Mathematical Institute

Master Thesis

Statistical Science for the Life and Behavioural Sciences

Methods for Clustering Data with Missing Values

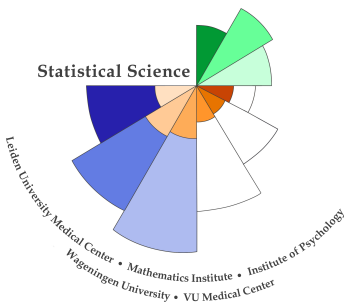
STEFAN E. WILSON

SUPERVISOR: DR. G. GORT (WUR)

2^{nd} SUPERVISOR: M. KAMPERT (LU)

DECEMBER 2015

Universiteit Leiden
The Netherlands



Abstract

In the world of clustering methodology, there exists a plethora of options. The choice becomes especially important when the number of clusters is not known a priori. Methods to handle missing data also have vast variation and these choices are often made based on the data missing mechanism. In this paper we seek to investigate the intersection of both these situations: Clustering, where one of the major objectives is cluster discovery, and doing so in the presence of missing values.

Model-based clustering estimates the structure of clusters, (number, size and distribution of clusters) using likelihood approaches. Likelihood methods also allow researchers to gain information from incomplete observations. In the following work, we will investigate adaptations of these likelihood estimations to infer cluster information about a given data set. Model-based clustering becomes the focal point because of the objectivity in cluster discovery, and for continuous data, its multivariate Gaussian density assumptions can be an asset to handling the problem of missing data.

An algorithm that utilises marginal multivariate Gaussian densities for assignment probabilities, was developed and tested versus more conventional ways of model-based clustering for incomplete data. These conventional methods included multiple imputation and using complete observations only. Assumptions of the data missing mechanism were important and taken into consideration during the testing of these methods. These assumptions were especially important for the model-based method when parameters had to be updated. All methods were tested using simulated data as well as real life publicly available data.

It was found that for cases with many observations, the complete case and multiple imputation have advantages over the marginal density method due to the increased availbiltity of disposable information and borrowable information respectively. Dimensionality and cluster separation were also important factors. Multiple Imputation was the preferred method when our data structure was more complicated (high dimensions, high cluster overlap), however in simpler settings, the marginal method worked best. The marginal method also showed significant promise in classifying observations to their clusters. The marginal method can be further adapted by making more robust parameter estimates and is discussed in this paper.

Contents

1 Introduction

It is often the goal of scientific research to group observations of certain phenomena based on their similarities, and simultaneously, separate groups of observations based on their differences. In various fields of study the execution of this ideal is needed: astronomy, character recognition, marketing, psychological analyses and many others. In this paper, we will be discussing the grouping mechanism known as clustering. For clustering, the objective is often to discover subpopulations that we may not have known to exist a priori. In this sort of exploratory research, a researcher may have little idea on how many groups exist. There are different types of clustering methods, and each method proposes a different criterion to determine how many separate groups exist in the data.

Clustering techniques are diverse and can be categorized in many different ways: hard versus fuzzy clustering, partitional versus hierarchical clustering and so on. Some clustering methods seek to cluster objects by analyzing the dissimilarity of the observations, or minimizing the distance between objects and a centroid measure. These are known as hierarchical and k-means clustering respectively and the two will be briefly discussed below. The drawbacks of some of the clustering methods are also highlighted below and because of this, our focus will be centered around model-based clustering which is well documented in the work done by McLachlan and Peel [11]. In clustering, one would like to arm the researcher with the power of discovery, and model based clustering has an objective criterion for doing this as can be seen in the work done by Fraley and Raftery [7]. We will take a closer look at the added advantages of model-based clustering with its abilities to model multiple covariance structures and test each, as well as its ability to handle data in high dimensions. Both of these ideals are carried out by first performing an eigenvalue decomposition of the covariance matrix. Outlines of this procedure will be discussed in this thesis and details can be found in the literature by Raftery and Fraley, as well as Bouveyron [8, 5].

The work in this thesis is not only focused on clustering but also the common occurrence of missing data. Extensive research has been done on the issue of missing data and these findings and conclusions can be found in various literature, where the nature of the missing values are discussed as well as treatments under different circumstances [9]. However, not very much has been said of combining the two problems of clustering in the presence of missing data. There is some work to be found in this intersection of topics

and it is suggested to utilize multiple imputation along with the desired clustering method to solve the problem [4]. There are also other discussions to utilize the EM algorithm to attain maximum likelihood estimates of the desired parameters in the presence of incomplete data [10]. Depending on the nature of the missing values, handling the problem in an unsuitable way can easily lead to biased results [2].

To handle the problem of clustering data with missing values, we look carefully at the method of multiple imputation combined with clustering [4], while keeping in mind the criticisms of multiple imputation [1]. Alternatively, we try a method that does not attempt to fill in the missing data. This method centers around the fact that for model-based clustering in more than one dimension, we assume that the the data comes from a joint multivariate Gaussian distribution. A useful property of the multivariate Gaussian, is that its marginal densities are also Gaussian. This property can be used to attain cluster assignment probabilities for an observation, even when full information is not available for that observation. In addition, model-based clustering requires the estimation of parameters, and we discuss methods for updating these parameters, where the update method largely depends on the nature of the missing data.

A simulation is then performed to address the ability to cluster with missing information using multiple imputation, marginal density approaches as well as the complete case method which deletes incomplete observations. In order for this to be a "fair" comparison, our simulation is done on data where the missing values are completely at random (complete case analysis is only suitable when data is missing at random [9]). The estimation of parameters utilizing the marginal probability approach, is performed also under the assumption that the missing data is a random phenomenon. In the development of this method, we shall also discuss an unbiased parameter estimation method for when the data is missing but not at random, via a maximum likelihood approach [10]. In this thesis we seek to discover which method is superior when one wishes to cluster data that has random missing values. Results of our simulation under random missingness will then be discussed. Lastly, we apply our methods to real life data, and assess their performances.

2 Motivation

In a study entitled "Unraveling the Malaria mosquito's sense of smell" (2011), done by Remco A. Suer, a former PhD student at the Entomology department of Wageningen University, the neuron activity of mosquitoes in response to odor stimuli was analyzed. The researcher wanted to uncover various functional types of neurons; neurons that behave differently for different odor compounds. In essence, he wanted to do a cluster analysis on the response of the neurons.

Previous studies have identified the presence of many different type of odor sensors on the mosquito antennae known as olfactory sensilla. Trichoid sensilla (TS) are the most abundant olfactory sensilla on the antennae of female mosquitoes [14]. Amongst the trichoid sensilla, researchers have found 5 distinct types labeled, A-E, however in this study only sensilla A, B and D were analyzed. This analysis consisted of exposing the mosquito to 132 compounds, and analyzing the electric pulses exhibited when each compound is passed, giving us the power to identify which odors are identified by the insect. According to the responses, each trichoid sensilla consisted of two olfactory neurons, hereby labeled neuron A and neuron B, however not to be confused with the different trichoid sensilla types. These two neurons did not exhibit any activity for trichoid sensilla A (TSA), so the analyses from the study were only feasible for TSB and TSD. Due to restrictions, the researcher was unable to answer the many questions concerning the raw data for TSD which were required for necessary data cleaning and comprehension. For this reason, for the remainder of this paper, all references to the mosquito study will pertain to TSB (neurons A and B).

The execution of this experiment required that a single mosquito had its legs removed and was taped to a surface. The antennae were fixed at a 90 degree angle, and two electrodes were put into contact with the mosquito: one at the base of the sensillum just piercing the cuticle, while the other electrode was placed in the eye of the mosquito. Many of the mosquitoes did not survive this initial process, and the rest could not survive these conditions for 132 chemical compounds. Also the neuron activity decreased as time passed due to the dwindling strength of the insect, so wisely the researcher randomized the order of induced compounds. However, these conditions still gave rise to a large amount of missing values due to the inability to test all surviving mosquitoes across all compounds, and a small sample space due to dead mosquitoes, a statistical challenge which was made more severe due to

the high dimensional aspect ($p > n$).

Figure 1 shows the missing patterns for the mosquito data, giving rise to 20 mosquitoes on the y-axis and their response or lack of response to 73 compounds on the x-axis. This is different from the initial objective of 132 compounds, and it shows that no mosquito was able to survive the entire process, as well as illustrating that the order of the compounds were not sufficiently randomized. The occurrence of missing values is extremely high; 36.4% of the data is missing. However it was observed that three mosquitoes only responded to 5 or less compounds out of the total 73, and were therefore deleted. After this deletion, our data comprises of 17 mosquitoes, 73 compounds (dimensions) with 26.3% of the information is missing.

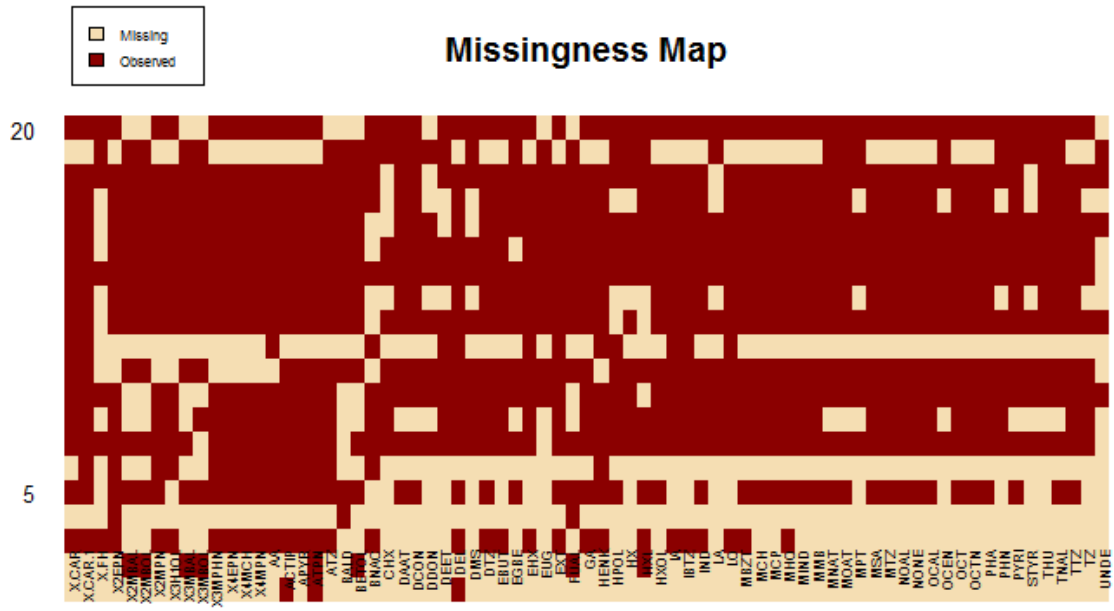


Figure 1: Degree of missing values. Individual mosquitoes (20) on the rows, and 73 tested compounds on the columns

Due to the high frequency of missing data, the researcher greatly reduced the sample space and dimensionality (12 mosquitoes with 40 compounds). A

hierarchical cluster analysis using squared Euclidean distances and Ward's method was utilized, but as discussed later in this thesis, there may be disadvantages to this method due to the somewhat subjective nature of choosing the number of clusters, and the treatment of the missing values. In fact, the critique of this mosquito study [14], pointed to these same issues: how do you justify the number of clusters and why not utilize all information? For TSB neuron A, 4 different response types were found [14]. The number of clusters were determined by plotting varying number of clusters against the within cluster sum of squared errors (W.S.S., explained further in the next section) and looking for the elbow as seen in Figure 2 (similar to a scree plot analysis). This method of determining the number of clusters utilizes a k-means framework (discussed later) and this comes with its own issues. The figures below show his results:

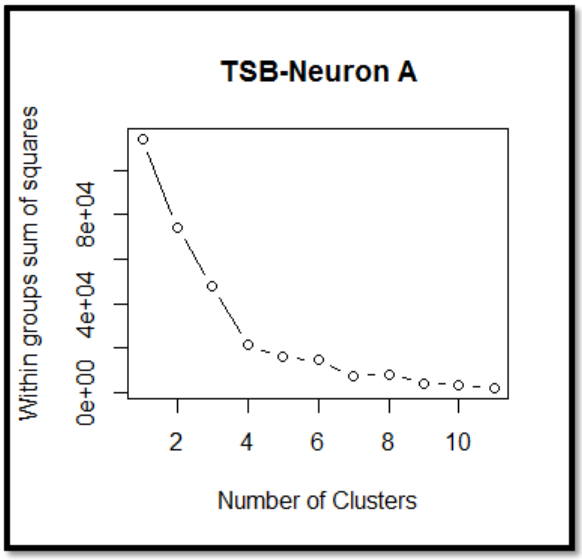


Figure 2: Plot of W.S.S. against number of clusters. Elbow at 4 clusters

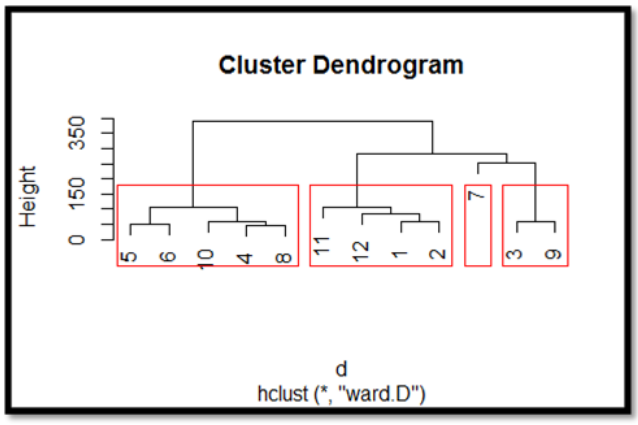


Figure 3: Dendrogram showing allocations based on 4 clusters

Accepting that this data for this experiment is quite problematic, it still gave rise to an interesting situation. The number of clusters can be found objectively via model-based clustering, as it compares various numbers of clusters and chooses the ideal fit based on fit statistics (more on this later). However, this method is unable to work with missing values. Hierarchical clustering can work with missing values but it makes the number of cluster determination slightly subjective, especially in cases when the W.S.S. plot does not give a clear indication of a suitable cut-off point. We shall take a look now at various clustering methods, their ability or inability to handle missing values, and what can be done to alleviate this problem.

3 Clustering Methods

In the world of cluster analysis, various methods are present. The choice of method is often made based on the nature of the research, the type of data or even the ease of execution of the clustering method. In this section we will discuss some of the types of clustering methods namely: hierarchical clustering, k-means clustering and model-based clustering.

