Topics in One-Way Supervised Biclustering
Using Gaussian Mixture Models
TOPICS IN ONE-WAY SUPERVISED BICLUSTERING
USING GAUSSIAN MIXTURE MODELS

BY
MONICA H.T. WONG, M.Sc.

A THESIS
SUBMITTED TO THE DEPARTMENT OF MATHEMATICS & STATISTICS
AND THE SCHOOL OF GRADUATE STUDIES
OF MCMASTER UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

© Copyright by Monica H.T. Wong, February 2017
All Rights Reserved
TITLE: Topics in One-Way Supervised Biclustering Using Gaussian Mixture Models

AUTHOR: Monica H.T. Wong, M.Sc., B.Sc.

SUPERVISORS: Dr. Paul D. McNicholas
Dr. David M. Mutch

NUMBER OF PAGES: xiii, 99
For my late grandma, who always encouraged my academic studies.
Abstract

Cluster analysis identifies homogeneous groups that are relevant within a population. In model-based clustering, group membership is estimated using a parametric finite mixture model, commonly the mathematically tractable Gaussian mixture model. One-way clustering methods can be restrictive in cases where there are suspected relationships between the variables in each component, leading to the idea of biclustering, which refers to clustering both observations and variables simultaneously. When the relationships between the variables are known, biclustering becomes one-way supervised. To this end, this thesis focuses on a novel one-way supervised biclustering family based on the Gaussian mixture model. In cases where biclustering may be overestimating the number of components in the data, a model averaging technique utilizing Occam’s window is applied to produce better clustering results. Automatic outlier detection is introduced into the biclustering family using mixtures of contaminated Gaussian mixture models. Algorithms for model-fitting and parameter estimation are presented for the techniques described in this thesis, and simulation and real data studies are used to assess their performance.
Acknowledgements

First and foremost, I would like to extend my heartfelt thanks to Dr. Paul McNicholas and Dr. David Mutch (University of Guelph) for providing me the opportunity to do my graduate studies under your joint supervision. Your constant support, guidance, patience, and encouragement aided immensely in my graduate journey, and your incredible achievements and knowledge are always great sources of inspiration and motivation for me. Thank you so much for always believing in me.

Thank you to Dr. Narayanaswamy Balakrishnan and Dr. Rob Deardon (University of Guelph/University of Calgary) for serving as members of my supervisory committees. Your insights were instrumental in the progress of my research. Thank you to Dr. Sunny Wang (Wilfred Laurier) for examining my thesis and Dr. Karl Freeman for chairing my thesis defence.

To the McNicholas and Mutch research groups past and present, thank you for balancing research and fun in the office and lab. Having enjoyable colleagues to interact with such as yourselves made me look forward to working in the lab and office every day.

To my Hamilton, Guelph, and Toronto friends, I cannot imagine how I would have completed this journey without your regular invitations (even if they were declined almost as regularly) to step away from my research and spend some time relaxing...
and socializing. Thank you for the great memories.

My most sincerest gratitude goes out to my family. Your unconditional love and support have guided me to where I am today, and I am forever grateful. Thank you for always emphasizing the importance of education and being the prime examples of hard workers. I send a special thank you to my late grandma, who was always my biggest academic supporter. I wish I could share this achievement with you in person, but I know you are watching over me and continue to support me from afar.
Publications

The following publications are based on the work presented in this thesis and have been published, submitted, or are in preparation for submission for publication:


Contents

Abstract iv

Acknowledgements v

Publications vii

1 Introduction 1

1.1 Bicluster Analysis ................................................. 1

1.1.1 Clustering .......................................................... 1

1.1.2 Biclustering ....................................................... 2

1.1.3 One-Way Supervised Model-Based Biclustering ............... 3

1.2 Thesis Outline ...................................................... 4

1.2.1 Chapter 2 .......................................................... 4

1.2.2 Chapter 3 .......................................................... 4

1.2.3 Chapter 4 .......................................................... 5

1.2.4 Chapter 5 .......................................................... 5

1.2.5 Chapter 6 .......................................................... 5

1.3 Contribution to the Literature .................................... 6
# Background

2.1 Finite Mixture Models ........................................... 8
2.2 Mixtures of Factor Analyzers and Extensions .................. 9
  2.2.1 Parsimonious Gaussian Mixture Models ..................... 10
2.3 Model-Based Biclustering ....................................... 11
2.4 The EM Algorithm and Extensions .............................. 13
  2.4.1 The EM Algorithm ............................................ 13
  2.4.2 The AECEM Algorithm ....................................... 13
  2.4.3 Convergence .................................................. 14
  2.4.4 Woodbury Identity ........................................... 15
2.5 Model Selection and Performance Assessment .................. 15
  2.5.1 The Bayesian Information Criterion ....................... 15
  2.5.2 The Adjusted Rand Index ................................. 16

# Two-Way Learning with One-Way Supervision for Gene Expression Data

3.1 Introduction .................................................... 18
3.2 Methods ........................................................ 20
  3.2.1 Covariance Structure ....................................... 20
  3.2.2 Parameter Estimation ...................................... 21
  3.2.3 Component Membership .................................... 26
3.3 Results ........................................................ 27
  3.3.1 Simulation Studies .......................................... 27
3.4 Application .................................................... 29
  3.4.1 Rat Data ..................................................... 29
5.3 Results ................................................................. 67
  5.3.1 Simulation Studies ............................................. 68
  5.3.2 Real Data ....................................................... 69
5.4 Discussion .......................................................... 73

6 Conclusions ......................................................... 76
  6.1 Summary ........................................................... 76
  6.2 Future Work ....................................................... 77
    6.2.1 Benchmark Biclustering Data Sets ......................... 77
    6.2.2 OSGaBi Performance ........................................ 78
    6.2.3 Model Selection ............................................. 78
    6.2.4 Model Averaging ............................................. 78
    6.2.5 Contaminated Data .......................................... 79

A AECEM Algorithm Details for the OSGaBi Family .......... 80
  A.1 Model UCU ....................................................... 80
  A.2 Model UUC ....................................................... 81
  A.3 Model UCC ....................................................... 82
  A.4 Model CUU ....................................................... 83
  A.5 Model CCU ....................................................... 84
  A.6 Model CUC ....................................................... 85
  A.7 Model CCC ....................................................... 86
List of Figures

3.1 Heatmaps of simulated data for CUC model. Examples of heatmaps of the four types of simulated data tested for the CUC model: low variance and good cluster separation (A), medium variance and good cluster separation (B), high variance and good cluster separation (C), and high variance with relatively close clusters (D). 38

3.2 Heatmaps of the rat data. (A) Heatmap of the rat data before biclustering. The red, yellow, and blue bars along the columns and represent liver, blood, and red TA samples, respectively. (B) Heatmap of the rat data after biclustering and subsequently rearranging the rows so that the observation clusters were contiguous. Black and grey bars along the columns and rows simply represent the presence of the clusters and do not indicate relationships between them. $G = 19$ for the observation clusters and along the columns, $q_1 = q_2 = 2$. 40
3.3 Heatmaps of the human data. (A) Heatmap of the human data before biclustering. The red, yellow, and blue bars along the columns and represent immune tissues, whole blood, and brain tissues respectively. (B) Heatmap of the human data after biclustering and subsequently rearranging the rows so that the observation clusters were contiguous. Black and grey bars along the columns and rows simply represent the presence of the clusters and do not indicate relationships between them. $G = 10$ for the observation clusters and along the columns, $q_1 = q_2 = 2$. 41
Chapter 1

Introduction

1.1 Bicluster Analysis

1.1.1 Clustering

Cluster analysis identifies underlying groups within data, whereby observations within a group are as similar to each other as possible, and observations across groups are as dissimilar as possible. It is an unsupervised technique because there is no knowledge of the underlying group structure \textit{a priori}, which reflects common real-life scenarios because observations are not usually accompanied by hints about their true groupings with respect to the variables. Some common clustering techniques include metric-based methods such as hierarchical clustering (Ward, 1963) and $k$-means clustering (MacQueen, 1967; Hartigan and Wong, 1979), and model-based methods that assume the data are generated from a finite mixture model (seminal work traced back to Wolfe, 1965; Edwards and Cavalli-Sforza, 1965). Metric-based methods remain in frequent use because they are easy to implement, but a major
drawback is their sensitivity to the distance metric selected (e.g., in gene expression analysis, Jaskowiak et al., 2014). Alternatively, model-based methods take one or more components in a mixture model as corresponding to a cluster (cf. McNicholas, 2016a). An advantage of model-based methods includes their systematic approach to selecting the parameterization of the model and the number of clusters (Fraley and Raftery, 1998).

1.1.2 Biclustering

One-way clustering methods previously described can be restrictive in certain applications. It is not always the case that the groups of patterns found in the observations are homogeneous across all the variables; rather, it could be the case that only a subset of the variables possesses these groupings. Consequently, biclustering techniques have been developed to address this recurring issue. Biclustering, first explored by Hartigan (1972), clusters both rows and columns simultaneously and results in biclusters, each of which consists of a subset of the variables and a subset of the observations. Ever since the seminal biclustering paper by Hartigan (1972), many biclustering techniques have emerged that tackle the problem in various ways and within a variety of applications. Examples of popular biclustering methods include plaid models (Lazzeroni and Owen, 2000), spectral biclustering (Kluger et al., 2003), and coupled two-way clustering (Getz et al., 2000).

One common way to classify biclustering methods is by the structure of biclusters that are allowed. In this classification scheme, there are two properties that define the structure of a bicluster: exhaustivity and exclusivity. Exhaustivity is the property that every observation (row) and/or variable (column) must belong to at least one
bicluster. Exclusivity is the property that a given observation and/or variable can only belong to at most one bicluster, such that if the method is non-exclusive, there is no overlap allowed among the biclusters. Spectral biclustering (Kluger et al., 2003) and the biclustering method by Hartigan (1972) is both row and column exhaustive and non-exclusive, whereas plaid models and coupled two-way clustering are both non-exhaustive and non-exclusive. The structural properties of biclusters is dependent on the application; therefore, it is important for researchers to confirm that the interpretation of the biclustering results makes sense when considering the structures of the biclusters.

Biclustering algorithms can also be classified by whether or not they utilize an evaluation measure for guiding the search for an optimal solution to the biclustering problem, i.e., metric- or non metric-based algorithms. Out of the above mentioned methods, only Hartigan’s method is a metric-based one. Although evaluation measures are flexible because they make little or no assumptions of the data, methods that utilize these types of algorithms run the risk of reaching a local optimal solution in the search space rather than a global optimal solution. Consequently, non metric-based algorithms may be better suited for high-dimensional data (Pontes et al., 2015).

1.1.3 One-Way Supervised Model-Based Biclustering

Biclustering is an unsupervised technique similar to its clustering counterpart because information regarding the underlying variable group structures and observation group structures are unknown a priori. The aforementioned biclustering methods are designed with this lack of information under consideration. Conversely, researchers may desire that the observations within biclusters satisfy a particular relationship
among the variables; the biclustering method would then be one-way supervised. Consider gene expression data: if the samples are the variables and the researcher knows in advance that certain samples (should) have relationships with each other (e.g., samples from the same treatment groups, taken from the same tissue type, etc.), this would be useful information for the algorithm to have \textit{a priori} if the researcher was interested in treatment- or tissue-specific clusters of genes.

1.2 Thesis Outline

1.2.1 Chapter 2

Background information is provided, which includes details on finite mixture models and factor analysis in model-based clustering. Model-based biclustering is discussed and techniques introduced. Details regarding the EM algorithm and variants thereof is provided in addition to model selection and performance assessment methods.

1.2.2 Chapter 3

A family of eight parsimonious Gaussian mixture models for the biclustering of gene expression data is proposed. Biclustering is accommodated by adopting a mixture of factor analyzers model with a binary, row-stochastic factor loadings matrix. Prior knowledge of the factor loadings matrix is useful in this application and is reflected in the one-way supervised nature of the algorithm. The family of models is demonstrated using both simulated and real data, including a previously unpublished microarray data set.
1.2.3 Chapter 4

A model averaging approach is introduced for the biclustering family previously developed in Chapter 3. In our approach, we consider cases when other models possess similar model selection criterion values to the best model, defined by Occam’s window. In cases where the models in Occam’s window do not have the same number of components, merging of the mixture components based on the adjusted Rand index is performed before model averaging. The model averaging approaches are validated using both simulated and real data.

1.2.4 Chapter 5

A family of eight parsimonious contaminated Gaussian mixture models for one-way supervised biclustering is introduced. It allows for automatic detection of outliers and minimizes the impact of said outliers on the component means. The family of models is demonstrated using both simulated and real data, and its performance is also compared with existing methods.

1.2.5 Chapter 6

The work presented in this thesis is summarized and suggestions for future direction are discussed.
1.3 Contribution to the Literature

The literature is currently lacking biclustering algorithms that handle scenarios such as the one described in Section 1.1.3. Additionally, although there exists numerous non-metric-based biclustering algorithms, the literature has not popularized parsimonious model-based biclustering families equivalent to those commonly found in the clustering paradigm. Broadly, the eight member family of McNicholas and Murphy (2008), cf. Section 2.2, is extended to the biclustering paradigm. This biclustering family has advantages not afforded by clustering and/or non metric-based biclustering algorithms, as mentioned previously. This family is subsequently extended to allow model averaging (Chapter 4) and to accommodate outliers (Chapter 5). The work in this thesis aims to pave the way for further research in the area of one-way supervision in model-based biclustering.

More specifically, the impact of the ideas proposed in this thesis on the current model-based biclustering literature can be summarized into the following points:

- A family of parsimonious Gaussian mixture models is introduced. It extends the mixture of factor analyzers model into the biclustering paradigm and utilizes one-way supervision to aid in the application of biomarker discovery using gene expression data. Application results indicate the effectiveness of this biclustering approach.

- A model averaging approach for biclustering is presented. It utilizes Occam’s window to determine the models to be averaged and merges mixture components if the models in the window do not have the same number of components. Simulated and real data indicate better biclustering results after applying this
approach to them.

- A family of parsimonious contaminated Gaussian mixture models is proposed that automatically detects outliers and minimizes the impact of these outliers on the component means. The family is demonstrated using simulated and real data and perform favourably compared to other available methods.
Chapter 2

Background

2.1 Finite Mixture Models

In model-based clustering, a random vector \( X \) arises from a parametric finite mixture distribution, whose density can be written as

\[
f(x \mid \vartheta) = \sum_{g=1}^{G} \pi_g h(x \mid \theta_g),
\]

where \( \pi_g \in (0, 1] \), such that \( \sum_{g=1}^{G} \pi_g = 1 \), is the mixing proportion for component \( g \), \( h(x \mid \theta_g) \) is the density of a multivariate random variable \( X \) with parameters \( \theta_g \), and \( \vartheta = (\pi_1, \ldots, \pi_G, \theta_1, \ldots, \theta_G) \). Commonly, the finite Gaussian mixture model is used in model-based clustering because of its mathematical tractability. This density is given by

\[
f(x \mid \vartheta) = \sum_{g=1}^{G} \pi_g \phi(x \mid \mu_g, \Sigma_g),
\]
where
\[ \phi(x \mid \mu_g, \Sigma_g) = \frac{1}{\sqrt{(2\pi)^d | \Sigma_g|}} \exp \left\{ -\frac{1}{2} (x - \mu_g) \Sigma_g^{-1} (x - \mu_g) \right\} \]
is the density of a multivariate Gaussian random variable $X$ with mean $\mu_g$ and covariance matrix $\Sigma_g$. An overview of finite mixture models is provided in McLachlan and Peel (2000a).

### 2.2 Mixtures of Factor Analyzers and Extensions

Consider a data matrix with observations as rows and variables as columns, where $n$ is the number of observations and $d$ is the number of variables. A $d$-dimensional random variable $X$ following a $G$-component Gaussian mixture model has
\[ G \frac{d(d+1)}{2} \]
free parameters from the covariance matrices. Thus, introducing parsimony into this structure would be beneficial. One method is via eigen-decomposition of the component covariance matrices as detailed in (Banfield and Raftery, 1993) and imposing constraints on the elements of the eigen-decomposed covariance structure to generate a family of Gaussian parsimonious clustering models (GPCM). Because this covariance structure is not the focus of this thesis, the reader is referred to the detailed discussion on this family of Gaussian mixture models provided in Celeux and Govaert (1995).

