Clustering Microarray Data Via a Bayesian Infinite Mixture Model

by

Dena Givari

A Thesis

Presented to

The University of Guelph

In partial fulfilment of requirements

for the degree of

Master of Science

in

Applied Statistics

Guelph, Ontario, Canada

ABSTRACT

Clustering Microarray Data Via a Bayesian Infinite Mixture Model

Dena Givari
University of Guelph, 2012

Advisors:
Dr. Paul McNicholas
Dr. Ryan Browne

Clustering microarray data is a helpful way of identifying genes which are biologically related. Unfortunately, when attempting to cluster microarray data, certain issues must be considered including: the uncertainty in the number of true clusters; the expression of a given gene is often affected by the expression of other genes; and microarray data is usually high dimensional. This thesis outlines a Bayesian infinite Gaussian mixture model which addresses the issues outlined above by: not requiring the researcher to specify the number of clusters expected, applying a non-diagonal covariance structure, and using mixtures of factor analyzers and extensions thereof to structure the covariance matrix such that it is based on a few latent variables. This approach will be illustrated on real and simulated data.

Acknowledgments

The author would like to thank the Newton Lab and especially Dr. Paul McNicholas and Dr. Sanjeena Subedi for their unwavering support and patience.

Table of Contents

List of Tables

Chapter 1

Introduction

What is to follow in the coming pages is a detailed description of a model-based Bayesian algorithm inspired by the work of Medvedovic and Sivaganesan (2002). The primary purpose of this algorithm is to cluster microarray data. This introduction will begin by outlining the nature of microarray data, followed by the work done by Medvedovic and Sivaganesan (2002), an overview of model-based clustering and finally the role of parsimonious Gaussian mixture models (McNicholas and Murphy, 2010, 2008; McNicholas et al., 2010; McNicholas, 2010) in the algorithm presented in this paper. The sections that follow will discuss the model applied to the data, the coding process of the algorithm, and an overview of the performance of the algorithm on five different data sets. Finally, a conclusion will aim to summarize the key ideas introduced in this work.

1.1 The Nature of Microarray Data

The effective assessment of clustering methods applied to genetic data requires an understanding of not only how microarray technology works but also why it works. The genetic information within eukaryotic cells, the building block of mam-

mals, is contained in molecules of DNA. The DNA is comparable to a library which holds records of the various possible functions a cell is able to carry out. Naturally, not all functions manifest under the same conditions. Specific manifestations occur in response to specific cellular conditions through a process of selective transcription of a gene or a group of genes which results in the production of mRNA that are responsible for creating specific proteins which ultimately carry out the function of genes.

Transcription is the process by which the DNA sequence of a gene is read by the cell to form a complementary mRNA molecule. The mRNA molecule produced is that DNA sequence's functional counterpart. If the role carried out by a given DNA sequence, or *gene*, is in high demand within a cell, that DNA sequence will be transcribed multiple times to produce multiple mRNA molecules. In this manner, the amount of a given mRNA molecule within a cell implies the level of expression of its corresponding gene. Microarray technology acknowledges this implication and measures the level of mRNA molecules in a cell to infer levels of gene expression. Genetic researchers have worked to create a database of *probes* which are short nucleotide sequences that have been documented to exist within the DNA sequence of various genes. Such probes serve various functions, one of which is to hybridize with mRNA that has been extracted from a cell. Microarray technology differentially labels these short nucleotide sequences with fluorescent dyes to allow researchers to compare levels of different mRNA within the cell. The fluorescent probes are left to hybridize with the mRNA from a cell for some time, this sample is then washed to remove unhybridized probes. The researcher then measures the different levels of fluorescence to

attain a direct indication of the amount of complementary mRNA. The higher the level of gene expression, the higher the number of mRNA and subsequently the higher the level of hybridized flourescent probes. Thus the level of fluorescence is a report of the level of gene expression (Mark et al., 1995).

Scientists can use microarray technology to study how the expression level of a gene changes under different conditions. Studying multiple genes in this manner allows the scientist to identify genes with similar expression patterns across conditions that demonstrate variable temporal, developmental, histological and physiological patterns. This gives insight to the biological functions of genes (Brown and Botstein, 1999) and their degree of inter-relatedness. The application of cluster analysis to microarray data is commonly used to satisfy this objective (D'haeseleer et al., 2000). Here, we will focus only on the analysis of gene expression data using model-based clustering.

1.2 Model-Based Clustering

Model-based approaches to clustering require the assumption that the data follow a mixture of probability distributions (Wolfe, 1963; Symons, 1981; McLachlan and Basford, 1988). Given a dataset of a multivariate random variable $\mathbf{y}$, such that $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ are n independent observations derived from G components, the following is the likelihood for the mixture model:

$$L(\theta_1, \ldots, \theta_G; \tau_1, \ldots, \tau_G | \mathbf{Y}) = \prod_{i=1}^{n} \sum_{g=1}^{G} \tau_g f_g(\mathbf{y}_i | \theta_g)$$

where f_g represents the density of the g^{th} component, θ_g represents the parameters of the g^{th} component and τ_g is the probability that any randomly selected observation belongs to the g^{th} component ($\tau_g > 0; \sum_{g=1}^{G} \tau_g = 1$). In the case of the Gaussian mixture model, f_g is parameterized by mean μ_g and covariance matrix $\mathbf{\Sigma}_g$, the likelihood for the mixture model becomes:

$$L(\mu_1, \ldots, \mu_G; \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_G; \tau_1, \ldots, \tau_G | Y) = \prod_{i=1}^{n} \sum_{g=1}^{G} \tau_g \phi_g(\mathbf{y}_i | \mu_g, \mathbf{\Sigma}_g),$$

where ϕ represents the multivariate Gaussian probability distribution.

The Gaussian mixture model can work well in both cases where the number of clusters are known and unknown (Yeung et al., 2001). However, the latter case is an added obstacle to clustering problems. Medvedovic and Sivaganesan (2002) recognized this obstacle and decided to bypass the need to account for the number of clusters by applying a Bayesian mixture model. Their procedure builds clusters based on the posterior probability distribution of the clusterings given the data.

One flaw in the work of Medvedovic and Sivaganesan (2002) is their assumption of an isotropic covariance structure. In order to successfully apply model-based clustering to microarray data, it is important for the model to reflect the true nature of the data. The insinuation that the expression level of a certain gene is independent of the expression level of all other genes in a cell is not consistent with what is currently known about gene expression. One advantage of not creating protein directly from DNA is that it allows the cell to have multiple levels of control over gene expression. One of the cell's primary methods for regulating a gene's expression level is the

expression level of other genes. In reality, the variation in a given gene's expression level is dependent on other gene's expression levels within the cell. Medvedovic and Sivaganesan (2002) were flawed in assuming independence among expression levels of genes and this has motivated the development of the Bayesian algorithm presented in this paper.