For k-means clustering methods, the a priori knowledge of number of clusters is required. In the mosquito research, and as a common practice, this is determined via a W.S.S. plot. To attain this plot, measures of the sum of squared deviations from each observation to a central measure (usually the mean), are plotted against the number of possible clusters. The sum of squared errors goes from its highest value when the number of clusters is 1, to zero when the number of clusters is equal to the number of observations. A disadvantage of using this plot to determine the number of clusters is that it is centered around the assumptions of spherical clusters assuming uncorrelated variables, equal weighting of variables plus the added hazard of outliers. Secondly, we cannot attain this plot when there are missing values. Lastly, sometimes the plot may be quite ambiguous and the "elbow" we seek to identify is quite unclear and subject to personal discretion. An example of the potential ambiguity can be seen below in the squared errors plot applied to the complete cases for TSB-neuron B. A researcher can argue for 2, 3, 4 or even 5 clusters:

There is hope however and it is found in the R-package NbClust.

NbClust: Determining the optimum number of clusters may be tricky, and various measures to estimate this have been proposed. This package computes the number of clusters according to 27 different measures, based on the clustering method desired, and gives the user the optimal number of clusters via the majority rule. Details of the measures, and the adjustments made for each clustering method are found in the documentation of the package. There are two disadvantages of this method for our situation: the algorithm cannot work with missing values and cannot handle high dimensional data (where $p > n$).

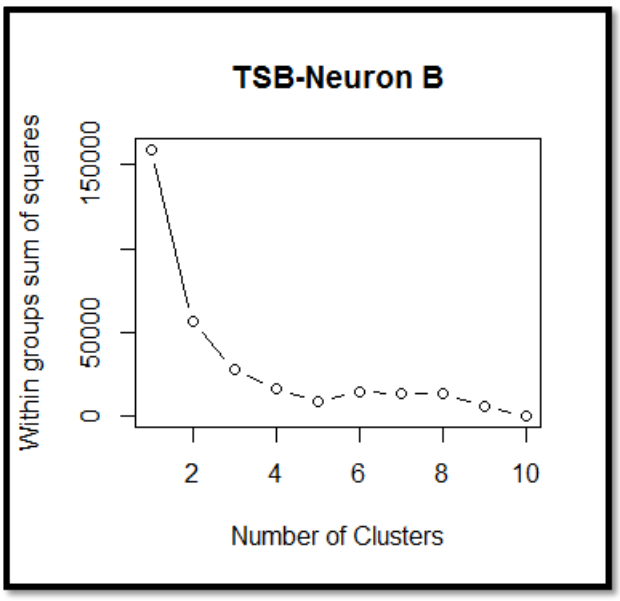


Figure 4: Plot of W.S.S. against number of clusters. Unclear Elbow

3.1 K-means (Partitioning Clustering)

This method of clustering seeks to assign observations to k groups. The number of groups, k, is specified before the algorithm takes place. These clusters are usually centered around the cluster mean, and the algorithm seeks to minimize the squared error between each observation and the cluster mean. The initial starting grouping for the observations is random, and observations are re-assigned depending on its relative closeness to a different cluster mean. The algorithm then updates the centroid measures based on the new assignments and the algorithm repeats the process until no improvement is realized with the squared error.

The most commonly used central measure is the mean. However, in the presence of outliers, the median may be preferred as a more robust measure for the centroid location. Regardless of the choice of centroid, the algorithm seeks to minimize:

$$SSE = \sum_{k=1}^{K} \sum_{x \in C_k} (x - c_k)^2 \tag{1}$$

Equation 1 gives the within cluster sum of squared errors, used in the plots

discussed previously (Figures 2 and 4). The index k specifies a given cluster, c_k is the central measure, and x are the observations belonging to that central measure, hence the subscript $x \in C_k$, where C_k are the observations that are part of the cluster centered by the particular centroid c_k. Whichever cluster centroid provides the smallest squared error with a particular observation, that observation is assigned to the respective cluster. The centroids are then updated based on the observations assigned to its cluster. Below we see the update equation where the centroid measure is the mean of a cluster, which is then substituted back into Equation 1.

$$c_k = \frac{1}{m_k} \sum_{x \in C_k} x \tag{2}$$

Criticism of standard k-means

- The K-means has the major flaw that it needs the user to specify how many clusters are present initially. There are however, some measures to approximate the number of clusters as seen in NbClust.
- The initial random assignment of observations can greatly affect the outcome of the clusters.
- The effect of outliers can greatly skew the central measure, especially when the mean is used.
- The clustering algorithm weighs each variable equally and this may not be the preferred approach with multivariate data.
- The clusters are always assumed to be spherically shaped, assuming uncorrelated variables, which may not be realistic as correlations that cause ellipsoid shapes are quite common.

K-means is still utilized because of its ease of comprehension and its low computational cost.

3.2 Hierarchical Clustering

This method will require a proximity measure, meaning how similar or different observations are. The idea is that data showing close proximity to each other will be in a cluster together and those with far proximity would

be in separate clusters. However before we discuss these measures, the type of data must be defined because this determines what proximity measure we use. Due to the data for the mosquito research, we will focus on quantitative, continuous data. For this type of data we can compute a distance metric known as the Minkowski metric. For two observations (i^{th} and j^{th} observations), the distance between them is given by:

$$d(x_i, x_j) = (|x_{i1} - x_{j1}|^g + |x_{i2} - x_{j2}|^g + |x_{i3} - x_{j3}|^g + ...|x_{ip} - x_{jp}|^g)^{\frac{1}{g}} \quad (3)$$

In this equation, p is the number of dimensions/variables and the value g governs the type of dissimilarity measure we use. For $g = 2$, we get the commonly used Euclidean distance whereas $g = 1$ gives the Manhattan distance. A value of $g = \infty$, gives the maximum or Chebychev distance. The choice of distance measure used can greatly affect the clustering analysis, and to avoid this dependence, data should first be standardized (L. Rokach and O. Maimon, 2005). The desired distance measure could then be applied to the Hierarchical Clustering algorithm.

As the name suggests, this is a hierarchical decomposition of the data from either the top to the bottom or vice versa. It can take two forms: agglomerative, where each data point starts as its own cluster, and observations are sequentially joined together, or divisive, where we start with all data points as one cluster and then sequentially separate them. For the agglomerative approach, because we need to merge the closest clusters we specify the distance criteria between two clusters as one of many types:

1. **Single Linkage:** The distance between two clusters is equal to the distance between its two closest points.

2. **Complete Linkage:** The distance between two clusters is equal to the distance between its two furthest points.

3. **Average Linkage:** The distance between two clusters is equal to the average of all distances between elements of each cluster.

4. **Ward's Minimum Variance:** The distance between two clusters is simply the increase in sum of squared errors when the clusters are merged. This methods tries to keep the "cost of merging" as low as possible.

The disadvantage with Single and Complete Linkage is that they are both easily affected by extreme values and can lead to elongated or compact clusters respectively. The Average Linkage method is therefore preferred as a suitable compromise between these two. Still the most commonly used distance criteria is Ward's Minimum Variance because of its ease of comprehension. Ward's method is performed in the research of the malaria mosquito as seen before.

Issues with Hierarchical Clustering

- This algorithm is computationally expensive, with a computation costs that increases quadratically with the number of observations.
- Once a partitioning or division has occurred, an observation cannot be reassigned making it a one-way algorithm. In fact, all partitions are nested within each other (going for 3 clusters to 4 clusters just means that one of the previous clusters is split into two parts).
- No objective function is directly minimized.
- The choice of number of clusters is determined by the analyst/researcher, and is quite subjective and therefore undesirable for exploratory research. The Agglomerative algorithm for example, continues to unify clusters until only one cluster remains of all the data, or until the user decides at what distance he/she would like to stop unifying clusters. Methods are present to determine the possible number of clusters like looking for the elbow when plotting the within cluster sum of squared error as discussed in previous sections. However, this does not perform ideally as seen in the 1996 study by Arabie, Hubert and Soete, where the number of clusters in 480 data sets, were correctly predicted 58% of the time.

3.3 Model-based Clustering

Model-based Clustering, as the name suggests, seek to cluster the data by modeling the behavior of the data as a mixture of distributions. When doing regular hierarchical and K-means cluster analysis, the clusters are assumed to be spherical, meaning each variable has the same variance and variables are uncorrelated. This of course is a very strong assumption, especially in the

presence of high dimensional data, where we expect variables to have some correlation thus giving a more ellipsoidal shape to the cluster distributions. The notion for the mixture model was first introduced by Karl Pearson in his groundbreaking paper, where he fitted two normal distributions with different means, variances and mixing proportions, to explain the distribution of crab data supplied by his colleague Weldon [13]. After plotting the histogram of 1000 measurements of forehead to body length ratio, Pearson detected some asymmetry in the distribution and thought it might be explained by some subpopulations [13]. This has led to the normal mixture model we know today, with density:

$$f(x, \theta) = \sum_{k=1}^{K} \pi_k \Phi(x, \theta_k) \tag{4}$$

with x a p-dimensional outcome, k indicates a specific cluster, π_k represents the mixing proportion of the k^{th} cluster where $\sum_{k=1}^{K} \pi_k = 1$, and Φ is the p-variate normal density with parameters:

$$\theta_k = (\mu_k, \Sigma_k) \tag{5}$$

corresponding to the vector of means, and covariance matrix for the k^{th} cluster.

It has been shown that various adaptations of this mixture distribution, can lead to various asymmetric distributions. The computational effort for applying the mixture methodology was extremely daunting at the time of its discovery [11]. However, with the advent of the EM algorithm and technological advances, the mixture modeling process can now cluster data and determine the likelihood of the data being described by the mixing distribution.

The EM algorithm treats the data as if information was missing; the missing information is a simple indicator variable that specifies if a particular observation belongs to a particular cluster. This algorithm is run and because it is centered on a likelihood approach, we can measure the fit of our model to the data. This not only includes the mixture proportions, but also the parameters, θ_k of each cluster. Each iteration has an E-step (expectation) and an M-step (maximization).

1. **The E-step at the s^{th} iteration**

   This step calculates the assignment probability of an observation i belonging to a cluster k

$$\pi_{ik}^{(s)} = \frac{\pi_k^{(s-1)} \Phi(x_i; \mu_k^{(s-1)}, \Sigma_k^{(s-1)})}{\sum_{k'=1}^{K} \pi_{k'}^{(s-1)} \Phi(x_i; \mu_{k'}^{(s-1)}, \Sigma_{k'}^{(s-1)})} \tag{6}$$

Based on the parameters found in a previous iteration, the expectation step calculates the probability of an observation belonging to cluster k. After this updated probability is calculated we proceed to the maximization step.

2. **The M-step at the s^{th} iteration**

   - First we calculate the proportion of each cluster:

$$\pi_k^{(s)} = \frac{1}{n} \sum_{i=1}^{n} \pi_{ik}^{(s)} \tag{7}$$

   - Then we update the mean of every cluster

$$\mu_k^{(s)} = \frac{\sum_{i=1}^{n} \pi_{ik}^{(s)} x_i}{\sum_{i=1}^{n} \pi_{ik}^{(s)}} \tag{8}$$

   - Finally we update the covariance matrix of each cluster (or the overall covariance matrix if we assume each cluster has the same covariance)

$$\Sigma_k^{(s)} = \frac{\sum_{i=1}^{n} \pi_{ik}^{(s)} (x_i - \mu_k^{(s)})(x_i - \mu_k^{(s)})'}{\sum_{i=1}^{n} \pi_{ik}^{(s)}} \tag{9}$$

     For a common covariance matrix we use (dropping the iteration superscript "s" for easier notation):

$$\Sigma = \frac{\sum_{k=1}^{K} (\sum_{i=1}^{n} \pi_{ik}) * \Sigma_k}{n} \tag{10}$$

     Substituting for Σ_k we get

$$\Sigma = [\sum_{k=1}^{K} (\sum_{i=1}^{n} \pi_{ik}) * \frac{\sum_{i=1}^{n} \pi_{ik} * (x_i - \mu_k)(x_i - \mu_k)'}{\sum_{i=1}^{n} \pi_{ik}}] \div n \tag{11}$$

Cancelling out the terms $\sum_{i=1}^{n} \pi_{ik}$ we get:

$$\Sigma = \frac{[\sum_{k=1}^{K} \sum_{i=1}^{n} \pi_{ik} * (x_i - \mu_k)(x_i - \mu_k)']}{n} \tag{12}$$

The algorithm continues until the difference

$$L(\Psi^{(s+1)}) - L(\Psi^{(s)}) \tag{13}$$

changes by a small amount, which we define as convergence. The entity L is the log-likelihood of the data given by

$$L(\Psi) = \sum_{i=1}^{n} log[\sum_{k=1}^{K} \pi_k \Phi(x_i, \theta_k)] \tag{14}$$

The procedure requires the calculation of the mixture proportions, the cluster means and full cluster covariances which need to be estimated from the data. However estimating all these parameters from a small data set can be problematic, and leads to overparameterization. To avoid this, we need to find a balance between the number of parameters and the generality of the model [5]. This can be done by the reparameterization of the covariance matrix according to its eigenvalue decomposition [3]. With this decomposition, we can indicate variable or constant shape, volume and orientation for the covariance matrices. This is implemented in the R-package mclust. The number of clusters and the parameterization of the model are chosen through minimization of the Bayesian Information Criteria (BIC score).

$$BIC(m) = -2(L) + v(m) * log(n) \tag{15}$$

Where L is the log-likelihood of the model m as calculated in equation 14, n is the number of observations and $v(m)$ is the number of parameters. So we can see that the number of parameters serves as a penalty.

Model Based Clustering also performs one task that the other two methods could not: fuzzy-clustering. K-means and hierarchical clustering assign observations to groups and give no measure as to the uncertainty of this assignment. Model based procedures however, because it is formulated around probability, can specify uncertainty. Classifying an observation is done by choosing the cluster that gives the highest assignment probability. However, even though an assignment is made based on this maximum, information is still available on the assignment probabilities for the other clusters, giving an indication of how uncertain the assignment could be. This is an added advantage of model-based clustering that will be touched upon later.

3.3.1 R-package mclust

This R-package seeks to add some flexibility to the assumptions made on the inherent structure of the data. Based on the definitions of the covariance updates seen above, two main generalizations of cluster covariances have emerged:

1. Each cluster has its own unique cluster specific covariance matrix as seen in equation 9.

2. The covariance matrix is constant across all clusters as seen in equation 10.

As proposed in literature, there may exist some structure that is not as restrictive as assuming a constant covariance matrix, but still not as free as having unique covariance matrices [3]. It seeks to find similarities or differences in other attributes of the covariances like, shape, volume and orientation. This is performed by first re-parameterizing the covariance matrix using eigenvalue decomposition:

$$\Sigma_k = D_k \Lambda_k D_k^T \tag{16}$$

where D_k is the orthogonal matrix of eigenvectors and Λ_k is the diagonal matrix of eigenvalues corresponding to the eigenvectors in D_k. The substitution $\Lambda_k = \lambda_k A_k$ where λ_k represents the first eigenvalue and A_k contains the factorized eigenvalues with the first value equal to one and the other values less than or equal to one and steadily decreasing. The orthogonal matrix D_k controls the orientation of the principal components of Σ_k; A_k determines the shape of the density contours (ellipsoid or spherical) and λ_k is a scalar quantity that determines the volume of the corresponding cluster. These characteristics of distribution are determined from the data and can be allowed to vary between clusters or constrained to be the same without going to the extremes of entire covariance equality or entire covariance uniqueness.