The factor analysis model (Spearman, 1904) assumes that the $d$-dimensional random vector $X_i$ can be modelled using a $q$-dimensional vector of latent factors $U_i$, where $q < d$. Factor analysis allows for a reduction in the number of parameters.
being estimated via a restriction in the covariance matrix, which is useful in high-dimensional data cases. The model can be written as

\[ X_i = \mu + \Lambda U_i + \epsilon_i, \]

where \( \Lambda \) is a \( d \times q \) matrix of factor loadings, the latent factors \( U_i \sim N(0, I_q) \) are independent, and the errors \( \epsilon_i \sim N(0, \Psi) \), where \( \Psi \) is a \( d \times d \) diagonal noise matrix, are independently distributed and independent of the \( U_i \). Thus, \( X_i \sim N(\mu, \Lambda \Lambda' + \Psi) \).

In the mixture of factor analyzers (MFA) model, developed by Ghahramani and Hinton (1997), different factor analysis models are allowed in different regions of the feature subspace, and the density is that of a Gaussian mixture model with component covariance structure \( \Sigma_g = \Lambda_g \Lambda_g' + \Psi_g \). Various other models have since been developed which extend the MFA model and further constrain the component covariance parameters. McLachlan and Peel (2000b) use a more general component covariance structure, \( \Sigma_g = \Lambda_g \Lambda_g' + \Psi_g \) and the mixtures of probabilistic principal component analyzers (MPPCA) developed by Tipping and Bishop (1999) assumes that the noise matrix is isotropic so that \( \Psi = \psi I_d \). In the next section, one more extension of the MFA model is detailed, namely the parsimonious Gaussian mixture models (PGMM).

### 2.2.1 Parsimonious Gaussian Mixture Models

McNicholas and Murphy (2008) developed the parsimonious Gaussian mixture model (PGMM) family of models to further extend the general MFA model with covariance \( \Sigma_g = \Lambda_g \Lambda_g' + \Psi_g \). It allows for various combinations of the constraints \( \Lambda_g = \Lambda, \Psi_g = \Psi, \) and the isotropic constraint \( \Psi_g = \psi I_d \) within the mixture of
factor analyzers model. This results in a family of eight parsimonious models. In Chapter 3, the PGMM family is extended for biclustering high-dimensional data.

2.3 Model-Based Biclustering

A recent review of biclustering on expression data by Pontes et al. (2015) classifies biclustering methods using various taxonomies. One taxonomy is based on bicluster structure, specifically whether or not the observations/variables must be assigned to a bicluster (exhaustivity) and whether or not the observations and/or variables can be assigned to multiple biclusters (exclusivity). When considering applications such as blood biomarker discovery discussed in Chapter 3, an implicit property of the biomarker is that its expression profile is highly correlated between the blood and tissue of interest, yet distinct from the rest of the tissues, indicating a unique biomarker for that tissue. Thus, the researcher would be interested in samples that are assigned to one bicluster only, in other words, non-overlapping column-exclusive biclusters. Examples of existing biclustering methods that adopt this property are plaid models developed by Lazzeroni and Owen (2000), biclustering via Gibbs sampling developed by Sheng et al. (2003), and Bayesian biclustering developed by Gu and Liu (2008). Briefly, plaid models incorporate additive two-way ANOVA models within the biclusters (Lazzeroni and Owen, 2000), biclustering via Gibbs sampling incorporates a frequency model for the bicluster patterns and Gibbs sampling for parameter estimation (Sheng et al., 2003), and Bayesian biclustering incorporates properties of both the plaid models and Gibbs sampling (Sheng et al., 2003). These are also examples of non metric-based probabilistic biclustering methods, based on another taxonomy provided in the review. They are non metric-based because they
do not utilize an evaluation measure and they are probabilistic because they use statistical analyses and probability theory to describe the data. The reader is referred to the review paper by Pontes et al. (2015) for a structured and detailed discussion on the available biclustering methods.

Under the probabilistic framework, Martella et al. (2008) propose a modified MFA technique for high-dimensional data for the simultaneous clustering of observations and variables. In this technique, variable cluster membership is represented by a binary row-stochastic matrix,

$$\Lambda_g = \{\lambda_{gjl}\} = \begin{cases} 1 & \text{if } Q(\cdot, \lambda_{gjl} = 1) = \max_h \{ Q(\cdot, \lambda_{gjh} = 1) \}, \\ 0 & \text{otherwise}, \end{cases}$$

where \(j = 1, \ldots, d, h, l = 1, \ldots, q, g = 1, \ldots, G\), and \(Q\) is the expected complete-data log-likelihood. We have \(X_i = \mu_g + \Lambda_g U_{ig} + \epsilon_{ig}\) with probability \(\pi_g\). In this case, \(U_{ig} \sim N(0, I_q)\) and \(X_i \sim N(\mu_g, \Lambda_g \Lambda_g' + \Psi_g)\), and the covariance structure is \(\Sigma_g = \Lambda_g \Lambda_g' + \Psi_g\). This particular form of factor loadings matrix results in a block-diagonal covariance matrix which is especially suitable in the biclustering framework because it models the grouped nature of the variables. Additionally, it results in the non-overlapping biclusters that are useful in applications such as blood biomarker discovery. Combinations of constraining or not constraining the elements of the covariance matrix across components results in a family of four models. This family will be referred to as MFABC (MFA for biclustering) from this point forward. The binary row-stochastic factor loadings matrix will be revisited in Chapter 3.

The equivalent terms “group” and “component” will frequently be encountered throughout this thesis. Because the topics are within the biclustering framework,
there may be ambiguity to the reader as to whether the term is relating to the observations or the variables. For the purpose of this thesis, “group” and “component” are to be taken to refer to observations unless otherwise specified.

2.4 The EM Algorithm and Extensions

2.4.1 The EM Algorithm

The expectation-maximization (EM) algorithm (Dempster et al., 1977) is an iterative procedure for computing the maximum likelihood estimates (MLE) when data are incomplete or treated as such. It is commonly used for fitting mixture models and estimating the parameters in model-based clustering. In clustering, the incomplete data consists of the unobserved group memberships and sometimes other latent variables. The EM algorithm is based on the complete-data, which consist of both observed and missing data. The algorithm begins with the expectation step (E-step), where the expected value of the complete-data log-likelihood ($Q$) is computed conditional on the current parameter estimates. In the maximization step (M-step), $Q$ is maximized with respect to the model parameters. These two steps are repeated until convergence. In mixture models, the algorithm is executed for each component. A detailed description of the EM algorithm is provided in Krishnan and McLachlan (1997).

2.4.2 The AECM Algorithm

The alternating expectation-conditional maximization (AECM) algorithm (Meng and van Dyk, 1997) is an extension of the EM algorithm that incorporates a series of
conditional maximization (CM) steps instead of a single M-step and also allows for
different specification of the complete-data in each cycle. Because CM steps replace
the M-step and thus potentially result in simpler calculations, this variant could be
computationally more efficient than the EM algorithm. The AECM algorithm is also
described in detail in Krishnan and McLachlan (1997).

2.4.3 Convergence

Convergence of the EM algorithm and its extensions is determined using the
Aitken’s acceleration (Aitken, 1926) to estimate the asymptotic maximum of the
log-likelihood at each iteration of the EM algorithm for a specific number of com-
ponents and a specific number of factors, as described in Krishnan and McLachlan
(1997). The Aitken’s acceleration at iteration \( t \) is

\[
a^{(t)} = \frac{l^{(t+1)} - l^{(t)}}{l^{(t)} - l^{(t-1)}},
\]

where \( l \) corresponds to the respective log-likelihood. The asymptotic estimate of the
log-likelihood at iteration \( (t + 1) \) is

\[
l^{(t+1)} = l^{(t)} + \frac{1}{1 - a^{(t)}}(l^{(t+1)} - l^{(t)})
\]

(Bohning et al., 1994). The stopping criterion is \( l^{(t+1)} - l^{(t)} < \epsilon \) (McNicholas et al.,
2010) provided that this difference is positive, where \( \epsilon = 0.1 \) is used in this thesis
(following McNicholas et al., 2010).
2.4.4 Woodbury Identity

When running the AECM algorithm, utilizing the Woodbury identity (Woodbury, 1950) avoids inverting any non-diagonal \( d \times d \) matrices that may be singular for \( d \gg n \). Suppose an \( n \times n \) matrix \( A \), an \( n \times q \) matrix \( H \), a \( q \times q \) matrix \( C \), and a \( q \times n \) matrix \( V \). The Woodbury identity states that

\[
(A + HCV)^{-1} = A^{-1} - A^{-1}H(C^{-1} + VA^{-1}H)^{-1}VA^{-1}.
\] (2.1)

Specifically for the AECM algorithm, setting \( H = \Lambda \), \( V = \Lambda' \), \( A = \Psi \), and \( C = I_q \) results in

\[
(\Psi + \Lambda\Lambda')^{-1} = \Psi^{-1} - \Psi^{-1}\Lambda(I_q + \Lambda'\Psi^{-1}\Lambda)^{-1}\Lambda'\Psi^{-1}.
\] (2.2)

Now, instead of inverting the \( d \times d \) covariance matrix on the left side of Equation 2.2, only the diagonal and \( q \times q \) matrices on the right side need to be inverted. With data where \( q \ll d \), this identity provides a major computational advantage. Another useful identity is for calculating the determinant of the covariance matrix in the AECM algorithm:

\[
|\Psi + \Lambda\Lambda'| = \frac{|\Psi|}{|I_q - \Lambda'(\Lambda\Lambda' + \Psi)^{-1}\Lambda'|}.
\]

2.5 Model Selection and Performance Assessment

2.5.1 The Bayesian Information Criterion

The Bayesian information criterion (BIC; Schwarz, 1978), which is an approximation to Bayes factors (Dasgupta and Raftery, 1998; Kass and Raftery, 1995), is
commonly used to select the best model in model-based clustering. Given a model with parameter vector $\theta$,

$$BIC = 2l(\hat{\theta}) - m \log n,$$

where $l(\hat{\theta})$ is the maximized log-likelihood, $\hat{\theta}$ is the MLE of $\theta$, $m$ is the number of free parameters in the model, and $n$ is the number of observations. Here, the model with the maximum BIC is chosen as the best model for the data. Previous studies have provided evidence that the BIC performs well as a model selection criterion for mixture models (Fraley and Raftery, 1999, 2002; McNicholas et al., 2010).

### 2.5.2 The Adjusted Rand Index

Throughout this thesis, biclustering performance is determined by utilizing data sets where the underlying groups are known \textit{a priori}. Comparing the known class labels to the estimated group memberships allows for assessing an algorithm’s performance. Although the true class labels are known, majority of the analyses are conducted as true biclustering problems; therefore, the labels are not used to aid in the analyses, unless otherwise noted.

The Rand index (RI; Rand, 1971) is a partitioning measure that cross-tabulates the true class labels with the estimated group memberships. Specifically, the RI is calculated by

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

where a pairwise agreement occurs when two observations belonging to the same cluster are assigned to the same group or when two observations belonging to different clusters are assigned to different groups. The RI can take on any value between 0
and 1, where 1 indicates perfect agreement. The limitation of the RI is that its expected value is greater than 1 and therefore can be difficult to interpret.

In light of this, Hubert and Arabie (1985) introduced the adjusted Rand index (ARI), which is the Rand index adjusted for chance and takes the form

\[
ARI = \frac{RI - \text{expected index}}{\text{maximum index} - \text{expected index}},
\]

where the expected index is equivalent to the expected value of the RI under random classification. The expected value of the ARI under random classification is zero. ARI > 0 is better agreement than would be expected by chance, ARI < 0 is worse agreement than would be expected by chance, and ARI = 1 indicates perfect agreement.
Chapter 3

Two-Way Learning with One-Way Supervision for Gene Expression Data

3.1 Introduction

In this chapter, a parsimonious family for one-way supervised biclustering using Gaussian mixture models is developed. Section 3.2 thoroughly details the properties of the family and how the parameters are estimated. Section 3.3 and 3.4 present results from the simulation studies and real data, respectively. Finally, Section 3.5 ends the chapter with a discussion of the work.

With the introduction of personalized medicine, the discovery of novel biomarkers via “omics” research plays a critical role in its advancement (Offit, 2011). A biomarker is defined as “a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacologic...
responses to a therapeutic intervention” (Ghosh and Poisson, 2009). The behaviour of a biomarker is expected to vary among individuals, thereby allowing treatment to be “personalized” depending on that individual’s (predicted) response. The ideal diagnostic tool is minimally invasive, leading researchers to investigate the use of peripheral blood cells as surrogate biopsy material, since blood is more easily accessible. The assumption is that the molecular profile of peripheral blood reflects a global overview of the physiological events occurring in different tissues throughout the body (Mohr and Liew, 2007).

When gene expression microarrays are used for biomarker discovery, the subset of identified genes acts as the set of biomarkers. Returning to the idea of peripheral blood as surrogate material, a gene that exhibits correlated expression profiles in blood and a given tissue may be a biomarker of interest. One popular way of identifying these subsets of correlated genes across the blood and the given tissue is via biclustering techniques (Ben-Dor et al., 2004).

As discussed in Chapter 1, biclustering is a useful technique when the researcher suspects biclusters of variables and observations in the data, but possesses no information about what properties of the variables define the biclusters. Thus it is an unsupervised technique similar to its clustering counterpart. The biclustering methods mentioned in Chapter 2 are designed with this lack of information under consideration. Conversely, researchers may desire that the observations within biclusters satisfy a particular relationship among the variables; the biclustering method would then become one-way supervised. This technique is particularly relevant for the blood biomarker discovery application mentioned earlier.
3.2 Methods

3.2.1 Covariance Structure

To accommodate biclustering we set the factor loadings matrix to be binary row-stochastic. To allow for supervision along the variable dimension, we provide the structure of this matrix to the algorithm. In our gene expression analysis case, the variables are the samples, thus we are setting a relationship between the samples in the data set and providing it to the algorithm during initialization and take it as constant. The constraints $\Lambda_g = \Lambda$, $\Psi_g = \Psi$, and $\Psi_g = \psi_g I_d$ create a family of eight one-way supervised Gaussian mixture models for biclustering (Table 3.1), which will be referred to as OSGaBi (one-way supervised Gaussian biclustering) hereafter.

Table 3.1: Properties of the OSGaBi family. The nomenclature, covariance structure, and number of covariance parameters for each member of the OSGaBi family. C, constrained; U, unconstrained.

<table>
<thead>
<tr>
<th>Model nomenclature</th>
<th>Covariance structure ($\Sigma_g$)</th>
<th>Covariance parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Lambda_g = \Lambda$</td>
<td>$\Psi_g = \Psi$</td>
<td>$\Psi_g = \psi_g I_d$</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>$\Lambda\Lambda' + \psi I_d$</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>$\Lambda\Lambda' + \Psi$</td>
</tr>
<tr>
<td>C</td>
<td>U</td>
<td>$\Lambda\Lambda' + \psi_g I_d$</td>
</tr>
<tr>
<td>C</td>
<td>U</td>
<td>$\Lambda\Lambda' + \Psi_g$</td>
</tr>
<tr>
<td>U</td>
<td>C</td>
<td>$\Lambda_g \Lambda_g' + \psi I_d$</td>
</tr>
<tr>
<td>U</td>
<td>C</td>
<td>$\Lambda_g \Lambda_g' + \Psi$</td>
</tr>
<tr>
<td>U</td>
<td>U</td>
<td>$\Lambda_g \Lambda_g' + \Psi_g I_d$</td>
</tr>
<tr>
<td>U</td>
<td>U</td>
<td>$\Lambda_g \Lambda_g' + \Psi_g$</td>
</tr>
</tbody>
</table>
3.2.2 Parameter Estimation

This section provides the mathematical details required to compute the parameter estimates for the eight members of the OSGaBi family. For convenience, the following notation is adopted. We denote the observed data as \( x \) and the unobserved latent parameters as \( U_{ig} \). We denote the missing group memberships as \( z \), where

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

for \( i = 1, \ldots, n, \ g = 1, \ldots, G \).