1.3 Parsimonious Gaussian Mixture Models

We can obtain a model that is more reflective of the true nature of microarray data by applying a non-diagonal covariance structure. However, another problem with microarray data is that it involves a usually large number of variables. This problem is referred to as *the curse of dimensionality*. This *curse* describes the following phenomenon: in very high dimensions (i.e., large number of measured variables), data points spread out such that they all seem equidistant to one another (Haque et al., 2004). In a more technical sense, the difference in the distance between the furthest points and the closest points approaches 0 as the dimension of the data approaches infinity (Beyer et al., 1999; Steinbach et al., 2003):

$$\lim_{d\to\infty} \frac{MaxDist - MinDist}{MinDist} = 0$$

McNicholas and Murphy (2008, 2010) proposed a solution to this problem by introducing a family of parsimonious Gaussian mixture models (hereafter referred as PGMMs). PGMMs generalize the mixtures of factor analyzers model (Ghahramani

and Hinton, 1997) to allow for the option of applying common structures in the covariance matrices across all components. This allows for a more parsimonious model and thus more stable estimates (McNicholas and Murphy, 2010).

McNicholas and Murphy (2008) generalized the covaraince structure of the factor analysis model by applying, or not applying, the following constraints: $\mathbf{\Lambda}_g = \mathbf{\Lambda}$, $\mathbf{\Psi}_g = \mathbf{\Psi}$ and $\mathbf{\Psi}_g = \psi\mathbf{I}_p$. McNicholas and Murphy (2010) further expanded this generalization by writing $\mathbf{\Psi}_g = \omega_g\Delta_g$, where $|\Delta_g| = 1$ thereby allowing constraints to be placed on, or not placed on, ω_g and Δ_g. The collective work resulted in a family of 12 parsimonious Gaussian mixture models. This paper presents the original 8 structures in Table 1.1.

Table 1.1: The following table presents the original family of 8 PGMMs.

$\mathbf{\Lambda}_g = \mathbf{\Lambda}$	$\mathbf{\Psi}_g = \mathbf{\Psi}$	Isotropic	Covariance Structure
C	C	C	$\mathbf{\Sigma}_g = \mathbf{\Lambda}\mathbf{\Lambda}^T + \psi\mathbf{I}_p$
C	C	U	$\mathbf{\Sigma}_g = \mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi}$
C	U	C	$\mathbf{\Sigma}_g = \mathbf{\Lambda}\mathbf{\Lambda}^T + \psi_g\mathbf{I}_p$
C	U	U	$\mathbf{\Sigma}_g = \mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi}_g$
U	C	C	$\mathbf{\Sigma}_g = \mathbf{\Lambda}_g\mathbf{\Lambda}^T{}_g + \psi\mathbf{I}_p$
U	C	U	$\mathbf{\Sigma}_g = \mathbf{\Lambda}_g\mathbf{\Lambda}^T{}_g + \mathbf{\Psi}$
U	U	C	$\mathbf{\Sigma}_g = \mathbf{\Lambda}_g\mathbf{\Lambda}^T{}_g + \psi_g\mathbf{I}_p$
U	U	U	$\mathbf{\Sigma}_g = \mathbf{\Lambda}_g\mathbf{\Lambda}^T{}_g + \mathbf{\Psi}_g$

The algorithm presented in this thesis has adopted the hierarchical Bayesian model outlined by Medvedovic and Sivaganesan (2002) and applied the CCU and CUU decomposition to allow for parsimonious non-diagonal covariance structures. In this way, a Bayesian algorithm was developed which bypasses the need for specifying the number of clusters. By applying PGMMs it reduces the number of free parameters and thus allows for better handling of high-dimensional microarray data. Also, the

algorithm reflects the true nature of gene expression by not assuming independence between genes.

Chapter 2

The Model

This chapter will begin by discussing the model in general. Afterwards we will implement the PGMM covariance structures and discuss specifically, the application of the CCU and CUU covariance structures to the model.

2.1 The Model without PGMMs

The initial model that the derived algorithm is based on is a hierarchical Gaussian mixture model with the structure in Figure 2.1.

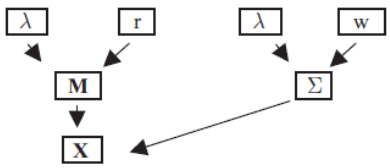

Figure 2.1: The nature of the dependencies of the model are represented above by a directed acyclic graph (Cowell and Spiegelhalter, 1999).

Suppose from G clusters, T observations have risen and for each observation p variables have been measured. The variable c_i will denote the cluster membership of non-breaking space; i.e., observation i. If it is assumed that each cluster is derived

from a component of a Gaussian mixture model, then the density of the model is:

$$p(\mathbf{x}_i \mid c_i = g, \boldsymbol{\mu}_g, \ldots, \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_g, \ldots \boldsymbol{\Sigma}_G) = \phi(\mathbf{x_i} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g),$$

where $\mathbf{x}_i$ is the vector reporting the measures of the p variables for observation i; c_i denotes the classification of observation i; $\boldsymbol{\mu}_g$ is the mean of the g^{th} component and $\boldsymbol{\Sigma}_g$ is the covariance matrix of component g. To cluster the observations an iterative algorithm will be used. First, the algorithm will aim to increase the likelihood as a function of the parameters given the current classification of observations. Secondly, the algorithm will update the classification by using the posterior distribution of the hyperparameters and parameters. In laymann terms, if there is room to increase this likelihood by re-classifying some or all observations, it will do so; otherwise, classification of observations will remain the same from iteration-to-iteration.

The algorithm is initialized by assigning all the observations to the same cluster. Then the mean and the covariance matrix based on the entire data set, $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$, are used as parameters for the prior distribution of the hyperparameters $\boldsymbol{\lambda}$, $\mathbf{r}$, β, $\mathbf{W}$. These prior distributions are:

$$p(\boldsymbol{\lambda} \mid \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) = \phi(\boldsymbol{\lambda} \mid \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x),$$

where $\boldsymbol{\lambda}$ follows a multivariate Gaussian distribution.

$$p(\mathbf{r} \mid \boldsymbol{\Sigma}_x, p) = f_{\mathbf{W}^{-1}}(\mathbf{r} \mid p, 2{\boldsymbol{\Sigma}_x}^{-1}),$$

where $\mathbf{r}$ follows an inverse Wishart distribution and p is the dimension of the data. β follows a Gamma distribution as noted below:

$$p(\beta) = f_{Gamma}(\beta \mid 1/2, 1/2),$$

and $\mathbf{W}$ follows an inverse Wishart distribution:

$$p(\mathbf{W} \mid \mathbf{\Sigma}_x, p) = f_{\mathbf{W}^{-1}}(\mathbf{W} \mid p, \mathbf{\Sigma}_x/2).$$

