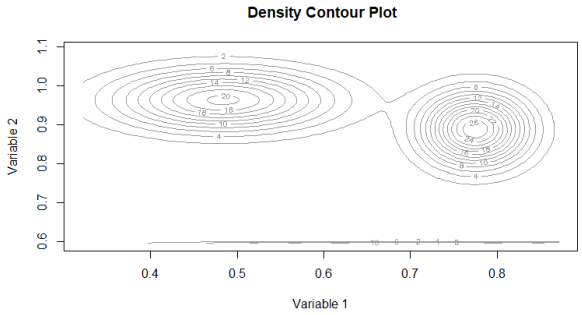
(a) Density plot 1.



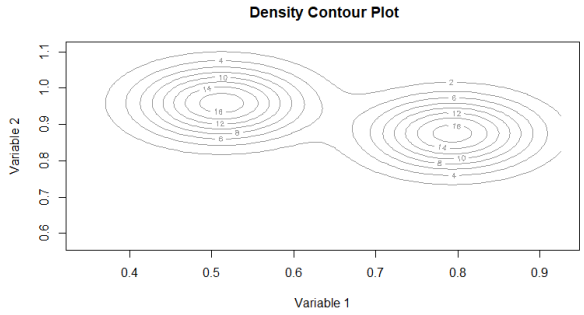
(b) Density plot 2.



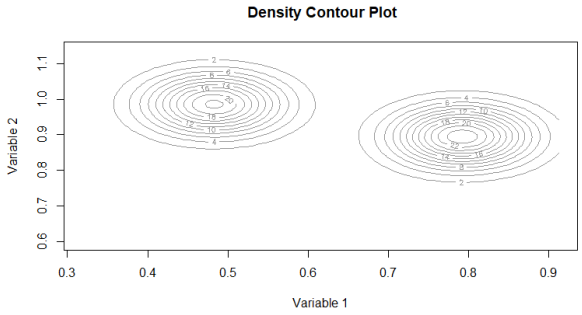
(c) Density plot 3.



(d) Density plot 4.



(e) Density plot 5.



(f) Density plot 6.

Figure (5.2) Density of spherical homogeneous clusters with outliers.

seen that the EM algorithm collapse for most of the models beyond 3 clusters. It is advised that after ill-conditioning has been encountered, the EM algorithm should not be continued if it was initialised using hierarchical clustering techniques (Fraley and Raftery, 1998).

In addition to this, the EM algorithm and BIC monitor an estimate of the reciprocal condition number of the resulting covariance matrices. This number ranges between 0 and 1 and as the number approaches zero the covariance matrices become more ill-conditioned (Golub and Van Loan, 1996). Therefore, the results for the BIC should be less reliable just before reaching the points of collapse of the algorithm. This is in line with observations made in the analysis and in Figure (5.3). The point just before the collapse (eg collapse occurs at 5 clusters in Figure (5.3)) is not the true number of clusters.

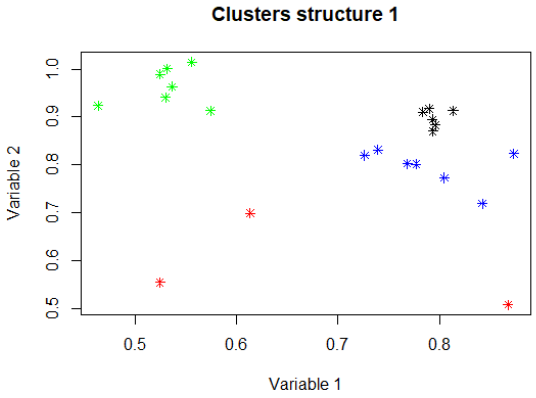
Some more complex computations have been proposed for cases in which the BIC is indeterminate. In some real and simulated clusters, the calculation of Bayesian factors using the Laplace-Mertopolis estimator in Bayesian inference with the Gibbs sampling were used (Bensmail et al., 1997).

5.3 Difference in loglikelihood

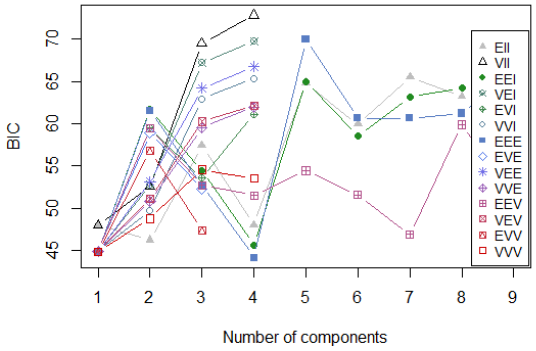
Of interest is also the realisation that the loglikelihood that results in the optimal BIC is lower when the clusters include outliers compared to when they don't. Firstly, considering the relationship between the BIC and likelihood function, Equation (5.1) below is the equation used to calculate the BIC values.

$$2\log f(S|G_i) \approx 2\log f(S|\theta_i, G_i) - v_i \log(n) = BIC_i \tag{5.1}$$

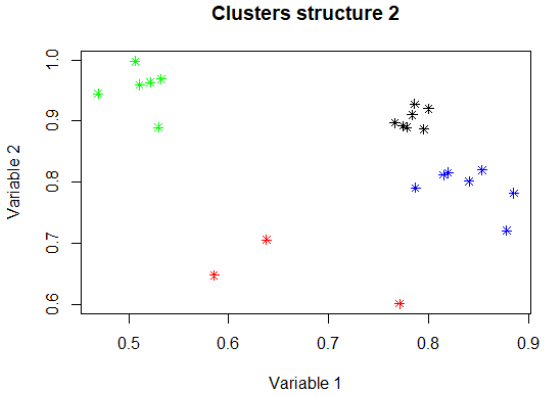
In the analysis, it was observed that the mean parameters were not affected as much as the variance estimates of variables within clusters when the outliers were present. The variance estimates were overestimated in most cases. Therefore, the low likelihood when outliers are present could be as a result of the probability model being affected by the increase in the variance. The increase in variance of the variables could



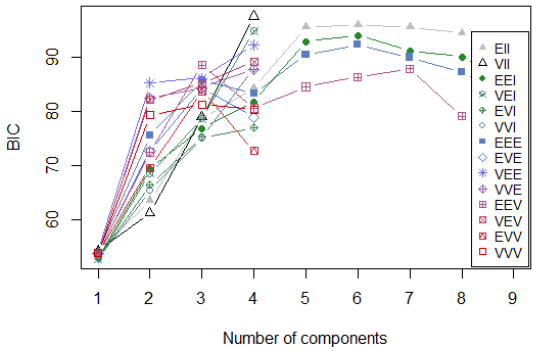
(a) Simulation 1



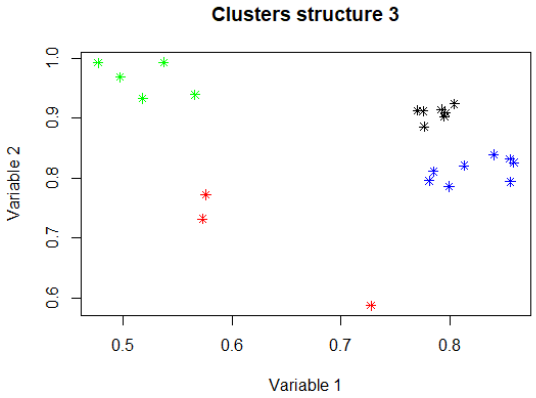
(b) BIC plot 1



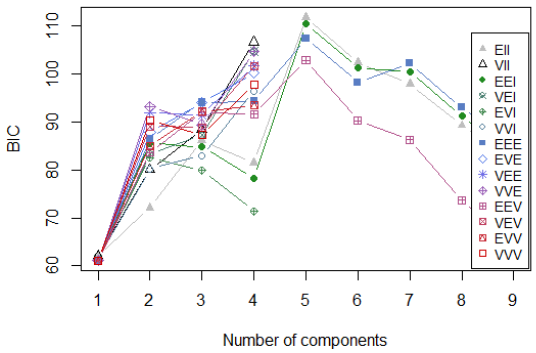
(c) Simulation 2



(d) BIC plot 2



(e) Simulation 3



(f) BIC plot 3

Figure (5.3) Three Spherical inhomogeneous clusters vs BIC curves

mean that the observations under the model are now sparse, therefore the model will give lower probabilities which would result in lower likelihood for observations within the clusters. Hence the lower loglikelihood values were observed when outliers are present. This would then mean the first term in Equation (5.1) would be smaller, thus producing lower BIC values. Therefore, the reduction in density, causes the likelihood function to also reduce. Hence the resulting difference between the loglikelihood for clusters with and without clusters. In (Lee et al., 2013) a similar observation was made where the algorithm was stopping at a local maximum in the presence of erroneous loop closures. To prevent this, the algorithm was run multiple times where the loop constraints with very low weights was removed after each run. Therefore, the outliers might be causing the algorithm to maximise at local maxima, therefore resulting in low probability values for all datapoints within the clusters.

However, regardless of this, the algorithm seems to only be slightly affected by the outliers when determining the best model to fit the data.

A similar study was done in (Lee et al., 2013), where robust pose-graph SLAM are studied using a classification Expectation Maximisation (EM) algorithm. There are erroneous loop-closure constraints that act as outliers that disrupt the algorithm. Within the EM iterations, the outlier loop closures are assigned low weights and therefore were less influential on the optimal results of the algorithm. Therefore, the probability assigned to the outliers might be low enough for the algorithm to still be able to recover the cluster structure.

5.4 Disruption of the algorithm by the prior distribution

As seen from the analysis it seems the prior distribution causes the effect of the outliers to not be clearly observed. This was because EM algorithm produced incorrect number of clusters, wrong parameter estimates when the outliers were removed from the

data. However, better parameter estimates were obtained when outliers were present. Therefore, it is of interest to understand how prior knowledge about the parameters in the mixture models would lead the EM algorithm to poor performance in the absence of anomalies.

Overfitting is the use of models that are more complicated than necessary, for example, that have more terms than necessary (Hawkins, 2004). Consider the parameter estimators that the EM algorithm uses when incorporating a prior distribution, these are tabulated in Table (2.2), in comparison to parameter estimators of the standard EM algorithm. The prior distribution uses additional parameters such as the mean, shrinkage and degrees of freedom of the prior distribution. Therefore, it is suspected that overfitting might be the reason why the EM algorithm produces worse results in the absence of outliers compared to how it performance in the presence of outliers. As a result, using the prior distribution, it might be difficult to understand the effect of outliers.

As a result of overfitting, the EM algorithm, could be reading incorrect number of clusters, and therefore, also feeding into the algorithm producing parameter estimates that are far from the true parameter values.

With the use of the prior distribution, it was realised that the likelihood of observations within clusters was lower in the absence of outliers than it was in the presence of outliers. This was contradicting the sensible findings obtained when the prior distribution was not used that, the likelihood was higher in the absence of outliers. Now, since the likelihood is dependent on the assumed probability model for a cluster, it is likely to produce incorrect results when the parameter estimates of the probability model are incorrect. Hence, the results in Figure (4.12) where the likelihood values from the EM algorithm in the absence of outliers is higher.