The AECM algorithm is used for parameter estimation. In the first cycle of the AECM algorithm, \((x_i, z_i)\) are the complete-data, where \( i = 1, \ldots, n \). During the CM-step, \( \pi_g \) and \( \mu_g \) are updated. During the E-step, the \( z_{ig} \) are replaced by their expected values

\[
E[z_{ig} \mid \hat{\pi}_g, \hat{\mu}_g, \hat{\Lambda}_g, \hat{\Psi}_g] = \frac{\hat{\pi}_g \phi(x_i \mid \hat{\mu}_g, \hat{\Lambda}_g, \hat{\Psi}_g)}{\sum_{h=1}^{G} \hat{\pi}_h \phi(x_i \mid \hat{\mu}_h, \hat{\Lambda}_h, \hat{\Psi}_h)} =: \hat{z}_{ig}
\]

leading to the calculation of the expected value of the complete-data log-likelihood, \( Q_1 \). In the CM-step, \( Q_1 \) is maximized to give

\[
\hat{\pi}_g = \frac{n_g}{n}
\]

and

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

where \( n_g = \sum_{i=1}^{n} \hat{z}_{ig} \).
In the second cycle of the AECM algorithm, \((x_i, z_i, U_{ig})\) are the complete-data for \(i = 1, \ldots, n, g = 1, \ldots, G\). Here, we focus on the CUU model first because it is the most probable case in real-life scenarios. During this CM-step, \(\Psi_g\) is updated. During this E-step, \(z_{ig}\) are replaced by \(\hat{z}_{ig}\) and \(U_{ig}\) and \(U_{ig}'\) are replaced by

\[
\mathbb{E}[U_{ig} | x_i, \hat{\mu}_g, \hat{\Lambda}, \hat{\Psi}_g] = \beta_g \sum_{i=1}^{n} \hat{z}_{ig}(x_i - \mu_g)
\]

\[
\mathbb{E}[U_{ig}' | x_i, \hat{\mu}_g, \hat{\Lambda}, \hat{\Psi}_g] = \mathbf{I}_q - \beta_g \Lambda + \beta_g \sum_{i=1}^{n} \hat{z}_{ig}(x_i - \mu_g)(x_i - \mu_g)'\beta_g',
\]

respectively, where \(\hat{\beta}_g = \Lambda' (\Lambda \Lambda' + \Psi_g)^{-1}\), to allow for the calculation of \(Q_2\). In the CM-step, the maximization of \(Q_2\) is specific for each model. Considering the CUU model,

\[
Q_2(\Lambda, \Psi_g) = C + \sum_{g=1}^{G} \frac{n_g}{2} \left[ \log |\Psi_g^{-1}| - \text{tr} \left\{ \Psi_g^{-1} S_g \right\} + 2\text{tr} \left\{ \Psi_g^{-1} \Lambda \beta_g S_g \beta_g' \right\} - \text{tr} \left\{ \Psi_g^{-1} \Theta_g \Lambda' \right\} \right],
\]

where \(C\) is a constant with respect to the unknown parameters,

\[
S_g = \frac{1}{n_g} \sum_{i=1}^{n} \hat{z}_{ig}(x_i - \mu_g)(x_i - \mu_g)'
\]

\[
\Theta_g = \mathbf{I}_q - \beta_g \Lambda + \beta_g S_g \beta_g'.
\]

The following score function is obtained when differentiating \(Q_2\) with respect
to \( \Psi_g \):

\[
S(\Lambda, \Psi_g) = \frac{\delta Q}{\delta \Psi_g} = \sum_{g=1}^{G} \frac{n_g}{2} \left[ \Psi_g - S_g + 2\Lambda \hat{\beta}_g S_g - \Lambda \Theta_g \Lambda' \right].
\]

Now, setting \( S(\Lambda, \Psi_g) = 0 \) and solving it gives the estimate

\[
\hat{\Psi}_g = \text{diag} \left\{ S_g - 2\Lambda \hat{\beta}_g S_g + \Lambda \Theta_g \Lambda' \right\}.
\]

Similarly for the UUU model,

\[
Q_2(\Lambda_g, \Psi_g) = C + \sum_{g=1}^{G} \frac{n_g}{2} \left[ \log |\Psi_g^{-1}| - \text{tr} \left\{ \Psi_g^{-1} S_g \right\} + 2 \text{tr} \left\{ \Psi_g^{-1} \Lambda_g \hat{\beta}_g S_g \right\}
- \text{tr} \left\{ \Psi_g^{-1} \Lambda_g \Theta_g \Lambda'_g \right\} \right],
\]

where \( \hat{\beta}_g = \Lambda'_g (\Lambda_g \Lambda'_g + \Psi_g)^{-1} \). The new estimate for this model is

\[
\hat{\Psi}_g^{\text{new}} = \text{diag} \left\{ S_g - \Lambda_g \hat{\beta}_g S_g \right\}.
\]

For the UCU model, where \( \Psi_g = \Psi \),

\[
Q_2(\Lambda_g, \Psi) = C + \sum_{g=1}^{G} \frac{n_g}{2} \left[ \log |\Psi^{-1}| - \text{tr} \left\{ \Psi^{-1} S_g \right\} + 2 \text{tr} \left\{ \Psi^{-1} \Lambda_g \hat{\beta}_g S_g \right\}
- \text{tr} \left\{ \Psi^{-1} \Lambda_g \Theta_g \Lambda'_g \right\} \right],
\]

23
where $\hat{\beta}_g = \Lambda_g' (\Lambda_g \Lambda_g' + \Psi)^{-1}$. The new estimate for this model is

$$
\hat{\Psi}_{\text{new}} = \sum_{g=1}^{G} \hat{\pi}_g \text{diag}\left\{ S_g - \Lambda_g \hat{\beta}_g S_g \right\}.
$$

For the UUC model, where $\Psi_g = \psi_g I_p$,

$$
Q_2(\Lambda_g, \psi_g) = C + \sum_{g=1}^{G} \frac{n_g}{2} \left[ d \log (\psi_g^{-1}) - \psi_g^{-1} \text{tr} \{ S_g \} + 2\psi_g^{-1} \text{tr} \left\{ \Lambda_g \hat{\beta}_g S_g \right\} \right. \\
- \left. \psi_g^{-1} \text{tr} \left\{ \Lambda_g \Theta_g \Lambda_g' \right\} \right],
$$

where $\hat{\beta}_g = \Lambda_g' (\Lambda_g \Lambda_g' + \psi_g I_d)^{-1}$. The new estimate for this model is

$$
\hat{\psi}_{g\text{new}} = \frac{1}{d} \text{tr} \left\{ S_g - \Lambda_g \hat{\beta}_g S_g \right\}.
$$

For the UCC model, $\Psi = \psi I_d$, and

$$
Q_2(\Lambda_g, \psi) = C + \sum_{g=1}^{G} \frac{n_g}{2} \left[ d \log (\psi^{-1}) - \psi^{-1} \text{tr} \{ S_g \} + 2\psi^{-1} \text{tr} \left\{ \Lambda_g \hat{\beta}_g S_g \right\} \right. \\
- \left. \psi^{-1} \text{tr} \left\{ \Lambda_g \Theta_g \Lambda_g' \right\} \right],
$$

where $\hat{\beta}_g = \Lambda_g' (\Lambda_g \Lambda_g' + \psi I_d)^{-1}$. The new estimate for this model is

$$
\hat{\psi}_{\text{new}} = \frac{1}{d} \sum_{g=1}^{G} \hat{\pi}_g \text{tr} \left\{ S_g - \Lambda_g \hat{\beta}_g S_g \right\}.
$$

The second half of the family constrains $\Lambda_g = \Lambda$. For the CCU model, $\Psi_g = \Psi$.
and

\[ Q_2(\Lambda, \Psi) = C + \sum_{g=1}^{G} \frac{n_g}{2} \left[ d \log |\Psi^{-1}| - \text{tr} \left\{ \Psi^{-1} S \right\} + 2 \text{tr} \left\{ \Psi^{-1} \hat{\Lambda} \hat{\beta} S \right\} - \text{tr} \left\{ \Psi^{-1} \Lambda \Theta \Lambda' \right\} \right] , \]

where \( \hat{\beta} = \Lambda'(\Lambda \Lambda' + \Psi)^{-1} \), \( S = \sum_{g=1}^{G} \hat{\pi}_g S_g \), and \( \Theta = I_q - \hat{\beta} \Lambda + \hat{\beta} S \hat{\beta} \). The new estimate for this model is

\[ \hat{\Psi}^\text{new} = \text{diag} \left\{ S - \Lambda \hat{\beta} S \right\} . \]

For the CUC model, \( \Psi_g = \psi_g I_d \), and

\[ Q_2(\Lambda, \psi_g) = C + \sum_{g=1}^{G} \frac{n_g}{2} \left[ d \log (\psi_g^{-1}) - \psi_g^{-1} \text{tr} \{ S_g \} + 2 \psi_g^{-1} \text{tr} \left\{ \Lambda \hat{\beta}_g S_g \right\} - \psi_g^{-1} \text{tr} \left\{ \Lambda \Theta_g \Lambda' \right\} \right] , \]

where \( \hat{\beta}_g = \Lambda'(\Lambda \Lambda' + \psi_g I_d)^{-1} \). The new estimate for this model is

\[ \hat{\psi}_g^\text{new} = \frac{1}{d} \text{tr} \left\{ S_g - 2 \Lambda \hat{\beta}_g S_g + \Lambda \Theta_g \Lambda' \right\} . \]

Finally, with the CCC model, where \( \Psi = \psi I_d \), and

\[ Q_2(\Lambda, \psi) = C + \sum_{g=1}^{G} \frac{n_g}{2} \left[ d \log (\psi^{-1}) - \psi^{-1} \text{tr} \{ S \} + 2 \psi^{-1} \text{tr} \left\{ \Lambda \hat{\beta} S \right\} - \psi^{-1} \text{tr} \left\{ \Lambda \Theta \Lambda' \right\} \right] , \]
where \( \hat{\beta} = \Lambda'(\Lambda\Lambda' + \psi I_d)^{-1} \). The new estimate for this model is

\[
\hat{\psi}^{\text{new}} = \frac{1}{d} \text{tr}\{S - \Lambda\hat{\beta}S\}.
\]

### 3.2.3 Component Membership

The predicted biclustering for each member of the OSGaBi family is given by the maximum \textit{a posteriori} (MAP) classification for the observations and the classifications originally provided for the variables. That is, the posterior predicted component membership of observation (i.e., gene) \( i \) is the value of \( g \) for which \( \hat{z}_{ig} \) is greatest. In the biological sense, this will identify which gene belongs to which subset, implying that the genes in each subset are related in some way. Component membership of variable (i.e., sample) \( j \) is already provided as \( \Lambda_g \) at the beginning of the algorithm, specifically

\[
\Lambda_g = \{\lambda_{gjl}\} = \begin{cases} 
1 & \text{if variable } j \text{ belongs to cluster } l, \\
0 & \text{otherwise},
\end{cases}
\]

for \( j = 1, \ldots, d, \ l = 1, \ldots, q, \) and \( g = 1, \ldots, G. \) In the biological view, we know \textit{a priori} that a certain set of samples are/should be related to each other, which is uncorrelated to another set of samples. A concrete example of how component membership is applied in microarray gene expression analysis is presented in Section 3.4.
3.3 Results

3.3.1 Simulation Studies

Simulation studies were carried out to validate the proposed biclustering algorithm. The ARI was used to evaluate the performance of the algorithm in recovering biclusters from the simulated data. Specifically, $z_i$ was compared with $\hat{z}_i$ after convergence was attained. Model selection was done via the BIC as previously described, although it can be noted that the integrated completed likelihood (ICL; Biernacki et al., 2000) and Akaike information criterion (AIC; Akaike, 1973) were used as comparison and produced the same outcomes.

Simulated data were generated with $G = 2, 3, \text{and } 4$ clusters for observations and $q = 2$ clusters for variables. This resulted in 4, 6, and 8 biclusters, respectively. Four cases were tested: low, medium, and high variance coupled with good cluster separation, and high variance coupled with relatively close clusters. For each case, 100 data sets were generated, where each set had $d = 8$ variables and $n = 200$ observations and was randomly generated from the same Gaussian distribution. Examples of heatmaps for each of the CUC cases visibly indicate that there are distinct biclusters in the simulated data (Figure 3.1). To reflect the one-way supervised nature of the algorithm, the true $\Lambda$ was provided. Twenty random starts were used for each run of the algorithm. Table 3.2 presents the results from these four simulation studies for the CUU and CUC models. It lists the average number of components selected, the most frequently chosen model, and the average ARI when fitting $G = 2, \ldots, 10$ observation clusters. Because the algorithm was sometimes overfitting for the number of components based on the model it chose, another analysis was included to show the
average ARI when the number of clusters was known (i.e., $G = 2, 3, 4$, depending on the case). These results are shown in the last column. The CUU and CUC models are focused on because they are the most probable cases in real scenarios considered later, and, additionally, they are the models most frequently selected when the number of clusters was known (results not shown).

For completeness, simulation studies were conducted on the remaining six OSGaBi models using simulated data with medium variance and good cluster separation and with the same properties as that used for the CUU and CUC models. As the true $\Lambda_g$ or $\Lambda$ was provided, it implied that for models with unconstrained $\Lambda$ (i.e., models UUU, UUC, UCU, and UCC), $G$ was known because the $\Lambda$ for each component would have been provided. Table 3.3 presents the average ARI, most frequently chosen model, and average number of clusters selected from this simulation study for each of the remaining six models when fitting $G = 2, \ldots, 10$ clusters for observations. Because the algorithm was once again sometimes overfitting for the number of components based on the model selected, this final analysis was included to show the algorithm’s performance when the number of clusters was fixed to $G = g_{\text{known}}$, where $g_{\text{known}}$ represents the number of observation clusters the data was generated from. The last column of the table presents the corresponding average ARI when fixing $G$ for the CCU and CCC models.

It is important to note that although the algorithm was sometimes overfitting for the number of components based on the model selected, the majority of the time the original components were simply being broken into smaller components. Table 3.4, a classification table from one of the simulation results, illustrates the very common occurrence.
In this specific result, Cluster 1 was broken up into three components by the algorithm, resulting in a total of four components. The final column of Tables 3.2 and 3.3 provide further evidence because once the algorithm is provided the correct number of components, the ARI indicates perfect or near perfect agreement.

3.4 Application

In this section, we present results from biclustering analyses on two different microarray data sets. Pre-processing for both data sets is done using the affy and oligo packages (Gautier et al., 2004; Carvalho and Irizarry, 2010), respectively, for R Bioconductor (R Core Team, 2016; Gentleman et al., 2004). To determine the set of differentially expressed genes for each analysis, the limma package (Smyth, 2004) is used. More specifically, statistically significant differentially expressed genes are determined using a moderated $F$ test that is based on a linear model fitted to the data. A detailed description of this linear model procedure designed for microarrays is provided in Smyth (2004).

3.4.1 Rat Data

We present the biclustering results from previously unpublished Affymetrix oligonucleotide array data. This study consisted of five male lean control rats and five male Zucker diabetic fatty (ZDF) rats, which are genetically predisposed to developing diabetes. Details regarding the original rat study are described in Beaudoin et al. (2013). From each animal, tissue was extracted from various tissue depots, including the liver and red tibialis anterior (red TA, a type of muscle). Blood was
also extracted, resulting in a total of 30 samples. RNA was extracted from these samples and used for the subsequent microarray gene expression analysis. After preprocessing, \( n = 19540 \) genes remained. We worked with the top 2000 differentially expressed genes between the red TA and liver \( (p < 0.01) \). For this analysis, we set the genes as the observations \( (n = 2000) \) and the samples as the variables \( (d = 30) \).