This package was utilized with our mosquito data. However, it seemed to suffer from the high-dimensionality of the data giving a warning of $n < p$. Due to the high-dimensionality, 490 parameters were estimated from 12 observations. For these observations, the unrealistic result of 11 clusters were found. The algorithm tested various covariance structures and selected the best model. The chosen model described the clusters as having equal shape

and volume, and diagonal orientations, allowing for correlation between variables (described as "EEI" in Appendix). Details can be found in the Appendix.

3.3.2 R-package HDclassif

For clustering in a high dimensional situation, it has been suggested that the data may be well approximated by working in subspaces with a dimension lower than that of the original space. These subspaces are found by first expressing the covariance matrix in its eigenvalue decomposition as performed with mclust. Afterwards, we split these eigenvalues into two levels, where the lower eigenvalues (noise) are represented by just one parameter, and each cluster therefore is modeled mainly according to the more important subspace [5]. The R package HDclassif, does this while applying the parameterization theory to capture the possible differences or similarities between clusters.

First let us rename the eigenvalue decomposition to follow the author of the package. For the covariance matrix Σ, we now rename its orthogonal matrices of its eigenvectors, Q, and its corresponding eigenvectors Δ. The matrix then becomes:

$$\Sigma_k = Q_k \Delta_k Q_k^{-1} \tag{17}$$

, (equivalent to equation 16 as seen in mclust).

Δ is the diagonal matrix containing the eigenvalues, and can be seen below (**warning**: Figure 5 is taken directly from literature and uses the index i to represent the cluster whereas in this paper we use k! [5])

$$\Delta_i = \begin{pmatrix} a_{i1} & & 0 & & \mathbf{0} \\ & \ddots & & & \\ 0 & & a_{id_i} & & \\ & & b_i & & 0 \\ \mathbf{0} & & & \ddots & \\ & & 0 & & b_i \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} d_i \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} (p - d_i) \end{matrix}$$

Figure 5: Diagonal matrix of eigenvalues for HDclassif

Reverting back to our index k to specify a cluster, we define the terms in the eigenvalue matrix of figure 5: d_k is the dimension of the cluster specific subspace for that cluster and is defined by the bigger eigenvalues $a_{k1}, a_{k2}...a_{kd_k}$; b_k is the noise parameter specific to the cluster and spans over the remaining dimensions $(p - d_k)$.

The method projects the data into two subspaces; one containing the main features of the group and the other containing the noisy or less important features of the group. The first thing that is determined, is the identification of the important dimensions for clustering, and this can be done by utilizing the scree analysis method to the eigenvalues, or by setting some cut-off criteria for the amount of explained variance captured by the higher eigenvalues. Where there seems to be a significant drop in the eigenvalues (determined by the proportion of variability captured by the larger eigenvalues or an obvious elbow in a scree plot), the smaller eigenvalues after the elbow (or cut-off) are averaged and denoted as b_k. The number of eigenvalues before the elbow make up the dimension d_k and the values themselves $(a_{k1}, a_{k2}...a_{kd_k})$ are the eigenvalues defining the important subspace. The projection onto the important subspace is defined by

$$P_k(x) = \tilde{Q}_k \tilde{Q}_k^T (x - \mu_k) + \mu_k \tag{18}$$

where $\tilde{Q}_k$ is made of the first d_k columns of Q_k and the $(p-d_k)$ zero columns. For the second subspace we get

$$P_k^{\perp}(x) = \bar{Q}_k \bar{Q}_k^T (x - \mu_k) + \mu_k \tag{19}$$

where $\bar{Q}_k = Q_k - \tilde{Q}_k$. The log-likelihood function is based on these two projections, and favors the location of an observation close to the mean of the important subspace. It saves on parameters by estimating the less important subspace through a single eigenvalue. Whereas some methods that utilize eigenvalue decomposition throw away the smaller values, we see that HDClassif still allows some expression of the noisy data.

This leads to the annotation specific to HDclassif known as $[a_{kj} b_k Q_k d_k]$. This parameterization is similar to that suggested by Banfield and Raftery [3], and similarly, tests models based on common orientations or unique orientations between clusters (Q vs Q_k); having the important dimensions be common or unique across clusters (d vs d_k); having the same noise across clusters or each cluster has its own noise parameter (b vs b_k). The important

eigenvalues (a) have two subscripts; j indicates the dimension and would allow each intrinsic dimension to have its own parameter while k specifies the cluster and allows for common parameters across clusters. We see here that the eigenvalue may be common to all important variables, but this common value may vary across clusters. Various adaptations of this notation give rise to various assumptions and differing number of parameters to be estimated.

Applied to our mosquito data for TSB neuron A, we found 2 clusters, and selected model $[a_{kj}b_kQ_kd]$. Each cluster has a unique orientation, noise parameter and important eigenvalues, and the only commonality between the clusters is they all have the same number of important dimensions: 3 common dimensions and 318 estimated parameters from the 12 mosquitoes. Details can be found in the Appendix.

4 Missing Values

It is important before trying to address the missing data problem, to examine the reason for the absent values. In general we want to discover if the data is missing randomly (Missing at Random, MAR or Missing Completely at Random, MCAR), or if the absent values are related to the response data (Not Missing at Random, NMAR). For the mosquito data, we can argue that the data is NMAR, as a reduced activity exhibited by the neurons is probably an indication of a weakening mosquito whose death may be soon, leading to missing values. The actual data predicts the missingness. An alternative argument is that the mosquito data is MAR and the missingness depends on the latent variable, time. There are a few ways we can deal with the missing values:

1. In the case of data showing MCAR patterns, we can just ignore the observations that contain missing data. This may be favorable in instances where the missingness is not severe, and we have a large enough sample space.

2. Replace the missing values with imputed values, and use them as if there were the observed ones.

3. Impute the missing values several times and pool the imputations while accounting for the fact that there is uncertainty in the imputation process (multiple imputation as performed by the R package MICE).

4. Maximum Likelihood techniques can also be implemented to get unbiased parameter estimates, for the parameters we are interested in [1]. For clustering these parameters include covariance matrices, cluster means and mixing proportions.

For a clustering analysis, simply throwing out the data can be an option. However, important discriminant information may be lost when we do so. In situations where the data does not agree with the assumption MCAR, this method is not suggested, and it becomes even more undesirable when the initial sample space is already small. Deletion has also been shown to give biased estimates when the MCAR assumption is violated [2]. Basic imputation may be implemented. However, treating imputed values as observed values has its disadvantages in that we are now binded to a given imputation,

as the uncertainty of the imputation technique is not taken into consideration. Imputation uses patterns in the data to replace the missing values, and may create artificial similarities as a result. This is something quite undesirable for cluster analysis because now some data that may have differed significantly in the missing dimension, may now be seen as similar.

MICE, developed by Stef van Buuren, is a useful tool and has the ability to impute missing values multiple times. It then pools the various imputations according to the research question and returns an output of parameter estimates alongside the uncertainty of the imputations. The package imputes each missing variable by conducting a regression on the other variables. In high dimensional settings we can run into over-fitting where 70 variables (for example) are predictors for a single target variable. For this reason regressors are picked via certain criteria. These variables are selected via correlation with the variable whose missing value we want to impute, as well as the proportion of usable cases which measures how many cases on the target variable actually have observed values on the predictor. Afterwards, multiple data sets are created, each with different imputed values for the missing data. The information can then be combined with the use of the scientific study question and is formatted to mainly handle regression based questions. The variability of information from all the imputed data sets are included in the regression coefficient measures. MICE includes the uncertainty due to imputation by combining the standard errors of the coefficients, with the estimate of the additional variability produced by the imputation process. The combined uncertainty for the pooled parameter estimate is given by:

$$S.E._{\bar{a}} = \sqrt{\frac{1}{M}\sum_{k=1}^{M} s_k^2 + (1+\frac{1}{M})(\frac{1}{M-1})\sum_{k=1}^{M}(a_k-\bar{a})^2} \tag{20}$$

where $S.E._{\bar{a}}$ is the standard error of the mean parameter estimate, M indicates the number of imputed data sets, k is the index for a particular imputed data set, s_k is the standard error of the parameter estimate for the k^{th} data set, a_k is the parameter estimate for the k^{th} data set and $\bar{a}$ is the mean of the parameter estimates across all imputed data sets [1].

For cluster analysis, the number of clusters is not a parameter to be estimated with associated standard errors. Therefore this pooling function cannot be applied for this type of analysis. It is suggested [4] that for each imputed data set, a cluster analysis is performed, and we choose the modal number of clusters. For example, for the mosquito data set, 40 imputations

were performed and it was found that for the imputed datasets, after using HDclassif, the modal number of clusters is 2, as seen in figure 6 below.

Multiple imputation leads to different results with every imputation whereas maximum likelihood using only the available data, will give the same result each time [1]. Another significant argument against multiple imputation, is the fact that the model used to impute values, and the scientific model of interest may be conflicting. This happens because the variable selection process in MICE cannot include interactions and quadratic terms that may be present in the analysis model. Maximum likelihood methods are preferred in these situations because everything (analysis and estimates when data is missing) happens under one model [1]. This parameter estimation of available data using maximum likelihood differs from multiple imputation in that no data is created. Instead, all available information is used as is to estimate the parameters most likely to be the source of the data [2]. Either way, when data violates MCAR assumptions, multiple imputation and maximum likelihood estimation will both be the more powerful tools for acquiring unbiased parameter estimates.

After using multiple imputation and HDClassif to find 2 clusters, the question of cluster assignment was eminent, and proved to be quite challenging. When printing out the cluster assignments we attain 40 different clustering assignments that are dependent on the imputed data. One may want to investigate these imputed sets to see if the same classification has occurred anywhere across the different imputations. The problem lies in the labeling whereas one group may be labeled 2 in one imputation and labeled 1 in another imputation but within these groups have similar members. After adjusting for the labeling problem, and similar to looking at the mode to find the number of clusters, the researcher must then find the modal assignment for observations across the imputations (so if an observation is put in a particular cluster most of the time we would conclude that this observation belongs to that cluster) [4]. This is feasible, however quite difficult if 100 imputations were performed.

With hierarchical clustering we do have one more option other than imputation and data deletion when trying to deal with missing values. We can still obtain a full distance matrix even though our data has missing values. This will lead to some information bias where one distance between two observations is based on all 73 variables while another distance between two other observations may be based on only 10 variables. To my knowledge, there is no mechanism to deal with the fact that some distance measures are

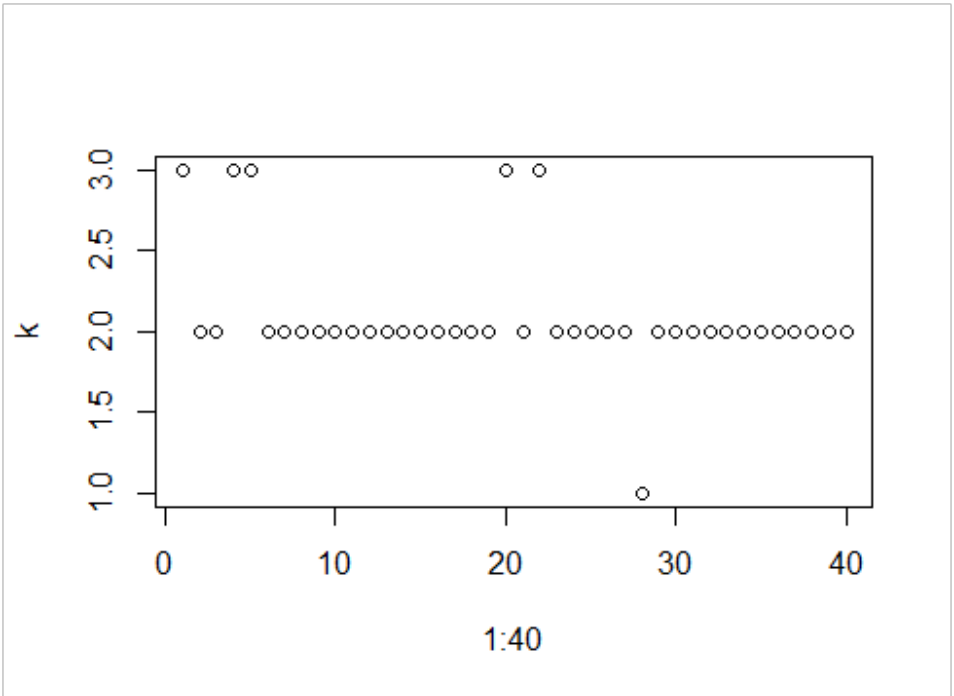


Figure 6: Number of clusters for 40 Imputed datasets

more informative than others, but besides this fact, this property of hierarchical clustering whereby clusters are determined via distance matrices helps to alleviate the problem of missing values. There still remains the concern of determining the number of clusters objectively as we cannot utilize NbClust with high dimensions or missing values , and we cannot perform a cluster vs sum of squared errors plot with incomplete data.

While working on this thesis, an idea came to mind in combining the use of multiple imputation to determine the number of clusters, with the property of hierarchical clustering to be able to handle missing values for the cluster assignment. The idea is, after we have figured out the modal number of clusters, we can then perform hierarchical clustering on the data and set the partition level to the number of clusters we found from the first step to attain the object classifications. If this method proves to be sound, we can classify our observations while avoiding the relabeling problem.

Another idea was model-based clustering which unlike hierarchical clustering, is unable to naturally handle the common problem of missing values without discarding or imputing information. The initial step is the use of marginal probabilities when calculating the assignment probabilities in the E-step of the algorithm, followed by the parameter estimation in the M-step.

5 Adapting the EM process in Mixture Modeling for Missing values

5.1 Marginal Probabilities for E-step

It is well known that marginal densities of a multivariate joint Gaussian density are also Gaussian [6]. This property of the multivariate Gaussian is useful to our missing value problem and will be illustrated below.