Upon sampling from the prior distributions of hyperparameters, sampling from the prior distributions of the parameters may proceed:

$$p(\boldsymbol{\mu}_g \mid \boldsymbol{\lambda}, \mathbf{r}) = \phi(\boldsymbol{\mu}_g \mid \boldsymbol{\lambda}, \mathbf{r}^{-1}),$$

$$p(\mathbf{\Sigma}_g \mid \beta, p, \mathbf{W}) = f_{\mathbf{W}^{-1}}(\mathbf{\Sigma}_g \mid \beta + p, \mathbf{W}),$$

where $\boldsymbol{\mu}_g$ follows a multivariate Gaussian distribution and $\mathbf{\Sigma}_g$ follows an inverse Wishart distribution. Sampling from the prior distribution of the parameters $\boldsymbol{\mu}_g$ and $\mathbf{\Sigma}_g$ (note that the algorithm is started with only one component) allows the re-evaluation of the classifcation of observations. This is done by calculating the probability of each observation given the sampled parameters. The probabilities of two observations belonging to a cluster whose component is parameterized by the sample $\boldsymbol{\mu}_g$ and $\mathbf{\Sigma}_g$ are much more similar for observations which in fact belong to the

same cluster as compared to observations which do not belong to the same cluster. As a simple example, suppose there are four observations: $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$. Observations $\mathbf{x}_1$ and $\mathbf{x}_2$ belong to cluster 1 and observations $\mathbf{x}_3$ and $\mathbf{x}_4$ belong to cluster 2. The probability of observation $\mathbf{x}_1$ belonging to a cluster whose component distribution is parameterized by the sampled $\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_1$ is much more similar to the probability of observation $\mathbf{x}_2$ belonging to cluster 1 than the probability of observation $\mathbf{x}_3$ or $\mathbf{x}_4$ belonging to cluster 1. Thus in the first iteration, observations are clustered together if their probabilities of being derived from the single existing component are similar.

At this point, observations have been reclassified, for each existing cluster the mean and covariance matrix of the data is computed and sampling from the posterior distribution of hyperparameters and parameters may proceed. In the iterations to follow, the algorithm computes the probability of observations belonging to each existing cluster in the following way:

$$p(c_i = g \mid c_{-i}, \mathbf{x}_i, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) = \frac{b\, n_{-i,g}}{T} \times \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g),$$

where b is a normalizing constant, $n_{-i,g}$ is the number of observations in cluster g not including observation i, and T is the number of observations. Observations are reassigned to the cluster for which they have the greatest probability of belonging. In general, a cluster is removed if all corresponding observations are lost to another cluster and a cluster is created if an observation is found to have a low probability of belonging to any existing cluster. At the end of each iteration, the algorithm may have moved observations from cluster to cluster and/or split the data set into more clusters.

This continues until a pre-determined maximum number of iterations is reached. By this point the number of clusters should not be changing from iteration-to-iteration and there should be minimal movement of observations between clusters.

As insinuated above, after the initialization phase is complete, hyperparameters and parameters may be sampled from their posterior distributions. It is important to note that the hyperparameter β does not have a conjugate prior distribution and we use importance sampling to overcome this difficulty. The following is a list the posterior distributions of the hyperparameters and parameters:

$$p(\mathbf{W}) = f_{\mathbf{W}^{-1}}(\mathbf{W} \mid p, \boldsymbol{\Sigma}_x/2 + 1\beta \times \sum_g \boldsymbol{\Sigma}_g),$$

where $\boldsymbol{\Sigma}_g$ is the covariance of component g,

$$p(\mathbf{r}) = f_{\mathbf{W}^{-1}}(\mathbf{r} \mid p + Q, 2{\boldsymbol{\Sigma}_x}^{-1} + Q),$$

where Q is the number of clusters which exist at the current iteration,

$$p(\boldsymbol{\lambda}) = \phi(\boldsymbol{\lambda} \mid (\boldsymbol{\Sigma}_x^{-1}Q + \mathbf{r}^{-1})^{-1} \times (\boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x^T + Q\mathbf{r}^{-1}\boldsymbol{\mu}_x^T), \boldsymbol{\Sigma}_x^{-1}Q + \mathbf{r}^{-1}),$$

$$p(\boldsymbol{\Sigma}_g) = f_{\mathbf{W}^{-1}}(\beta + p \times n_g, \mathbf{W} + \boldsymbol{\Sigma}_g),$$

where n_g denotes the number of observations assigned to cluster g,

$$p(\boldsymbol{\mu}_g) = \phi(\boldsymbol{\mu} \mid ((\mathbf{r} + \boldsymbol{\Sigma}_g \times n_g)^{-1} \times \mathbf{r} \times \boldsymbol{\lambda} + (\boldsymbol{\Sigma}_g \times n_g) \times (\boldsymbol{\mu}_g))^T, \mathbf{r} + \boldsymbol{\Sigma}_g \times n_g),$$

where $\boldsymbol{\mu}_g$ is the mean of component g.

Using the newly-sampled hyperparameter and parameter values, the reclassification of each observation, the removal of cluster(s) and addition of a cluster is carried out (see chapter three for details on how this is done). All proceeding iterations will make use of the stated posterior distributions until convergence is reached.

2.2 The Model with PGMMs

The application of PGMMs suggests the behaviour of p measured variables is due to the behaviour of q latent variables such that $q \ll p$. This allows the expression of $\mathbf{x}_i$ to be written as:

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

which results in the following marginal distribution of $\mathbf{x_i}$:

$$\mathbf{x}_i \sim N(\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}),$$

incorporating this to the model described above produces the following dependencies:

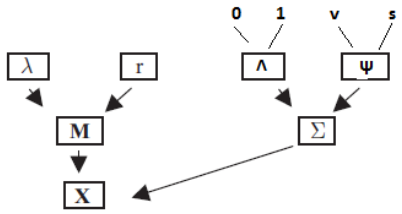

Figure 2.2: The nature of the dependencies of the model when PGMMs are applied are represented above by a directed acyclic graph (Cowell and Spiegelhalter, 1999).

Notice the only change made by the application of PGMMs is in the components of the covariance structure. Since, the structure of the component covariance matrices have been decomposed, β and $\mathbf{W}$ are no longer used as hyperparameters. Instead, the covariance matrices are derived based on samples from the prior and posterior distributions of the factor weights matrix $\mathbf{\Lambda}$ and the noise matrix $\mathbf{\Psi}$. All other factors of the model remain the same except for an additional step which computes a matrix $\mathbf{U}$ which reports latent factor measurements of all observations. Lopes and West (2004) described prior and posterior distributions for the weights matrix, $\mathbf{\Lambda}$, and the diagonal noise matrix $\mathbf{\Psi}$. The prior distribution of the $\mathbf{\Lambda}$ matrix is:

$$p(\mathbf{\Lambda})_{ij} = \phi(\mathbf{\Lambda}_{ij} \mid 0, 1)\mathbf{1}\{i = j, \mathbf{\Lambda}_{ij} > 0\}.$$

Such that the diagonal elements of $\mathbf{\Lambda}$ are positive. The prior distribution of the diagonal elements of $\mathbf{\Psi}$ is:

$$p(\psi_j \mid v, s^2) = f_{IG}(\psi_j \mid v/2, vs^2/2),$$

where f_{IG} is the $p.d.f$ of the Inverse Gamma distribution. Before sampling from the posterior distributions of $\mathbf{\Lambda}$ and $\mathbf{\Psi}$, a matrix $\mathbf{U}$ is created. $\mathbf{U}$ contains T vectors reporting sampled latent variable measures for each observation and is used when sampling from the posterior distributions of $\mathbf{\Lambda}$ and $\mathbf{\Psi}$. The vector $\mathbf{U}_i$ has the following multivariate Gaussian distribution (Lopes and West, 2004):

$$p(\mathbf{U}_i) = \phi(\mathbf{U}_i \mid (\mathbf{I}_q + \mathbf{\Lambda}^T\mathbf{\Psi}^{-1}\mathbf{\Lambda})^{-1}\mathbf{\Lambda}^T\mathbf{\Psi}^{-1}\mathbf{x}_i, (\mathbf{I}_q + \mathbf{\Lambda}^T\mathbf{\Psi}^{-1}\mathbf{\Lambda})^{-1}).$$