In addition, the prior distribution did not seem to change the model fit on the data in the presence or absence of outliers, in both the simulated and real datasets. This knowledge can be used by a researcher to bear in mind that, as much as the model assumed by the EM algorithm might be the same in the presence and absence of outliers, the parameter estimates might still be different and hence the likelihood of observations

within clusters will be different too. The ability of the EM algorithm to recover the structure of the clusters seems indifferent to the use of the prior distribution.

Chapter 6

Conclusion

This study has looked into how outliers affect the expectation maximization algorithm's ability to recover clusters using simulated as well as real datasets. The study used the number of clusters, the structure of the clusters using the Bayesian Information Criterion, as well as the parameter estimates. The study showed convincing results that the algorithm's ability to recover the number of clusters is linked to the position of the outliers relative to the cluster as well as the density of existing clusters towards the centre. When the outliers lie further from the clusters, they are grouped within their own cluster(s). As the density towards the centre of existing clusters is high, then a seperate cluster will be made for the outliers.

The Expectation Maximization algorithm is known to collapse when covariance matrices are singular. The study showed that the collapse of the algorithm can be informative. In the absence of outliers the algorithm collapses sooner or at lower cluster numbers than when the outliers are present. This has appeared to occur not because the outliers are data points that are part of the data set, but because they are outlying/abnormal data points. The study also shows that the collapse does not occur, in any case, before the actual number of clusters in the dataset.

The results from this study also showed that, even though the number of clusters recoverd from the data were incorrect due to outliers, the structure of the clusters were still recovered. In addition, the fact that the algorithm had instances were it collapsed

did not appear to affect the ability of the algorithm to recover the cluster structure. Therefore, from the results obtained from this study, the existence of the outliers did not affect the algorithm's ability to recover the cluster of the clusters.

The study also shows convincing results that the robustness of the expectation maximisation algorithm when using the prior distribution can negatively alter the results from the EM algorithm. The number of clusters and the structure of clusters when outliers are not present can be incorrectly estimated to a significant degree. This is suggested to be as a result of overfitting because the expectation maximisation algorithm with the prior distribution incorporates more parameters than without the prior distribution, therefore allowing for overfitting.

The mean values of the clusters studied seemed to deviate from the true mean for one variable and move closer to the true mean for the other variable. Therefore, this indicated a negative and positive effect of the outliers. However, one must be cautious as the dimensionality of the data could affect the algorithm. The algorithm had underestimated the variance of the variables used to generate the clusters. However, as the outliers moved further from the clusters, the variance estimates approached the true variance.

Results are primarily for 2 dimensions, with fixed number of clusters i.e 2 and cluster sizes. More clusters, dimensions, methods of simulating outliers and other methods of dealing with outliers could be investigated.

References

Aggarwal, C. C. and Philip, S. Y. (2005). An effective and efficient algorithm for high-dimensional outlier detection. *The VLDB journal*, 14(2):211–221.

Azzalini, A. and Bowman, A. W. (1990). A look at some data on the old faithful geyser. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 39(3):357–365.

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

Barnett, V. and Lewis, T. (1974). *Outliers in statistical data*. Wiley.

Bay, S. D. and Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 29–38. ACM.

Beaubien, J. M. and Baker, D. P. (2004). The use of simulation for training teamwork skills in health care: how low can you go? *BMJ Quality & Safety*, 13(suppl 1):i51–i56.

Ben-Gal, I. (2005). Outlier detection. In *Data mining and knowledge discovery handbook*, pages 131–146. Springer.

Benaglia, T., Chauveau, D., Hunter, D., and Young, D. (2009). mixtools: An r package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29.

Bensmail, H., Celeux, G., Raftery, A. E., and Robert, C. P. (1997). Inference in model-based cluster analysis. *statistics and Computing*, 7(1):1–10.

Bholowalia, P. and Kumar, A. (2014). Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9).

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.

Bock, H. H. (1996). Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis*, 23(1):5–28.

Bouchet, F. and Kandrup, H. E. (1985). Particle-mesh simulations of clustering in cosmology. *The Astrophysical Journal*, 299:1–4.

Bowers, K. J., Chow, D. E., Xu, H., Dror, R. O., Eastwood, M. P., Gregersen, B. A., Klepeis, J. L., Kolossvary, I., Moraes, M. A., Sacerdoti, F. D., et al. (2006). Scalable algorithms for molecular dynamics simulations on commodity clusters. In *SC'06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, pages 43–43. IEEE.

Bozdogan, H. (1994). Choosing the number of clusters, subset selection of variables, and outlier detection in the standard mixture-model cluster analysis. In *New approaches in classification and data analysis*, pages 169–177. Springer.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM.

Buchanan, J. A. (2001). Use of simulation technology in dental education. *Journal of dental education*, 65(11):1225–1231.

Cebeci, Z. and Yildiz, F. (2015). Comparison of k-means and fuzzy c-means algorithms on different cluster structures. *AGRÁRINFORMATIKA/JOURNAL OF AGRICULTURAL INFORMATICS*, 6(3):13–23.

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

Cheeseman, P. C., Stutz, J. C., et al. (1996). Bayesian classification (autoclass): theory and results. *Advances in knowledge discovery and data mining*, 180:153–180.

Chen, H., Chen, J., and Kalbfleisch, J. D. (2004). Testing for a finite mixture model with two components. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(1):95–115.

Chen, H., Chiang, R. H., and Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4).

Dang, X. and Serfling, R. (2010). Nonparametric depth-based multivariate outlier identifiers, and masking robustness properties. *Journal of Statistical Planning and Inference*, 140(1):198–213.

Dasgupta, S. (1999). Learning mixtures of gaussians. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 634–644. IEEE.

Do, C. B. and Batzoglou, S. (2008). What is the expectation maximization algorithm? *Nature biotechnology*, 26(8):897.

DuMouchel, W. and Schonlau, M. (1998). A fast computer intrusion detection algorithm based on hypothesis testing of command transition probabilities. In *KDD*, pages 189–193.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868.

Ertöz, L., Steinbach, M., and Kumar, V. (2003). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 47–58. SIAM.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, 90(430):577–588.

Fayyad, U. and Smyth, P. (1997). From massive data sets to science catalogs: applications and challenges. *Statistics and Massive Data Sets: Report to the Committee on Applied and Theoretical Statistics, eds. J. Kettenring and D. Pregibon, National Research Council*.

Fraley, C., Raftery, A., and Wehrens, R. (2005). Incremental model-based clustering for large datasets with small clusters. *Journal of Computational and Graphical Statistics*, 14(3):529–546.

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.

Fraley, C. and Raftery, A. E. (2000). Model-based clustering, discriminant analysis. Technical report, density estimation. Technical Report 380, University of Washington ....

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.

Fraley, C. and Raftery, A. E. (2005). Bayesian regularization for normal mixture estimation and model-based clustering. Technical report, Washington Univ Seattle Dept of Statistics.

Fraley, C. and Raftery, A. E. (2006). Mclust version 3 for r: Normal mixture modeling and model-based clustering. Technical report, Citeseer.

Fraley, C. and Raftery, A. E. (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of classification*, 24(2):155–181.

Frühwirth-Schnatter, S. and Malsiner-Walli, G. (2019). From here to infinity: sparse finite versus dirichlet process mixtures in model-based clustering. *Advances in data analysis and classification*, 13(1):33–64.

Gao, J., Hu, W., Zhang, Z. M., Zhang, X., and Wu, O. (2011). Rkof: robust kernel-based local outlier detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 270–283. Springer.

Gelman, A. (2006). Prior distribution. *Encyclopedia of environmetrics*, 4.

Gersho, A. and Gray, R. M. (2012). *Vector quantization and signal compression*, volume 159. Springer Science & Business Media.

Ghosh, D. and Vogt, A. (2012). Outliers: An evaluation of methodologies. In *Joint statistical meetings*, pages 3455–3460. American Statistical Association San Diego, CA.

Golub, G. H. and Van Loan, C. F. (1996). Matrix computations the john hopkins university press. *Baltimore and London*.

Gosain, A. and Dahiya, S. (2016). Performance analysis of various fuzzy clustering algorithms: a review. *Procedia Computer Science*, 79:100–111.

Gough, I. (2001). Social assistance regimes: a cluster analysis. *Journal of European social policy*, 11(2):165–170.

Hansen, M. H., Madow, W. G., and Tepping, B. J. (1983). An evaluation of model-dependent and probability-sampling inferences in sample surveys. *Journal of the American Statistical Association*, 78(384):776–793.

Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.

Hawkins, D. M. (2004). The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12.

Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126.

Jain, A. K. and Dubes, R. C. (1988). Algorithms for clustering data.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In *European Conference on Artificial Life*, pages 704–720. Springer.

Johnson, R. A., Wichern, D. W., et al. (2002). *Applied multivariate statistical analysis*, volume 5. Prentice hall Upper Saddle River, NJ.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.

Kadane, J. B., Shmueli, G., Minka, T. P., Borle, S., Boatwright, P., et al. (2006). Conjugate analysis of the conway-maxwell-poisson distribution. *Bayesian analysis*, 1(2):363–374.

Kangas, J., Kohonen, T., et al. (1996). Developments and applications of the self-organizing map and related algorithms. *Mathematics and Computers in Simulation*, 41(1):3–12.

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

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the american statistical association*, 90(430):773–795.

Knorr, E. M., Ng, R. T., and Tucakov, V. (2000). Distance-based outliers: algorithms and applications. *The VLDB Journal—The International Journal on Very Large Data Bases*, 8(3-4):237–253.

Kwon, Y., Nunley, D., Gardner, J. P., Balazinska, M., Howe, B., and Loebman, S. (2010). Scalable clustering algorithm for n-body simulations in a shared-nothing cluster. In *International Conference on Scientific and Statistical Database Management*, pages 132–150. Springer.

Lampinen, J. and Oja, E. (1992). Clustering properties of hierarchical self-organizing maps. *Journal of Mathematical Imaging and Vision*, 2(2-3):261–272.

Lazarevic, A. and Kumar, V. (2005). Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166. ACM.

Lee, G. H., Fraundorfer, F., and Pollefeys, M. (2013). Robust pose-graph loop-closures with expectation-maximization. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 556–563. IEEE.

Likas, A., Vlassis, N., and Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461.

Lin, X. and Zhu, Y. (2004). Degenerate expectation-maximization algorithm for local dimension reduction. In *Classification, Clustering, and Data Mining Applications*, pages 259–268. Springer.

Maitra, R. and Melnykov, V. (2010). Simulating data to study performance of finite mixture modeling and clustering algorithms. *Journal of Computational and Graphical Statistics*, 19(2):354–376.

Mangiameli, P., Chen, S. K., and West, D. (1996). A comparison of som neural network and hierarchical clustering methods. *European Journal of Operational Research*, 93(2):402–417.

Meilă, M. and Heckerman, D. (2001). An experimental comparison of model-based clustering methods. *Machine learning*, 42(1-2):9–29.

Melnykov, V., Chen, W.-C., and Maitra, R. (2012). MixSim: An R package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, 51(12):1–25.