For example, let us assume we have data, where each observation has three variables: $X_i = [x_{1i}, x_{2i}, x_{3i}]$. For model based clustering to work, we assume that all the data come from the density in equation 4. For this dimensional setting, the mean vector

$$\mu = [\mu_1, \mu_2, \mu_3]$$

and covariance matrix

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

Now recall the E-step of the algorithm where we have to calculate the probability of assignment as seen in equation 6. For this calculation, in both the numerator and denominator the density: $\Phi(x_i; \mu_k, \Sigma_k)$ has to be calculated. Below this calculation is represented for our three variable example, without indicating the specific cluster k:

$$p(x; \mu, \Sigma) = \frac{1}{2\pi \begin{vmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{vmatrix}^{1/2}} exp(-\frac{1}{2} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \\ x_3 - \mu_3 \end{bmatrix}^T \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}^{-1} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \\ x_3 - \mu_3 \end{bmatrix})$$

Suppose we encounter an observation that has a missing value concerning the second variable. Because this vector only contains $[x_1, x_3]$, we will run into problems because now we have no value to subtract with the second mean (μ_2). However, because of the property of the multivariate Gaussian in that its marginals are also Gaussian, we can model the bivariate distribution as Gaussian having the parameters:

$$\mu = [\mu_1, \mu_3]$$

and covariance matrix

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{13} \\ \sigma_{31} & \sigma_{33} \end{bmatrix}$$

Now, we can simply find the marginal density of belonging to a cluster based on the information available.

$$p(x; \mu, \Sigma) = \frac{1}{2\pi \begin{vmatrix} \sigma_{11} & \sigma_{13} \\ \sigma_{31} & \sigma_{33} \end{vmatrix}^{1/2}} exp(-\frac{1}{2} \begin{bmatrix} x_1 - \mu_1 \\ x_3 - \mu_3 \end{bmatrix}^T \begin{bmatrix} \sigma_{11} & \sigma_{13} \\ \sigma_{31} & \sigma_{33} \end{bmatrix}^{-1} \begin{bmatrix} x_1 - \mu_1 \\ x_3 - \mu_3 \end{bmatrix})$$

Even though it can be argued that some probabilities are based on partial information while others are fully informative, it is good to note that the assignment probabilities for an observation is based on its density of belonging to a given cluster divided by the sum of the densities of belonging to all clusters. For a given observation where information is limited due to missingness, the probability of belonging to cluster $1, 2, 3....k$, are all based on partial information. The assignment probability is a measure that is relative to all these marginal densities, and still gives the desired result of which cluster is the most likely to be the source of our observation. We do not compare marginal densities to full densities, but compare marginals to marginals based on the same limited information.

With this marginal multivariate Gaussian, we can overcome the E-step of the model based procedure, without removing or imputing data. Next we have to discuss our maximization steps, where the proportion of clusters, the cluster means, and the cluster covariances are to be updated, and how to do so in the presence of missing data.

5.2 M-step adapted for missing values

At this moment, we again need to consider the mechanism of missingness in our data. The M-step recalculates and finds likelihood maximizing estimates for cluster proportions, means and covariances. Unbiased parameter estimates for the mean and covariance are easily acquired when the data is MCAR, and is shown below. However when data violates this MCAR assumption, we need more robust procedures to generate unbiased estimates, which is discussed at the end of this section.

For the proportion of each cluster there is no special adaptation required, because there are no missing values in our assignment probability (π_{ik}) information. We can just find the mean of all the assignment probabilities for a given cluster as given in equation 7.

For the calculation of the mean for each cluster, given in equation 8 we multiply each observation (x_i) by it's corresponding assignment probability (also referred to as 'weights') to that cluster. If there are any missing values, under the MCAR assumption, we can just ignore it. When dividing by the sum of weights for the cluster, each variable is divided by the sum of weights used; those weights that were not used because of a missing value, are not included in denominator summation for that corresponding variable. This is easily implemented in R by applying a weighted mean to the columns of our data, where our weights are the assignment probabilities, and instructing the command that na.rm="TRUE", re-iterating again that this will provide unbiased estimate of these means when the data is MCAR.

The update of the covariance matrix seen in equation 9, is not as straight forward. The first thing is to investigate how traditional software handles this problem which lead to the investigation of the "pairwise" argument used in R when wanting to calculate covariances from data with missing values. The covariance between two variables is then calculated only using the complete pairs of observations between those two variables and averaged over the number of pairs present. It is known that covariance estimation using this pairwise method is unbiased under MCAR assumptions [2, 9]. This covariance update is done using matrix multiplication where the whole covariance matrix is calculated for all variables in one step, as opposed to calculating covariances between pairs of variables individually. How do we implement this strategy with matrix multiplication? Let us illustrate with an example.

A covariance matrix needs to be calculated frm the data below

X_1	X_2
1	4
-1	NA
6	2

Covariance calculation in R is done by:

$$\frac{\sum^{n}(x-\mu)(x-\mu)'}{n-1}$$

In our example $n = 3$ and our vector of means for X_1 and X_2 is $(2, 3)$. For the first observation $(x - \mu)(x - \mu)' =$

$$\begin{pmatrix} -1 \\ 1 \end{pmatrix} \times \begin{pmatrix} -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

For the third observation $(x - \mu)(x - \mu)' =$

$$\begin{pmatrix} 4 \\ -1 \end{pmatrix} \times \begin{pmatrix} 4 & -1 \end{pmatrix} = \begin{pmatrix} 16 & -4 \\ -4 & 1 \end{pmatrix}$$

For the second observation, containing a missing value $(x - \mu)(x - \mu)' =$

$$\begin{pmatrix} -3 \\ NA \end{pmatrix} \times \begin{pmatrix} -3 & NA \end{pmatrix} = \begin{pmatrix} 9 & NA \\ NA & NA \end{pmatrix}$$

In order for the summation to take place, we replace the NA values in any individual squared deviance matrix by a zero (0). After doing this we can sum our three matrices, fulfilling the numerator of our covariance calculation.

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} + \begin{pmatrix} 16 & -4 \\ -4 & 1 \end{pmatrix} + \begin{pmatrix} 9 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 26 & -5 \\ -5 & 2 \end{pmatrix}$$

Our last step is dividing by $n - 1$ or in essence averaging. However for each total, like the calculation for mean, we want to divide by the number of observations used to acquire the sum. So for the total of 26, we used three values to get this sum so $3 - 1 = 2$ will be it's denominator. For the other totals, we only added two observations so they will be divided by $2 - 1 = 1$. In the end our result will be:

$$\begin{pmatrix} 26 & -5 \\ -5 & 2 \end{pmatrix} \div \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 13 & -5 \\ -5 & 2 \end{pmatrix}$$

This is the same result attained if you input the data in R and ask for a covariance to be calculated using pairwise observations.

This simple example is the blueprint for our calculation of the cluster covariances in this missing data environment.

- First we calculate the deviation, $x_i - \mu_k$, for every observation.
- We then multiply each deviation by the square root of its assignment probability, for that observation belonging to the k^{th} cluster

$$(\sqrt{\pi_{ik}})(x_i - \mu_k)$$

- Before we square this weighted deviation and sum over all observations, we first replace all NA values by the number 0

- Square the deviances and sum over all observations. This is easily implemented in R with the command crossprod

- To divide by the sum of weights, we do something similar to the simple example where we only sum over those weights actually used in calculating the weighted deviances. This is done easily with matrix algebra and we can take a look at our example. By creating a matrix of 1's, if data is present and 0's if data is missing, we can attain our matrix for number of observed values per covariance calculation. From our example, the matrix showing absence/presence of a value is:

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

  To attain the denominator we did in our example we can simply take the crossproduct of this matrix:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 2 & 2 \end{pmatrix}$$

  For our simple example we used $n - 1$ hence the slight difference. To attain a corresponding measure for sum of weights as required by the denominator of our covariance matrix update, we simply need to insert a diagonal matrix containing the assignment probabilities between the indicator matrices above, and we would get the desired sum of weights for all combinations of variables, in matrix form with same dimensions as the covariance matrix.

- Often this method will lead to a covariance matrix that is not positive definite. This is solved by adding a small constant to the diagonal of the covariance matrix to make this covariance matrix invertible. The R command nearPD easily calculates the nearest positive definite matrix for any given matrix, and is therefore programmed into the algorithm.

Theoretically, the steps above seem to avoid the problem of missing values in model based clustering, by making small adjustments to the E-step, and to the covariance update of the M-step. However our M-step requires the strict MCAR assumption, which is usually not the case in the real world. Another way of attaining the estimates for the mean and covariance matrices is through maximum likelihood, which is more robust when data is MAR or even NMAR.

5.2.1 Maximum Likelihood to attain Means and Covariances

When our data is missing in a systematic manner, the method above will be flawed and lead to biased estimates of the covariance matrix. However, we can still determine a way to calculate the most likely parameters to fit the data we have available. This is done via Maximum Likelihood and it is even been preferred to multiple imputation in some literature [1]. Suppose we have full information on the first m observations, and missing information on y_1 and y_2 on the remaining $n - m$ observations, (which is known as a monotone missing pattern), the likelihood looks like:

$$L = \Pi_{i=1}^{m} f(y_{i1}, y_{i2}..., y_{ip}, \theta)\Pi_{m+1}^{n} f(y_{i3}, y_{i4}..., y_{ip}, \theta) \tag{21}$$

This likelihood can then be maximized to attain parameter estimates of θ [1], which in our case are our cluster means and covariances. Similarly to the clustering algorithm which depends on the EM algorithm, this step also requires the EM algorithm which now has to deal with two sources of missing data. If the only source of missing data was the indicator variable for whether an object belonged to a cluster a not, we simply would have the steps shown above in the section Model-based clustering. With two sources of missing information, the EM algorithm can still theoretically be adapted to handle the situation. Details can be found in the paper by Ghahramani [10]. The execution of this ideal is quite troublesome when trying to apply the formulas found in the literature. Attempts were made to use established mixed model software [1] to obtain maximum likelihood estimates of covariance matrices in the case of missing values. Because the mixed model software in R (packages nlme, lme4) is less developed than in SAS, we used the latter to set and extract the covariance matrix. However, the algorithm performed poorly for reasons unknown.. However, we think that after reading literature on how mixed models can be used to attain maximum likelihood estimates when there is missing data, we can apply this knowledge to attain our desired

parameters [1]. The mixed model software in R is somewhat messy and there are multiple packages that perform poorly, especially when applying the weights, setting a covariance structure and extracting the covariance matrix. Attempts were made to incorporate SAS as seen in the literature, but still the algorithm performed poorly when used in conjunction with SAS.

We were hoping that within this thesis, we would have incorporated this theory into our algorithm but our attempts were unfortunately in vain. This solution of using maximum likelihood estimates to produce unbiased estimates of our parameters is something that we eagerly want to test, and hope to develop in the future.

6 Simulation Study

We maintain our focus on model based clustering, because of its automated way of identifying clusters, its ability to perform fuzzy-clustering, as well as its adaptability to high dimensional data. Our aim is to add a further adaption when there is missing data.

6.1 Algorithm

Firstly, a function was written to reduce the mean of vectors and covariance matrix based on only available values, and thus calculate the marginal multivariate probabilities if data is missing. Next, the updates for cluster weights, means and covariances were programed as described above, also allowing for missing values. The likelihoods were calculated for varying number of clusters and the optimal value chosen via a BIC score. Lacking the sophistication of established algorithms, the algorithm developed along with this thesis has not been expanded as yet to test various covariance structures. Because of this, a common covariance matrix between clusters is assumed within the algorithm.

Before testing this algorithm for its performance with incomplete data, its performance in comparison to the R package mclust was tested. At first the algorithm was severely outperformed by mclust however upon investigation it was found that this was due to the starting values. The R package uses agglomerative hierarchical clustering partitions as its source of starting values. When this procedure was included in our algorithm, the results were the same as mclust. However, can we still attain these "healthy" starting values when there is missing data? The answer is yes, because as stated before, hierarchical clustering can work with incomplete data once the distance matrix is complete. A complete distance matrix just requires that there must at least one complete pair of variables, for all pairs of observations. In higher dimensions this is quite probable because among a large number of variables it is highly likely an observed pair occurs. We are not worried about the information bias of the distance matrix as we are only using it for starting values. In the event that we do get a missing value in our distance matrix, it is usually an extremely low percentage (example mosquito data: n=20, p=73, missing=36.4%, distance matrix had 1 NA), and can be easily imputed. This distance matrix imputation feature was therefore added as one of the steps in the algorithm.

The complete algorithm can be seen in the Appendix.

6.2 Data Generation

Testing our theories requires the creation of multivariate data, where the mixing distributions are known, giving us the power to check which method performs best. The package MixSim was developed just for this purpose. Presented to the public in 2012 by authors Melnykov, Chen and Maitra, this package allows the user to generate data from a Gaussian mixture of distributions, while allowing the user to specify a certain amount of overlap. This pairwise overlap is the sum of two misclassification probabilities and can be utilized to manipulate the clustering complexity of a dataset [12]. The overlap between the k^{th} and m^{th} component can be defined as

$$\omega_{km} = \omega_{k|m} + \omega_{m|k} \tag{22}$$

where $\omega_{k|m}$ is the probability of misclassifying an observation X to the k^{th} cluster when it truly belongs to the m^{th} component.

$$\omega_{k|m} = Pr[\pi_m \Phi(X, \mu_m, \Sigma_m) < \pi_k \Phi(X, \mu_k, \Sigma_k) | X \sim N_p(\mu_m, \Sigma_m)] \tag{23}$$

The package allows the user to specify a maximum and/or average overlap as well as many other useful arguments that can greatly vary the nature of the data, and thus test the clustering methods in different situations. These arguments include:

- The number of components
- The number of dimensions
- Noisy observations
- Spherical or non-spherical components
- Homogeneous or heterogeneous components (identical or different covariance matrices)
- Equal proportions or not

Due to the limitations of our algorithm, we restricted the data to have homogeneous components in this simulation. Secondly we set the components to be non-spherical (correlated variables), and our number of components to be fixed at 3. Allowed to vary was the dimensionality, the pairwise overlap and the amount of missing data. Missing data were generated by random sampling from a uniform distribution, and the amount of missing data took the values $10\%, 15\%, 20\%, 25\%$ and 30%. The number of dimensions did not extend to the scope of the mosquito data, due to the instability of pairwise covariance calculations in higher dimensions. Therefore dimensions took the values: $p = 3, 5, 8, 12$. Motivated by the mosquito analysis, where the removal of observations was not favorable because of the small sample space, the observation count was set at a relatively low value of 60 observations. In practice, if there is a very large sample space then removing incomplete objects will quite likely have little effect on the analysis. For these situations, there would be no need for a marginal or imputation approach, however for the sake of this study and future small sample studies, we maintain $N = 60$.

Lastly, a choice of the pairwise overlap had to be made and based on the manual for the MixSim package, values of $\bar{\omega} = 0.001$ will lead to well separated clusters, and $\bar{\omega} = 0.05$ will lead to clusters with substantial overlap. Based on this we chose to use values of $\bar{\omega}$ of 0.001, 0.015, and 0.05.

6.3 Tests

6.3.1 Cluster Discovery

Three methods were tested for their ability to correctly identify the number of components of the data in the presence of MCAR missing values.

1. Complete Case analysis with mclust, where any observation containing at least one missing value is removed.

2. Multiple Imputation, where 45 imputed data sets were created and each data set analyzed with mclust and the modal number of clusters chosen.