Once the matrix of latent factors has been created, sampling may proceed from the posterior distributions of $\mathbf{\Lambda}$ and $\mathbf{\Psi}$. The posterior distribution of the first q rows of $\mathbf{\Lambda}$ is the following multivariate Gaussian distribution (Lopes and West, 2004):

$$p(\mathbf{\Lambda}_j) = \phi(\mathbf{\Lambda}_j \mid (\mathbf{I}_j + \psi_j^{-1}\mathbf{U}_j^{'}\mathbf{U}_j)^{-1}(\mathbf{\Lambda}_{0,j}\mathbf{1}_j + \psi_j^{-1}\mathbf{U}_j^{'}\mathbf{x}_j), (\mathbf{I}_j +_j^{-1} \mathbf{U}_j^{'}\mathbf{U}_j)^{-1}),$$

where $\mathbf{x_j}$ is a vector reporting the measures of variable j for all observations and $\mathbf{\Lambda}_{0,j}$ is a vector reporting the first j elements of row j in the $\mathbf{\Lambda}$ matrix sampled from the prior distribution. The following is the multivariate Gaussian posterior distribution of rows $q+1, \ldots, p$ in the $\mathbf{\Lambda}$ matrix (Lopes and West, 2004):

$$p(\mathbf{\Lambda}_j) = \phi(\mathbf{\Lambda}_j \mid (\mathbf{I}_q + \psi_j^{-1}\mathbf{U}^{'}\mathbf{U})^{-1}(\mathbf{\Lambda}_{0,j}\mathbf{1}_k + \psi_j^{-1}\mathbf{U}^{'}\mathbf{x}_j), (\mathbf{I}_q + \psi_j^{-1}\mathbf{U}^{'}\mathbf{U})^{-1}).$$

Finally, the inverse Gamma posterior distribution of the diagonal elements of $\mathbf{\Psi}$ is:

$$p(\psi_j) = f_{IG}(\psi_j \mid (v+T)/2, (vs^2 + d_j)/2)$$

where ψ_j is the j^{th} diagonal element of $\mathbf{\Psi}$ and,

$$d_j = (\mathbf{x}_j - \mathbf{U}\mathbf{\Lambda}_j^{'})^{'}(\mathbf{x}_j - \mathbf{U}\mathbf{\Lambda}_j^{'}).$$

Note, $\mathbf{\Lambda}$ is sampled in the same way for both CCU and CUU, $\mathbf{\Psi}$ is sampled differently in the two cases. In both cases, each diagonal element of $\mathbf{\Psi}$ is sampled independently, however in CCU all clusters are assigned the same $\mathbf{\Psi}$ matrix whereas in CUU each cluster has its own unique $\mathbf{\Psi}_g$ matrix. Thus, the posterior distribution of $\mathbf{\Psi}$ is slightly different in the two cases. When CCU is implemented, $\mathbf{x}_j$ in the posterior distribution of $\mathbf{\Psi}$ refers to the j^{th} variable of every observation in the data set. However when CUU is implemented, $\mathbf{x}_j$ refers to the observations in the cluster for which $\mathbf{\Psi}$ is being sampled.

2.3 Model Selection: BIC

The criteria used for model selection is the Bayesian Information Criterion (BIC; Schwarz, 1978):

$$\text{BIC} = -2\ln(\mathcal{L}) + k\log(n),$$

where $\mathcal{L}$ is the likelihood of the model given the data, k is the number of free parameters and n is the number of observations (Anderson and Burnham, 2004). The BIC seeks to favor the most probable model given the data (Cavanaugh, 2009) and does so by considering the likelihood of the data and the number of free parameters. The better the model, the lower the BIC value (Seltman, 2012). In this way, the BIC is used to select the structure of the covariance matrix, the number of latent factors (q) and the number of clusters (G).

Chapter 3

Methodology

In this chapter, we will review the required software to run the algorithm and move on to a detailed explanation of the implemented procedure given a non-diagonal covariance structure. This will be followed by a discussion of the changes required in order to apply the CCU and CUU covariance structures.

3.1 Software

This algorithm was coded in *R* version 2.10.0 and requires the following libraries: `QRMlib`, `MCMCpack`, `mclust`, `bayesSurv`, `cubature`, and `mvtnorm`. The `QRMlib` package allows for operations such as column sum and mean as required by the algorithm. `MCMCpack` allows sampling from the various hyperparameter and parameter distributions. `mclust` allows the use of the *maximum a posteriori estimation* in order to update the classification of the observations. The `bayesSurv` package allows sampling from the inverse Wishart distribution. Finally, the `mvtnorm` package permits sampling from the multivariate Gaussian distribution.

3.2 The Procedure in General

3.2.1 Set-Up

The procedure begins by first uploading the required libraries and the data into R. The following pseudo-code describes the set-up of the procedure:

- Create a $T \times m$ matrix C such that T is the number of observations and m is the maximum number of iterations the algorithm will run. Each column in C represents a given iteration and reports the classification of all observations in that iteration.
- $C[, 1] = \mathbf{1}$ since all observations in iteration 1 belong to the same cluster.
- Create a vector **Q** to report the number of clusters at each iteration. Set $Q[1] = 1$ since at iteration 1 there is only one cluster.

3.2.2 Initialization

The initialization phase primarily involves sampling hyperparameters and parameters from their prior distributions. Because the distributions of the hyperparameters depend on the mean and covariance of the data set, these statistics are the first to be computed.

- **avg** holds the computed mean of the given data.
- **Sigma2** is the computed covariance matrix of the data.

Now the hyperparameters may be sampled from their prior distributions.

- $\boldsymbol{\lambda}$ is sampled from $multivariateGaussian(\mathbf{avg}, \mathbf{Sigma2})$.
- $\mathbf{r}$ is sampled from $inverseWishart(p, 2 \times \mathbf{Sigma2^{-1}})$.
- β is sampled from $Gamma(0.5, 0.5)$.
- $\mathbf{W}$ is sampled from $inverseWishart(p, \frac{\mathbf{Sigma2}}{2})$.

Once samples of the hyperparameters are obtained, the parameters of the sole component in iteration 1 may be sampled from their prior distributions:

- 1000 samples of the mean are taken from $multivariateGaussian(\boldsymbol{\lambda}, \mathbf{r}^{-1})$ and stored in a matrix $mmue$.
- 1000 samples of the covariance matrix are taken from $inverseWishart(\beta+p, \mathbf{W})$ and stored in an array $msgima$.