Milligan, G. W. (1985). An algorithm for generating artificial test clusters. *Psychometrika*, 50(1):123–127.

Murtagh, F. and Raftery, A. E. (1984). Fitting straight lines to point patterns. *Pattern recognition*, 17(5):479–483.

Nguyen, H. V., Ang, H. H., and Gopalkrishnan, V. (2010). Mining outliers with ensemble of heterogeneous detectors on random subspaces. In *International Conference on Database Systems for Advanced Applications*, pages 368–383. Springer.

Park, H.-S. and Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341.

Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In *ACM Sigmod Record*, volume 29, pages 427–438. ACM.

Rasmussen, C., de la Cruz, B., Ghahramani, Z., and Wild, D. (2008). Modeling and visualizing uncertainty in gene expression clusters using dirichlet process mixtures. *IEEE/ACM transactions on computational biology and bioinformatics*, 6(4):615–628.

Rasmussen, C. E. (2000). The infinite gaussian mixture model. In *Advances in neural information processing systems*, pages 554–560.

Salgado, C. M., Azevedo, C., Proença, H., and Vieira, S. M. (2016). Noise versus outliers. In *Secondary Analysis of Electronic Health Records*, pages 163–183. Springer.

Sander, J., Ester, M., Kriegel, H.-P., and Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 2(2):169–194.

Schoenharl, T. W. and Madey, G. (2008). Evaluation of measurement techniques for the validation of agent-based simulations against streaming data. In *International Conference on Computational Science*, pages 6–15. Springer.

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

Steinbach, M., Ertöz, L., and Kumar, V. (2004). The challenges of clustering high dimensional data. In *New directions in statistical physics*, pages 273–309. Springer.

Stevens, J. P. (1984). Outliers and influential data points in regression analysis. *Psychological Bulletin*, 95(2):334.

Tiao, G. C. and Zellner, A. (1964). On the bayesian estimation of multivariate regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):277–285.

Titterington, D. M., Smith, A. F., and Makov, U. E. (1985). *Statistical analysis of finite mixture distributions*. Wiley,.

Tong, Y. L. (2012). *The multivariate normal distribution*. Springer Science & Business Media.

Williams, G., Baxter, R., He, H., Hawkins, S., and Gu, L. (2002). A comparative study of rnn for outlier detection in data mining. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 709–712. IEEE.

Xu, D. and Tian, Y. (2015a). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193.

Xu, D. and Tian, Y. (2015b). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193.

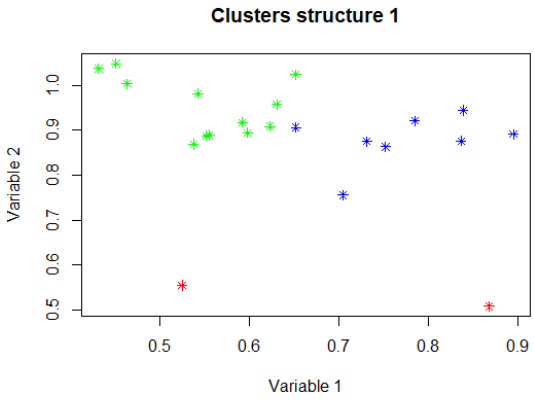
Xu, R. and Wunsch, D. C. (2005). Survey of clustering algorithms.

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.

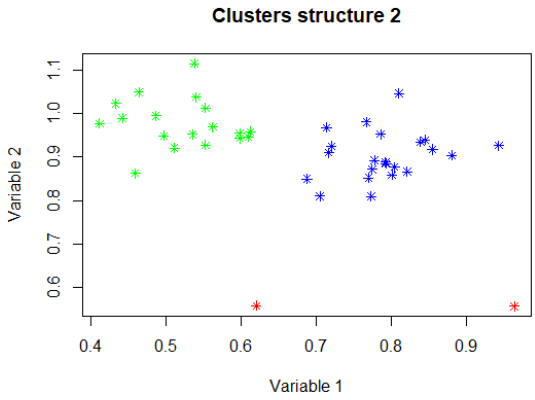
Zhang, T., Ramakrishnan, R., and Livny, M. (1997). Birch: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182.

Appendix A

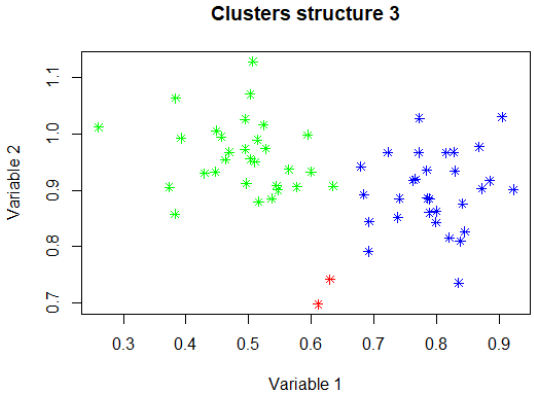
Simulations



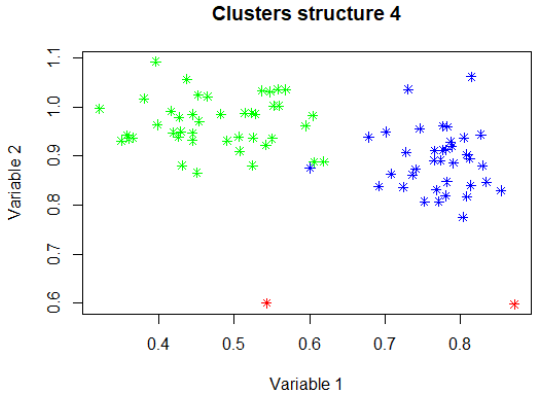
(a) Simulation with 20 datapoints



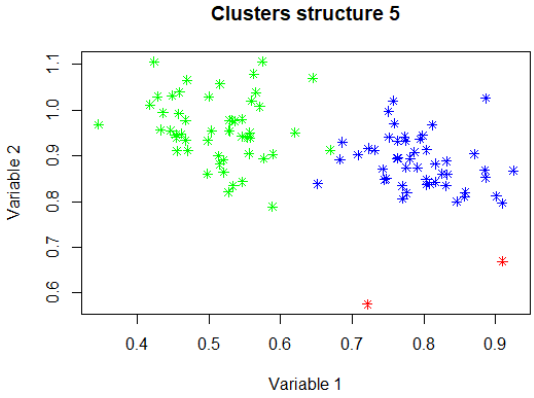
(b) Simulation with 40 datapoints



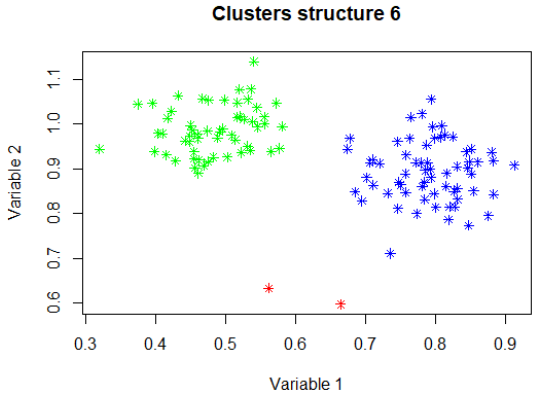
(c) Simulation with 60 datapoints



(d) Simulation with 80 datapoints

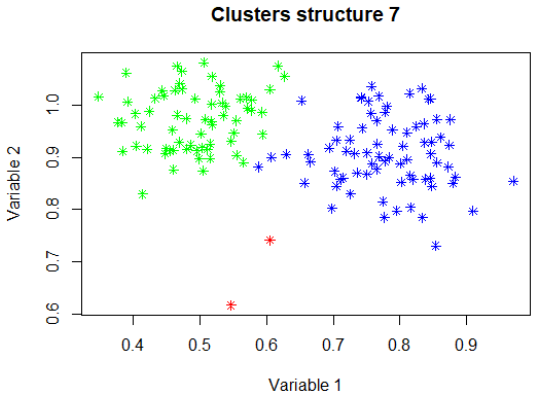


(e) Simulation with 100 datapoints

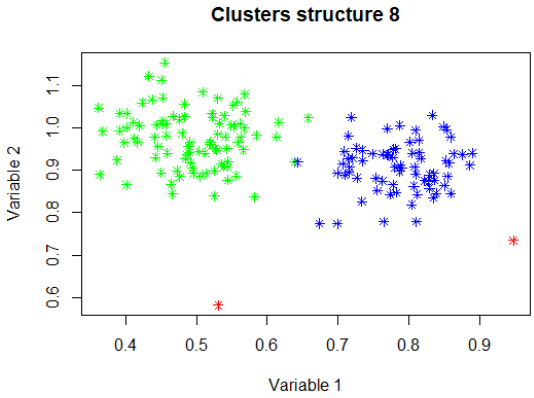


(f) Simulation with 120 datapoints

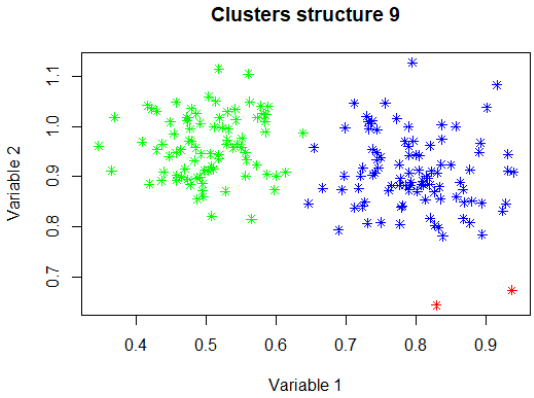
Figure (A.1) Simulations of spherical clusters with 2 outliers