3. The algorithm developed in this thesis that utilizes marginal probabilities and generates parameters assuming that the data is missing randomly.

In the end, we wish to see how the accuracy of detecting 3 clusters varies with dimension, extent of missing values and cluster separation, for all three methods.

6.3.2 Classification Accuracy

Next, the ability to correctly classify observations was tested using three methods, when the number of clusters are known a priori. The complete case analysis classification was not tested, as we often had less than half of the observations left after row wise deletion. This comparison would have been unfair to the other methods which kept all 60 observations, giving more opportunities for misclassification. The following methods were compared.

1. Hierarchical clustering which as stated, has the natural ability to cluster when there are missing values by attaining a full distance matrix.

2. Multiple Imputation that creates 45 imputed datasets as before, and calculates the mean accuracy of the 45 data sets. This is different to how it is done in practice, where for each observation we would look for its modal classification after solving the relabeling issue [4].

3. The cluster assignment generated from the algorithm developed in this paper, taking the maximum assignment probability (Equation 6) and assigning the object to this cluster.

We seek to identify how missingness, dimensionality and cluster separation, affect classification accuracy.

6.3.3 Real Life Data

All three methods were then tested with the publicly available seeds dataset. This dataset consists of 7 measures on the kernel of three different types of wheat seed. Details are given in the results section below.

Cluster discovery and classification accuracy were both outcome measures for these tests as the extent of missing information was allowed to vary.

6.4 Results

6.4.1 Cluster Discovery

A preliminary simulation was done at first and it was found that the overlap measure had an immense impact. For this preliminary test, 300 simulations were done, where 100 of them were simulated data sets with significant overlap ($\bar{\omega} = 0.05$).

The ability to detect three clusters in this preliminary test can be seen below.

Table 1: Preliminary simulation showing the effect of cluster separation on cluster discovery

Method	$\bar{\omega} = 0.001, 0.015$	$\bar{\omega} = 0.05$
C.C. with Mclust	32%	15%
Multiple Imputation	75%	43%
Marginal Method	72%	26%

Table 1 gives us some indication that the multiple imputation method seems better suited for cases where conditions are not favorable, like when clusters are not well separated. However, cluster overlap cannot be seen in real data, while dimensions and missingness rate are observable. For this reason we decided to vary the overlap very little (0.001 and 0.015) and pool these results.

After determining this trend, we ran 2000 simulations. For each combination of missingness and dimension, 100 simulations were conducted. Of these 100 simulations, 50 were done with well separated clusters and the other 50 were not as well separated. Table 2 below illustrates this layout, for only one level of overlap (example $\bar{\omega} = 0.001$).

Table 2: Number of simulations for each combination of dimension and missingness

$\bar{\omega} = 0.001$				
	Dimensions			
% Miss	3	5	8	12
10	50	50	50	50
15	50	50	50	50
20	50	50	50	50
25	50	50	50	50
30	50	50	50	50

For all simulations, the cluster discovery accuracy results can be seen in the tables below.

Table 3: Cluster discovery for $\bar{\omega} = 0.001$

$\bar{\omega} = 0.001$												
	3			5			8			12		
	C.C.	M.I.	M.M.	C.C.	M.I.	M.M.	C.C.	M.I.	M.M.	C.C.	M.I.	M.M.
10%	88	90	94	82	92	98	58	82	98	32	90	96
15%	90	90	98	62	90	98	38	88	98	14	88	94
20%	80	92	100	54	94	100	24	86	98	16	92	90
25%	78	76	96	52	90	94	20	92	92	26	92	82
30%	72	80	92	30	94	90	22	94	88	20	80	82

Table 4: Cluster discovery for $\bar{\omega} = 0.015$

$\bar{\omega} = 0.015$												
	3			5			8			12		
	C.C.	M.I.	M.M.	C.C.	M.I.	M.M.	C.C.	M.I.	M.M.	C.C.	M.I.	M.M.
10%	80	82	82	50	80	86	38	64	56	22	70	30
15%	72	76	82	36	68	62	30	70	40	16	62	42
20%	62	62	70	38	66	62	16	62	42	14	52	14
25%	62	66	78	34	68	76	26	72	50	24	60	44
30%	50	60	74	22	52	50	14	66	66	12	58	44

In tables 3 and 4, the row headings represent the amount of missing information $(10\%, 15\%, 20\%, 25\%, 30\%)$, while the bigger column headings

represent the different dimensions (3,5,8 and 12). The acronym C.C. represents the complete case cluster detection accuracies, M.I represents the multiple imputation cluster detection accuracies and M.M. represents the marginal method cluster detection accuracies. All numbers within the cells are percentages. Figure 7 illustrates these results further:

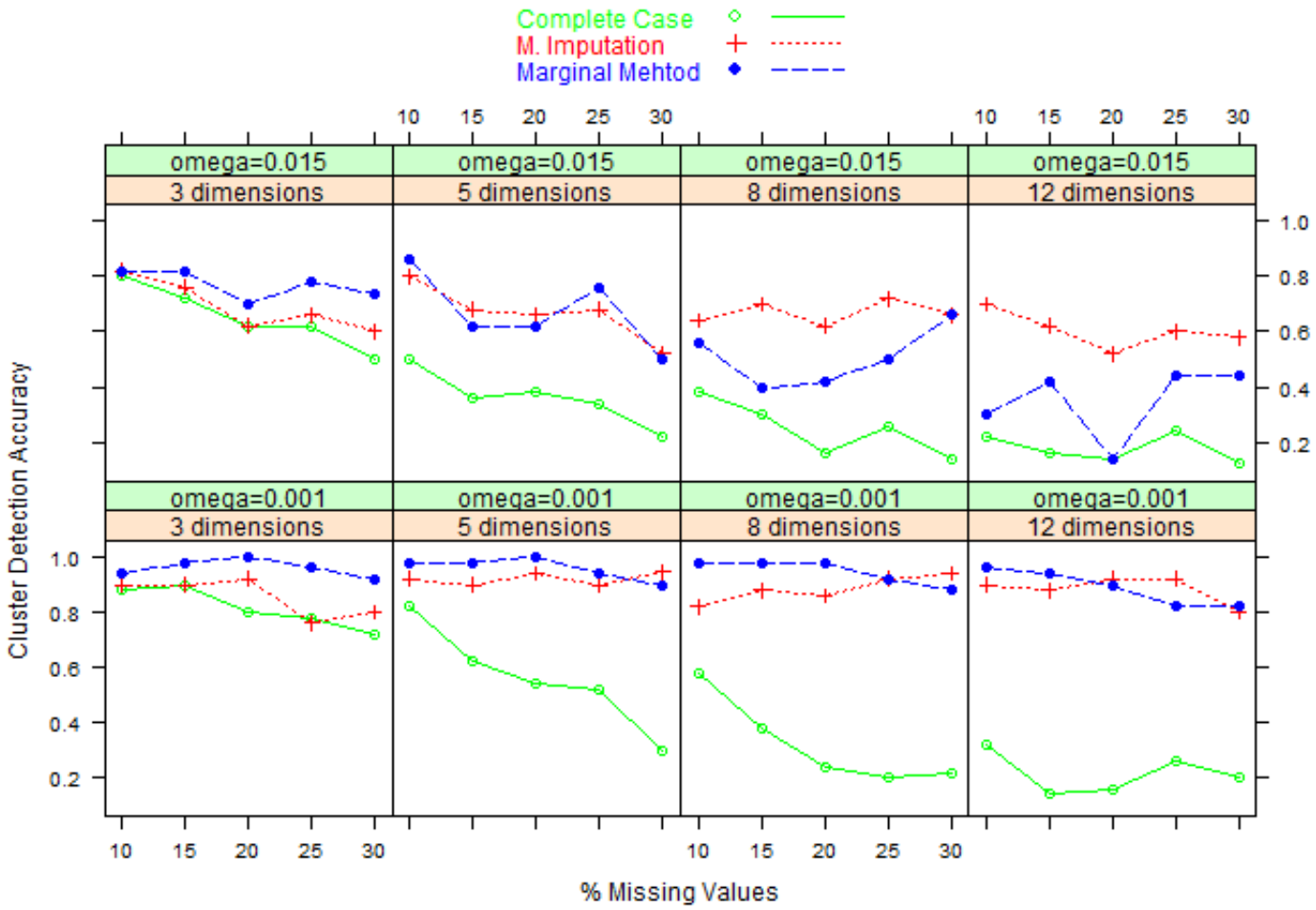


Figure 7: Cluster detection accuracy

As expected, all three methods show a general decline in detection ability as missingness increases. As detected in the preliminary simulations, there seems to be a significant effect from the overlap, and now we can also see a

significant impact caused by dimensionality. Summarizing over all scenarios we get the results in table 5 below:

Table 5: Cluster Discovery for all simulations

Method	Cluster Detection Accuracy
Complete Case with Mclust	41.9%
Multiple Imputation w/ Mclust and modal cluster	77.2%
Marginal Method	75.7%

Overall it seems that the complete case method is quite inferior, however it must be stressed that this is expected in smaller sample spaces. From this summarized table 5 we may be led to believe that there is only a small difference between the imputation and the marginal approach.

Before we delve into the the effects on this cluster detection accuracy based on dimension and missingness, we would like to take another look at how much our methods differ with this seemingly slight difference in cluster separation: $\bar{\omega} = 0.001$, and $\bar{\omega} = 0.0015$)

Table 6: The effect of Cluster Separation on Cluster Discovery

Method	$\bar{\omega} = 0.001$	$\bar{\omega} = 0.015$
C.C. with Mclust	47.9%	35.9%
Multiple Imputation	88.6%	65.8%
Marginal Method	93.9%	57.5%

Table 6 shows the harsh impact of the overlap measure, with the biggest victim being the marginal method falling by 36.4%. Imputation and the complete case analyses fall by 22.8% and 12% respectively. There is evidence here suggesting that for the marginal density method, no information is better than making inferences from partial information as the clusters become more cohesive.

Next we wish to discover the impact of dimensionality and missingness rate on the different methods.

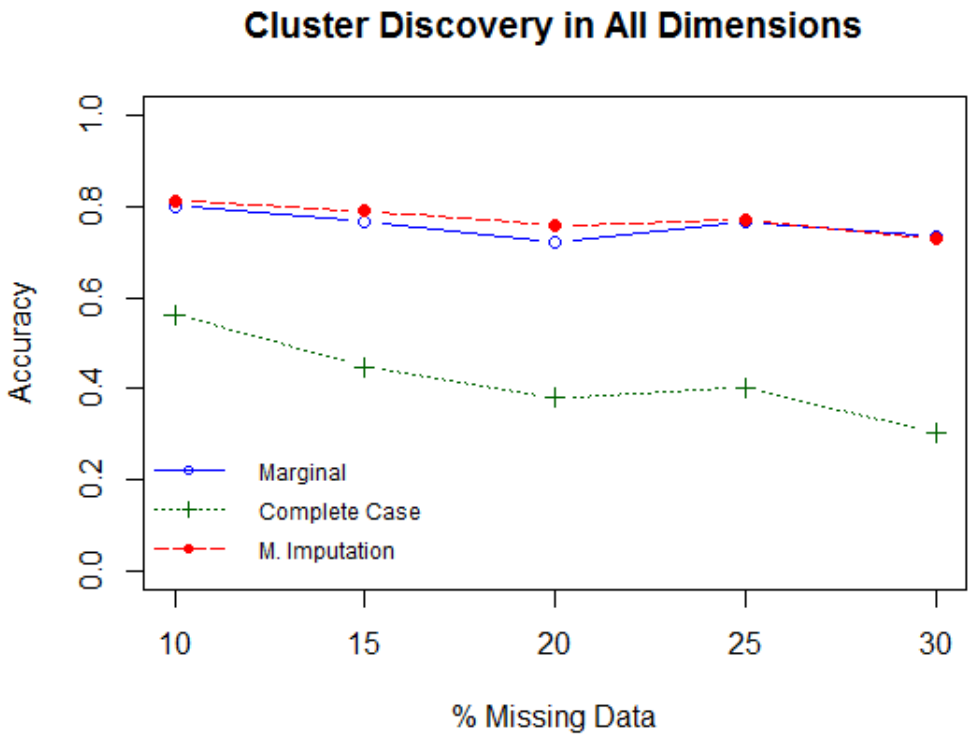


Figure 8: Accuracy of cluster detection for different values of missingness averaged over all levels of overlap

Figure 8 confirms that the complete case method is not suitable for this situation, even though the missing data mechanism is random. Also, we notice the slight superiority of multiple imputation over the marginal method, which does not widen or decrease significantly as the rate of missingness increases. They both appear to be roughly equal in correctly predicting the number of clusters. All three methods in fact do not show any drastic decline as missingness increases

The results showing the effect of the dimensionality can be seen in the next figure.

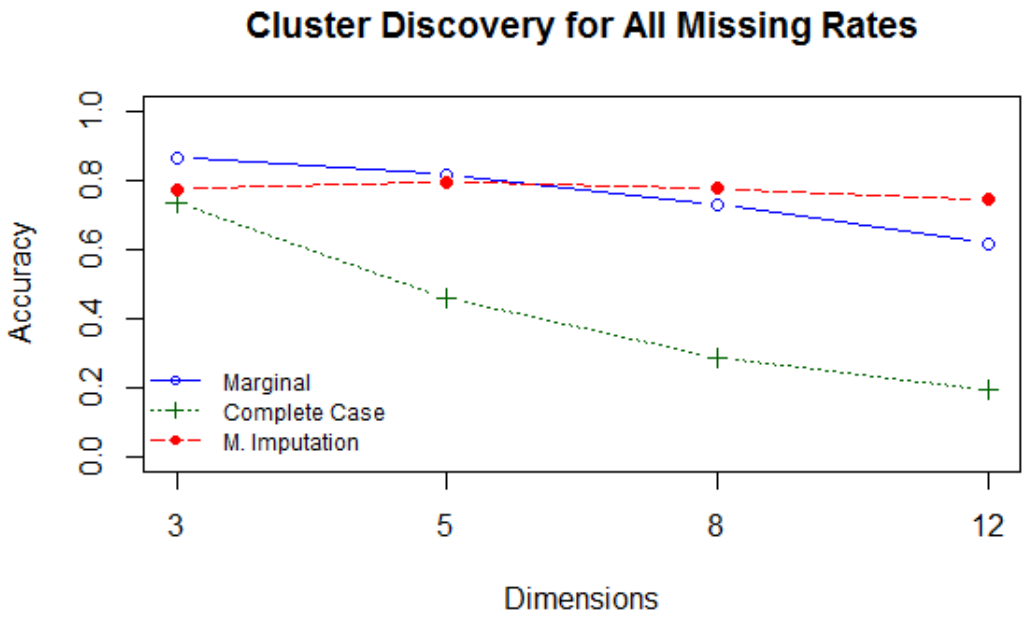


Figure 9: Accuracy of cluster detection for different dimensions averaged over all levels of overlap

Along with cluster separation, dimensionality seems to be a major factor affecting the ability of these methods to detect clusters in the presence of missing data. Taking a closer look, the marginal method seems to outperform multiple imputation when the dimensions are quite low. This may be due to the fact that because imputation seeks to borrow information in order to fill in values, a smaller dimensional space would have less information from which to borrow from. Thereby a missing value is predicted from 2 variables, would not be as accurate as a missing value predicted by 7 variables. Compounding this issue is the relatively small sample space.