The reason 1000 samples of the parameters are taken is to compute the average probability of each observation given the sampled parameters and hyperparameters. The vector **matm** reports the average probability of each observation given the parameters and hyperparameters. Observations which belong to the same cluster have similar probabilities of belonging to the only existing component. This information is now used to cluster the data in the following way:

- The probability of each observation is divided by the minimum probability (i.e., each value in **matm** is divided by $min(\mathbf{matm})$) and stored in a vector **faz**. Thus, the value $\mathbf{faz}[i]$ may be thought of as the factor by which the probability of observation i is greater than the minimum probability.

- Observations whose **faz** values are similar will be clustered together.

In each iteration, once classifications of the observations have been updated, there are two matters which need to be addressed. First, the numbers identifying each existing cluster must be sequenced as $1, 2, \ldots, G$. This is addressed in the following way:

- A vector **s** is created to list the numbers identifying each cluster in order from lowest to highest, i.e., if at a given iteration there exist three clusters $4, 3, 1$, then $\mathbf{s} = (1, 3, 4)$.

- **s2** is created to list a sequence of numbers from 1 to G.

- For each observation i and each cluster g, if $C[i, it+1] = s[g]$ then $C[i, it+1] < -s2[g]$. In this way, the existing clusters are numbered in the appropriate sequence.

The second matter which needs to be addressed in each iteration is the size of the clusters. It is necessary to ensure no cluster is too small. There are two reasons why the size of the clusters must be controlled. First, if any cluster has only 1 observation a measure of variance cannot be computed. Second, sampling from the posterior distribution of β (which is done further along in the algorithm) is hindered by error messages if covariance matrices are of small clusters. To avoid these issues, the following procedure takes place at each iteration after observations have been re-classified:

- A vector **jud** is created to report the number of observations in each newly formed cluster.

- For cluster i, if $\mathbf{jud}[i] \leq (0.03 \times T)$, the observations belonging to cluster i will be moved to the cluster with the most number of observations. That is, each existing cluster will hold more than 3% of the total number of observations.

Please note the minimum percent of observations in each cluster is subjectively determined. 3% was used when implementing the algorithm to the five data sets presented in this paper. However the percentage can easily be changed depending on the size of the data set of interest. The more observations contained in the data set, the lower the percentage needs to be.

The initialization phase is now complete and the algorithm is prepared to enter the second iteration.

3.2.3 Updates

The iterations which follow the initialization phase will all conduct the procedure to be explained in this section. Recall at the end of the initialization phase the observations were split to form clusters. The second iteration begins by computing the mean and covariance matrices of each existing cluster as these statistics will be used to sample hyperparameters and consequently parameters from their posterior distributions.

- The rows of the matrix *meannew* report the computed mean of each existing cluster.

- The matrices in the array d report the computed covariance matrices of the existing clusters.

Once *meannew* and d have been developed, the hyperparameters may be sampled. β is the first parameter to be sampled. Because β does not have a conjugate prior distribution, sampling is done via importance sampling.

- 1000 samples of β are taken from the prior distribution of β: Gamma(0.5, 0.5) and sorted in order from lowest to highest.
- Next, $p(d \mid \beta[k])$, that is the likelihood of the computed covariance matrices given each sampled β, is computed and stored in a vector **weight**. This vector then gets normalized, i.e., divided by the sum of **weight**. In this way, **weight** reports an approximate posterior probability density for each sampled β.
- A second vector **sumweight** is created to report an approximate posterior cumulative probability for each sampled β.
- 1000 samples, $\mathbf{u}$ are now taken from $uniform[0, 1]$.
- If $\mathbf{sumweight}_{i-1} \leq u < \mathbf{sumweight}_i$, then the β which corresponds to $\mathbf{sumweight}_{i-1}$ is taken to be a sample of β from the posterior distribution.

In this way, 1000 samples of β are taken from its posterior distribution. The average of these 1000 samples will eventually be used to sample from the posterior distribution of the component covariance matrices. The remaining hyperparameters and parameters have conjugate prior distributions and so sampling is much less complicated. 1000 samples from the posterior distributions of each, $\mathbf{W}$, $\mathbf{r}$ and $\boldsymbol{\lambda}$ are taken and averaged. The average of the samples of the hyperparameters are now applied to sample from the posterior distributions of the component mean and covariance matrices. At this

point in the algorithm, for each existing component, the covariance matrix and mean ($\boldsymbol{\Sigma}_g$ and $\boldsymbol{\mu}_g$) have been sampled from their posterior distributions. The next step will be to update the classifications of observations as described below:

- If there is more than one existing cluster: the algorithm simply assigns observations to clusters to which they have the highest probability of belonging.
- If there is only one existing cluster: the aim of the algorithm is to create multiple clusters so clustering will proceed as it did during the first iteration.

New clusters are created if a) the second of the two above cases occurs, and if b) the maximum probability of an observation belonging to any of the existing groups is less than a pre-determined value. However, recall that new clusters will only carry on to the next iteration if they obtain more than 3% of the data. Clusters with less than 3% of the data will be absolved before the next iteration begins. This is simply a measure taken to ensure the program may compute future covariance matrices and marks the end of a given iteration.

3.3 Procedure with PGMMs

3.3.1 CCU

Recall that the implementation of PGMMs means the covariance matrix is decomposed in terms of the loading matrix $\boldsymbol{\Lambda}$ and noise matrix $\boldsymbol{\Psi}$. In the case of the CCU covariance matrix structure, factor weights and the noise matrix are held constant across groups. Thus the initialization phase does not sample the hyper-

parameters $\mathbf{W}$ and β and consequently does not sample the covariance matrices of components from an inverse Wishart distribution. Instead, to obtain a prior sample of 1000 covariance matrices, the algorithm takes 1000 samples of $\mathbf{\Lambda}$, the factor weights matrix and 1000 samples of $\mathbf{\Psi}$ matrices from their prior distributions. It then computes 1000 covariance matrices based on the sampled factor weights and noise matrices. As before, the algorithm now continues by sampling the hyperparameters $\mathbf{r}$ and $\boldsymbol{\lambda}$ to subsequently take 1000 samples of $\boldsymbol{\mu}$. The probabilities of the observations are computed given the computed covariance structures and sampled $\boldsymbol{\mu}$ vectors. Splitting of the data is carried out in the same fashion described above. The next major change in the algorithm is in sampling parameters from their posterior distributions. As the structure of the covariance matrix has changed, there is no need to sample from the posterior distribution of β and $\mathbf{W}$. Instead, the factor weights matrix $\mathbf{\Lambda}$ and noise matrix $\mathbf{\Psi}$ are sampled from their posterior distributions 1000 times to generate 1000 posterior covariance structures. The average of these covariance structures becomes the posterior sampled covariance structure for all groups. The fact that the posterior covariance structure is the same for all components is an important difference to note between the CCU and CUU version of the algorithm. Once the posterior covariance structure is sampled, the algorithm continues as before. The hyperparameters $\mathbf{r}$ and $\boldsymbol{\lambda}$ are sampled from their posterior distributions to allow sampling of $\boldsymbol{\mu}_g$.

3.3.2 CUU

The final version of this algorithm adopts a CUU covariance structure. The initialization of this algorithm is no different than the initialization of the algorithm with the CCU structure. What is different in this version is that the noise matrix, $\mathbf{\Psi}$, is unique for each existing cluster. Thus 1000 samples of $\mathbf{\Psi}$ are taken from its posterior distribution for each existing cluster. In this way, each component obtains a unique covariance matrix. The algorithm is the same as that of the CCU version in all other aspects.