The goal of the biclustering analysis was to identify sets of genes within the blood that possess similar expression profiles within the distinct tissues. Thus, we aimed to match biclusters containing genes that had similar expression profiles that were unique for blood and a specific tissue type. We focus here on genes with similar expression profiles between blood and liver. Downstream, these candidate genes can be tested to determine if they can function as blood biomarkers of metabolic status in individuals in different contexts (i.e., response to interventions, different disease states, etc); however, this subsequent analysis goes beyond the scope of this thesis.

We constrained the structure of \( \Lambda_g \) because we knew the relationships required among the three sample types. Specifically, we wanted correlated expression between blood and liver only, implying that expression between blood and red TA were uncorrelated and expression between liver and red TA were uncorrelated as well. The other (extraneous) relationship characterized by the block-diagonal covariance matrix was the correlated nature of the expression strictly among the liver samples. Consequently, \( q = 2 \) for the number of variable clusters (i.e., the two relationships described previously). Sample types were constant across all components, i.e., \( \Lambda_g = \Lambda \), and thus we limited the algorithm to fit model CUU and CUC. These two \( \Lambda \)-constrained models were chosen based on the results from the simulation studies previously mentioned. We normalized the data and fitted the range of \( G = 2, \ldots, 30 \).
The BIC selected a CUU model with $G = 19$ observation clusters for the blood-liver analysis. As seen from the heatmaps before and after biclustering and subsequent rearranging, there are definitive biclusters in the data (Figure 3.2). We inputted the gene lists for each of the 19 biclusters into the online functional annotation tool DAVID (Database for Annotation, Visualization and Integrated Discovery; Huang et al., 2008, 2009) to elucidate potential biological processes that are dominant in each bicluster. DAVID functional annotation results indicated that the largest proportions of genes in the blood-liver biclusters have roles in protein metabolic and modification processes, the carboxylic metabolic process, the oxaloacid metabolic process, and intracellular signal transduction (biological processes as defined by the Gene Ontology Consortium, 2015), all biological processes of which have previously been shown to have an involvement in diabetes and obesity, and some processes within the liver (Issad et al., 2010; Saltiel and Kahn, 2001; Nawrocki et al., 2006; Laffel, 1999; Yeaman, 1989; Virkamäki et al., 1999; Taniguchi et al., 2006). These processes account for approximately 20–43% of the genes in the various biclusters and are all statistically significantly enriched ($p < 0.05$). There is also a general inference that insulin resistance occurs at different times in insulin sensitive tissues such as muscle and liver (Kraegen et al., 1991; Samuel et al., 2010); therefore, it is not surprising that the expression profiles between the liver and red TA are not similar. Additionally, it has been previously established that the peripheral blood transcriptome reflects changes in various tissues throughout the body (Mohr and Liew, 2007), a property that is illustrated in the biclusters of interest.
3.4.2 Human Data

The second data set we analyzed is another Affymetrix oligonucleotide array retrieved from the Gene Expression Omnibus (GEO; Edgar et al., 2002), accession number GSE1133. The original study aimed to profile 79 human and 61 mouse tissues in terms of their transcriptomes under normal conditions (Su et al., 2004). Here, we focus on the human arrays, specifically the tissues related to the immune system (20 tissue types) and the brain (16 tissue types), and also whole blood, for a total of 37 tissue types. Each tissue had two replicates, giving a total of 74 samples. After pre-processing and removing genes without Entrez gene identifiers, \( n = 3867 \) genes remained, of which 2148 genes were differentially expressed between brain and immune tissues \((p < 0.01)\). Similar to the rat data, we set the genes as the observations \((n = 2148)\) and the samples as the variables \((d = 74)\).

The goal of this biclustering analysis was to identify sets of genes within the blood that possess similar expression within the distinct groups of tissues. Thus, we aimed to match biclusters containing genes that had similar expression that were unique for blood and a specific group of tissues. We focus here on genes with similar expression between blood and immune tissues. Subsequent work can involve determining which of these candidate genes can function as blood biomarkers of normal immune function in individuals.

Similar to the rat data, we constrained the structure of \( \Lambda_g \) because we knew the relationships required among the three sample types. Specifically, we wanted correlated expression only between blood and immune tissues. This implied that expression between blood and brain tissues were uncorrelated, and expression between
immune and brain tissues were uncorrelated as well. The other (extraneous) relationship characterized by the block-diagonal covariance matrix was the correlated nature of the expression strictly among the immune tissue samples. Consequently, $q = 2$ for the number of variable clusters (i.e., the two relationships described previously). Samples were constant across all components, i.e., $\Lambda_g = \Lambda$, and thus we again limited the algorithm to fit model CUU and CUC. We normalized the data and fitted the range of $G = 2, \ldots, 30$.

The BIC selected a CUU model with $G = 10$ observation clusters for the blood-immune analysis. As seen from the heatmaps before and after biclustering and subsequent rearranging, there are again definitive biclusters in the data (Figure 3.3). DAVID functional annotation results indicate that the largest portion of genes in each bicluster have roles in the nucleobase-containing small molecule metabolic process, macromolecule metabolic process, microtubule-based process, microtubule cytoskeletal organization, response to DNA damage stimulus, and transmembrane transport; all biological processes that have been linked to immune responses (Ishii and Akira, 2008; Neefjes et al., 2011; Paludan and Bowie, 2013; Parcej and Tampe, 2010). These processes account for anywhere between 4 and 51% of the genes in the various biclusters, and are all statistically significantly enriched ($p < 0.05$). Furthermore, blood acts as a transporter for the immune system by transporting immune cells throughout the body, thus blood can provide an extensive view of the immune status of an individual (Chaussabel et al., 2010). This property is reflected in the biclusters of interest because there is a correlation among the expression between the blood and the immune tissues.
3.5 Discussion

One specific taxonomy of biclustering methods for gene expression data aims to retrieve non-overlapping biclusters characterized by one specific sample type (in this case, “sample” could refer to a type of treatment, tissue, disease state, etc.) along the variable dimension, as reviewed in Pontes et al. (2015). This is useful in the completely unsupervised case where the labels of the samples are unknown and the goal of biclustering is to find sets of genes with expression profiles unique to a specific sample type. Alternatively, when identifying novel tissue-specific blood biomarkers, knowledge of the tissue type allows for setting \textit{a priori} the relationship desired between the various samples (i.e., variables) in the data set. Consequently, one-way supervision (i.e., supervised along the variable dimension) during biclustering enables these discoveries. To the best of our knowledge, biclustering methods currently available under the taxonomy of non-overlapping biclusters do not provide the option of one-way supervision along the variable dimension to aid in tissue-specific applications such as blood biomarker discovery.

Another advantage of approaching tissue-specific blood biomarker discovery through the use of biclustering is the ability to identify groups of genes that are potentially related to each other through their biological pathways. Commonly, correlation analysis between blood and a tissue is conducted using the available gene list in its entirety, e.g. Sullivan et al. (2006), consequently not revealing any information about genes related by biological pathways that a cluster analysis would provide. In our OSGaBi family, setting the labels for the variable clusters and subsequently biclustering conditional on this information allows us to handle this limitation of correlation analysis.
Simulation study results show that the BIC (and other popular model selection criteria) is sometimes leading to the overfitting of the simulated data because the ARI indicates perfect or near perfect agreement once the number of observation clusters is treated as fixed. While the BIC has been shown to be unreliable in higher dimensions (e.g., Chen and Chen, 2008) — and this may suggest that further research on an optimal model selection criteria for this family of biclustering models is warranted — it is quite possible that the selection of larger values of $G$ is simply a result of lack of concentration around the modes at higher variances. The inclusion of results for fixed $G$ follows McNicholas and Murphy (2010) and McLachlan et al. (2002), who carried out mixture model analysis of gene expression data by treating $G$ as fixed and known. Note that, in Martella et al. (2008) where the binary row-stochastic factor loadings matrix is a property of their MFABC family, the authors report simulation results but do not mention the model selection criterion or the range of number of observation clusters fitted; therefore, it is not known if the authors treated $G$ as fixed. Conversely, the authors mention the use of the BIC and AIC for model selection in their real data study with gene expression data, supporting the use of the BIC for our analyses until the optimal model selection criteria is determined.

We have presented biclustering results using the OSGaBi family on two real microarray gene expression data sets. The first one was a previously unpublished rat microarray gene expression data set, where identified biclusters corresponded to genes whose expression profiles were correlated between liver and blood (and not between red TA and blood, or liver and red TA). Identified biclusters were enriched in genes related to biological processes known to play a role in insulin resistance and obesity in a tissue-specific manner. The second data set was a subset of a microarray
gene expression data set from the GEO database that aimed to profile the human transcriptome under normal conditions. In this analysis, identified biclusters corresponded to genes whose expression correlated between immune tissues and blood (and not between brain tissues and blood, or immune and brain tissues). Identified biclusters contained genes related to biological processes previously associated with the immune system. Although further biological experimental analysis and interpretation need to be conducted to determine the best candidate gene(s) in both preliminary analyses, the initial results show promise in using the OSGaBi biclustering family for discovering novel blood biomarkers to act as surrogate tissue material in the maintenance of health and the prevention of disease.
Table 3.2: Simulation study results for model CUC and CUU. Values in brackets represent the respective standard deviation. \( \text{var} = \) variance; \( \text{sep} = \) separation.

<table>
<thead>
<tr>
<th>Case</th>
<th>( G = 2, \ldots, 10 )</th>
<th>( G = g_{\text{known}} )</th>
<th>Average ( G )</th>
<th>Most chosen model</th>
<th>Average ARI</th>
<th>Average ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUC, ( g_{\text{known}} = 2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low var, good cluster sep</td>
<td>2.5 (0.8)</td>
<td>CUU</td>
<td>0.955 (0.083)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid var, good cluster sep</td>
<td>2.4 (0.8)</td>
<td>CUU</td>
<td>0.955 (0.094)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, good cluster sep</td>
<td>4.0 (1.0)</td>
<td>CUC</td>
<td>0.708 (0.106)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, close clusters</td>
<td>6.4 (1.2)</td>
<td>CUC</td>
<td>0.502 (0.135)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUU, ( g_{\text{known}} = 2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low var, good cluster sep</td>
<td>2.4 (0.7)</td>
<td>CUU</td>
<td>0.969 (0.072)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid var, good cluster sep</td>
<td>2.5 (0.8)</td>
<td>CUU</td>
<td>0.964 (0.071)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, good cluster sep</td>
<td>4.0 (1.0)</td>
<td>CUC</td>
<td>0.705 (0.103)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, close clusters</td>
<td>6.4 (1.3)</td>
<td>CUC</td>
<td>0.485 (0.138)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUC, ( g_{\text{known}} = 3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low var, good cluster sep</td>
<td>3.5 (0.7)</td>
<td>CUU</td>
<td>0.981 (0.039)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid var, good cluster sep</td>
<td>3.4 (0.7)</td>
<td>CUU</td>
<td>0.984 (0.033)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, good cluster sep</td>
<td>5.1 (1.0)</td>
<td>CUC</td>
<td>0.864 (0.081)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, close clusters</td>
<td>8.8 (1.1)</td>
<td>CCC</td>
<td>0.601 (0.066)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUU, ( g_{\text{known}} = 3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low var, good cluster sep</td>
<td>3.5 (0.6)</td>
<td>CUU</td>
<td>0.984 (0.028)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid var, good cluster sep</td>
<td>3.4 (0.7)</td>
<td>CUU</td>
<td>0.975 (0.050)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, good cluster sep</td>
<td>5.0 (1.1)</td>
<td>CUC</td>
<td>0.866 (0.079)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, close clusters</td>
<td>8.8 (1.0)</td>
<td>CUC</td>
<td>0.590 (0.070)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUC, ( g_{\text{known}} = 4 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low var, good cluster sep</td>
<td>4.4 (0.7)</td>
<td>CUU</td>
<td>0.989 (0.254)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid var, good cluster sep</td>
<td>4.3 (0.5)</td>
<td>CUU</td>
<td>0.992 (0.020)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, good cluster sep</td>
<td>6.2 (1.0)</td>
<td>CUC</td>
<td>0.887 (0.048)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, close clusters</td>
<td>9.7 (0.5)</td>
<td>CUC</td>
<td>0.658 (0.045)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUU, ( g_{\text{known}} = 4 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low var, good cluster sep</td>
<td>4.4 (0.9)</td>
<td>CUU</td>
<td>0.989 (0.031)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid var, good cluster sep</td>
<td>4.4 (0.7)</td>
<td>CUU</td>
<td>0.989 (0.024)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, good cluster sep</td>
<td>4.6 (0.8)</td>
<td>CUU</td>
<td>0.970 (0.048)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High var, close clusters</td>
<td>9.8 (0.5)</td>
<td>CUC</td>
<td>0.653 (0.046)</td>
<td>1.0 (0.0)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 3.1: Heatmaps of simulated data for CUC model. Examples of heatmaps of the four types of simulated data tested for the CUC model: low variance and good cluster separation (A), medium variance and good cluster separation (B), high variance and good cluster separation (C), and high variance with relatively close clusters (D).
Table 3.3: Simulation study results for the other six models. Values in brackets represent the respective standard deviation. var = variance; sep = separation.

<table>
<thead>
<tr>
<th>Model</th>
<th>$G = 2, \ldots, 10$</th>
<th>$G = g_{\text{known}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average $G$</td>
<td>Most chosen model</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$g_{\text{known}} = 2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UUU</td>
<td>2</td>
<td>UUC</td>
</tr>
<tr>
<td>UUC</td>
<td>2</td>
<td>UUC</td>
</tr>
<tr>
<td>UCU</td>
<td>2</td>
<td>UCC</td>
</tr>
<tr>
<td>UCC</td>
<td>2</td>
<td>UCC</td>
</tr>
<tr>
<td>CCU</td>
<td>2.3 (0.7)</td>
<td>CCU</td>
</tr>
<tr>
<td>CCC</td>
<td>2.4 (1.2)</td>
<td>CCU</td>
</tr>
<tr>
<td>$g_{\text{known}} = 3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UUU</td>
<td>3</td>
<td>UUC</td>
</tr>
<tr>
<td>UUC</td>
<td>3</td>
<td>UUC</td>
</tr>
<tr>
<td>UCU</td>
<td>3</td>
<td>UCC</td>
</tr>
<tr>
<td>UCC</td>
<td>3</td>
<td>UCC</td>
</tr>
<tr>
<td>CCU</td>
<td>4.0 (1.3)</td>
<td>CCU</td>
</tr>
<tr>
<td>CCC</td>
<td>4.3 (1.4)</td>
<td>CCU</td>
</tr>
<tr>
<td>$g_{\text{known}} = 4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UUU</td>
<td>4</td>
<td>UUC</td>
</tr>
<tr>
<td>UUC</td>
<td>4</td>
<td>UUC</td>
</tr>
<tr>
<td>UCU</td>
<td>4</td>
<td>UCC</td>
</tr>
<tr>
<td>UCC</td>
<td>4</td>
<td>UCC</td>
</tr>
<tr>
<td>CCU</td>
<td>5.1 (1.3)</td>
<td>CCU</td>
</tr>
<tr>
<td>CCC</td>
<td>5.1 (1.1)</td>
<td>CCU</td>
</tr>
</tbody>
</table>

Table 3.4: Example of a classification table from a simulation result that shows the overfitting of the data.

<table>
<thead>
<tr>
<th></th>
<th>true</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>estimated</td>
<td>1</td>
<td>56</td>
<td>32</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 3.2: Heatmaps of the rat data. (A) Heatmap of the rat data before biclustering. The red, yellow, and blue bars along the columns and represent liver, blood, and red TA samples, respectively. (B) Heatmap of the rat data after biclustering and subsequently rearranging the rows so that the observation clusters were contiguous. Black and grey bars along the columns and rows simply represent the presence of the clusters and do not indicate relationships between them. \( G = 19 \) for the observation clusters and along the columns, \( q_1 = q_2 = 2 \).
Figure 3.3: Heatmaps of the human data. (A) Heatmap of the human data before biclustering. The red, yellow, and blue bars along the columns and represent immune tissues, whole blood, and brain tissues respectively. (B) Heatmap of the human data after biclustering and subsequently rearranging the rows so that the observation clusters were contiguous. Black and grey bars along the columns and rows simply represent the presence of the clusters and do not indicate relationships between them. $G = 10$ for the observation clusters and along the columns, $q_1 = q_2 = 2$. 
Chapter 4

Mixture Model Averaging for One-Way Supervised Biclustering

4.1 Introduction

In this chapter, a model averaging method is proposed for use within the biclustering framework. The framework is built in Section 4.2 and the performance of the averaging method is evaluated using simulated and real data sets in Section 4.4. A discussion of the work presented in this chapter is found in Section 4.5.