From the above plot it seems that the marginal method outperforms the imputation method until about 6 or 7 dimensions. All three methods show declines as the dimensionality is increased. Imputation, however, seems to be the most stable while complete case analysis performs very poorly very quickly, as dimensions are increased. This is due to the fact that more and more observations are discarded as the number of dimensions increase.

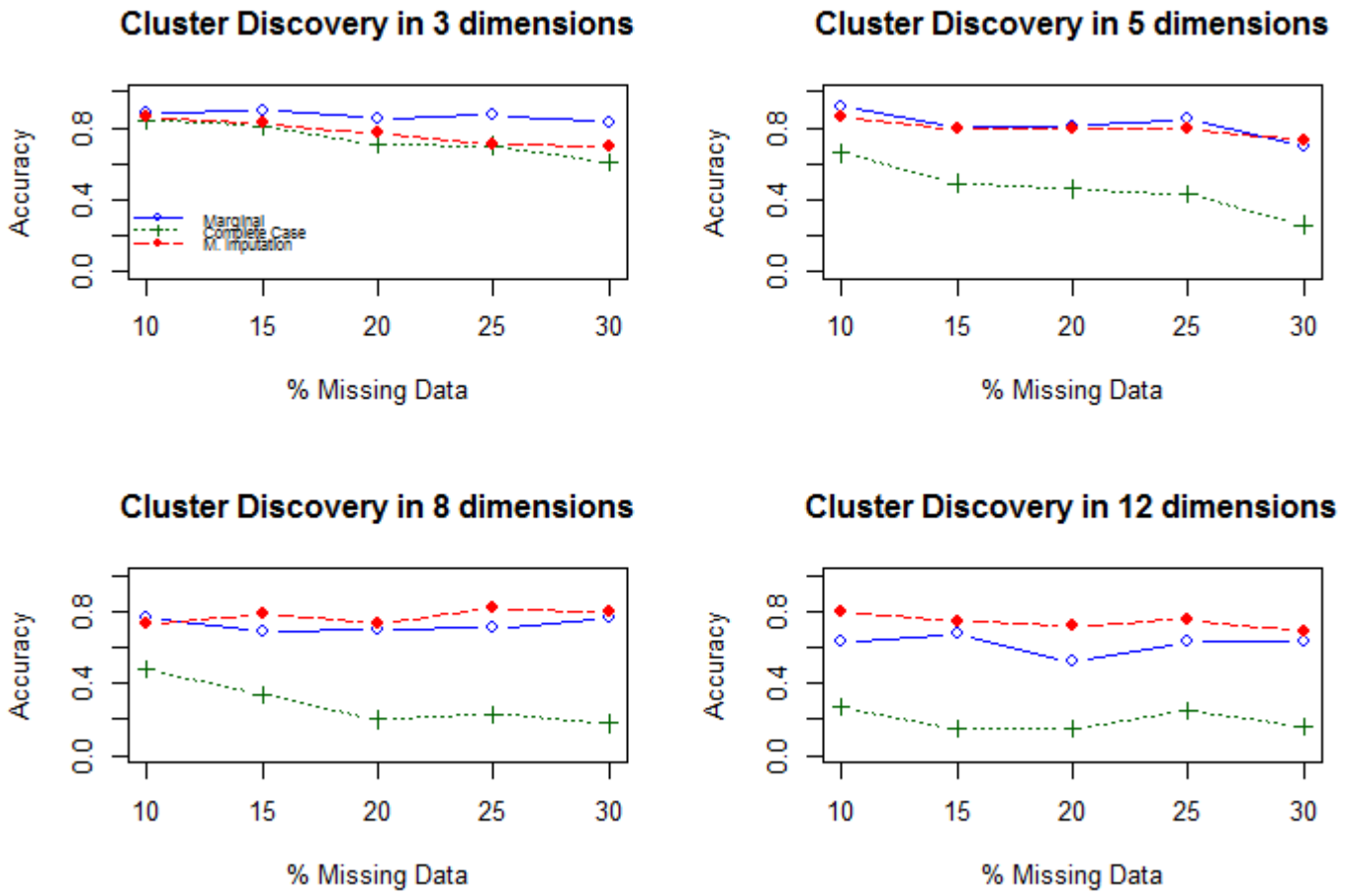


Figure 10: In depth look at the effect of dimensionality and missingness

Figure 10 confirms that in the lower dimensions, the marginal method outperforms the multiple imputation method even as missingness increases. In fact, in three dimensions, multiple imputation suffers from increasing missing information while the marginal method remains stable. Multiple Imputation starts to show its superiority around 8 dimensions and is completely favorable in dimensions larger than this, giving evidence that multiple imputation is the more robust method.

Keeping the missing rate constant at 10%, we can truly see the effect of dimensionality.

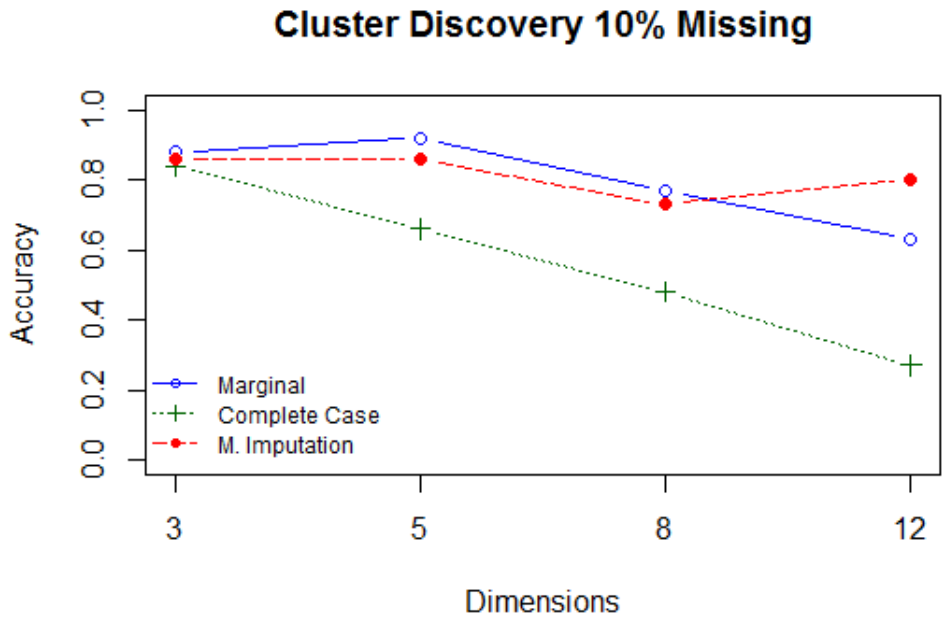


Figure 11: Cluster detection accuracy for different Dimensions with 10% Missing values

It can be seen that imputation is on average, the better choice for cluster discovery. This may be caused by the pairwise covariance matrix becoming more unstable in higher dimensions as well as the lack of technical versatility of the marginal algorithm, which does not contain the advanced programming of the well developed mclust.

6.4.2 Classification Accuracy

Classification accuracy is the ability to correctly assign observations to their cluster. The effect of cluster separation, dimensionality and missingness were tested against the hierarchical, imputation and marginal clustering methods. The number of simulations at different combinations of factors were exactly the same as seen in Table 3, and tested at the same two levels of separation. The accuracy results can be seen in the tables below:

In tables 7 and 8, the row headings represent the amount of missing

Table 7: Classification accuracy for $\bar{\omega} = 0.001$

$\bar{\omega} = 0.001$												
	3			5			8			12		
	H.C.	M.I.	M.M.	H.C.	M.I.	M.M.	H.C.	M.I.	M.M.	H.C.	M.I.	M.M.
10%	96.4	96.1	98	97.7	97.2	98.9	97.8	97.4	99	97.9	98.2	99.6
15%	94.4	94.7	97.2	96.1	96.2	99	96.2	95.8	98.7	97.3	97.1	98.6
20%	91.7	92.3	93.9	94.6	94.4	97.6	95.8	95.8	98.2	96.7	97.1	99
25%	91.3	90.6	95.6	92.5	92.8	97.2	94.4	94	97.5	95.2	95.3	98
30%	90.6	89	95.2	89.9	91.4	95.9	92.4	92.4	97	93.1	92.7	97.6

Table 8: Classification accuracy for $\bar{\omega} = 0.015$

$\bar{\omega} = 0.015$												
	3			5			8			12		
	H.C.	M.I.	M.M.	H.C.	M.I.	M.M.	H.C.	M.I.	M.M.	H.C.	M.I.	M.M.
10%	90.5	92	95	91.4	91.7	95.6	89.2	89.5	93.2	87.3	89.5	91.4
15%	88.2	89.9	93	87.9	88.4	92.9	88.8	88.4	93.2	86.9	89.2	91.3
20%	86.4	86.5	90.9	86	86.4	92.5	86.1	86.4	91.5	85.2	86.5	89.6
25%	82.3	84.3	87.5	84.3	83	90.7	84.8	82.9	89.8	86.3	85.2	90.9
30%	82.8	82.4	88.6	83.3	83.1	90	82.5	81.8	88.7	82.8	82.4	88.6

information $(10\%, 15\%, 20\%, 25\%, 30\%)$, while the bigger column headings represent the different dimensions (3,5,8 and 12). The acronym H.C. represents the hierarchical clustering classification accuracies, M.I represents the multiple imputation classification accuracies and M.M. represents the marginal method classification accuracies. All numbers within the cells are percentages.

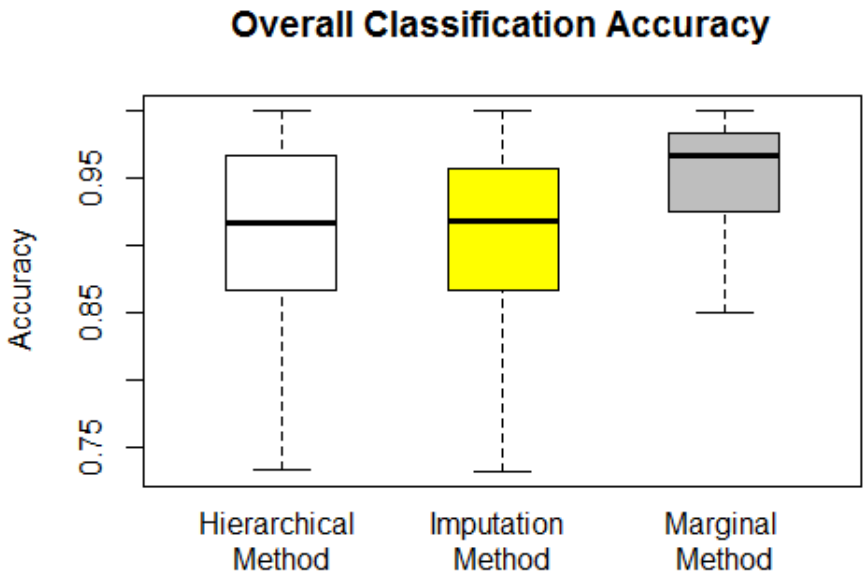


Figure 12: Classification Accuracy for all cases

Figure 12 shows that the marginal method seems to work very well at classifying objects, and has more certainty. Multiple imputation and the hierarchical method both seem to vary very widely. It is a surprising result for multiple imputation as each data point is based on an average of classification accuracies for 45 imputed datasets. One would think that this would lend itself to a much smaller level of variability than shown in the figure.

We then took a closer look at the factors that were varied and the impact this had on our methods.

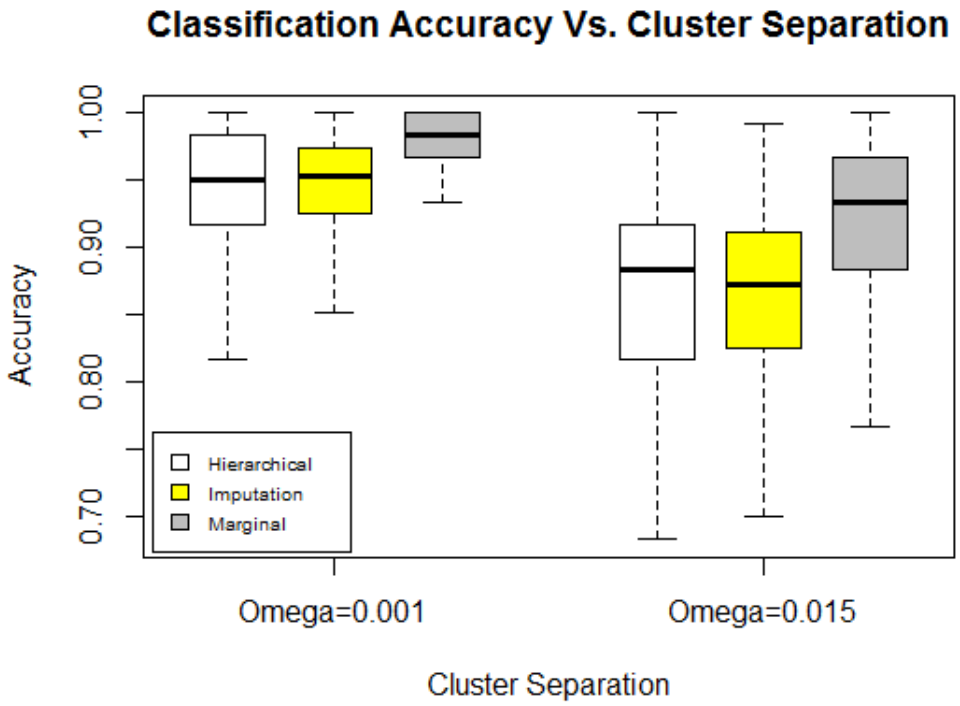


Figure 13: The effect of cluster separation on classification accuracy

There is an expected decline in classification accuracy as the clusters become less separated, however the greatest decline does not come from the marginal method as seen before for cluster discovery (Table 5). The decline in this method is less than the decline in the other two. The change in cluster separation, not only affects the average classfication accuracy of our methods, but shows also a negative effect on the variance of the classification accuracies. The greater the cluster overlap, the greater the uncertainty of classification accuracy. This can be seen across all three methods, hierarchical clustering being the most extreme.

	row.names	hc.1	imp.1	mar.1	hc.2	imp.2	mar.2
1	lower tail	0.817	0.851	0.933	0.683	0.700	0.767
2	lower quartile	0.917	0.925	0.967	0.817	0.826	0.883
3	median	0.950	0.954	0.983	0.883	0.873	0.933
4	upper quartile	0.983	0.974	1.000	0.917	0.912	0.967
5	max	1.000	1.000	1.000	1.000	0.992	1.000

Figure 14: R output of Boxplot statistics from Figure 12

Figure 14 gives the numerical details of figure 13. In figure 14, hc.1 and hc.2 are the hierarchical clustering boxplot accuracy statistics for the two levels of separation. Imp and Mar stand for the Imputation and Marginal approach respectively. The interquartile range for hierarchical clustering went from 6.6% to 10%; Imputation went from 4.9% to 8.6% while the interquartile range for the marginal method rose from 3.3% to 8.4%. The relative increase in interquartile range is actually greater for the marginal method even though it is still more accurate under the decreased cluster separation. It seems that our method can perform better, but shows signs of deteriorating at a faster rate.