Chapter 4

Results

This chapter will describe the performance of the three versions of the algorithm: non-PGMM, PGMM CUU, and PGMM CCU, on simulated and real data. The following are simulated data: Simulated Data 1 with three clusters, Simulated Data 2 with 4 clusters, and Simulated Data 3 with two clusters in the shape of a figure-8. The real data sets are the Wisconsin Breast Cancer Data (Faraway, 2009; Bennett and Mangasarian, Bennett and Mangasarian) and the Alon data (Alon et al., 1999; McNicholas and Murphy, 2010).

- For all runs of the algorithm, if the maximum probability of belonging to any existing group is less than 10^{-20}, a new cluster is created.
- For runs of the algorithm which implement PGMMs $v = 1$ and $s^2 = 1$.

4.1 Simulated Data 1

Simulated Data 1 is a five dimensional data set of 600 observations from 3 different clusters. This section will discuss, the performance of the algorithm when PGMMs have not been applied, then the performance of the algorithm when PGMMs have been applied. Note, each cluster is derived from a multivariate Gaussian

distribution and below are the parameters of each of the three distributions:

Table 4.1: This table displayes the mean parameter of the components from which the three clusters were derived.

Cluster	Mean
1	$0, 0, 5, 4, 6$
2	$100, 100, 25, 34, 57$
3	$50, 50, 80, 670, 880$

The covariance matrix for all three clusters is:

$$\begin{pmatrix} 1.0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1.0 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1.0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1.0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 1.0 \end{pmatrix}$$

4.1.1 Simulated Data 1: Version without PGMMs

Perfect classification was obtained for Simulated Data 1 when PGMMs were not implemented. The BIC of this model is 1180.0.

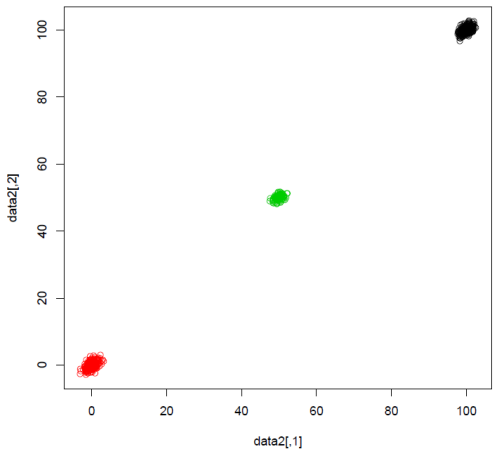

Figure 4.1: The figure above displays the three clusters in Simulated Data 1. The different colours indicate the different clusters.

4.1.2 Simulated Data 1: PGMMs

Recall the PGMMs applied are the CCU and CUU. For Simulated Data 1, the number of latent factors tested are $q = 1, 2$. Thus there were four models fitted to the data, all of which except the model CUU $q = 1$ obtained perfect clustering. In regards to the CUU, $q = 1$ model, the algorithm grouped the first and second clusters together and identified the third cluster as the outliers of the data.

Based on the BIC values of all 4 models, the most parsimonious model with the best fit is the one which implements the CCU structure, assumes 1 latent factor ($q = 1$) and discovers $G = 3$.

Table 4.2: The table below displays the BIC values of the best fitting models which implemented PGMMs.

PGMM Structure	q	G	BIC
CUU	2	3	705.1
CCU	2	3	-164.3
CCU	1	3	-687.4

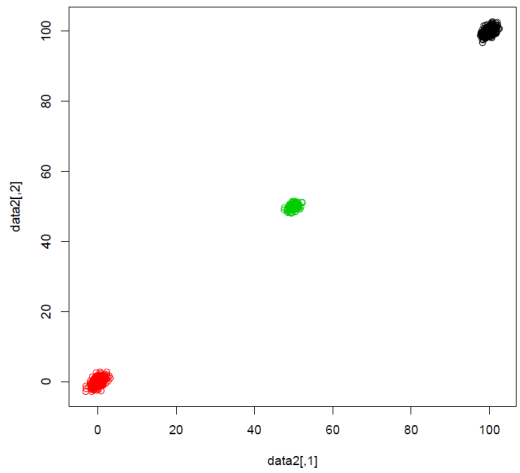

Figure 4.2: The figure above displays the clustering results of Simulated Data 1 when the model CCU $q = 1$ was applied.

4.2 Simulated Data 2

Simulated Data 2 is a 7 dimensional data set of 700 observations from 4 different clusters. As before, this section will discuss, the performance of the algorithm when PGMMs have not been applied, then the performance of the algorithm when PGMMs have been applied. Note, each cluster is derived from a multivariate Gaussian distribution and below are the parameters of each of the four distributions:

Table 4.3: This table displayes the mean parameter of the components from which the four clusters were derived.

Cluster	Mean
1	$0, 0, 5, 4, 6, 40, 40$
2	$10, 10, 25, 34, 57, 100, 100$
3	$50, 50, 80, 67, 88, 0, 0$
4	$150, 150, 200, 200, 195, 300, 300$

The covariance matrix for all four clusters is:

$$\begin{pmatrix} 1.0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1.0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1.0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1.0 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 1.0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 1.0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 1.0 \end{pmatrix}$$

4.2.1 Simulated Data 2: Version without PGMMs

Perfect classification was obtained for Simulated Data 2 when PGMMs were not implemented. The BIC of this model is 3115.4

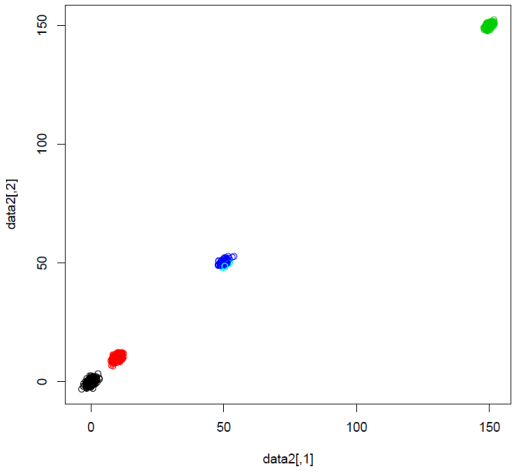

Figure 4.3: The figure above displays the four clusters in Simulated Data 2. The different colours indicate the different clusters.

4.2.2 Simulated Data 2: PGMMs

Recall the PGMMs applied are the CCU and CUU. For Simulated Data 2, the number of latent factors tested are $q = 1, 2, 3$. Thus there were six models fitted to the data. Four of the six models attained perfect clustering. The two models which did not have perfect clustering results were the model which implemented the CCU structure and set $q = 2$ as well as the model which implemented the CUU structure and set $q = 1$. In regards to the CCU, $q = 2$ model, the algorithm grouped clusters 1 and 2 together. And with respect to the CUU, $q = 1$ model, the algorithm classified the fourth cluster as the outliers of the data set. Based on the BIC values of all 6 models, the most parsimonious model with the best fit is the one which implements the CCU structure and assumes 1 latent factor ($q = 1$) and reveals 4 clusters ($G = 4$).