4.2 Background

When applying the OSGaBi family to a data set for biclustering, it is common to fit each of the models for a range of the number of components, $G$. Subsequently, the best model is selected based on the criterion such as the BIC and the corresponding component memberships for each observation are reported. Although previous
studies have shown that the BIC performs well as a model selection criterion for mixture models (Fraley and Raftery, 1999, 2002; McNicholas et al., 2010), other studies have shown that the BIC does not always result in the best predicted component memberships or number of components (see Chapter 3 and Biernacki et al., 2000).

4.2.1 A Posteriori Merging of Components

In model-based clustering using Gaussian mixture models, the definition of “cluster” is commonly synonymous with that of “component”, although there are cases where a Gaussian distribution consists of multiple Gaussian subpopulations (McNicholas, 2016b). In such scenarios, a cluster contains multiple Gaussian components and consequently the BIC may select a model with a larger number of clusters than what is to be expected from the data because it is taking into account all the individual components within each cluster (Hennig, 2010). If this overestimating happens, components can be merged a posteriori to form clusters, and various methods have been proposed to perform this, such as Baudry et al. (2010), Hennig (2010), and more recently Wei and McNicholas (2015). Baudry et al. (2010) propose a merging method based on a hierarchical combination of the mixture components selected by the BIC that minimizes the entropy of the resulting clustering. Hennig (2010) also merges components hierarchically, but the merging criterion can be based on unimodality or misclassification probability estimation.

Wei and McNicholas (2015) present a merging method that maximizes the ARI when comparing the merged model with a pre-defined reference model. To define the reference model, consider a $G$-component mixture model, whose components we want to merge to produce an $H$-component mixture model, where $H < G$. The density
before and after merging can be written as:

\[ f(x) = \sum_{j=1}^{H} \pi_j f^*_j(x) = \sum_{g=1}^{G} \pi_g \phi(x|\mu_g, \Sigma_g), \]  

where each \( \pi^*_j \) is the sum of one or more of \( \pi_1, \ldots, \pi_G \), and each \( f^*_j(x) \) is a mixture of one or more of the component densities \( \phi(x|\mu_1, \Sigma_1), \ldots, \phi(x|\mu_G, \Sigma_G) \). The authors introduce a mixture model component merging criterion based on the ARI using a reference model. They consider two separate cases:

**Case I** The reference model is the model with the largest BIC, and all models with a smaller number of components are discarded.

**Case II** The reference model is the model with the smallest number of components.

They choose the merged mixture model with the best ARI value when compared to the reference model.

**4.2.2 Model Averaging**

After the process of model selection in clustering using a criterion such as the BIC, it is common to proceed with the selected model and ignore the other models, thereby not taking into account the uncertainty that is introduced by having multiple models to choose from. This issue is most apparent in cases when two or more models have very similar model-selection criterion values. The technique of model averaging acknowledges the presence of model uncertainty by averaging parameter estimates across the models deemed as “close”. A common technique is that of Bayesian model
averaging (BMA; Hoeting et al., 1999). Consider models $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K$, the quantity of interest $\Delta$, and data $D$. The posterior probability of $\Delta$ given $D$ is

$$
P(\Delta|D) = \sum_{k=1}^{K} P(\Delta|\mathcal{M}_k, D) P(\mathcal{M}_k|D),
$$

(4.2)

where $P(\Delta|\mathcal{M}_k, D)$ is the posterior distribution of $\Delta$ under model $\mathcal{M}_k$, and $P(\mathcal{M}_k|D)$ is the posterior probability for model $\mathcal{M}_k$ given by

$$
P(\mathcal{M}_k|D) = \frac{P(D|\mathcal{M}_k) P(\mathcal{M}_k)}{\sum_{l=1}^{K} P(D|\mathcal{M}_l) P(\mathcal{M}_l)},
$$

(4.3)

where

$$
P(D|\mathcal{M}_k) = \int P(D|\theta, \mathcal{M}_k) P(\theta|\mathcal{M}_k) d\theta_k
$$

(4.4)

is the integrated likelihood of model $\mathcal{M}_k$, $\theta_k$ is the vector of parameters of $\mathcal{M}_k$, $P(\theta_k|\mathcal{M}_k)$ is the prior density of $\theta_k$ under $\mathcal{M}_k$, $P(D|\theta_k, \mathcal{M}_k)$ is the likelihood, and $P(\mathcal{M}_k)$ is the prior probability that $\mathcal{M}_k$ is the true model given that one of the models considered is true.

Using a logarithmic scoring rule, Madigan and Raftery (1994) showed that BMA gives better predictive ability than using any single model. The shortcomings of BMA in application are two-fold: the posterior model probabilities in Equation 4.3 involve very high-dimensional integrals and are consequently difficult to compute, and the number of models in the summation in Equation 4.4 can be immense. To address the latter shortcoming, Madigan and Raftery (1994) propose the use of Occam’s window to select the set of models, implementing the idea that models should be discarded if they fit the data far worse than the best model. Under this proposed method, models
are discarded if they do not belong to
\[ A' = \left\{ \mathcal{M}_k : \max_l \left\{ \frac{P(\mathcal{M}_l|D)}{P(\mathcal{M}_k|D)} \right\} \leq c \right\}, \]  
(4.5)

for some positive constant \( c \). The authors use \( c = 20 \) to mimic a \( p \)-value of \( 0.05 \), which will be the threshold this chapter adopts in future sections.

To address the first shortcoming, the BIC can be used to approximate the integral in Equation 4.4 (Kass and Raftery, 1995; Dasgupta and Raftery, 1998), i.e.
\[ P(D|\mathcal{M}_k) = \exp \left\{ \frac{1}{2} \text{BIC}_k \right\}, \]
where \( \text{BIC}_k \) is the BIC for model \( \mathcal{M}_k \). Thus, Equation 4.3 can be calculated as
\[ P(\mathcal{M}_k|D) = \frac{\exp \left\{ \frac{1}{2} \text{BIC}_k \right\}}{\sum_{k=1}^{K} \exp \left\{ \frac{1}{2} \text{BIC}_k \right\}}, \]  
(4.6)

and Occam’s window (Equation 4.5) can be written as
\[ \{ \mathcal{M}_k : \max_l \{ \text{BIC}_l \} - \text{BIC}_k \leq 2 \log c \}. \]  
(4.7)

### 4.3 Methodology

In this section, we provide the details for our merging mixture components and model averaging procedures.
4.3.1 Merging Mixture Components

The merging procedure follows the procedure introduced by Wei and McNicholas (2015). In the case where Occam’s window contains models with differing numbers of components, it is necessary to choose a reference model and perform merging on each of the remaining models to achieve the same number of components, $G_R$. In Case I, the maximum BIC does not always correspond to the model with the smallest number of components and instead will have $H$ components. If this happens, models with $H < G_R$ are discarded and models with $H > G_R$ are each merged to have $G_R$ components. In Case II, models do not need to be discarded because the number of components in the reference model, $G_R$, refers to the smallest number of components within Occam’s window. For simplicity, we will refer to the number of components in the reference model, $G_R$, as $G$.

The merging procedure proceeds as follows. First, a $\binom{H}{G} \times G$ dimension matrix is created to store in each row, all the $\binom{H}{G}$ possibilities of where $G$ of the components are assigned after merging. In the second $G^{H-G} \times (H - G)$ dimension matrix, each row will correspond to a possibility where the remaining $(H - G)$ original components can be merged to. Finally, a third $\binom{H}{G} \times G^{H-G}$ dimension matrix will contain the ARI values comparing each merging combination with the reference model classifications. The best ARI value will determine the best merging combination. Model averaging can be performed only once the models in Occam’s window are each merged to contain $G$ components.
4.3.2 Averaging *A Posteriori* Probabilities

In model-based clustering, component membership for the observations is denoted by $z_{ig}$ where

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

To obtain these hard values, the *a posteriori* probabilities

$$\hat{z}_{ig} = \frac{\hat{\pi}_g \phi(x_i | \hat{\mu}_g, \hat{\Sigma}_g)}{\sum_{h=1}^{G} \hat{\pi}_h \phi(x_i | \hat{\mu}_h, \hat{\Sigma}_h)}$$

are hardened to 0 or 1 based on maximum *a posteriori* (MAP) probabilities, i.e., MAP($\hat{z}_{ig}$) = 1 if $\max_h \{\hat{z}_{ih}\}$ occurs in component $h = g$, and MAP($\hat{z}_{ig}$) = 0 otherwise.

If merging has been done on the models within Occam’s window previously, $\hat{z}_{ij}^*$ will simply be the sum of the original *a posteriori* probabilities of the components that were merged to create the new component $j$.

After any models are discarded from Occam’s window and any necessary merging is performed, we compute a weighted average of the *a posteriori* probabilities (AAP) using Equation 4.3 as weights for each model. MAP probabilities can then be determined as described previously.

4.4 Results

In all analyses, the ARI is used to determine performance before and after merging (if applicable) and before and after AAP. The ARI is calculated between the true
classifications and the classifications after clustering and again after AAP.

### 4.4.1 Simulated Data

We consider several scenarios for our simulation studies: model CUU with low, mid, high variances and well-separated components, and high variance and closer component means; models CUC, CCU, and CCC with low, mid, and high variances and well-separated components. For each scenario, we generate \( d = 10 \) variables and \( n = 50 \) observations from the described Gaussian model, with each data set containing \( G = 2 \) components. We run OSGaBi on each data set, providing \( \Lambda_g \) to perform one-way supervised biclustering using five random starts. We fit \( G = 2, \ldots, 15 \) for the CUU simulations and \( G = 2, \ldots, 20 \) for the remaining simulations and the corresponding model as the model subset. Because our model subset consists of one model and \( \Lambda_g \) (and therefore \( q \)) is fixed, there will never be a situation where Occam’s window will contain models with the same number of components; consequently, merging must always be performed before AAP. The four OSGaBi models that contain unrestricted \( \Lambda_g \) are not applicable for AAP because only one model is fitted. That is, because each \( \Lambda_g \) is provided, the algorithm will know in advance how many components are in the model, therefore merging and model averaging are not required.

We also conduct a study using high-dimensional data simulated from the CUU model with mid variance and well-separated components, specifically \( d = 100 \) variables and \( n = 50 \) observations within \( G = 3 \) components. The one-way supervised biclustering settings are the same as described previously for CUU scenarios.

All scenarios produced results for AAP Case II, whereas not all scenarios produced results for AAP Case I (Tables 4.1 and 4.2). Comparing the ARI values between the
best result after biclustering, Case II always improved the result, whereas Case I (if applicable) resulted in the same value. Of particular note is the marked improvement in the ARI values with the scenarios involving CUU and CUC using Case II. Running Case II AAP on the CCU and CCC scenarios still resulted in a large overestimation of the number of components, although there was slight improvement in the classifications.

4.4.2 Real Data

Consider a multivariate data set matrix, where each row is an observation and each column is a variable. The “standard” multivariate data set contains data points that could potentially have drastically different units. Take, for instance, the Australian athletes (ais) data set found in the DAAG package (Maindonald and Braun, 2015) in R. The data consists of measurements on various blood characteristics and anthropometric variables from a group of athletes. It is evident that units of measurement between the blood characteristics and the anthropometric variables will be different, and even within each group of variables there will be differences. Alternatively, the “less conventional” multivariate data set can be thought of as having data points with the same unit of measurement and same range of values. Some examples of this type of data set are microarray gene expression data sets (where each gene is an observation and each sample is a variable) and other data sets where the data points are from the same spectrum of values, such as percentages/proportions. This latter, “less conventional” type of multivariate data set is used in the OSGaBi algorithm. Additionally, besides cluster labels for observations, natural groupings within the variables have to be evident in order for variable clusters to be established.
Upon working with the two microarray data sets from Chapter 3, it is noted that Occam’s window only contains one model in both cases, thus model averaging cannot be applied to those data sets. Instead, we use two different oil data sets to illustrate the performance of our model averaging technique in the following sections.

**Olive Oil Fatty Acid (FA) Composition**

One such benchmark clustering data set that satisfies these criteria is the olive oil data found in the `pgmm` package (McNicholas et al., 2015) in R. The data set is made up of 572 observations on the percent (×100) composition of olive oil with respect to eight fatty acids (palmitic, palmitoleic, stearic, oleic, linoleic, linolenic, arachidic, and eicosenoic acids) found by lipid fractionation from nine areas across three regions in Italy. The fatty acids can be divided into three groups based on the length of their carbon chain: 16 carbons (palmitic and palmitoleic acids), 18 carbons (stearic, oleic, linoleic, and linolenic acids), and 20 carbons (arachidic and eicosenoic acids). These three groups will correspond to the three variable clusters. We use the classifications of the nine areas as the true labels for the data. To differentiate this data set from the next one, we simply refer to this data set as the “FA” data hereafter.

To minimize the subspace the algorithm searches in, semi-supervised biclustering is performed by randomly providing the class labels for approximately half of the observations. We fit $G = 9, \ldots, 15$ and model subset \{CUU, CUC, CCU, CCC\} using five random starts. The biclustering results indicate that there are two models in Occam’s window, both model CUU but the top BIC-scoring one with $G = 10$ components and the second with $G = 9$ components. Because the model with the
highest BIC has the larger number of components, only AAP Case II is applicable. The best ARI value immediately after biclustering is 0.9240, with an increase to 0.9366 after AAP (Table 4.3).

**Triacylglyceride (TG) Composition in Oils**

As discussed at the beginning of this section, biclustering data sets require natural groupings for both the observations and the variables. In the following, the variable groupings are set, but the observation groupings are less evident. The data set consists of triacylglyceride (TG) profiles from a total of 120 oil samples of three types: various types and grades of olive oils, non-olive vegetable oils, and non-olive vegetable oils mixed with olive oil (de la Mata-Espinosa et al., 2011). In our biclustering analysis, the 120 samples act as the variables (i.e., $d = 120$) with the three types as the variable clusters, thus serving as a high-dimensional example. The profiles come from high-performance liquid chromatography analyses performed on each sample, and the observations correspond to the time points at which a peak height was recorded. For illustrative purposes, we sample $n = 21$ time points spaced out evenly throughout the resulting profiles. Specific regions and the TG peaks within these regions are commonly used to identify the type of oil in the sample (see Holcapek et al., 2005; de la Mata-Espinosa et al., 2011). The time points were classified based on this information; each time point falling within a useful TG peak was placed into its own cluster, time points falling within a useful region were placed into respective clusters, and non-informative time points were placed into a final cluster. For simplification and clarification, we will refer to this data set as the “TG” data.
We fit $G = 2, \ldots, 20$ and model subset \{CUU, CUC, CCU, CCC\} using five random starts. The biclustering results indicate that there are two models in Occam’s window, both model CCC but the top BIC-scoring one with $G = 14$ components and the second with $G = 18$ components. Table 4.4 presents the model averaging results. The best ARI value immediately after biclustering is 0.4554, with an increase to 0.5316 after AAP Case I, but decreases slightly to 0.4241 after AAP Case II.

4.5 Discussion

We propose a shift from the conventional “single best model” paradigm commonly seen within the model-based biclustering literature to using a model averaging approach that utilizes Occam’s window. The approach averages a posteriori probabilities (AAP), which requires the models within Occam’s window to possess the same number of components. If models do not possess the same number of components, merging of components is proposed using a method developed based on the ARI. The ARI calculation requires a “reference” classification, and this can be obtained from the model with the highest BIC or lowest number of components within Occam’s window, then labelled as the reference model. Although model averaging has been proposed on a number of occasions in the model-based clustering framework, this is the first mention of applying the procedure in the biclustering context to the best of the authors’ knowledge.