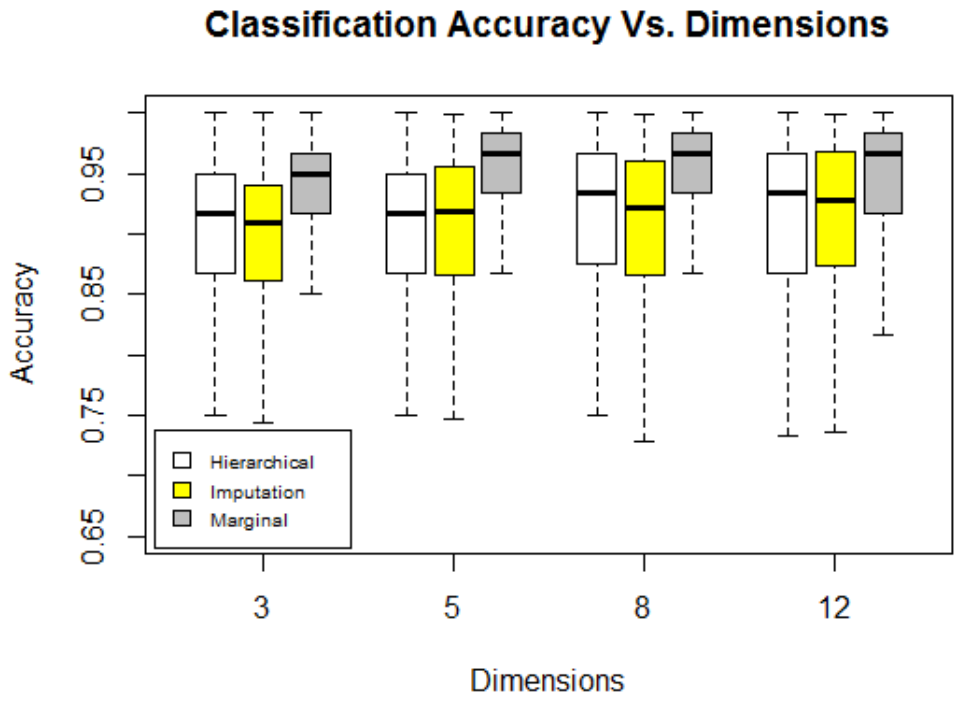


Figure 15: The effect of dimensionality on cluster accuracy

As seen again in figure 15, the marginal method outperforms the other two methods in terms of average classification accuracy and the variability of these accuracy measures. However, across dimensions the interquartile ranges and general spread of the accuracies for the imputation and hierarchical methods seem to remain fairly constant. The marginal method shows some deviation when going for 8 to 12 clusters. One possible explanation, as seen with cluster discovery, is the larger covariance matrix being estimated from 60 observations may be causing some instability.

Lastly we look at the impact of missing values on our methods.

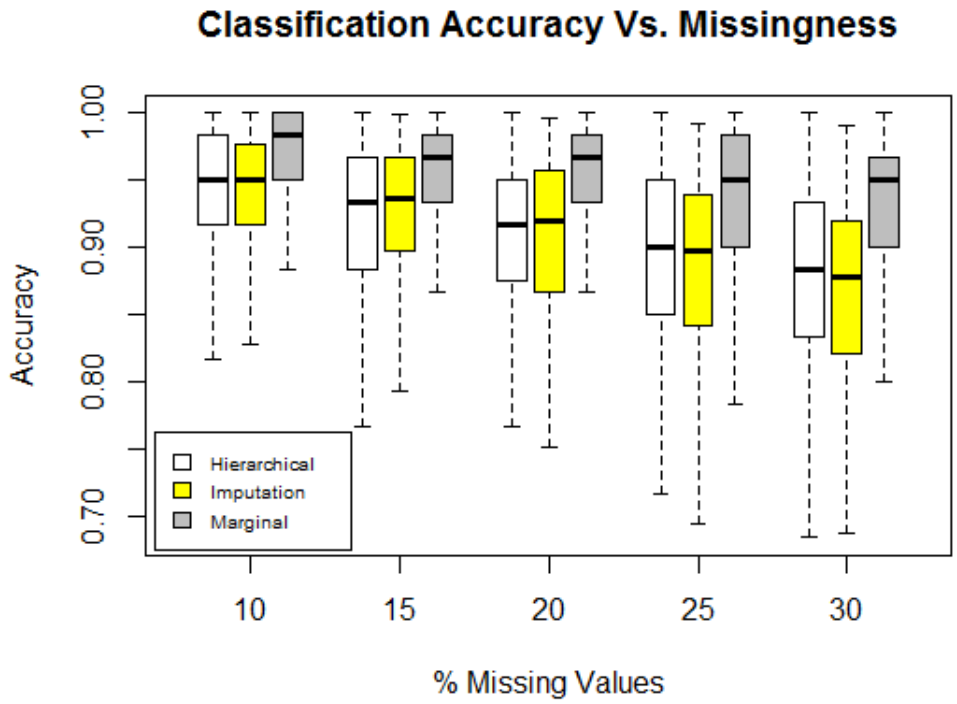


Figure 16: The effect of missingness on classification accuracy

The varying missingness seems to have an equal effect on all the methods. All three methods steadily decrease in accuracy, and increase in uncertainty as the extent of missing information increases. They all seem to deteriorate together. With that being said, the marginal method has shown superiority in classification accuracy over all methods regardless of dimension, cluster separation or missingness, specifically in this neighborhood of sample size and dimensionality (we have neither tested larger sample sizes, or adapted the marginal method to cope with high dimensions).

6.4.3 Real Data: Seeds

The motivation for this thesis came from a dataset with 17 observations, 73 variables and 26% missing information. The algorithm developed in this study is not advanced enough to handle this problem in such high dimensions.

However, we have seen some promise for this algorithm in lower dimensions and wish to test the methods used above, in their ability to discover clusters and classify the seeds. Also being tested is the ability to classify observations with our marginal approach and the hierarchical approach. This data set consists of 7 variables:

1. Area A
2. Perimeter P
3. Compactness $C = 4 * \pi * \frac{A}{P^2}$
4. Length of kernel
5. Width of kernel
6. Asymmetry coefficient
7. Length of kernel groove

All the variables are real-valued continuous.

Firstly, observing that compactness was a linear combination of 2 other variables. This linear dependency may be harmful to our model-based procedures, therefore we removed the first 2 variables, and our analysis was done on only the last 5 variables. This suited us well based on our theory above, that the marginal method should be preferred in 5 or less dimensions.

The algorithm was tested alongside mclust (with no data missing), and both methods found 3 clusters present, with the exact same BIC score. Also, quite fortunately for our algorithm, mclust results showed that the best estimated covariance structure was a common covariance between clusters.

In the simulation, usingMixSim, multivariate normal data were created, for which we found our algorithm performed well even when 30% of the information was missing. However the seeds data, coming from the real world did not have this ideal design, so for just 10% missing information, our log-likelihoods showed some odd behavior; steadily increasing and converging before a rapid descent to -Inf. The marginal method algorithm, performed directly in tune with mclust with complete data, both approaches giving the same positive log-likelihoods. For some unknown reason however, significant values of missingness were not feasible with the marginal method algorithm. and levels of missingness were restricted to $1.5\%, 2\%, 2.5\%$, 3% and 3.5%.

For each level of missingness, 25 simulations were performed. The results for cluster detection can be seen below:

Table 9: Cluster detection for the seeds dataset

Missingness	Marginal	Imputation	Complete Case
1.5%	88%	100%	96%
2%	76%	100%	68%
2.5%	64%	100%	92%
3%	72%	100%	72%
3.5%	80%	100%	80%
Total	76%	100%	81.6%

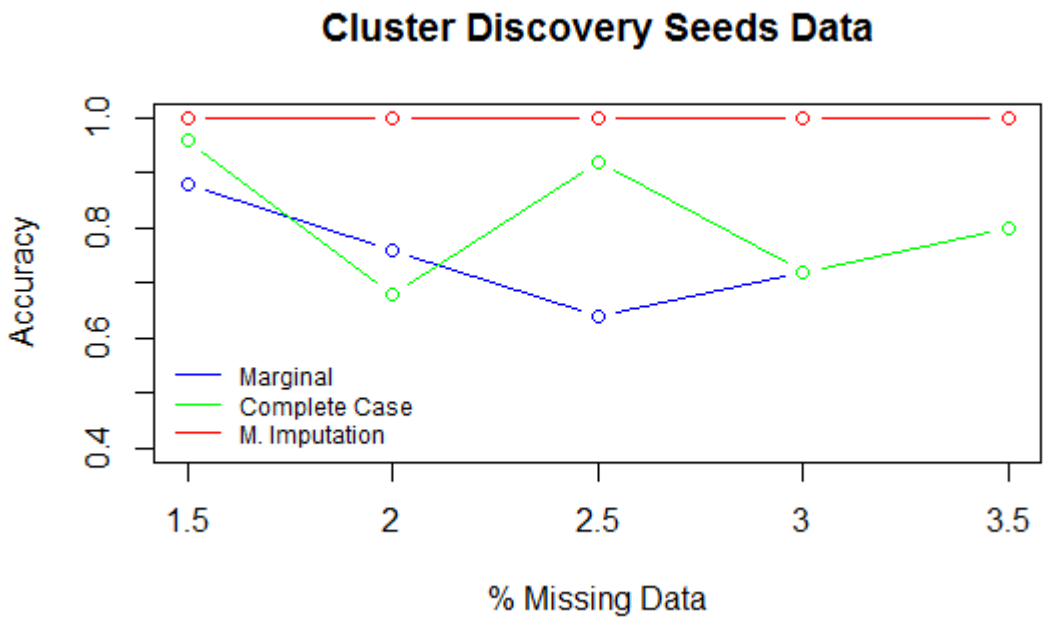


Figure 17: Real Data Cluster Discovery with Missing values

Complete case analysis seems to work much better. A large part is due to the fact that there are 210 seeds in this dataset. Imputation also benefits from

the larger sample space, with more information to predict missing values. This led to the result, opposed to our findings in the simulation study, that the marginal method was not superior in this 5 dimensional setting. Also, we were not able to test at substantial levels of missingness as with the simulated data.

However, there is one advantage that the marginal method has over both these methods. With complete case analysis, those observations that are discarded have no chance of being classified. With imputation, there is also the added relabeling issue to be handled, before we can classify observations. After this you classify observations based on their modal grouping and with 210 observations across 45 imputed datasets, this may turn out to be a daunting task. Achieving fuzzy clustering with the multiple imputation method may be possible. One could average the assignment probability of an observation belonging to a cluster, across all imputed data sets. However, with the marginal method, no observation is left out and therefore unclassified, no relabeling is necessary and fuzzy-clustering is easier to apply to the analysis. In fact, the marginal method was able to classify the objects in the dataset quite accurately.

The ability to classify the objects were compared using the same methods described in section 6.3.2 of this paper.

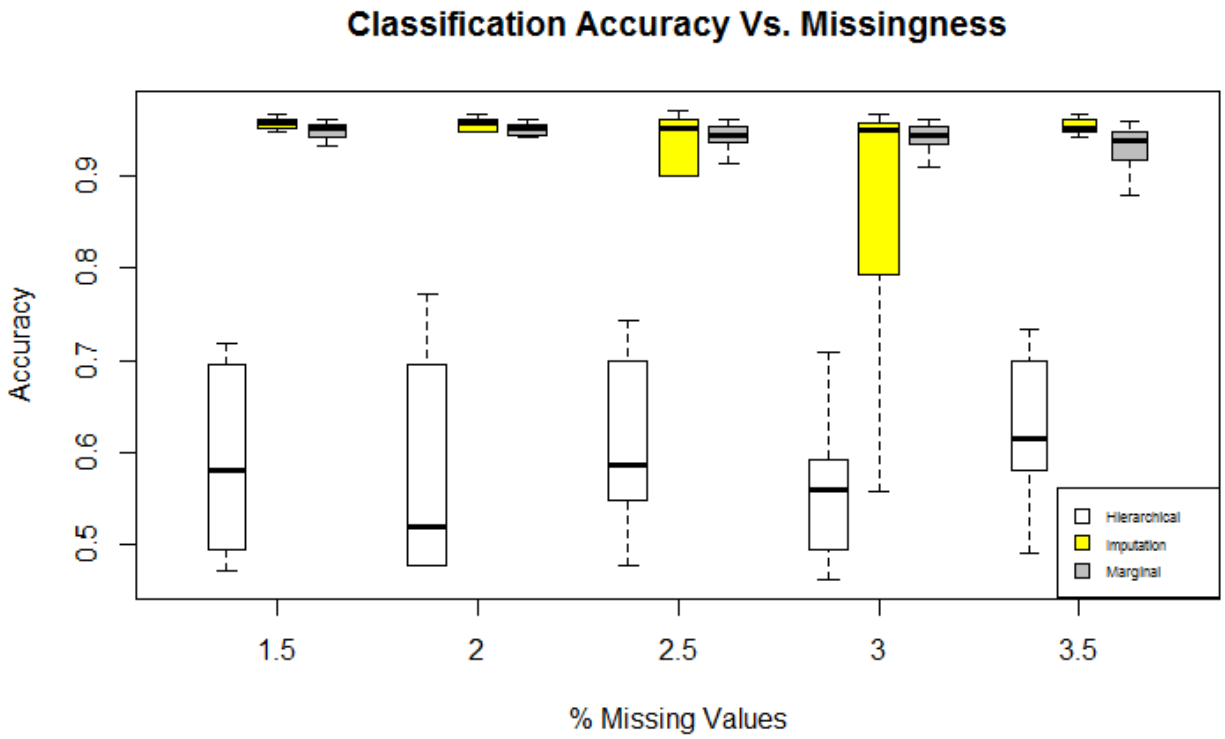


Figure 18: Classification of Seeds with missing values

For the marginal method all classifications were made with an accuracy above 90%, where every observation is classified regardless of having a missing value or not. In one example where there is 2% missing values, 9 out of 210 observations were misclassified. Taking a closer look at the assignment probabilities, it was seen that at least 5 of these misclassified observations, were a 'close call', meaning that the probabilities were between 0.5 and 0.65, thus giving a significant chance of belonging to another cluster. Of these 9 misclassified observations only 2 were incomplete observations. Multiple imputation also proved to be quite effective, with just a small sign of unwanted variability. Hierarchical clustering of this data worked quite poorly. The pairwise overlap for this data was calculated to be $\bar{\omega} = 0.021$, and as seen in figure 12 of the previous section, the significant overlap may be the cause of this poor performance. However this overlap does not seem to have the same effect on imputation and the marginal approach.