Table 4.4: The table below displays the BIC values of the best fitting models which implemented PGMMs.

PGMM Structure	q	G	BIC
CUU	3	4	118.9
CUU	2	4	-798.6
CCU	3	4	-673.7
CCU	1	4	-870.9

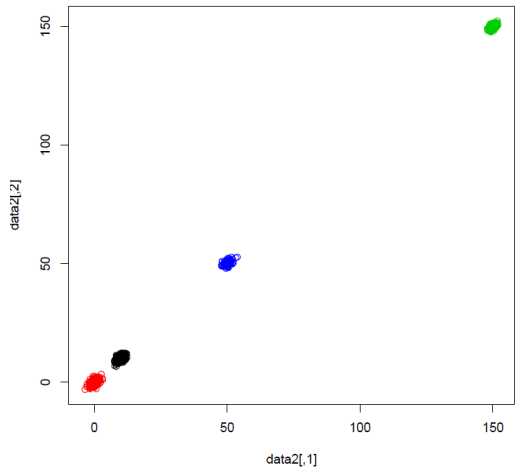

Figure 4.4: The figure above displays the clustering results of Simulated Data 2 when the model CCU $q = 1$ was applied. Note, the different colours indicate the different clusters.

4.3 Simulated Data 3

Simulated Data 3 is a 3 dimensional data set of 500 observations. There are two clusters in the data in form of the figure-8. As before, this section will discuss first, the performance of the algorithm when PGMMs have not been applied, then the performance of the algorithm when PGMMs have been applied. Note, each cluster is derived from a multivariate Gaussian distribution and below are the parameters of each of the two distributions:

Table 4.5: This table displayes the mean parameter of the components from which the two clusters were derived.

Cluster	Mean
1	$-1, -1, -1$
2	$3, 3, 3$

The covariance matrix for both clusters is:

$$\begin{pmatrix} 1.0 & 0.5 & 0.5 \\ 0.5 & 1.0 & 0.5 \\ 0.5 & 0.5 & 1.0 \end{pmatrix}$$

4.3.1 Simulated Data 3: Version without PGMMs

This version of the algorithm correctly determined the number of mixture components, however, the final results misclassified 5 observations and so perfect clustering was not achieved. The BIC of this model is 375.8.

Table 4.6: Below is the final clustering results of Simulated Data 3 when the model applied did not implement PGMMs.

Classification	Cluster 1	Cluster 2
True Cluster 1	1	249
True Cluster2	246	4

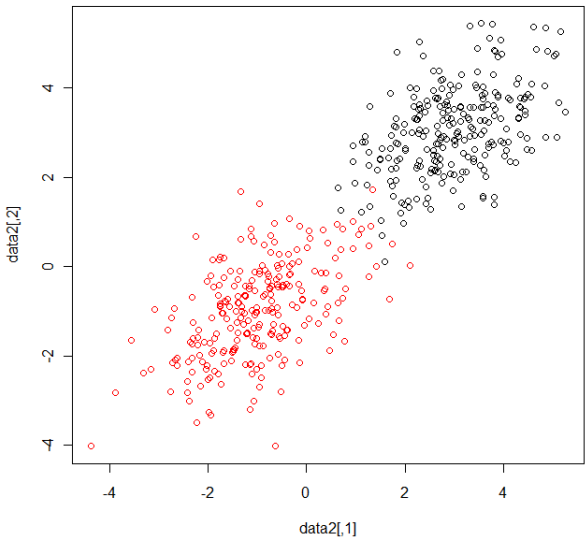

Figure 4.5: The above figure displays the clustering results of Simulated Data 3 when PGMMs were not implemented. Note, the different colours indicate the different clusters.

4.3.2 Simulated Data 3: PGMMs

Recall the PGMMs applied are the CCU and CUU. For Simulated Data 3, the number of latent factors tested are $q = 1, 2$. Thus there were four models fitted to the data, none of which obtained perfect classification. Based on the BIC values of all 4 models, the most parsimonious model with the best fit is the one which implements the CCU structure, assumes 1 latent factor ($q = 1$) and uncovers 2 clusters ($G = 2$).

Table 4.7: The table below displays the BIC values of the best fitting models which implemented PGMMs.

PGMM Structure	q	G	BIC
CUU	2	2	-292.4
CUU	1	2	-293.4
CCU	2	2	-238.0
CCU	1	2	-591.3

Table 4.8: Below is the final clustering results of Simulated Data 3 when the model CCU $q = 1$ was applied to the data.

Classification	Cluster 1	Cluster 2
True Cluster 1	246	4
True Cluster 2	9	241

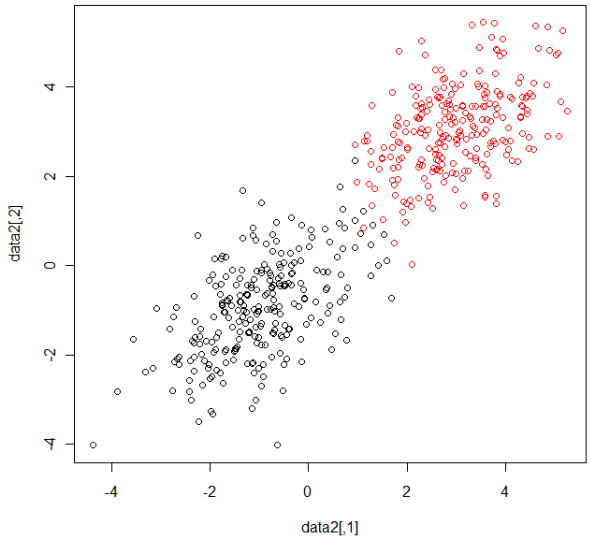

Figure 4.6: The above figure displays the clustering results of Simulated Data 3 when the model CCU $q = 1$ was applied to the data. Note, the different colours indicate the different clusters.

4.4 Wisconsin Breast Cancer Data

The Wisconsin Breast Cancer Data (WBCA) is data on 681 tumors of which 238 are malignant. For each tumor the following 9 variables were measured: marginal adhesion, bare nuclei, bland chromatin, epithelial cell size, mitoses, normal nucleoli, clump thickness, cell shape uniformity and cell size uniformity. The purpose of WBCA was to test the effectiveness of the *fine needle aspiration* procedure. This procedure

takes a sample of the tumor tissue in order to determine its malignancy (Faraway, 2009). Thus the question of interest is the classification of a tumor as malignant or benign. As with the simulated data, this section will discuss first, the performance of the algorithm when PGMMs have not been applied, then the performance of the algorithm when PGMMs have been applied.

4.4.1 WBCA: Version without PGMMs

The algorithm correctly determined the number of mixture components, however, the final results misclassified 63 observations and so perfect clustering was not achieved. The BIC of this model is 8487.2

Table 4.9: Below is the final clustering results of WBCA when the model applied did not implement PGMMs.

Classification	Cluster 1	Cluster 2
Malignant	41	197
Benign	421	22

4.4.2 WBCA: PGMMs

Recall the PGMMs applied are the CCU and CUU. For WBCA, the number of latent factors tested are $q = 1, 2, 3, 4$. Thus there were eight models fitted to the data, none of which obtained perfect classification. Based on the BIC values of all 8 models, the most parsimonious model with the best fit is the one which implements the CCU structure and assumed 3 latent factor ($q = 3$).