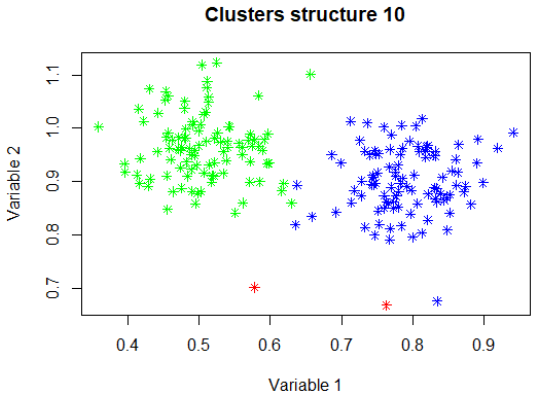
(a) Simulation with 140 datapoints



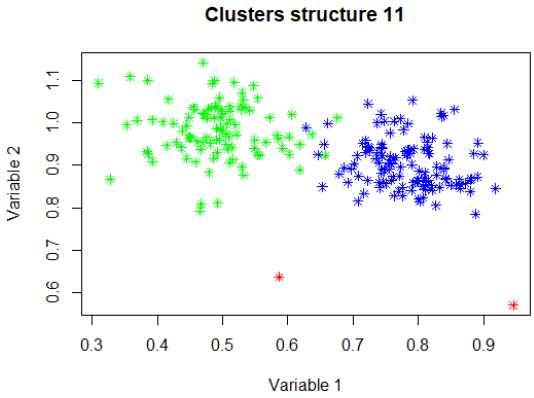
(b) Simulation with 160 datapoints



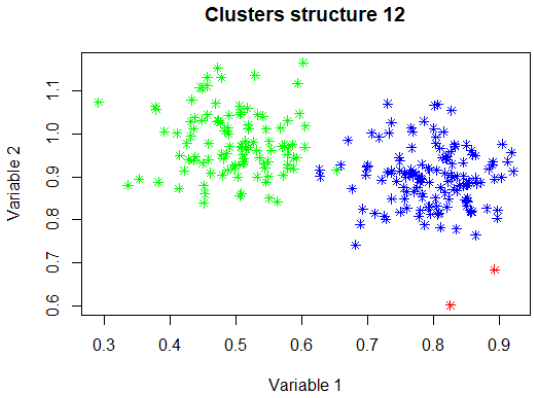
(c) Simulation with 180 datapoints



(d) Simulation with 200 datapoints

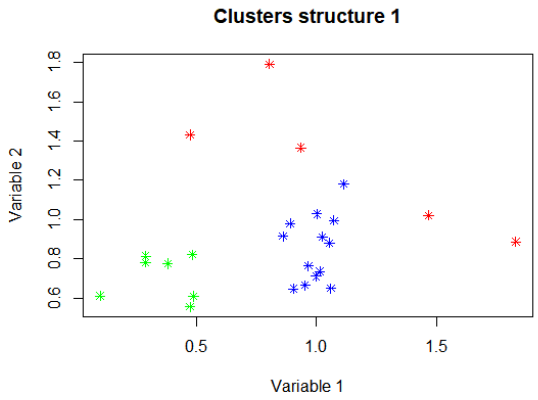


(e) Simulation with 220 datapoints

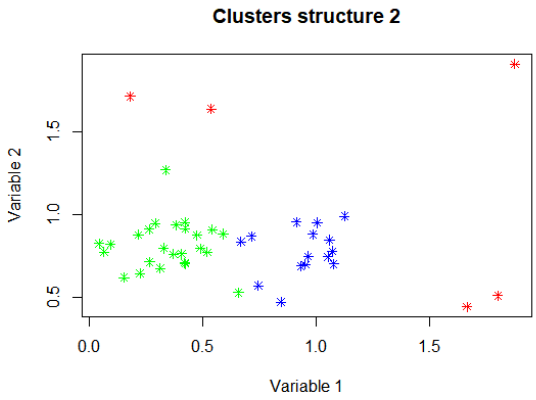


(f) Simulation with 240 datapoints

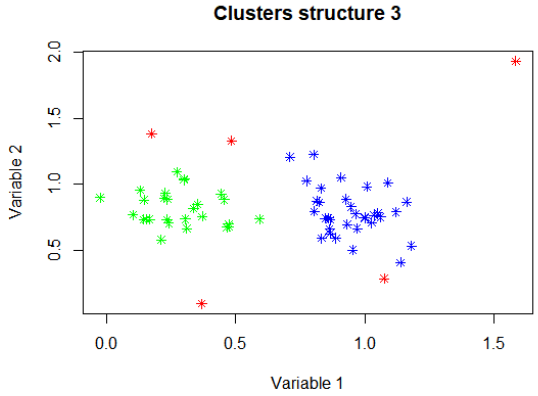
Figure (A.2) Simulations of spherical homogeneous clusters with 2 outliers continued.



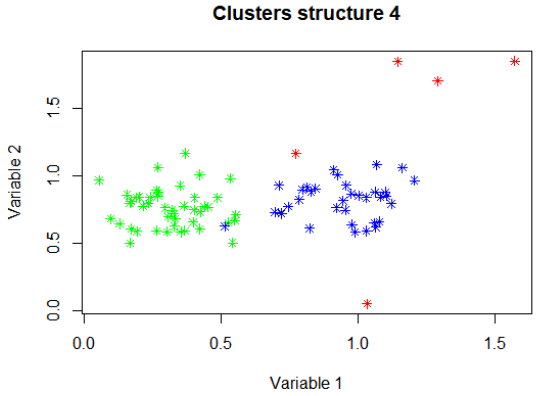
(a) Simulation with 20 datapoints



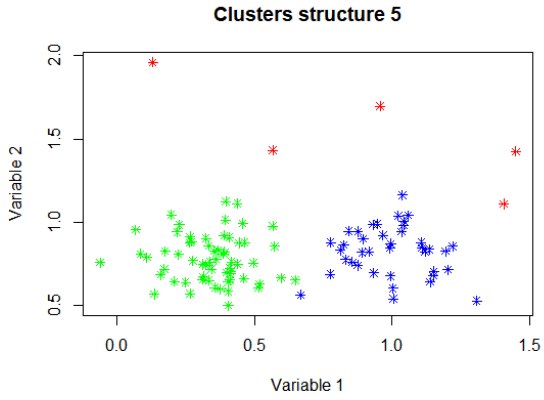
(b) Simulation with 40 datapoints



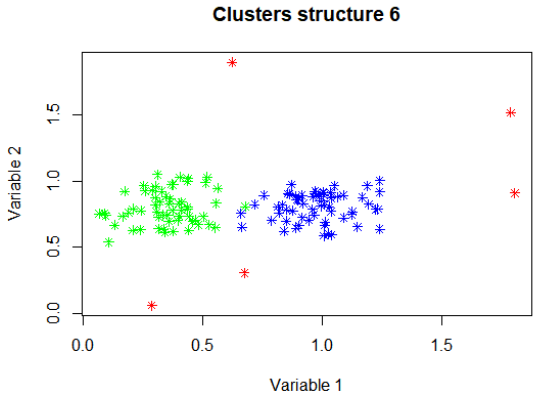
(c) Simulation with 60 datapoints



(d) Simulation with 80 datapoints

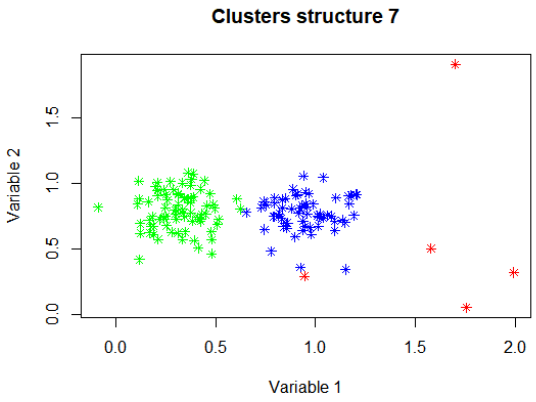


(e) Simulation with 100 datapoints

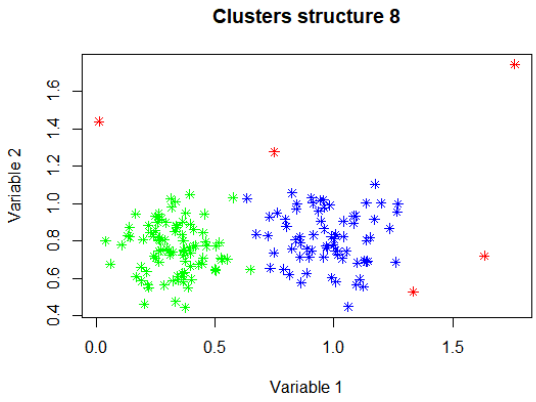


(f) Simulation with 120 datapoints

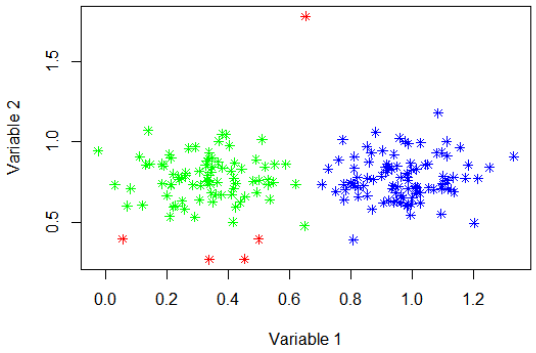
Figure (A.3) Simulations of spherical clusters with 5 outliers



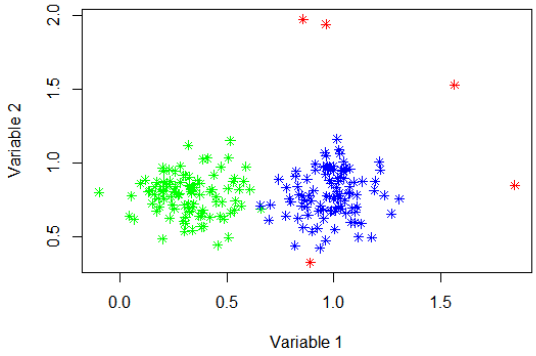
(a) Simulation with 140 datapoints



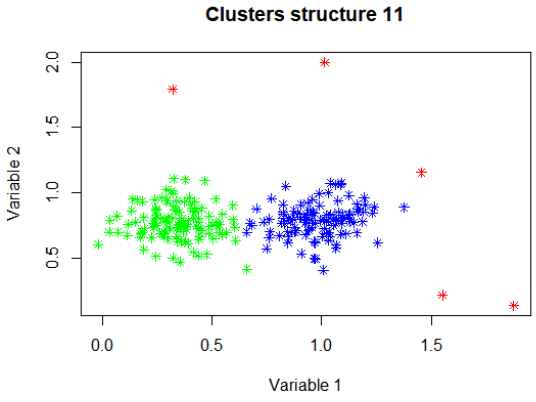
(b) Simulation with 160 datapoints



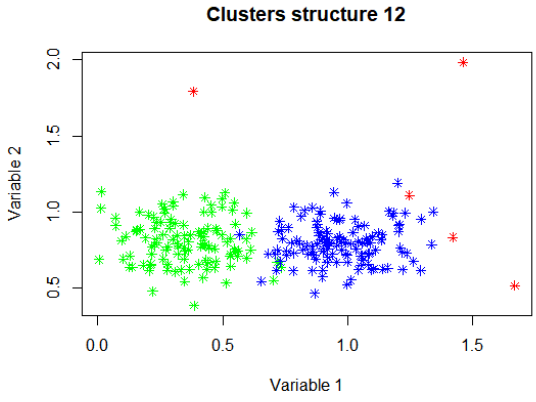
(c) Simulation with 180 datapoints



(d) Simulation with 200 datapoints



(e) Simulation with 220 datapoints



(f) Simulation with 240 datapoints

Figure (A.4) Simulations of spherical homogeneous clusters continued with 5 outliers continued

Appendix B

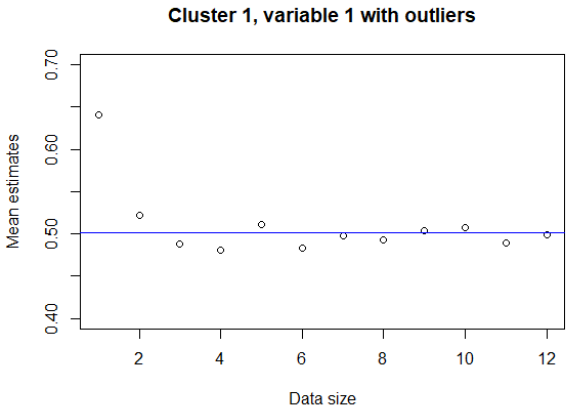
Parameter estimates

Table (B.1) Variance estimates of variable 2 in clusters recovered by the EM algorithm on the data sets with 20, 40, 60, 80,100,120,140 and 160 datapoints with 2 outliers.