When considering the models within Occam’s window for model averaging, it is not always necessary to use all models available. During AAP, two cases are examined. In Case I, models with at least the same number of components as the reference model are used, whereas in Case II, all models are used. The performance of AAP relies
heavily on the number of components selected; therefore, within Occam’s window, the method of selecting the number of components is very important. An additional factor that affects the performance of AAP, regardless of case, is the effectiveness of the merging procedure.

We illustrated the performance of AAP in the biclustering framework using both low- and high-dimensional simulation scenarios. AAP Case II always had superior results compared to when solely biclustering, as measured by the ARI. In the scenarios where AAP Case I are applicable (i.e., the model with the best BIC was not the model with the largest number of components), it performed comparably to when only biclustering. The last two scenarios involving OSGaBi models CCU and CCC were still grossly overestimating the number of components despite the improvement in ARI after model averaging. This suggests that an alternative method for selecting the number of components for the final averaged model may allow for better overall performance of the procedure, including when scenarios like this arise.

We also tested the performance of AAP using a benchmark clustering data set. Although there are many benchmark clustering data sets available, finding ones that could be used as a benchmark biclustering data set suited to our needs is not as straightforward a task. Characteristics of a suitable data set included data points that originate from a spectrum of values, regardless of variable. Additionally, the variables have to possess natural groupings in order to facilitate biclustering. The FA data measured percent composition of fatty acids, therefore satisfying the former criteria. Grouping the fatty acids by their carbon chain length aided in satisfying the latter criteria. Results based on semi-supervised biclustering indicated some improvement after AAP Case II was applied (Case I was not applicable). To our knowledge,
this is the first time the FA data has been biclustered and the variables used in this fashion. We use semi-supervised biclustering which steers away from a true biclustering problem. Semi-supervision is supported because in many real-life scenarios, information regarding the classes of some (even if only a small fraction of) observations is known. These labeled data substantially aid in the efficiency and accuracy of the clustering algorithm by guiding it to a good region of the search space (Basu et al., 2002).

Our final TG data set illustrates the AAP technique in a high-dimensional environment. It is not the ideal data set because it does not possess definitive natural groupings for (what we define as) the observations that we can compare our results with, but researching the properties of typical TG profiles reveals a crude classification system sufficient for proof of concept. We see increased agreement with the crude classes after AAP Case I, but not with Case II, and we also observe that the ARI values are not particularly close to 1 regardless of utilizing AAP or not. This does not necessarily mean that our model averaging algorithm has no potential in the high-dimensional case, but rather could imply that our proposed classification system is indeed crude and thus can be further refined, resulting in improved and larger ARI values.

There are potential trade-offs among the two AAP cases, some of which have already been highlighted through the simulation studies and the real data set. Except for the high-dimensional TG data, AAP Case II has consistently improved the classification results, indicating that it has the most impact on classification performance. The trade off, because it chooses as the reference model the one with the
least number of components, is that it may underestimate the number of components. Conversely, with AAP Case I, underestimating the number of components is less likely, but improvement in classification performance may be limited or nil, as seen in our analyses.
Table 4.1: A summary of the models in Occam’s window, along with ARI values for the true labels versus predicted classifications from the best model and from averaging 
\textit{a posteriori} probabilities for the simulation studies using $n = 50$ observations and $d = 10$ variables.

| Scenario            | Occam’s Window | $P(\mathcal{M}_i|D)$ | ARI |
|---------------------|----------------|-----------------------|-----|
|                     | $G$  | BIC       | Case I | Case II | Best | Case I | Case II |
| CUU, low variance   | 3    | -2468.711 | 0.5008  | 0.5008   | 0.9016 | 0.9016 | 0.9412  |
|                     | 4    | -2468.718 | 0.4992  | 0.4992   |        |        |         |
| CUU, mid variance   | 5    | -2360.603 | N/A     | 0.5347   | 0.8022 | N/A    | 1       |
|                     | 2    | -2360.881 | N/A     | 0.4653   |        |        |         |
| CUU, high variance  | 4    | -2593.142 | 0.6509  | 0.6058   | 0.8775 | 0.8775 | 1       |
|                     | 5    | -2594.388 | 0.3491  | 0.3249   |        |        |         |
|                     | 2    | -2597.479 | N/A     | 0.0693   |        |        |         |
| CUU, high variance, | 4    | -2550.901 | N/A     | 0.7399   | 0.5761 | N/A    | 0.9200  |
| closer means        | 2    | -2550.992 | N/A     | 0.2601   |        |        |         |
| CUC, low variance   | 8    | -2175.498 | N/A     | 0.6853   | 0.5568 | N/A    | 1       |
|                     | 7    | -2144.024 | N/A     | 0.3147   |        |        |         |
| CUC, mid variance   | 3    | -2987.339 | N/A     | 0.9438   | 0.9412 | N/A    | 1       |
|                     | 2    | -2992.979 | N/A     | 0.0562   |        |        |         |
| CUC, high variance  | 4    | -2981.492 | 0.9319  | 0.9319   | 0.8426 | 0.8426 | 1       |
|                     | 6    | -2792.295 | 0.0681  | 0.0681   |        |        |         |
| CCU, low variance   | 16   | -2312.461 | 0.7773  | 0.7773   | 0.2183 | 0.2183 | 0.3793  |
|                     | 18   | -2323.689 | 0.2227  | 0.2227   |        |        |         |
| CCU, mid variance   | 18   | -2684.379 | 0.6890  | 0.6890   | 0.1619 | 0.1619 | 0.1930  |
|                     | 19   | -2685.970 | 0.3110  | 0.3110   |        |        |         |
| CCU, high variance  | 20   | -2858.263 | N/A     | 0.5549   | 0.1665 | N/A    | 0.2782  |
|                     | 19   | -3047.554 | N/A     | 0.4451   |        |        |         |
| CCC, low variance   | 18   | -2171.283 | 0.8811  | 0.8811   | 0.2748 | 0.2748 | 0.3747  |
|                     | 19   | -2120.263 | 0.1189  | 0.1189   |        |        |         |
| CCC, mid variance   | 19   | -2659.069 | 0.7928  | 0.7928   | 0.2610 | 0.2610 | 0.2863  |
|                     | 20   | -2661.753 | 0.2072  | 0.2072   |        |        |         |
| CCC, high variance  | 19   | -2804.484 | 0.8873  | 0.8873   | 0.2391 | 0.2391 | 0.2690  |
|                     | 20   | -2783.145 | 0.1127  | 0.1127   |        |        |         |
Table 4.2: A summary of the models in Occam’s window, along with ARI values for the true labels versus predicted classifications from the best model and from averaging *a posteriori* probabilities for the high-dimensional simulation study using $n = 50$ observations and $d = 100$ variables.

| Scenario          | Occam’s Window | $P(M_i|D)$ | ARI       |
|------------------|---------------|---------|-----------|
|                  | $G$ | BIC     | Case I | Case II | Best | Case I | Case II |
| CUU, mid variance| 3   | -23342.6 | N/A    | 0.8940  | 0.7842 N/A | 1       |
|                  | 2   | -23374.5 | N/A    | 0.1060  |       |         |         |

Table 4.3: A summary of the models in Occam’s window, along with ARI values for the true labels versus predicted classifications from the best model and from averaging *a posteriori* probabilities for the FA data set.

| Model | Occam’s Window | $P(M_i|D)$ | ARI       |
|-------|---------------|---------|-----------|
|       | $G$ | BIC     | Case I | Case II | Best | Case I | Case II |
| CUU   | 10  | -43931.59| N/A    | 0.8380  | 0.9240 N/A | 0.9366 |
| CUU   | 9   | -43934.88| N/A    | 0.1620  |       |         |         |

Table 4.4: A summary of the models in Occam’s window, along with ARI values for the true labels versus predicted classifications from the best model and from averaging *a posteriori* probabilities for the TG data set.

| Model | Occam’s Window | $P(M_i|D)$ | ARI       |
|-------|---------------|---------|-----------|
|       | $G$ | BIC     | Case I | Case II | Best | Case I | Case II |
| CCC   | 14  | 7162.024| 0.5000  | 0.5000  | 0.4554 0.5316 | 0.4241 |
| CCC   | 18  | 7162.023| 0.5000  | 0.5000  |       |         |         |
Chapter 5

Two-Way Clustering with One-Way Supervision Using Contaminated Gaussians

5.1 Introduction

In many applications, the data generated are often contaminated with outliers, spurious points, or noise (herein collectively referred to as “bad” points), thus robust distributions are required to deal with this type of contaminated data. As the current literature has provided evidence of successfully handling bad points in a model-based manner within the clustering paradigm yet not within the biclustering paradigm, this chapter extends the problem of contamination to the one-way supervised model-based biclustering paradigm described in Chapter 3. In the following sections a one-way supervised contaminated Gaussian mixture model family for biclustering when dealing with bad points is described (Section 5.2), the biclustering family’s performance on
simulated and real data contaminated with bad points is demonstrated (Section 5.3), and the family and its potential application are discussed (Section 5.4).

The multivariate Gaussian distribution is commonly used to model data from a continuous random variable because of its mathematical tractability. Frequently, the multivariate $t$-distribution is used to deal with bad points, where the bad points can be accommodated due to the degrees of freedom parameter and the resulting heavier tails of the distribution (Lange et al., 1989; Kotz and Nadarajah, 2004). This method was first developed for clustering by McLachlan and Peel (1998) and Peel and McLachlan (2000). Another option involves convenient robust estimates of the means and covariance matrices of the multivariate Gaussian distribution (see Campbell, 1984; McLachlan and Basford, 1988; De Veaux and Krieger, 1990; Markatou, 2000). An alternative method is to add a distribution to model the bad points alongside the Gaussian distribution that models the “good” points. One such mixture is the contaminated Gaussian distribution. As the name implies, it consists of a Gaussian distribution to model the good points with a large prior probability (i.e., greater than 0.5), and it also contains a second Gaussian component with the same mean and an inflated covariance, with a smaller prior probability, to model the bad points (Tukey, 1960; Aitkin and Wilson, 1980).

The advantage of the contaminated Gaussian distribution is that it allows for automatic outlier detection, whereas the $t$-distribution will just “absorb” the bad points into the distribution and not identify them. Additionally, the mean of the two components is a weighted mean and is calculated in such a way that the bad points have reduced impact on the estimate (see Little, 1988, for further discussion on down-weighting for contaminated Gaussian distributions).
In model-based clustering, the multivariate Gaussian mixture model is commonly used because of its mathematical tractability. More robust ways of clustering, such as using the $t$ mixture model (examples include McLachlan and Peel, 1998; Peel and McLachlan, 2000; Andrews and McNicholas, 2011a,b) or the contaminated Gaussian mixture model mentioned above (examples include Punzo and McNicholas, 2016; Blostein, 2016), are conventional ways of dealing with bad points.

Similar to clustering, bad points can be present in the data when biclustering. Current biclustering techniques that accommodate contamination utilize robust double $k$-means methods (Farcomeni, 2009; Ferraro and Vichi, 2015), robust methods involving the mean-squared residue (Leng and Hong, 2010; Wang et al., 2002), and a relaxation-based geometric technique (Zhao et al., 2009), among others.

### 5.2 Methodology

#### 5.2.1 The Contaminated Gaussian Mixture Model

Here, we adopt the multivariate contaminated Gaussian distribution

$$
\phi_{CN}(\mathbf{x}; \vartheta_g) = \alpha_g \phi(\mathbf{x}; \mu_g, \Sigma_g) + (1 - \alpha_g) \phi(\mathbf{x}; \mu_g, \eta_g \Sigma_g),
$$

where $\alpha_g \in (0.5, 1)$, the degree of contamination $\eta_g > 1$, and $\vartheta_g = \{\alpha_g, \mu_g, \Sigma_g, \eta_g\}$ for $g = 1, \ldots, G$. The constraint $\alpha_g \in (0.5, 1)$ is consistent with the notion in robust statistics that at least half of the observations are good. The degree of contamination can be interpreted as the increase in variability due to the bad observations, as reflected by the constraint $\eta_g > 1$, and can also be called the inflation parameter. The density of the mixture of multivariate contaminated Gaussian distributions is
\[ p_{\text{CN}}(x; \vartheta_g) = \sum_{g=1}^{G} \pi_g \phi(x; \mu_g, \Sigma_g) + (1 - \alpha_g) \phi(x; \mu_g, \eta_g \Sigma_g). \]

Although our contaminated model has \(2G\) components because each \(g\) has one Gaussian component and one contaminated Gaussian component, the use of the term “component” or “cluster” will refer to \(G\) in the rest of the chapter, unless otherwise stated.

These models are extensions of the OSGaBi family from Chapter 3 and thus have the same nomenclature, covariance structure, and number of covariance parameters as described in Table 3.1. The family will be referred to as OSCGaBi (one-way supervised contaminated Gaussian biclustering) hereafter.

### 5.2.2 AECM Algorithm

The AECM algorithm is used for parameter estimation. In the algorithm, we separate the parameters into two groups \(\vartheta = \{\vartheta_1, \vartheta_2\}\), corresponding to each cycle, specifically \(\vartheta_1 = \{\pi_g, \mu_g, \alpha_g, \eta_g\}\) and \(\vartheta_2 = \{\Psi_g\}\). The first group, \(\vartheta_2\), is further split into two subgroups \(\vartheta_1 = \{\vartheta_{11}, \vartheta_{12}\}\), corresponding to the two CM steps in the first cycle, where \(\vartheta_{11} = \{\pi_g, \mu_g, \alpha_g\}\) and \(\vartheta_{12} = \{\eta_g\}\).

### 5.2.3 First Cycle

In the first AECM cycle, we take \(v_1, \ldots, v_n\) and \(z_1, \ldots, z_n\) as the missing data, where \((v_i = v_{i1}, \ldots, v_{iG})'\). Thus, \((x_1, \ldots, x_n, v_1, \ldots, v_n, z_1, \ldots, z_n)\) are the complete data and the complete-data log-likelihood can be written as

\[ l_{1c}(\vartheta_1) = l_{1c_1}(\{\pi_g\}_{g=1}^{G}) + l_{1c_2}(\{\alpha_g\}_{g=1}^{G}) + l_{1c_3}(\{\mu_g, \eta_g\}_{g=1}^{G}) \]
where

\[
\begin{align*}
 l_{1c1} \left( \{\pi_g\}_{g=1}^G \right) &= \sum_{i=1}^n \sum_{g=1}^G z_{ig} \log(\pi_g) \\
 l_{1c2} \left( \{\alpha_g\}_{g=1}^G \right) &= \sum_{i=1}^n \sum_{g=1}^G z_{ig} \left[ v_{ig} \log(\alpha_g) + (1 - v_{ig}) \log(1 - \alpha_g) \right] \\
 l_{1c3} \left( \{\mu_g, \eta_g\}_{g=1}^G \right) &= -\frac{1}{2} \sum_{i=1}^n \sum_{g=1}^G \left[ z_{ig} \log|\Sigma_g^{(k)}| + d z_{ig} (1 - v_{ig}) \log(\eta_g) \\
 &+ z_{ig} \left( v_{ig} \frac{1 - v_{ig}}{\eta_g} \right) (x_i - \mu_g)' \left( \Sigma_g^{(k)} \right)^{-1} (x_i - \mu_g) \right],
\end{align*}
\]

with \( \Sigma_g^{(k)} = \Lambda_g^{(k)} \Lambda_g^{(k)\prime} + \Psi_g^{(k)} \) for the \( k \)th iteration.