7 Discussion

In this thesis, we focused on the ability to cluster when data is missing at random. The data is assumed to be multivariate, quantitative, continuous data so that it is suitably analyzed using Gaussian mixture modeling clustering techniques. Therefore the scope of these results are restricted to quantitative continuous data whose data is missing at random. Theoretical direction is given for cases when the missing data violates randomness. For data that violates the Gaussian assumptions, like discrete data, a mixture of discrete distributions is possible.

Overlap had the strongest effect on all methods, with the most severe victim being the marginal method. The marginal method runs the risk of calculating assignment probabilities, based on the available variables that may not contain the decisive information for clustering. In these cases it may be better to have less information than have misleading information because the latter can lead to the skewing of the cluster properties. Figure 19 shows two clustering examples: one with poorly separated clusters, the other with well separated clusters.

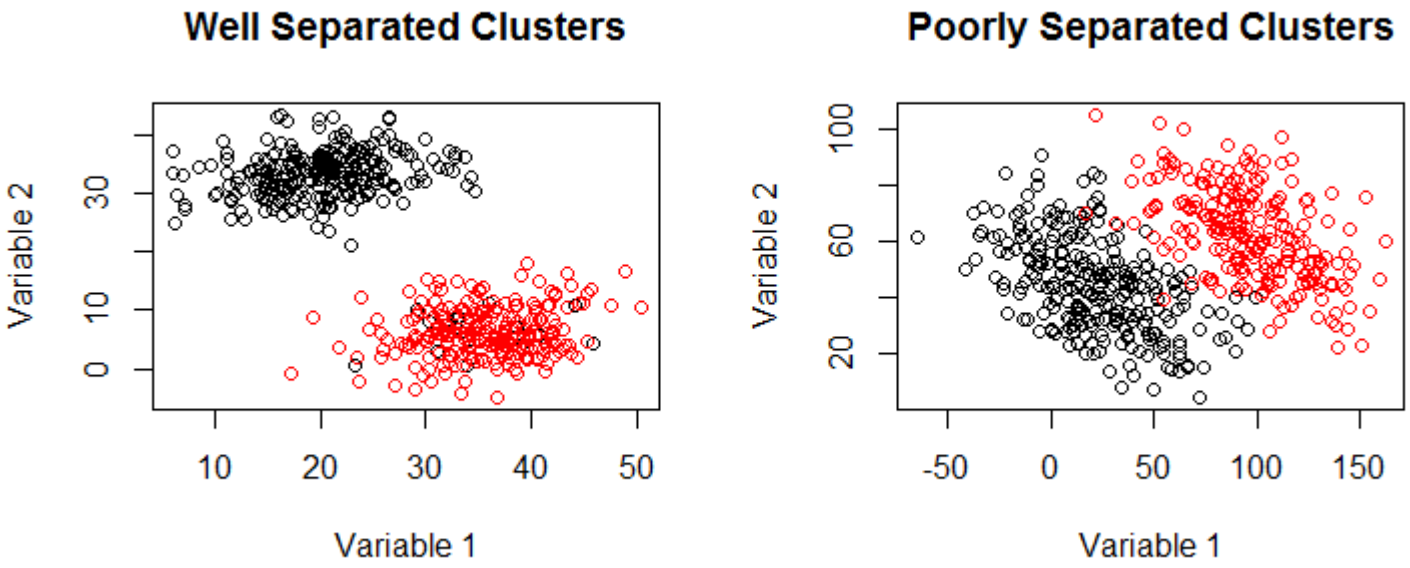


Figure 19: Cluster comparison

If for both these examples, variable 2 was missing, there is still quite a lot of discriminating power in variable 1 to correctly cluster the data. However, in the reverse situation where variable 1 is missing, variable 2 on its own has a very hard time distinguishing between the clusters when looking at the poorly separated clusters. Of course this is not the only way clusters become closer, that is, one dimension at a time. Also, well separated clusters may not be as powerfully discriminant in one or a few variables as shown in the illustration. However after investigating the pattern between poor and well separated clusters, it can be argued that this loss of important discriminant variable, is the reason for the significant decline in the performance of the marginal method as cluster overlap increased.

Multiple imputation seeks to borrow information from other fully observed objects, and when these objects occur in clusters, the borrowed information also contains the cluster trends. For this reason imputation was the most stable even when the clusters became more overlapping. Multiple imputation also remained the strongest performer as the dimensions increased, as this allowed for more predictor information. Dimensionality also had a harsh effect on the marginal method. One reason could be the estimate of the covariance matrix using the pairwise method. The added strain of dimensionality and this pairwise covariance calculation, may have resulted in a covariance matrix that is not positive definite. To alleviate this, as mentioned before we add the constant to the diagonal. It may be the case that the size of the constant required to make the matrix positive definite, sends the covariance estimate very far from its true value, having a skewing effect on all other parameters to be estimated. Thus as the dimensionality increases and so does the constant, the marginal method deteriorates.

The poor performance by the complete case method in the simulated data was probably due to the small sample size, coupled with the high dimensionality which causes more observations to be deleted. With the real data however, these conditions were the opposite, and the complete case performed quite well. Thus, for large sample sizes, in low dimensions with few missing values, the complete case analysis for clustering would be suitable. However, the advent of this thesis was inspired by something much different and hence we tried the marginal approach.

To answer the question of handling the mosquito data, multiple imputation followed by HDclassif is the only option. This step is to be followed by the relabeling of cluster assignments, and finding the modal assignment for every observation. We think, based on our simulations, that this method

will give better results than the reduced data analysis done in the PhD thesis. Note, however, in the PhD thesis, the data was reduced according to incomplete observations and incomplete variables, something we did not do in the simulation. Regardless, we still put more confidence in the multiple imputation method for the mosquito data, because of the stability shown across dimensionality and cluster separation.

All hope is not lost for the marginal method proposed in this thesis. Partial information was still useful to accurately predict the number of clusters 78% of the time for simulated data, as well as the classification precision for the real life data above 95%. However a much better cluster detection accuracy was expected with the real life data. This may be due to the pairwise overlap. The overlap was higher (0.021) in the real data than the simulated data (0.001 and 0.015). We have already discussed the possible reason to why this greater overlap impacts the marginal method so significantly. If the variables missing were the only variables discriminating between clusters, this lack of decisive information can make an incomplete observation more harmful than helpful to the cluster analysis.

Lastly, the classification by hierarchical clustering seems to be a bad gamble. Classification accuracies can go below 52% for this type of multivariate data even when full information is given. The hierarchical clustering algorithm's ability to cluster with missing value will remain a gift for starting values, but not as a final classification.

8 Future Work

Many steps can be taken in the technical aspect. The function and algorithm can probably be coded more efficiently and expanded to do more. Attaining the maximum likelihood parameter estimates in the M-step of the algorithm is also necessary in the development of the marginal method. When this is done, testing can be done when the data is not missing at random.

The performance of this marginal methodology may be improved by borrowing ideology from the high dimensional clustering methods mentioned above. Through eigenvalue decomposition, the important dimensions can be determined. If an observation has missing information in one of these dimensions, then that observation should be deleted, otherwise the missing information would just be from the noisy dimensions and removing it for marginal density calculation would not lead to misleading information.

References

[1] Paul D. Allison. *Handling Missing Data by Maximum Likelihood.* SAS Global Forum, Statistics and Data Analysis, Paper 312-2012, 2012

[2] Amanda N. Baraldi and Craig K. Enders. *An introduction to modern missing data analyses.* Journal of School Psychology, 48, 2010, 5-37

[3] Jeffrey D. Banfield and Adrian E. Raftery. *Model-Based Gaussian and Non-Gaussian Clustering.* Biometrics, Volume 49, No.3, 1993

[4] Xavier Basagaña, Jose Barrera-Gómez, Marta Benet, Josep M. Antó and Judith Garcia-Aymerich. *A Framework for Multiple Imputation in Cluster Analysis.* American Journal Of Epidemiology, 2013

[5] C. Bouveyron, S. Girard and C. Schmid. *High Dimensional Data Clustering.* Computational Statistics and Data Analysis, Vol. 52, No. 1, 2007

[6] Chuong B. Do. *More on Multivariate Gaussians.* Unpublished Manuscript, 2008

[7] Chris Fraley and Adrian E. Raftery. *How many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis.* The Computer Journal, Vol. 41, No. 8, 1998.

[8] Chris Fraley and Adrian E. Raftery. *Model-based clustering, discriminant analysis and density estimation.* Journal of the American Statistical Association, 2002; 97, 458; pg611

[9] John W. Graham. *Missing Data: Analysis and Design.* Springer Science and Business Media, New York, 2012

[10] Zoubin Ghahramani and Michael I. Jordan. *Supervised Learning from incomplete data via an EM approach.* Advances in Neural Information Processing Systems 6, Morgan Kaufmann Publishers, California, 1994

[11] Geoffrey McLachlan and David Peel. *Finite Mixture Models.* Wiley Series, New York, 2000.

[12] Volodymyr Melnykov, Wei-Chen Chen and Ranjan Maitra. *Mixsim: An R package for Simulating data to Study Performance of Clustering Algorithms.* Journal of Statistical Software, Vol. 51, No. 12, 2012

[13] Karl Pearson. *Contributions to the mathematical theory of evolution.* Phil. Trans. R. Soc. Lond. A 1894 185 71-110, 1894

[14] Remco A. Suer. *Unraveling the malaria mosquito's sense of smell.* PhD Thesis, Wageningen University, 2011

9 Appendix: R code

Mclust applied to Mosquito data:

```
library(mclust)

fit <-  Mclust(Amat, G=1:15)

> fit$G
[1] 11

> fit$modelName
[1] "EEI"

> fit$bic
[1] -2143.443
```

HDclassif applied to Mosquito data:

```
library(HDclassif)

fit <- hddc(Amat,model="all", algo="SEM", d="Cattell")

> fit$K
[1] 2

> fit$model
[1] "AKJBKQKD"

> fit$BIC
[1] -3279.779
```

Functions for marginal densities:

```
library(mvtnorm)

dmvnorm.nm <- function(x,mu,covm, log=FALSE){
x.nm <- x[!is.na(x)]
mu.nm <- mu[!is.na(x)]
covm.nm <- covm[!is.na(x),!is.na(x)]
if (length(covm.nm)==1) covm.nm <- matrix(covm.nm)
dmvnorm(x.nm, mu.nm, covm.nm, log=log)
}

dmvnorm.nm.a <- function(y,mu,covm, log=FALSE){
apply(y, 1, dmvnorm.nm, mu, covm, log)
}
```

Algorithm for model-based allowing missing values:

```
library(ape)  ##For imputing distance matrix

d <- dist(y, method="euclidean")
d2 <- ultrametric(d)  ##impute distance matrix
d <- as.dist(d2)
fit <- hclust(d, method="ward.D2")

# Some starting values for mu, sd, and pi

iv <- cutree(fit, k=2)  #Testing for 2 clusters

mu1 <- apply(y[which(iv==1),], 2,mean, na.rm=T)
mu2 <- apply(y[which(iv==2),], 2,mean, na.rm=T)

covm <- 0.5*diag(p)  #p is number of dimensions

pi1 <- sum(iv==1)/N
pi2 <- sum(iv==2)/N

LL <- rep(0, 2)
```

```
c <- 500

while (abs(c) > 0.0000001){

(LL[1] <- sum(log(pi1*dmvnorm.nm.a(y, mu1, covm) + pi2*dmvnorm.nm.a(

p1 <- dmvnorm.nm.a(y, mu1, covm)
p2 <- dmvnorm.nm.a(y, mu2, covm)

postp <- cbind(pi1*p1,pi2*p2) / rowSums(cbind(pi1*p1,pi2*p2))

# M-step

pi1 <- mean(postp[,1])
pi2 <- mean(postp[,2])

w1 <- postp[,1]
mu1 <- apply(y, 2, weighted.mean, w1, na.rm=TRUE)

w2 <- postp[,2]
mu2 <- apply(y, 2, weighted.mean, w2, na.rm=TRUE)

yc1 <- t((t(y)-mu1))
yc2 <- t((t(y)-mu2))

w.yc1 <- sqrt(w1)*yc1
w.yc2 <- sqrt(w2)*yc2

df <- crossprod(!is.na(y))-1

sumw1 <- t(!is.na(y)) %*% diag(w1) %*% !is.na(y)
sumw2 <- t(!is.na(y)) %*% diag(w2) %*% !is.na(y)

w.yc1[is.na(w.yc1)] <- 0
w.yc2[is.na(w.yc2)] <- 0

covm <- (crossprod(w.yc1) + crossprod(w.yc2) )/(sumw1+sumw2)
```

```
(LL[2] <- sum(log(pi1*dmvnorm.nm.a(y, mu1, covm) + pi2*dmvnorm.nm.a(

c <- LL[2] - LL[1]
}

par.2 <- 2*p + (2-1) + p*(p+1)/2
BIC.2 <- -2*LL[2] +(log(N)*par.2)
```

Creating data and missing values:

```
library(MixSim)

K <- 3
p <- 8
N <- 60
p.ms <- 0.25  #Percentage missing values

Q <- MixSim(BarOmega = 0.015, K = K, p = p, sph=FALSE, hom=TRUE, PiL
A <- simdataset(n = N, Pi = Q$Pi, Mu = Q$Mu, S = Q$S)

y <- A$X

ms <- p.ms*p*N

na <- sample(c(1:N), ms, replace=TRUE)
k <- sample(c(1:p), ms, replace=TRUE)

for (i in 1:ms){
y[na[i],k[i]]<-NA
}

cbind(y,A$id)
```

Code for imputation and complete case methods:

```
library(mice)

### complete case

y.cc <- na.omit(y)
fit2 <- Mclust(y.cc, G=2, modelNames="EEE")
fit2$bic
fit2$G

##Imputation

imp <- mice(y, m=45)
k <- rep(0,45)

for(i in 1:45){

fit3<- Mclust(complete(imp,i),G=2:6)

k[i]<- fit3$G
}

plot(1:45, k)
```