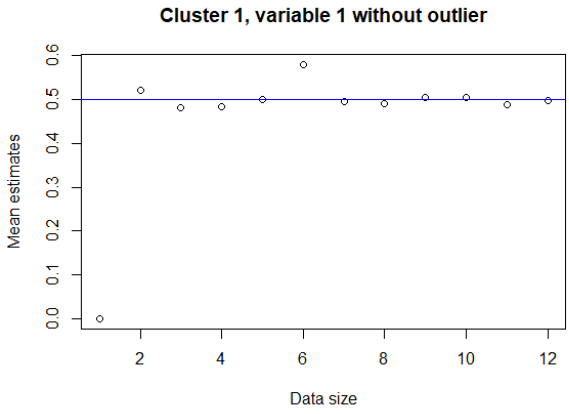
Data size	20	40	60	80	100	120	140	160
Cluster 1	0.0006	0.0031	0.0055	0.0027	0.0046	0.0033	0.0049	0.0039
Cluster 2	0.0006	0.0030	0.0055	0.0038	0.0046	0.0033	0.0049	0.0039
Cluster 3	0.0006	0.0000	-	0.0000	-	0.0033	-	0.0039
Cluster 4	0.0006	-	-	-	-	0.0033	-	-

Table (B.2) Variance estimates of variable 2 in clusters recovered by the EM algorithm on the data sets with 20, 40, 60, 80,100,120,140 and 160 datapoints without outliers.

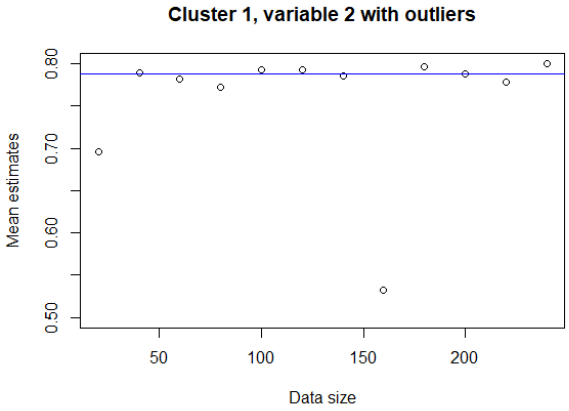
Data size	20	40	60	80	100	120	140	160
Cluster 1	0.0047	0.0034	0.0046	0.0027	0.0039	0.0033	0.0043	0.0038
Cluster 2	-	0.0034	0.0046	0.0038	0.0039	0.0033	0.0043	0.0038



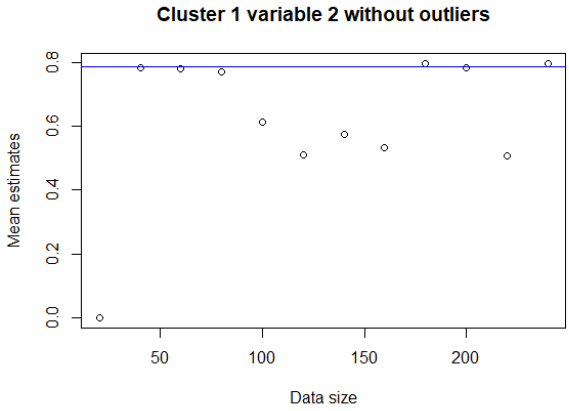
(a) Mean estimates with outliers



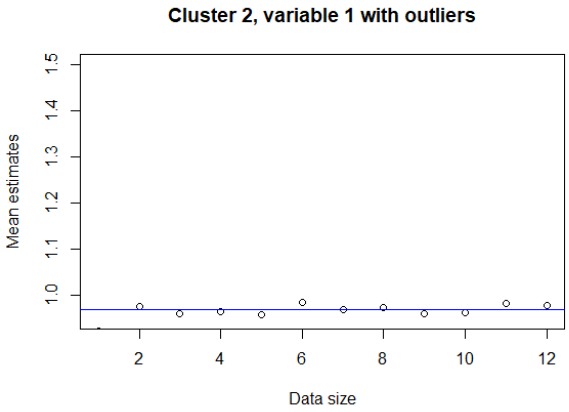
(b) Mean estimates without outliers



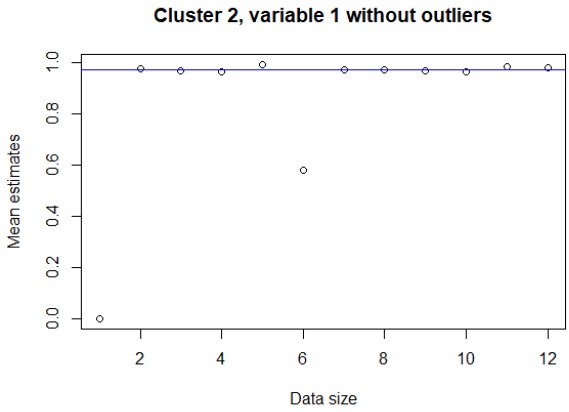
(c) Mean estimates with outliers



(d) Mean estimates without outliers

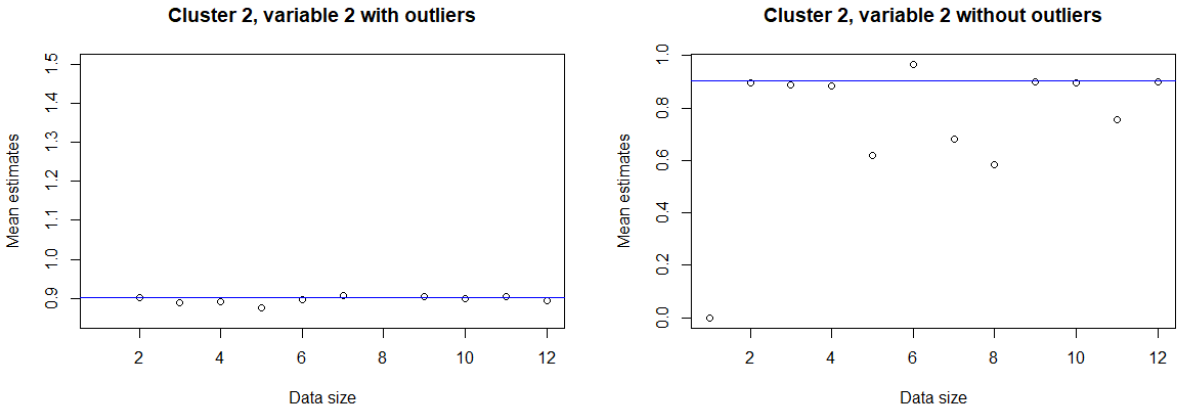


(e) Mean estimates with outliers



(f) Mean estimates without outliers

Figure (B.1) Mean estimates for spherical homogeneous clusters with 2 outliers vs without the outliers with the prior distribution.

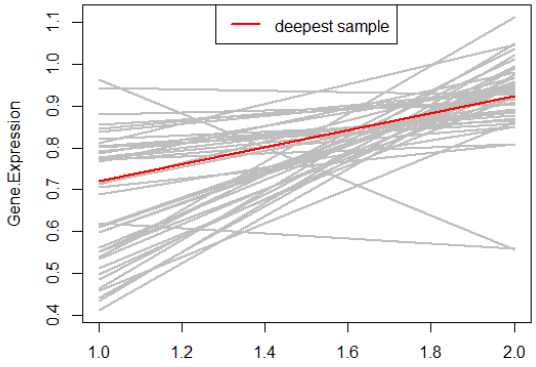


(a) Mean estimates with outliers (b) Mean estimates without outliers

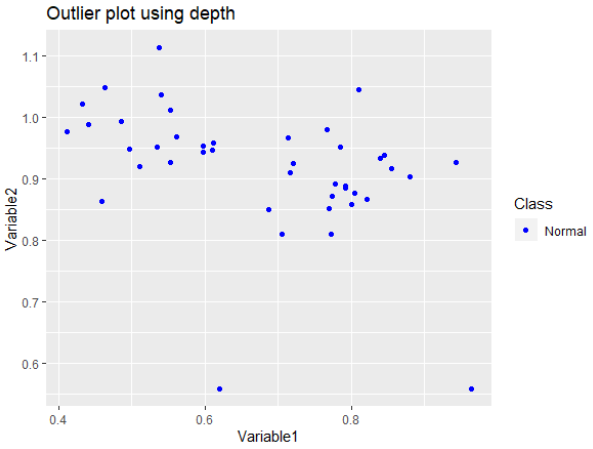
Figure (B.2) Mean estimates of spherical homogeneous clusters with 2 outliers vs without the outliers with the prior distribution continued.

Appendix C

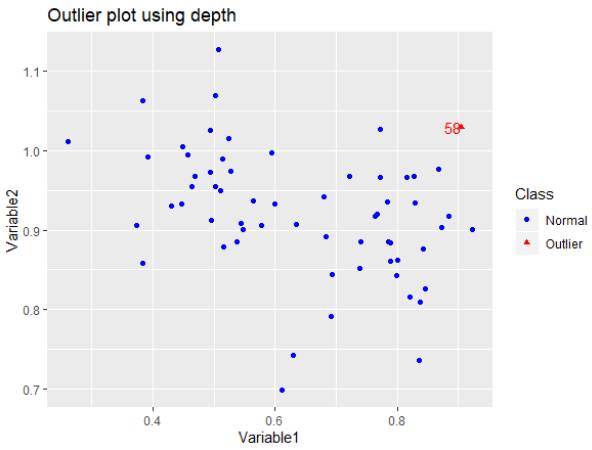
Depth-based outlier detection algorithm results



(a) Invalid results for cluster set 1



(b) Outlier detection results for cluster set 2



(c) Outlier detection results for cluster set 3

Figure (C.1) Classification plot for the Depth-based algorithm with threshold of 0.04