**E Step**

During the E step of the first cycle for iteration \((k + 1)\), we calculate \( Q_1 \), the expectation of \( l_{1c} \) given the observed data and \( \vartheta^{(k)} \). Here we require

\[
\begin{align*}
 z_{ig}^{(k)} &= \mathbb{E} \left( Z_{ig}|x_i, \vartheta^{(k)} \right) = \frac{\pi_g^{(k)} \ p_{\text{CN}}(x_i; \mu_g^{(k)}, \Sigma_g^{(k)}, \alpha_g^{(k)}, \eta_g^{(k)})}{\sum_{j=1}^G \pi_j^{(k)} \ p_{\text{CN}}(x_i; \mu_j^{(k)}, \Sigma_j^{(k)}, \alpha_j^{(k)}, \eta_j^{(k)})} \\
v_{ig}^{(k)} &= \mathbb{E} \left( V_{ig}|x_i, \vartheta^{(k)} \right) = \frac{\alpha_g^{(k)} \phi(x_i; \mu_g^{(k)}, \Sigma_g^{(k)})}{p_{\text{CN}}(x_i; \mu_g^{(k)}, \Sigma_g^{(k)}, \alpha_g^{(k)}, \eta_g^{(k)})},
\end{align*}
\]

where \( Z_{ig} \) and \( V_{ig} \) are the random variables corresponding to \( z_{ig} \) and \( v_{ig} \), respectively.
CM Step 1

During the first CM step, we maximize $Q_1$ with respect to $\vartheta_{11}$, fixing $\vartheta_{12} = \vartheta_{12}^{(k)}$, leading to the following updates

$$
\pi_{g}^{(k+1)} = \frac{n_g^{(k+1)}}{n}; \quad \alpha_{g}^{(k+1)} = \frac{1}{n_g^{(k+1)}} \sum_{i=1}^{n} z_{ig}^{(k)} v_{ig}^{(k)}; \quad \mu_{g}^{(k+1)} = \frac{\sum_{i=1}^{n} z_{ig}^{(k)} \left( v_{ig}^{(k)} + \frac{1 - v_{ig}^{(k)}}{\eta_{g}^{(k)}} \right) x_i}{\sum_{i=1}^{n} z_{ig}^{(k)} \left( v_{ig}^{(k)} + \frac{1 - v_{ig}^{(k)}}{\eta_{g}^{(k)}} \right)};
$$

where $n_g^{(k)} = \sum_{i=1}^{n} z_{ig}^{(k)}$.

CM Step 2

During the second CM step, we maximize $Q_1$ with respect to $\vartheta_{12}$, fixing $\vartheta_{11} = \vartheta_{11}^{(k+1)}$. Specifically,

$$
Q_1 = \frac{d}{2} \sum_{i=1}^{n} z_{ig}^{(k)} \left( 1 - v_{ig}^{(k)} \right) \log(\eta_{g}) - \frac{1}{2} \sum_{i=1}^{n} z_{ig}^{(k)} \left( 1 - v_{ig}^{(k)} \right) \left( x_i - \mu_{g}^{(k+1)} \right)' \left( \Sigma_{g}^{(k)} \right)^{-1} \left( x_i - \mu_{g}^{(k+1)} \right)
$$

needs to be maximized with respect to $\eta_{g}$ with the constraint $\eta_{g} > 1, g = 1, \ldots, G$.

The optimize() function of the R package stats is used to determine the analytical solution because a closed form solution is not possible.

5.2.4 Second Cycle

In the second AECM cycle, we take $v_1, \ldots, v_n, z_1, \ldots, z_n$, and the latent factors $u_1, \ldots, u_n$ as the missing data, where $u_{ig} = u_{i1}, \ldots, u_{iQ}$, for $g = 1, \ldots, G$. Thus,
\( \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{v}_1, \ldots, \mathbf{v}_n, \mathbf{z}_1, \ldots, \mathbf{z}_n, \mathbf{u}_1, \ldots, \mathbf{u}_n \) are the complete data and the complete-data log-likelihood can be written as

\[
l_{2c}(\vartheta) = C + \sum_{g=1}^{G} \left\{ \frac{-n_g}{2} \log |\Psi_g| - \frac{n_g}{2} \text{tr} \left( \Psi_g^{-1} S_g^{(k+1)} \right) \right. \\
+ \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) \left( \mathbf{x}_i - \mu_g^{(k+1)} \right)' \Psi_g^{-1} \Lambda_g u_{ig} \\
- \frac{1}{2} \text{tr} \left[ \Lambda_g' \Psi_g^{-1} \Lambda_g \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) u_{ig} u_{ig}' \right] \}
\]

where \( n_g = \sum_{i=1}^{n} z_{ig} \), \( C \) is a constant with respect to \( \vartheta_2 \), and

\[
S_g^{(k+1)} = \frac{1}{n_g} \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) \left( \mathbf{x}_i - \mu_g^{(k+1)} \right) \left( \mathbf{x}_i - \mu_g^{(k+1)} \right)'.
\]

**E Step**

During the E step of the second cycle for iteration \((k + 1)\), we calculate \( Q_2 \), the expectation of \( l_{2c} \) given the observed data and \( \vartheta^{(k+1/2)} = \{ \vartheta_1^{(k+1)}, \vartheta_2^{(k)} \} \). Here we require replacing \( z_{ig} \) and \( v_{ig} \) with

\[
\begin{align*}
\vartheta_1^{(k+1/2)} = \mathbb{E} \left( Z_{ig} | \mathbf{x}_i, \vartheta^{(k+1/2)} \right) & \quad \text{and} \\
\vartheta_2^{(k+1/2)} = \mathbb{E} \left( V_{ig} | \mathbf{x}_i, \vartheta^{(k+1/2)} \right),
\end{align*}
\]

respectively, and

\[
\mathbb{E}_{\vartheta^{(k+1/2)}} \left[ Z_{ig} \left( V_{ig} + \frac{1 - V_{ig}}{\eta_g^{(k+1)}} \right) U_{ig} | \mathbf{x}_i \right] = \vartheta_1^{(k+1/2)} \left( V_{ig}^{(k+1/2)} + \frac{1 - V_{ig}^{(k+1/2)}}{\eta_g^{(k+1)}} \right) \beta_g^{(k)} \left( \mathbf{x}_i - \mu_g^{(k+1)} \right)
\]
\[ \mathbb{E}_{\varphi^{(k+1/2)}}(Z_{ig}V_{ig}U_{ig}U'_{ig}|x_i) = z_{ig}^{(k+1/2)}v_{ig}^{(k+1/2)} \begin{bmatrix} I_g - \beta_g^{(k)} \Lambda_g^{(k)} \\ + \beta_g^{(k)}(x_i - \mu_g^{(k+1)})(x_i - \mu_g^{(k+1)})' \beta_g^{(k)'} \end{bmatrix} \]

\[ \mathbb{E}_{\varphi^{(k+1/2)}} \left[ Z_{ig} \left( \frac{1 - V_{ig}}{\eta_g^{(k+1)}} \right) U_{ig} U'_{ig}|x_i \right] = z_{ig}^{(k+1/2)} \left( \frac{1 - v_{ig}^{(k+1/2)}}{\eta_g^{(k+1)}} \right) \begin{bmatrix} I_g - \beta_g^{(k)} \Lambda_g^{(k)} \\ + \beta_g^{(k)}(x_i - \mu_g^{(k+1)})(x_i - \mu_g^{(k+1)})' \beta_g^{(k)'} \end{bmatrix}, \]

where \( \beta_g^{(k)} = \Lambda_g^{(k)'} \left( \Lambda_g^{(k)} \beta_g^{(k)} + \Psi_g^{(k)} \right)^{-1} \). Thus, the expected complete-data log-likelihood is

\[ Q_2(\varphi) = C + \sum_{g=1}^{G} n_g^{(k+1/2)} \left\{ \frac{1}{2} \log |\Psi_g^{(k+1/2)}| - \frac{1}{2} \text{tr} \left( \Psi_g^{(k+1/2)} S_g^{(k+1)} \right) \right. \]

\[ + \left. \text{tr} \left( \Psi_g^{(k+1/2)} \Lambda_g^{(k)} \beta_g^{(k)} S_g^{(k+1)} \right) - \frac{1}{2} \text{tr} \left[ \Lambda_g^{(k)'} \Psi_g^{(k+1/2)} \Lambda_g^{(k)} \Theta_g^{(k+1/2)} \right] \right\}, \]

where \( n_g^{(k+1/2)} = \sum_{i=1}^{n} z_{ig}^{(k+1/2)} \) and \( \Theta_g^{(k+1/2)} = I_g - \beta_g^{(k)} \Lambda_g^{(k)} + \beta_g^{(k)} S_g^{(k+1)} \beta_g^{(k)'} \) is a symmetric \( q \times q \) matrix.

**CM Step**

During the CM step of the second cycle, we maximize \( Q_2 \) with respect to \( \varphi \), fixing \( \varphi_1 = \varphi_1^{(k+1)} \), leading to the following update

\[ \Psi_g^{(k+1)} = \text{diag} \left( S_g^{(k+1)} - \Lambda_g^{(k+1)} \beta_g^{(k)} S_g^{(k+1)} \right). \]
The equations and updates derived in this section are for the most unconstrained model, UUU. As the updates for the remaining seven models are similar, we provide those details in Appendix A.

5.2.5 Component Membership and Outlier Identification

The predicted biclustering for each member of the OSCGaBi family is given by the MAP classification for the observations and the classifications originally provided for the variables, similar to the procedure for the OSGaBi family.

To classify observations in each component as good or bad, we consider the \( \text{a posteriori} \) probability, \( v_{ig} \). If \( v_{ig} > 0.5 \), the observation \( x_i \) is considered good; otherwise, \( x_i \) is labelled as bad.

5.3 Results

To investigate the performance of the OSCGaBi family, we compare our results with some other biclustering and clustering methods found in R, namely:

1. OSGaBi, the non-contaminated equivalent of the proposed family in Chapter 3,
2. PGMM, model-based clustering using Gaussian factor analyzers (McNicholas and Murphy, 2008),
3. MCLUST, model-based clustering using Gaussian mixtures parameterized by eigenvalue decomposition (Fraley and Raftery, 2002; Fraley et al., 2012), and
4. MMtFA, model-based clustering using \( t \) factor analyzers (Andrews and McNicholas, 2011a,b; Andrews et al., 2016).
OSGaBi, PGMM, and MMtFA were selected for comparison because they also utilize mixtures of factor analyzers. MCLUST was selected because it is a popular model-based clustering algorithm, and MMtFA was selected because it uses the multivariate t distribution, which is one of the alternative ways of handling bad points mentioned in Section 5.1. OSGaBi was selected because it is the closest biclustering algorithm to OSCGaBi. We used the default settings for the above algorithms, unless otherwise noted. For methods dealing with factor analyzers (i.e., OSCGaBi, OSGaBi, PGMM, and MMtFA), only models with constrained $\Lambda_g$ were run. For MCLUST, only multivariate mixture models were run. To keep consistent with the one-way supervision of the proposed algorithm, methods utilizing factor analysis (i.e., PGMM and MMtFA) were set to $q$ factors depending on the known number of variable clusters in the data set being analyzed. The model with the best BIC is presented for each result and the ARI is used to determine performance.

5.3.1 Simulation Studies

To test our method, we simulated 100 data sets from the CUU model of the OSGaBi family. Each data set consisted of $n = 400$ observations and $d = 20$ variables, broken into $g = 4$ observation clusters and $q = 3$ variable clusters. Out of the total number of observations, 20% were outliers with inflation factor $\eta_g = 2$. These outliers were distributed throughout the components. We ran OSCGaBi and the other methods on the resulting data set, fitting the range of $G = 2, \ldots, 10$.

We present the simulation study results in Table 5.1. OSCGaBi has an average ARI of 0.980 and outperforms the other (bi)clustering methods in retrieving the correct observation clusters. Compared to OSGaBi, the contaminated family result
is favoured for one-way supervised biclustering. It is noted that MMtFA essentially performed equivalently to OSCGaBi based on the very similar ARI values and average estimated number of components.

Table 5.1: Simulation study results. Standard deviations are found in the brackets. The highest-scoring ARI is in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average $G$</th>
<th>Most chosen model</th>
<th>Average ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSCGaBi</td>
<td>4.03 (0.171)</td>
<td>CCC</td>
<td><strong>0.980</strong> (0.013)</td>
</tr>
<tr>
<td>OSGaBi</td>
<td>9.79 (0.433)</td>
<td>CCU</td>
<td>0.609 (0.093)</td>
</tr>
<tr>
<td>PGMM</td>
<td>4.58 (0.781)</td>
<td>CCU</td>
<td>0.933 (0.067)</td>
</tr>
<tr>
<td>MCLUST</td>
<td>3.85 (0.359)</td>
<td>EEE</td>
<td>0.941 (0.097)</td>
</tr>
<tr>
<td>MMtFA</td>
<td>4 (0)</td>
<td>CCIC</td>
<td>0.979 (0.010)</td>
</tr>
</tbody>
</table>

5.3.2 Real Data

To further validate the performance of the OSCGaBi family, we applied it to two real data sets. Both data sets look at fatty acid composition, but one is within olive oil and the other is within various commercial oils. Outliers were added to these data sets to introduce contamination. Additionally, we compared our results with the (bi)clustering algorithms mentioned previously. For each algorithm and each data set, the range of $G = 2, \ldots, 15$ was fitted.

Olive Oil Fatty Acid Composition

The first data set is the same olive oil data used in Chapter 4. Similarly, we used the classifications of the nine areas as the true labels for the data. To introduce contamination, we added 143 points to the data set with an inflation parameter of $\eta_g = 2$ so that it corresponded to 20% of the total number of observations. To
simulate from the correct contaminated distribution, OSGaBi was run on the original data set with the classifications provided, and the resulting parameter estimates were retained for the contaminated distribution. The outliers were distributed among the nine components. We ran OSCGaBi and the other methods on the resulting data set, which we refer to as the “olive oil” data hereafter.

Table 5.2 presents the results from the analyses on the olive oil data. OSCGaBi is the clear top performer in this case with an ARI of 0.810. It underestimated the number of components, and when looking at the classification table, it was mainly the case that two of the true components were being combined into other components to decrease the number of components to seven (results not shown). This was also seen for the other methods, but they further underestimated (MM\(_t\)FA) or overestimated the number of components (OSGaBi, PGMM, and MCLUST).

Table 5.2: (Bi)clustering results using the olive oil data with noise added to make up 20% of the total number of observations. The highest-scoring ARI is in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>(G)</th>
<th>Model</th>
<th>ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSCGaBi</td>
<td>7</td>
<td>CUU</td>
<td>0.810</td>
</tr>
<tr>
<td>OSGaBi</td>
<td>11</td>
<td>CCU</td>
<td>0.689</td>
</tr>
<tr>
<td>PGMM</td>
<td>14</td>
<td>CUU</td>
<td>0.591</td>
</tr>
<tr>
<td>MCLUST</td>
<td>14</td>
<td>VVE</td>
<td>0.449</td>
</tr>
<tr>
<td>MM(_t)FA</td>
<td>5</td>
<td>CCUU</td>
<td>0.668</td>
</tr>
</tbody>
</table>

Table 5.3 indicates that the MAP classifications have a tendency to label samples corresponding to its origin in terms of region, indicating that perhaps olive oil from areas within a region are somehow similar with respect to their fatty acid composition when taking into consideration the lengths of the fatty acid carbon chain. For instance, almost all Sardinia region samples were placed into the same component, regardless of being from the coast or inland. In the region of south Italy, the majority
of the Sicily samples were not grouped together, but were mainly divided between Calabria and North Apulia.

Table 5.3: Cross-tabulation of the MAP classifications associated with the selected OSCGaBi model against true classes for the olive oil data.

<table>
<thead>
<tr>
<th>Region</th>
<th>Area</th>
<th>Estimated Component</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>Sardinia</td>
<td>Coastal Sardinia</td>
<td>48 0 0 0 0 0 0</td>
</tr>
<tr>
<td></td>
<td>Inland Sardinia</td>
<td>78 0 0 2 0 0 0</td>
</tr>
<tr>
<td>South Italy</td>
<td>Calabria</td>
<td>0 64 4 2 0 1 0</td>
</tr>
<tr>
<td>Sicily</td>
<td></td>
<td>0 26 18 7 0 0 0</td>
</tr>
<tr>
<td>North Apulia</td>
<td></td>
<td>0 2 35 0 2 1 0</td>
</tr>
<tr>
<td>South Apulia</td>
<td></td>
<td>0 4 0 217 0 0 0</td>
</tr>
<tr>
<td>North Italy</td>
<td>Umbria</td>
<td>0 0 2 0 68 4 0</td>
</tr>
<tr>
<td>East Liguria</td>
<td></td>
<td>0 0 1 0 3 60 1</td>
</tr>
<tr>
<td>West Liguria</td>
<td></td>
<td>0 0 0 0 0 3 62</td>
</tr>
</tbody>
</table>

Commercial Oil Fatty Acid Composition

The second real data set is found in the caret package in R (Brodnjak-Vončina et al., 2005). The oil data set consists of concentrations (in percentages) of seven fatty acids measured in 96 commercial oil samples. The samples consisted of pumpkin, sunflower, peanut, olive, soybean, rapeseed, and corn oils. These seven oil origins were used as the true classifications of the observations. The fatty acids were a mix of 16-carbon (palmitic acid), 18-carbon (stearic, oleic, linoleic, and linolenic acids), and 20-carbon (eicosanoic and eicosenoic acids) fatty acids; these three carbon-chain lengths were used to define the variable clusters.