Table 4.10: The table below displays the BIC values of the best fitting models which implemented PGMMs.

PGMM Structure	q	G	BIC
CCU	4	2	1098.3
CCU	3	2	937.7

Table 4.11: Below is the final clustering results of WBCA when the model CCU $q = 3$ was implemented.

Classification	Cluster 1	Cluster 2
Malignant	44	194
Benign	427	16

4.5 Alon Data

The Alon data used is a reduced gene expression data from 62 samples of Colon tissue and has 461 measured variables (Alon et al., 1999; McNicholas and Murphy, 2010). This reduced version of the data was sourced from www.paulmcnicholas.info. The original data contained 2000 variables. As described in McNicholas and Murphy (2010), McLachlan et al. (2002) reduced the number of variables in this data by applying the EMMIX-GENE approach. Details on this approach can be found in McNicholas and Murphy (2010). The data set contains 40 tumor tissues and 22 normal tissues, thus the two clusters are tumor and normal.

4.5.1 Alon: Without PGMMs

When PGMMs were not implemented in the algorithm, the procedure identified 3 clusters ($G = 3$). The BIC of this model is 1567.2.

Table 4.12: Below is the final clustering results of Alon when PGMMs were not implemented.

Classification	Cluster 1	Cluster 2	Cluster 3
Tumor	9	19	12
Normal	1	14	7

4.5.2 Alon: PGMMs

The number of latent factors tested for both the CCU and the CUU version of the algorithm were $1, 2, 3$ and 4. None of the 8 models applied obtained perfect clustering results. However, based on the BIC values of the models, the model with the most appropriate fit was the CCU, $q = 1$ version which uncovered 2 cluster.

Table 4.13: The table below displays the BIC values of the best fitting models which implemented PGMMs.

PGMM Structure	q	G	BIC
CUU	4	2	2581.4
CCU	4	2	1799.2
CCU	3	2	2128.8
CCU	2	2	1688.9
CCU	1	2	1421.2

Table 4.14: Below is the final clustering results of Alon when the model CCU $q = 1$ was implemented.

Classification	Cluster 1	Cluster 2
Tumor	26	14
Normal	5	17

At this point, it is important to note that although McNicholas and Murphy (2010) had better clustering results of the Alon data, their best model implemented the equivalent of the CUC covariance matrix structure. Because this study implemented the CCU and the CUU structures, the results presented here cannot be

compared to those presented in McNicholas and Murphy (2010).

Chapter 5

Conclusion

Microarray technology provides data on the expression levels of different genes in a sample of tissue. Clustering methods are applied to microarray data in order to distinguish different types of tissue based on the expression levels of genes. One of the common clustering approaches is model-based clustering. Medvedovic and Sivaganesan (2002) applied a Bayesian mixture model to cluster microarray data which removed the need to specify the number of suspected components. The issue with their model is that it applies a diagonal covariance matrix structure. A diagonal covariance matrix structure does not account for the inter-relatedness of genes and so cannot be deemed appropriate.

The Bayesian mixture model presented in this paper aimed to improve the model presented in Medvedovic and Sivaganesan (2002). The first step taken to improve their approach was to apply a non-diagonal covariance matrix to the Gaussian mixture model. This allows researchers to account for covariance between variables. Based on the clustering results of the five datasets presented in this paper, the non-diagonal covariance structure was fairly successful in clustering data where $n > p$; that is, data where the number of observations is greater than the number of variables measured. However, microarray data is usually high dimensional data such that

$p > n$. For this reason the model had to be revised.

The implementation of PGMMs reduces the number of free parameters to be estimated and thus allow for better handling of high dimensional data. The Alon data represents a common occurrence in microarray data, where $p \gg n$. As the results demonstrated, the implementation of PGMMs produced more viable results than when PGMMs were not implemented, that is, the covariance structure was not decomposed. The algorithm presented here removes the need to specify a suspected number of clusters and by allowing a non-diagonal covariance structure, it accounts for co-variation between genes. The algorithm produces acceptable clustering results without the need to implement PGMMs when $p < n$. However, since the majority of microarray data is high dimensional such that $p \gg n$, the implementation of PGMMs is indeed necessary.

Future work on this algorithm should consider the develpment of a systematic approach in determining the minimum number of observations each cluster should retain.

Bibliography

Alon, U., N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America 96*(12), 6745–6750.

Anderson, D. R. and K. Burnham (2004). Multimodel inference: Understanding aic and bic in model selection. *Sociological Methods Research 33*(2), 261–304.

Bennett, K. P. and O. L. Mangasarian. Advances in optimization and parallel computing: Neural network training via linear programming.

Beyer, K. S., J. Goldstein, R. Ramakrishnan, and U. Shaft (1999). When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory*, pp. 217–235.

Brown, P. and D. Botstein (1999). Exploring the new world of the genome with dna microarrays. *Nature Genetics 21*, 33–37.

Cavanaugh, J. (2009). 171:290 model selection lecture VI: The Bayesian information criterion. University Lecture.

Cowell, R.G., D. P. L. S. and D. J. Spiegelhalter (1999). *Probabilistic Networks and Expert Systems*. New York: Springer.

D'haeseleer, P., S. Liang, and R. Somogyi (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics 16*(8), 707–726.

Faraway, J. (2009). Faraway: Functions and datasets for books by julian faraway. http://www.maths.bath.ac.uk/ jjf23/. R package version 1.0.4.

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

Haque, E., H. Liu, and L. Parsons (2004). Subspace clustering for high dimensional data: A review. *Sigkdd Explorations 6*(1), 90–105.

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

Mark, S., D. Shalon, R. Davis, and P. Brown (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science, New Series 270*, 467–470.

McLachlan, G. J. and K. E. Basford (1988). *Mixture Models: Inference and applications to clustering*. New York: Marcel Dekker Inc.

McLachlan, G. J., R. W. Bean, and D. Peel (2002). A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics 18*(3), 412–422.

McNicholas, P. D. (2010). Model-based classification using latent Gaussian mixture models. *Journal of Statistical Planning and Inference 140*(5), 1175–1181.

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

McNicholas, P. D. and T. B. Murphy (2010). Model-based clustering of microarray expression data via latent gaussian mixture models. *Bioinformatics 26*(21), 2705–2712.

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

Medvedovic, M. and S. Sivaganesan (2002). Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics 18*(9), 1194–1206.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics 6*, 31–38.

Seltman, H. (2012). Experimental design and analysis (15.7.1): Penalized likelihood methods for model selection. University Lecture.

Steinbach, M., L. Ertz, and V. Kumar (2003). The challenges of clustering high-dimensional data. In *In New Vistas in Statistical Physics: Applications in Econophysics, Bioinformatics, and Pattern Recognition*. Springer-Verlag.

Symons, M. (1981). Clustering criteria and multivariate normal mixtures. *Biometrics 37*, 35–43.

Wolfe, J. H. (1963). Object cluster analysis of social areas. Master's thesis, University of California, Berkley.

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