Appendix D

R codes

D.1 Simulations

```
> #Packages
> library(MixSim)
> library(mclust)
> set.seed(11111)
> set.I=MixSim(MaxOmega = 0.02,K=2,p=2,sph = TRUE,ecc = 1,hom = TRUE)
> #########SIMULATIONS
> sample.I.1=simdataset(20,Pi=set.I$Pi,Mu=set.I$Mu,S=set.I$S,n.out = 2,alpha =
    0.01,max.out =1000)
> sample.I.2=simdataset(40,Pi=set.I$Pi,Mu=set.I$Mu,S=set.I$S,n.out = 2,alpha =
    0.01,max.out =1000)
> sample.I.3=simdataset(60,Pi=set.I$Pi,Mu=set.I$Mu,S=set.I$S,n.out = 2,alpha =
    0.01,max.out =1000)
> sample.I.4=simdataset(80,Pi=set.I$Pi,Mu=set.I$Mu,S=set.I$S,n.out = 2,alpha =
    0.01,max.out =1000)
> sample.I.5=simdataset(100,Pi=set.I$Pi,Mu=set.I$Mu,S=set.I$S,n.out = 2,alpha =
    0.01,max.out =1000)
> sample.I.6=simdataset(120,Pi=set.I$Pi,Mu=set.I$Mu,S=set.I$S,n.out = 2,alpha =
    0.01,max.out =1000)
> ########SAMPLE PLOTS
> colors=c("red", "green", "blue")
> par(mfrow=c(1,1))
> par(mar=c(4,4,4,4))
```

```
> plot(sample.I.1$X,main ="Clusters structure 1",xlab ="Variable 1",ylab ="
    Variable 2",col = colors[sample.I.1$id+1], pch = 8, cex = 0.9,axes = TRUE)
> plot(sample.I.2$X,main ="Clusters structure 2",xlab ="Variable 1",ylab ="
    Variable 2",col = colors[sample.I.2$id+1], pch = 8, cex = 0.9,axes = TRUE)
> plot(sample.I.3$X,main ="Clusters structure 3",xlab ="Variable 1",ylab ="
    Variable 2",col = colors[sample.I.3$id+1], pch = 8, cex = 0.9,axes = TRUE)
> plot(sample.I.4$X,main ="Clusters structure 4",xlab ="Variable 1",ylab ="
    Variable 2",col = colors[sample.I.4$id+1], pch = 8, cex = 0.9,axes = TRUE)
> plot(sample.I.5$X,main ="Clusters structure 5",xlab ="Variable 1",ylab ="
    Variable 2",col = colors[sample.I.5$id+1], pch = 8, cex = 0.9,axes = TRUE)
> plot(sample.I.6$X,main ="Clusters structure 6",xlab ="Variable 1",ylab ="
    Variable 2",col = colors[sample.I.6$id+1], pch = 8, cex = 0.9,axes = TRUE)
```

D.2 Model Fitting

```
> #####WITHOUT PRIOR DISTRIBUTION
> em.I.1=Mclust(sample.I.1[["X"]])
fitting ...
|======================================================================| 100%
> em.I.2=Mclust(sample.I.2[["X"]])
fitting ...
|======================================================================| 100%
> em.I.3=Mclust(sample.I.3[["X"]])
fitting ...
|======================================================================| 100%
> em.I.4=Mclust(sample.I.4[["X"]])
fitting ...
|======================================================================| 100%
> em.I.5=Mclust(sample.I.5[["X"]])
fitting ...
|======================================================================| 100%
> em.I.6=Mclust(sample.I.6[["X"]])
fitting ...
|======================================================================| 100%
> #####WITH PRIOR DISTRIBUTION
> em.I.1.prior=Mclust(sample.I.1[["X"]],prior = priorControl())
```

```
fitting ...
|======================================================================| 100%
Warning message:
In mclustBIC(data = c(0.597461901250151, 0.554711044156686, 0.542122961384872,
    :
The presence of BIC values equal to NA is likely due to one or more of the
    mixture proportions being estimated as zero, so that the model estimated
    reduces to one with a smaller number of components.
> em.I.2.prior=Mclust(sample.I.2[["X"]],prior = priorControl())
fitting ...
|======================================================================| 100%
> em.I.3.prior=Mclust(sample.I.3[["X"]],prior = priorControl())
fitting ...
|======================================================================| 100%
> em.I.4.prior=Mclust(sample.I.4[["X"]],prior = priorControl())
fitting ...
|======================================================================| 100%
> em.I.5.prior=Mclust(sample.I.5[["X"]],prior = priorControl())
fitting ...
|======================================================================| 100%
Warning message:
In mclustBIC(data = c(0.515024567418933, 0.455280390738373, 0.467225794348577,
    :
The presence of BIC values equal to NA is likely due to one or more of the
    mixture proportions being estimated as zero, so that the model estimated
    reduces to one with a smaller number of components.
> em.I.6.prior=Mclust(sample.I.6[["X"]],prior = priorControl())
fitting ...
|======================================================================| 100%
> ####removing the outliers
> sample.II.1=as.data.frame(sample.I.1)
> sample.II.2=as.data.frame(sample.I.2)
> sample.II.3=as.data.frame(sample.I.3)
> sample.II.4=as.data.frame(sample.I.4)
> sample.II.5=as.data.frame(sample.I.5)
> sample.II.6=as.data.frame(sample.I.6)
> ########WITHOUT PRIOR DISTRIBUTION
> em.II.1=Mclust(sample.II.1[sample.II.1$id!=0,][,-3])
fitting ...
|======================================================================| 100%
> em.II.2=Mclust(sample.II.2[sample.II.2$id!=0,][,-3])
```

```
fitting ...
|=============================================================| 100%
> em.II.3=Mclust(sample.II.3[sample.II.3$id!=0,][,-3])
fitting ...
|=============================================================| 100%
> em.II.4=Mclust(sample.II.4[sample.II.4$id!=0,][,-3])
fitting ...
|=============================================================| 100%
> em.II.5=Mclust(sample.II.5[sample.II.5$id!=0,][,-3])
fitting ...
|=============================================================| 100%
> em.II.6=Mclust(sample.II.6[sample.II.6$id!=0,][,-3])
fitting ...
|=============================================================| 100%
> #####WITH PRIOR DISTRIBUTION
> em.II.1.prior=Mclust(sample.II.1[sample.II.1$id!=0,][,-3],prior = priorControl
    ())
fitting ...
|=============================================================| 100%
Warning message:
In mclustBIC(data = c(0.597461901250151, 0.554711044156686, 0.542122961384872,
    :
The presence of BIC values equal to NA is likely due to one or more of the
    mixture proportions being estimated as zero, so that the model estimated
    reduces to one with a smaller number of components.
> em.II.2.prior=Mclust(sample.II.2[sample.II.1$id!=0,][,-3],prior = priorControl
    ())
fitting ...
|=============================================================| 100%
> em.II.3.prior=Mclust(sample.II.3[sample.II.1$id!=0,][,-3],prior = priorControl
    ())
fitting ...
|=============================================================| 100%
> em.II.4.prior=Mclust(sample.II.4[sample.II.1$id!=0,][,-3],prior = priorControl
    ())
fitting ...
|=============================================================| 100%
> em.II.5.prior=Mclust(sample.II.5[sample.II.1$id!=0,][,-3],prior = priorControl
    ())
fitting ...
|=============================================================| 100%
```

```
> em.II.6.prior=Mclust(sample.II.6[sample.II.1$id!=0,][,-3],prior = priorControl
    ())
fitting ...
|============================================================| 100%
> ######EM PLOTS with OUTLIERS
> ###########BIC PLOTS WIThout PRIOR DISTRIBUTION
> plot(em.I.1,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> plot(em.I.2,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> plot(em.I.3,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> plot(em.I.4,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> plot(em.I.5,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> plot(em.I.6,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> ######BIC PLOTS WITH PRIOR DISTRIBUTION
> plot(em.I.1.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> plot(em.I.2.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> plot(em.I.3.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> plot(em.I.4.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> plot(em.I.5.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> plot(em.I.6.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> ###########BIC PLOTS WIThout PRIOR DISTRIBUTION
> plot(em.II.1,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> plot(em.II.2,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> plot(em.II.3,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> plot(em.II.4,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
```

```
> plot(em.II.5,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> plot(em.II.6,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> ######BIC PLOTS WITH PRIOR DISTRIBUTION
> plot(em.II.1.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> plot(em.II.2.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> plot(em.II.3.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> plot(em.II.4.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> plot(em.II.5.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
> plot(em.II.6.prior,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
```

D.3 Extraction of Models

```
> ######WITHOUT PRIOR DISTRIBUTION
> models.I=list(em.I.1,em.I.2,em.I.3,em.I.4,em.I.5,em.I.6,em.I.7,em.I.8,em.I.9,
    em.I.10,em.I.11,em.I.12)
> mod.names.I=rep(0,12)
> loglikes.I=rep(0,12)
> for (i in 1:length(mod.names.I)) {
+   mod.names.I[i]=paste(models.I[[i]]$modelName)
+   loglikes.I[i]=round(models.I[[i]]$loglik,2)}
> mod.names.loglikes.I=as.data.frame(cbind(mod.names.I,loglikes.I))
> mod.names.loglikes.I
mod.names.I loglikes.I
1           EEI       45.09
2           VVI       92.42
3           EII      107.06
4           VVI      178.23
5           EII      191.42
6           EII      258.45
```

```
7           EII      260.17
8           EII       325.4
9           VII       344.6
10          EII      417.59
11          VII      448.67
12          EEI      454.95
> #WITH PRIOR DISTRUBTION
> models.prior.I=list(em.I.1.prior,em.I.2.prior,em.I.3.prior,em.I.4.prior,em.I
    .5.prior,em.I.6.prior,em.I.7.prior,em.I.8.prior,em.I.9.prior,em.I.10.prior,
    em.I.11.prior,em.I.12.prior)
> mod.names.prior.I=rep(0,12)
> loglikes.prior.I=rep(0,12)
> for (i in 1:length(mod.names.prior.I)) {
+   mod.names.prior.I[i]=paste(models.prior.I[[i]]$modelName)
+   loglikes.prior.I[i]=round(models.prior.I[[i]]$loglik,2)}
> mod.names.loglikes.prior.I=as.data.frame(cbind(mod.names.prior.I,loglikes.
    prior.I))
> mod.names.loglikes.prior.I
mod.names.prior.I loglikes.prior.I
1                 EEV             37.67
2                 EVI             84.34
3                 EII            106.86
4                 EVI            169.57
5                 EII             191.3
6                 EII            258.31
7                 EII            269.32
8                 EII            325.29
9                 VII             344.5
10                EII            417.53
11                EII            442.69
12                EEI            454.91
> ######WITHOUT PRIOR DISTRIBUTION
> models.II=list(em.II.1,em.II.2,em.II.3,em.II.4,em.II.5,em.II.6,em.II.7,em.II
    .8,em.II.9,em.II.10,em.II.11,em.II.12)
> mod.names.II=rep(0,12)
> loglikes.II=rep(0,12)
> for (i in 1:length(mod.names.II)) {
+   mod.names.II[i]=paste(models.II[[i]]$modelName)
+   loglikes.II[i]=round(models.II[[i]]$loglik,2)}
> mod.names.loglikes.II=as.data.frame(cbind(mod.names.II,loglikes.II))
> mod.names.loglikes.II
```

```
mod.names.II loglikes.II
1            XXX        40.38
2            EII        87.29
3            EII       112.77
4            VVI       176.24
5            EII       203.93
6            EII       261.46
7            EII       273.72
8            EII       332.09
9            EII       351.96
10           EII        428.7
11           EII       456.08
12           EII       461.83
> #WITH PRIOR DISTRUBTION
> models.prior.II=list(em.II.1.prior,em.II.2.prior,em.II.3.prior,em.II.4.prior,
    em.II.5.prior,em.II.6.prior,em.II.7.prior,em.II.8.prior,em.II.9.prior,em.II
    .10.prior,em.II.11.prior,em.II.12.prior)
> mod.names.prior.II=rep(0,12)
> loglikes.prior.II=rep(0,12)
> for (i in 1:length(mod.names.prior.II)) {
+   mod.names.prior.II[i]=paste(models.prior.II[[i]]$modelName)
+   loglikes.prior.II[i]=round(models.prior.II[[i]]$loglik,2)}
> mod.names.loglikes.prior.II=as.data.frame(cbind(mod.names.prior.II,loglikes.
    prior.II))
> mod.names.loglikes.prior.II
mod.names.prior.II loglikes.prior.II
1                  XXX              39.48
2                  EVI              82.28
3                  EII              98.43
4                  EVI             157.33
5                  EII             178.22
6                  EII             242.06
7                  EII             245.56
8                  EII             297.25
9                  VII             320.89
10                 EII              380.5
11                 EEI             415.05
12                 EII             421.07
```

D.4 Parameter estimates

```
> ###### WITH OUTLIERS
> models=list(em.I.1,em.I.2,em.I.3,em.I.4,em.I.5,em.I.6,em.I.7,em.I.8,em.I.9,em.
    I.10,em.I.11,em.I.12)
> mu.I.11=rep(0,12)
> mu.I.12=rep(0,12)
> mu.I.21=rep(0,12)
> mu.I.22=rep(0,12)
> for (i in 1:length(models)) {
+   mu.I.11[i]=models[[i]][["parameters"]][["mean"]][1,1]
+   mu.I.12[i]=models[[i]][["parameters"]][["mean"]][1,2]
+   mu.I.21[i]=models[[i]][["parameters"]][["mean"]][2,1]
+   mu.I.22[i]=models[[i]][["parameters"]][["mean"]][2,2]
+ }
> mu.I.11
[1] 0.6843162 0.5222578 0.4882314 0.4839767 0.5118005 0.4829479 0.4995861
[8] 0.4927745 0.5033965 0.5069347 0.4919680 0.4994065
> plot(mu.I.11,type = "p",main = "Cluster 1, variable 1 with outliers",ylim = c
    (0.4,0.7))
> abline(h=set.I$Mu[1,1],col="blue")
> mu.I.12
[1] 0.5117079 0.7893086 0.7821216 0.7746909 0.7931024 0.7925557 0.7840114
[8] 0.5309217 0.7968969 0.7884483 0.7810422 0.8000791
> plot(clustersize,xlab = "Cluster size",mu.I.12,type = "p",main = "Mean
    estimates of variable 2 of cluster with outliers",ylab = "Mean estimates",
    ylim = c(0.5,0.8))
> abline(h=set.I$Mu[2,1],col="blue")
> mu.I.21
[1] 0.8997399 0.9763359 0.9609175 0.9640513 0.9590821 0.9859441 0.9627410
[8] 0.9746911 0.9611354 0.9625607 0.9824069 0.9783939
> plot(mu.I.21,type = "p",main = "Cluster 2, variable 1 with outliers",ylim = c
    (0.95,1.5))
> abline(h=set.I$Mu[1,2],col="blue")
> mu.I.22
[1] 1.0173458 0.9026782 0.8885920 0.8894254 0.8752388 0.8950699 0.9056385
[8] 0.5821305 0.9038405 0.8988842 0.9050018 0.8950238
> plot(mu.I.22,type = "p",main = "Cluster 2, variable 2 with outliers",ylim = c
    (0.85,1.5))
> abline(h=set.I$Mu[2,2],col="blue")
```

```
> models.II=list(em.II.2,em.II.3,em.II.4,em.II.5,em.II.6,em.II.7,em.II.8,em.II
    .9,em.II.10,em.II.11,em.II.12)
> mu.II.11=rep(0,11)
> mu.II.12=rep(0,11)
> mu.II.21=rep(0,11)
> mu.II.22=rep(0,11)
> for (i in 1:length(models.II)) {
+   mu.II.11[i]=models.II[[i]][["parameters"]][["mean"]][1,1]
+   mu.II.12[i]=models.II[[i]][["parameters"]][["mean"]][1,2]
+   mu.II.21[i]=models.II[[i]][["parameters"]][["mean"]][2,1]
+   mu.II.22[i]=models.II[[i]][["parameters"]][["mean"]][2,2]
+ }
> mu.II.11
[1] 0.5221340 0.4877073 0.4840477 0.5111839 0.4829774 0.4978942 0.4925588
[8] 0.5053139 0.5063084 0.4898240 0.4993805
> plot(mu.II.11,type = "p",main = "Cluster 1, variable 1",ylim = c(0.4,0.7))
> abline(h=set.I$Mu[1,1],col="blue")
> mu.II.12
[1] 0.7894291 0.7928415 0.7747235 0.7921951 0.7920578 0.7859843 0.7837103
[8] 0.7972634 0.7892107 0.7795251 0.7988932
> plot(clustersize,c(NA,mu.II.12),xlab = "Cluster size",ylab = " Mean estimates"
    ,type = "p",main = "Mean estimates of variable 2 in cluster without outliers
    ",ylim = c(0.5,0.8))
> abline(h=set.I$Mu[2,1],col="blue")
> mu.II.21
[1] 0.9763373 0.9627150 0.9640464 0.9590469 0.9859395 0.9688309 0.9747094
[8] 0.9605548 0.9646476 0.9827212 0.9785726
> plot(mu.II.21,type = "p",main = "Cluster 2, variable 1",ylim = c(0.95,1.5))
> abline(h=set.I$Mu[1,2],col="blue")
> mu.II.22
[1] 0.9026717 0.8979929 0.8894000 0.8858507 0.8934312 0.9072065 0.9077456
[8] 0.9088178 0.9015750 0.9054079 0.8987720
> plot(mu.II.22,type = "p",main = "Cluster 2, variable 2",ylim = c(0.85,1.5))
> abline(h=set.I$Mu[2,2],col="blue")
```

D.5 Outlier detection

```
> library(OutlierDetection)
> maha(sample.I.1$X,cutoff = 0.99)
$`Outlier Observations`
[1] 0.5243535 0.5543725

$`Location of Outlier`
[1] 22

$`Outlier Probability`
[1] 0.9921448

$`Scatter plot`

> maha(sample.I.2$X,cutoff = 0.99)
$`Outlier Observations`
[,1]      [,2]
[1,] 0.9643121 0.5572394
[2,] 0.6202292 0.5583100

$`Location of Outlier`
[1] 41 42

$`Outlier Probability`
[1] 0.9977415 0.9996573

$`Scatter plot`

> maha(sample.I.3$X,cutoff = 0.99)
$`Outlier Observations`
[1] 0.6108471 0.6982496

$`Location of Outlier`
[1] 61

$`Outlier Probability`
[1] 0.9914425

$`Scatter plot`

> depthout(sample.I.1$X,cutoff = 0.04, boottimes = 100)
Error in quantile.default(bootlbnorm, cutoff) :
```

```
missing values and NaN's not allowed if 'na.rm' is FALSE
In addition: Warning messages:
1: In max(aa) : no non-missing arguments to max; returning -Inf
2: In max(aa) : no non-missing arguments to max; returning -Inf
> depthout(sample.I.2$X,cutoff = 0.04, boottimes = 100)
$`Outlier Observations`
[1] V1 V2
<0 rows> (or 0-length row.names)

$`Location of Outlier`
integer(0)

$`Outlier Probability`
NULL

$`Scatter plot`

> depthout(sample.I.3$X,cutoff = 0.04, boottimes = 100)
$`Outlier Observations`
[1] V1 V2
<0 rows> (or 0-length row.names)

$`Location of Outlier`
integer(0)

$`Outlier Probability`
NULL

$`Scatter plot`
```

```
> library(readxl)
> Mammography_dataset=read_excel("Masters Research/Mammography dataset.xlsx")
> summary(Mammography_dataset)
attr1              attr2               attr3              attr4
Min.   :-0.7844   Min.   :-0.47019   Min.   :-0.5916   Min.   :-0.8596
1st Qu.:-0.7844   1st Qu.:-0.47019   1st Qu.:-0.5916   1st Qu.:-0.8596
Median :-0.1086   Median :-0.39499   Median :-0.2310   Median :-0.8596
Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
3rd Qu.: 0.3139   3rd Qu.:-0.07649   3rd Qu.: 0.2198   3rd Qu.: 0.8202
Max.   :31.5084   Max.   : 5.08585   Max.   :29.4778   Max.   : 9.5912
```

```
attr5              attr6              class
Min.   :-0.3779   Min.   :-0.9457   Min.   :-1.0000
1st Qu.:-0.3779   1st Qu.:-0.9457   1st Qu.:-1.0000
Median :-0.3779   Median :-0.9457   Median :-1.0000
Mean   : 0.0000   Mean   : 0.0000   Mean   :-0.9535
3rd Qu.:-0.3779   3rd Qu.: 1.0166   3rd Qu.:-1.0000
Max.   :23.6171   Max.   : 1.9490   Max.   : 1.0000
>
> ####OUTLIER DETECTION
> mahalanobis=maha(Mammography_dataset[,-7],cutoff = 0.99997)
> mahalanobis$`Outlier Observations`
# A tibble: 268 x 6
attr1  attr2   attr3  attr4  attr5  attr6
<dbl>  <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
1 -0.784 -0.444   5.67  -0.860 -0.378 -0.946
2  7.47  -0.448  -0.592 -0.860 -0.378 -0.946
3  0.336 -0.121   0.175  5.50  -0.378  0.639
4  6.90  -0.430  -0.547 -0.860 -0.378 -0.946
5 -0.335 -0.311  20.7    2.54   0.358  0.651
6  5.45  -0.408  -0.501 -0.860 -0.378 -0.946
7 -0.784 -0.453   6.17  -0.860 -0.378 -0.946
8 -0.784 -0.439   9.06  -0.860 -0.378 -0.946
9  1.05   5.04   -0.186 -0.860 -0.378 -0.946
10  7.22  -0.453  -0.501 -0.860 -0.378 -0.946
# ... with 258 more rows
> mahalanobis$`Location of Outlier`
[1]      3     19    129    259    360    377    425    484    541    558    626
[12]   682    744    812    841    878    916   1052   1060   1067   1097   1098
[23]  1099   1103   1107   1110   1150   1231   1264   1279   1284   1302   1324
[34]  1380   1410   1483   1551   1561   1611   1625   1629   1758   1761   1837
[45]  1896   1918   1956   1985   2041   2054   2103   2220   2221   2222   2223
[56]  2226   2227   2235   2245   2314   2398   2474   2560   2664   2699   2724
[67]  2797   2883   2899   2930   2976   3122   3190   3333   3334   3336   3338
[78]  3346   3348   3353   3357   3389   3406   3449   3546   3608   3615   3694
[89]  3710   3765   3784   3800   3884   4009   4036   4127   4161   4181   4379
[100]  4394   4445   4453   4456   4465   4470   4471   4475   4504   4549   4677
[111]  4684   4791   4826   4909   4912   4973   5057   5114   5175   5184   5208
[122]  5241   5260   5266   5442   5538   5543   5571   5575   5578   5583   5587
[133]  5588   5626   5719   5807   6010   6032   6035   6074   6245   6246   6311
[144]  6372   6394   6399   6488   6559   6675   6687   6688   6690   6698   6700
[155]  6704   6708   6711   6717   6758   6788   6842   6899   6992   7112   7319
```

```
[166]  7321  7335  7350  7354  7415  7421  7451  7453  7511  7530  7658
[177]  7797  7814  7816  7817  7820  7821  7823  7825  7918  7921  7942
[188]  8023  8040  8109  8147  8275  8279  8355  8377  8399  8416  8421
[199]  8427  8498  8695  8835  8868  8871  8901  8922  8928  8930  8931
[210]  8936  8938  8942  8943  8945  8947  8957  8999  9031  9062  9100
[221]  9111  9202  9241  9267  9268  9379  9381  9387  9449  9496  9668
[232]  9705  9758  9857  9893  9898  9900  9947 10005 10047 10048 10053
[243] 10155 10159 10206 10322 10351 10396 10476 10511 10658 10675 10853
[254] 10929 10943 10994 11009 11052 11143 11160 11166 11167 11172 11173
[265] 11175 11177 11181 11183
> mahalanobis$`Outlier Probability`
[1] 0.9999995 1.0000000 1.0000000 1.0000000 1.0000000 0.9999999 1.0000000
[8] 1.0000000 0.9999768 1.0000000 1.0000000 1.0000000 0.9999741 1.0000000
[15] 1.0000000 0.9999824 1.0000000 0.9999882 0.9999896 1.0000000 1.0000000
[22] 1.0000000 1.0000000 0.9999851 1.0000000 0.9999998 0.9999797 1.0000000
[29] 0.9999995 1.0000000 1.0000000 0.9999921 1.0000000 1.0000000 0.9999759
[36] 0.9999842 0.9999781 1.0000000 0.9999740 0.9999941 1.0000000 1.0000000
[43] 1.0000000 1.0000000 1.0000000 0.9999889 1.0000000 1.0000000 1.0000000
[50] 0.9999795 0.9999723 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[57] 0.9999965 1.0000000 1.0000000 1.0000000 0.9999999 0.9999777 0.9999888
[64] 1.0000000 1.0000000 1.0000000 0.9999998 0.9999773 1.0000000 1.0000000
[71] 1.0000000 0.9999827 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[78] 1.0000000 0.9999994 1.0000000 1.0000000 0.9999996 0.9999800 1.0000000
[85] 0.9999726 1.0000000 1.0000000 0.9999782 0.9999993 1.0000000 0.9999743
[92] 1.0000000 1.0000000 0.9999829 1.0000000 0.9999993 0.9999869 0.9999971
[99] 1.0000000 0.9999999 0.9999807 1.0000000 1.0000000 1.0000000 1.0000000
[106] 1.0000000 1.0000000 1.0000000 0.9999896 0.9999997 0.9999868 1.0000000
[113] 1.0000000 0.9999933 1.0000000 0.9999970 0.9999981 0.9999999 0.9999820
[120] 0.9999983 0.9999999 0.9999905 1.0000000 0.9999747 1.0000000 0.9999861
[127] 1.0000000 1.0000000 1.0000000 1.0000000 0.9999984 1.0000000 1.0000000
[134] 1.0000000 0.9999705 0.9999836 0.9999768 1.0000000 0.9999706 1.0000000
[141] 0.9999723 0.9999745 0.9999805 0.9999751 0.9999999 0.9999998 0.9999901
[148] 0.9999998 0.9999829 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
[155] 0.9999967 1.0000000 1.0000000 1.0000000 1.0000000 0.9999994 0.9999921
[162] 0.9999999 0.9999964 0.9999927 0.9999997 0.9999991 1.0000000 1.0000000
[169] 0.9999994 1.0000000 0.9999907 1.0000000 0.9999839 1.0000000 1.0000000
[176] 0.9999780 1.0000000 0.9999991 1.0000000 1.0000000 0.9999877 1.0000000
[183] 1.0000000 1.0000000 1.0000000 0.9999880 0.9999738 0.9999957 1.0000000
[190] 1.0000000 1.0000000 1.0000000 1.0000000 0.9999960 0.9999859 0.9999999
[197] 1.0000000 1.0000000 1.0000000 0.9999975 0.9999945 0.9999834 0.9999998
[204] 0.9999959 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.9999923
```

```
[211] 0.9999996 1.0000000 1.0000000 1.0000000 1.0000000 0.9999731 1.0000000
[218] 0.9999710 1.0000000 0.9999957 0.9999978 1.0000000 1.0000000 1.0000000
[225] 0.9999920 0.9999866 1.0000000 0.9999788 1.0000000 0.9999949 1.0000000
[232] 0.9999720 0.9999783 1.0000000 1.0000000 1.0000000 1.0000000 0.9999889
[239] 1.0000000 1.0000000 0.9999957 0.9999959 0.9999822 0.9999854 0.9999814
[246] 0.9999753 1.0000000 1.0000000 1.0000000 0.9999913 0.9999922 1.0000000
[253] 0.9999821 1.0000000 1.0000000 0.9999754 0.9999986 1.0000000 0.9999998
[260] 1.0000000 1.0000000 0.9999999 0.9999999 1.0000000 1.0000000 0.9999998
[267] 1.0000000 1.0000000
> plot(mahalanobis$`Location of Outlier`,mahalanobis$`Outlier Probability`, main
     = "Outlier location and probability for a Mammography dataset",xlab = "
    Location of outlier",ylab = "Outlier Probability",col="blue")
> length(mahalanobis$`Outlier Probability`)
[1] 268
>
> depth1=depthout(Mammography_dataset[,-7], rnames = FALSE, cutoff = 0.028,
    boottimes = 100)
> plot(depth1$`Location of Outlier`,depth1$`Outlier Probability`, main = "
    Outlier location and probability for a Mammography dataset",xlab = "Location
     of outlier",ylab = "Outlier Probability",col="blue")
> length(depth1$`Outlier Observations`)
[1] 6
> length(depth1$`Outlier Probability`)
[1] 284
> library(mclust)
> real.data=Mclust(Mammography_dataset[,-7])
fitting ...
|============================================================| 100%
> plot(real.data,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol
    =1,cex=0.8,inset=0.005))
> real.data$modelName
[1] "EEV"
> real.data$loglik
[1] -23309.87
> real.data1$modelName
[1] "VVV"
> real.data1$loglik
[1] 72441.71
> real.data1=Mclust(Mammography_dataset[Mammography_dataset$class!=1,][,-7],
    prior = priorControl())
fitting ...
```

```
|============================================================| 100%
Warning message:
In mclustBIC(data = c(0.23001961, 0.15549112, -0.78441482, 0.54608818,  :
The presence of BIC values equal to NA is likely due to one or more of the
    mixture proportions being estimated as zero, so that the model estimated
    reduces to one with a smaller number of components.
> plot(real.data1,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol
    =1,cex=0.8,inset=0.005))
>
> nonoutliers=Mammography_dataset[Mammography_dataset$class!=1,]
> outliers=Mammography_dataset[Mammography_dataset$class!=-1,]
>
> index=sample(1:nrow(nonoutliers),260)
> newnonoutliers=nonoutliers[-index,]
> newMammography=rbind(newnonoutliers,outliers)
> nrow(newMammography)
[1] 10923
> newmodel=Mclust(newMammography[,-7])
fitting ...
|============================================================| 100%

> ##############################################################################
> Lymphography=read_excel("Masters Research/Lymphography.xlsx")
> lymph.mahalanobis=maha(Lymphography[,-1],cutoff = 0.995)
> lymph.mahalanobis$`Location of Outlier`
[1]  15  25  37  42  53  66 140
> Lymphography[lymph.mahalanobis$`Location of Outlier`,]
# A tibble: 7 x 19
class lymphatics `block of affer~ `bl. of lymph. ~ `bl. of lymph. ~
<dbl>      <dbl>            <dbl>             <dbl>             <dbl>
1     0          3                2                 2                 2
2     3          3                1                 1                 1
3     0          3                1                 1                 1
4     3          2                2                 2                 1
5     2          3                1                 1                 1
6     3          4                1                 1                 1
7     0          3                1                 1                 1
# ... with 14 more variables: `by pass` <dbl>, extravasates <dbl>,
#   `regeneration of` <dbl>, `early uptake in` <dbl>, `lym.nodes
#   dimin` <dbl>, `lym.nodes enlar` <dbl>, `changes in lym.` <dbl>, `defect
#   in node` <dbl>, `changes in node` <dbl>, `changes in stru` <dbl>,
```

```
#   `special forms` <dbl>, `dislocation of` <dbl>, `exclusion of no` <dbl>,
#   `no. of nodes in` <dbl>
> plot(lymph.mahalanobis$`Location of Outlier`,lymph.mahalanobis$`Outlier
    Probability`, main = "Outlier location and probability for a Lymphography
    dataset",xlab = "Location of outlier",ylab = "Outlier Probability",col="blue
    ")
> lymph.depth=depthout(Lymphography[,-1], rnames = FALSE, cutoff = 0.1,
    boottimes = 100)
> lymph.depth$`Location of Outlier`
[1]   3  15  37  45  48  59  79  91 132 137 140 145
> Lymphography[lymph.depth$`Location of Outlier`,]
# A tibble: 12 x 19
class lymphatics `block of affer~ `bl. of lymph. ~ `bl. of lymph. ~
<dbl>      <dbl>             <dbl>             <dbl>             <dbl>
1      3           3                 2                 2                 2
2      0           3                 2                 2                 2
3      0           3                 1                 1                 1
4      0           3                 2                 2                 2
5      2           4                 1                 1                 1
6      3           4                 2                 2                 2
7      0           1                 1                 1                 1
8      3           4                 2                 2                 1
9      3           4                 2                 2                 2
10      0           1                 1                 1                 1
11      0           3                 1                 1                 1
12      2           2                 1                 1                 1
# ... with 14 more variables: `by pass` <dbl>, extravasates <dbl>,
#   `regeneration of` <dbl>, `early uptake in` <dbl>, `lym.nodes
#   dimin` <dbl>, `lym.nodes enlar` <dbl>, `changes in lym.` <dbl>, `defect
#   in node` <dbl>, `changes in node` <dbl>, `changes in stru` <dbl>,
#   `special forms` <dbl>, `dislocation of` <dbl>, `exclusion of no` <dbl>,
#   `no. of nodes in` <dbl>
> plot(lymph.depth$`Location of Outlier`,lymph.depth$`Outlier Probability`, main
     = "Outlier location and probability for a Lymphography dataset",xlab = "
    Location of outlier",ylab = "Outlier Probability",col="blue")
> lymph.em=Mclust(Lymphography[,-1],prior = priorControl())
fitting ...
|=============================================================| 100%
> LymphoGraph=Lymphography[Lymphography$class!=1,]
> lymph.em.nout=Mclust(LymphoGraph[LymphoGraph$class!=4,][,-1],prior =
    priorControl())
```

```
fitting ...
|=================================================================| 100%
> plot(lymph.em,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",ncol=1,
    cex=0.8,inset=0.005))
> plot(lymph.em.nout,main = TRUE,what = "BIC",legendArgs=list(x="bottomright",
    ncol=1,cex=0.8,inset=0.005))
```