To contaminate the data set, we added 24 points with an inflation parameter of \( \eta_g = 2 \) to result in 20% of the data being contaminated. Similar to the olive oil data, OSGaBi was run on the original data set with the classifications provided in order to
simulate from the correct contaminated distribution. The resulting parameter estimates were used for the contaminated distribution, and the outliers were distributed among the seven components. We then ran OSCGaBi and the other methods on the final data set, which will be referred to as the “commercial oil” data hereafter.

Table 5.4 shows the performance of the methods on the oil data set. MM$_{t}$FA performed the best based on its ARI value of 0.788, but it grossly overestimated the number of components compared to the rest of the methods. The classification table indicated that besides mislabelling some observations, the algorithm broke down the true components further, sometimes with only one observation in a component (results not shown). The second best performer was OSCGaBi, with a slightly lower ARI value of 0.742, but did not overestimate the number of components. Furthermore, the classification error rate of OSCGaBi was 13.3%, misclassifying 16 of the observations (the same calculation cannot be done for the MM$_{t}$FA results because the number of true classes and the number of estimated classes are not equal).

Table 5.4: (Bi)clustering results using the commercial oil data with noise added to make up 20% of the total observations. The highest-scoring ARI is in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>$G$</th>
<th>Model</th>
<th>ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSCGaBi</td>
<td>7</td>
<td>CUC</td>
<td>0.742</td>
</tr>
<tr>
<td>OSGaBi</td>
<td>7</td>
<td>CUU</td>
<td>0.457</td>
</tr>
<tr>
<td>PGMM</td>
<td>8</td>
<td>CUU</td>
<td>0.635</td>
</tr>
<tr>
<td>MCLUST</td>
<td>6</td>
<td>VEE</td>
<td>0.646</td>
</tr>
<tr>
<td>MM$_{t}$FA</td>
<td>15</td>
<td>CCCC</td>
<td><strong>0.788</strong></td>
</tr>
</tbody>
</table>
5.4 Discussion

In clustering, data contaminated with bad points are often dealt with using robust model-based methods, commonly involving $t$ mixture models or using an extra component to model the bad points, such as the contaminated Gaussian mixture model. The latter method is advantageous in that it allows for automatic outlier detection and reduces the impact the bad points have on the component means. While the focus on robust model-based methods in the current literature is within the clustering paradigm, this paper has introduced a family of contaminated Gaussian mixture models for dealing with bad points within the biclustering paradigm, specifically for one-way supervised biclustering.

Simulation study results indicate that OSCGaBi outperformed the other clustering and biclustering methods, with MM$t$FA coming in a very close second. It can be argued that MM$t$FA resulted in the same performance as OSCGaBi since the difference in ARI was small, but MM$t$FA is a clustering family and thus does not allow the user to set the relationships between the variables. The user can indeed preset the number of factors, $q$ (which was done in our case as well), but this is not the same as setting the variable relationships in biclustering because the factor loadings matrix has become binary and row-stochastic and now represents the variable cluster memberships in this biclustering method (Chapter 3; Martella et al., 2008).

Although there are numerous benchmark data sets for clustering, these are not as pronounced for biclustering. Furthermore, the data sets must include observations that originate from a spectrum of values, regardless of variable (e.g., percentages), and the variables must possess natural groupings in order to allow for biclustering. The real data analyses looked at two different datasets containing fatty acid compositions
within olive oil and within various commercial oils, respectively, both with bad points added manually to introduce contamination. With the olive oil data, using the length of the carbon chains of the fatty acids as the variable cluster labels was a way to group the variables that made biological sense. The results indicated that OSCGaBi performed the best in retrieving the area that each sample originated from, based on the lengths of the carbon chains of the various fatty acids measured. OSGaBi performed next best according to the ARI, reinforcing the idea that the carbon-chain length could be used to cluster the variables, but that the bad points made it difficult for the algorithm to perform well. The two factor analyzer families, MMtFA and PGMM, and MCLUST did not perform as well, providing further evidence that the variable labels aid in retrieving the sample region.

With the commercial oil data, the carbon-chain length of the fatty acids was once again used to group the variables. ARI results indicated that MMtFA performed the best in retrieving the type of oil the sample was from. Although this was the case, it overestimated the number of components by more than two-fold, at times placing single observations into its own separate components. On the other hand, OSCGaBi performed slightly worse but accurately estimated the number of components. The other methods also decently estimated the number of components compared to MMtFA. The fact that OSCGaBi performed almost as well as MMtFA and additionally accurately estimated the number of components indicates that perhaps the variable cluster labels aid in guiding the algorithm to the correct number of components.

Comparisons were successfully made between OSCGaBi and various robust model-based methods, but it would have been optimal to include other robust methods that
are not model-based. Unfortunately, the various methods mentioned in Section 5.1 could not be tested mainly due to the fact that some of the software was not made publicly available. Furthermore, some of the algorithms allow for non-exhaustive and non-exclusive biclusters (as classified by Madeira and Oliveira, 2004; Pontes et al., 2015), which are not properties of the biclusters possible with OSCGaBi and the applications demonstrated herein. Finally, these algorithms do not allow for setting how the variables are related to each other (i.e., do not allow for setting the variable cluster memberships), which is a key property of the OSCGaBi algorithm.
Chapter 6

Conclusions

6.1 Summary

This thesis has proposed a parsimonious Gaussian family, a model averaging method, and a parsimonious contaminated Gaussian family that accommodates outliers, all within the biclustering framework. The work developed in this thesis represents contributions to the growing literature on model-based biclustering and its applications.

In Chapter 3, a family of parsimonious Gaussian mixture models for the biclustering of gene expression data is proposed. These models work in a one-way supervised fashion in that the variable labels are known. The binary and row-stochastic factor loadings matrix results in a block-diagonal covariance matrix, which can be a useful property in biclustering applications for dictating the relationships between the variables. A promising application for the method is in the discovery of novel peripheral blood biomarkers for use as surrogate biopsy material.

In Chapter 4, the idea of model averaging is introduced into the model-based
biclustering framework. The model averaging approach is established via the BMA framework, Occam’s window, and the BIC. Although model averaging has been proposed on a number of occasions in the model-based clustering framework, this is the first mention of applying the procedure in the biclustering context.

In Chapter 5, a one-way supervised contaminated Gaussian biclustering family is introduced for the handling of contamination in the biclustering paradigm. Simulated and real data studies demonstrate its superior performance compared to other available clustering and biclustering methods. Furthermore, results from the real data studies indicate that clustering results could improve if the algorithm is provided the variable cluster memberships and is more robust to outliers. This work sets the stage for further research in model-based methods for accommodating contaminated data during bicluster analyses.

6.2 Future Work

6.2.1 Benchmark Biclustering Data Sets

It is crucial to find and/or identify benchmark biclustering data sets that satisfy the characteristics required for the OSGaBi family and other biclustering methods (e.g., satisfy the exclusivity and exhaustivity properties of said method), or at least to develop solid, literature-supported classifications for existing data sets that have promise to be benchmark biclustering data sets. These discoveries will aid immensely in evaluating the performance of biclustering methods.
6.2.2 OSGaBi Performance

Future work should aim to compare performance of the OSGaBi family to that of other model-based biclustering algorithms capable of detecting non-overlapping clusters and allowing for one-way supervision. Current methods are available for the former (as mentioned previously in Chapter 3), but do not allow for the latter criteria. This limitation in the existing methods makes it difficult to compare the observations that are found in the biclusters to those found using the OSGaBi family because they do not always correspond to the intended subset of variables.

6.2.3 Model Selection

Simulation studies in Chapter 3 and 5 as well as real data comparisons in Chapter 5 indicate that investigation of alternatives to BIC for model selection might be warranted. Additionally, as stated earlier, the BIC has been shown to be unreliable in higher dimensions, suggesting that further research on an alternative model selection criteria for the OSGaBi family may prove beneficial.

6.2.4 Model Averaging

Although the focus in this thesis is the OSGaBi family of mixture models, the model averaging approaches presented here can be applied to other biclustering families and mixtures based on non-Gaussian densities. The model averaging approach is established via the BMA framework, Occam’s window, and the BIC, but other options can be investigated. This might lead to different sets of models to be averaged and also to different calculations for the weights of each contributing model.
6.2.5 Contaminated Data

Though considerable research has gone into model-based approaches for handling contaminated data in the clustering paradigm, the same cannot be said for within the biclustering paradigm. The work described in Chapter 5 sets the stage for further insight into one-way supervised biclustering involving contamination. Future work could involve exploring other distributions within the elliptical family besides the multivariate Gaussian distribution for modeling the contaminated data. Depending on the data, another direction could involve using one distribution to model the good points and a different distribution to model the bad points. For instance, previously in the clustering case, the multivariate Gaussian was used to model the good points and a multivariate uniform for the bad points (Banfield and Raftery, 1993). This model could be extended for biclustering.
Appendix A

AECM Algorithm Details for the OSCGaBi Family

The details of the AECM algorithm for the OSCGaBi family are derived similarly to that for the OSGaBi family from Chapter 3, thus only the constraints and key equations and parameter estimates (mainly for the second cycle) are provided here.

A.1 Model UCU

Constraint: $\Psi_g = \Psi$

$$l_{2e}(\vartheta) = C + \sum_{g=1}^{G} \left\{-\frac{n_g}{2} \log |\Psi| - \frac{n_g}{2} \text{tr} \left( \Psi^{-1} S^{(k+1)}_g \right) + \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) \left( x_i - \mu_g^{(k+1)} \right) ' \Psi^{-1} \Lambda_g u_{ig} ight. \right.$$  

$$\left. - \frac{1}{2} \text{tr} \left[ \Lambda_g ' \Psi^{-1} \Lambda_g \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) u_{ig} u_{ig}' \right] \right\}$$
\[ \beta_g^{(k)} = \Lambda_g^{(k)'} \left( \Lambda_g^{(k)} \Lambda_g^{(k)'} + \Psi^{(k)} \right)^{-1} \]

\[ Q_2(\vartheta_2) = C + \sum_{g=1}^{G} n_g^{(k+1/2)} \left\{ \frac{1}{2} \log |\Psi^{-1}| - \frac{1}{2} \text{tr} \left( \Psi^{-1} S_g^{(k+1)} \right) \right. \]
\[ + \text{tr} \left( \Psi^{-1} \Lambda_g \beta_g^{(k)} S_g^{(k+1)} \right) \left. - \frac{1}{2} \text{tr} \left[ \Lambda_g' \Psi^{-1} \Lambda_g \Theta_g^{(k+1/2)} \right] \right\} \]

\[ \Psi^{(k+1)} = \sum_{g=1}^{G} \hat{\pi}_g \text{diag} \left\{ S_g^{(k+1)} - \Lambda_g \beta_g^{(k)} S_g^{(k+1)} \right\} \]

### A.2 Model UUC

**Constraint:** \( \Psi_g = \psi_g I_d \)

\[ l_2c(\vartheta) = C + \sum_{g=1}^{G} \left\{ - \frac{n_g}{2} d \log \psi_g - \frac{n_g}{2} \psi_g^{-1} \text{tr} \left( S_g^{(k+1)} \right) \right. \]
\[ + \psi_g^{-1} \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_{(k+1)}} \right) \left( x_i - \mu_g^{(k+1)} \right)' \Lambda_g u_{ig} \]
\[ - \frac{1}{2} \psi_g^{-1} \text{tr} \left[ \Lambda_g' \Lambda_g \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_{(k+1)}} \right) u_{ig} u_{ig}' \right] \}

\[ \beta_g^{(k)} = \Lambda_g^{(k)'} \left( \Lambda_g^{(k)} \Lambda_g^{(k)'} + \psi_g I_d^{(k)} \right)^{-1} \]

\[ Q_2(\vartheta_2) = C + \sum_{g=1}^{G} n_g^{(k+1/2)} \left\{ d \log (\psi_g^{-1}) - \frac{1}{2} \psi_g^{-1} \text{tr} \left( S_g^{(k+1)} \right) \right. \]
\[ + \psi_g^{-1} \text{tr} \left( \Lambda_g \beta_g^{(k)} S_g^{(k+1)} \right) \left. - \frac{1}{2} \psi_g^{-1} \text{tr} \left[ \Lambda_g' \Lambda_g \Theta_g^{(k+1/2)} \right] \right\} \]

\[ \psi_g^{(k+1)} = \frac{1}{d} \text{tr} \left\{ S_g^{(k+1)} - \Lambda_g \beta_g^{(k)} S_g^{(k+1)} \right\} \]

81
A.3 Model UCC

Constraint: $\Psi = \psi I_d$

\[
 l_{2e}(\vartheta) = C + \sum_{g=1}^{G} \left\{ -\frac{n_g}{2} d \log \psi - \frac{n_g}{2} \psi^{-1} \text{tr} \left( S_g^{(k+1)} \right) \\
 + \psi^{-1} \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_{ig}^{(k+1)}} \right) \left( x_i - \mu_{g}^{(k+1)} \right)^t \Lambda_g u_{ig} \\
 - \frac{1}{2} \psi^{-1} \text{tr} \left[ \Lambda_g^t \Lambda_g \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_{ig}^{(k+1)}} \right) u_{ig} u_{ig}^t \right] \right\}
\]

\[
 \beta_g^{(k)} = \Lambda_g^{(k)^t} \left( \Lambda_g^{(k)} \Lambda_g^{(k)^t} + \psi I_d^{(k)} \right)^{-1}
\]

\[
 Q_2(\vartheta_2) = C + \sum_{g=1}^{G} n_g^{(k+1)/2} \left\{ \frac{d}{2} \log \left( \psi^{-1} \right) - \frac{1}{2} \psi^{-1} \text{tr} \left( S_g^{(k+1)} \right) \\
 + \psi^{-1} \text{tr} \left( \Lambda_g \beta_g^{(k)} S_g^{(k+1)} \right) - \frac{1}{2} \psi^{-1} \text{tr} \left[ \Lambda_g^t \Lambda_g \Theta_g^{(k+1)/2} \right] \right\}
\]

\[
 \hat{\psi}^{(k+1)} = \frac{1}{d} \sum_{g=1}^{G} \hat{\pi}_g \text{tr} \left\{ S_g^{(k+1)} - \Lambda_g \beta_g^{(k)} S_g^{(k+1)} \right\}
\]
A.4 Model CUU

Constraint: $\Lambda_g = \Lambda$

$$l_{2e}(\vartheta) = C + \sum_{g=1}^{G} \left\{ -\frac{n_g}{2} \log|\Psi_g| - \frac{n_g}{2} \text{tr} \left( \Psi_g^{-1} S_g^{(k+1)} \right) ight.$$  
$$+ \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) \left( x_i - \mu_g^{(k+1)} \right)' \Psi_g^{-1} \Lambda u_{ig}$$  
$$- \frac{1}{2} \text{tr} \left[ \Lambda' \Psi_g^{-1} \Lambda \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) u_{ig} u_{ig}' \right] \right\}$$

$$\beta_g^{(k)} = \Lambda_g^{(k)' \Lambda_g^{(k)}}^{-1} \left( \Lambda_g^{(k)} + \Psi_g^{(k)} \right)$$

$$Q_2(\vartheta_2) = C + \sum_{g=1}^{G} n_g^{(k+1)/2} \left\{ \frac{1}{2} \log|\Psi_g^{-1}| - \frac{1}{2} \text{tr} \left( \Psi_g^{-1} S_g^{(k+1)} \right) ight.$$  
$$+ \text{tr} \left( \Psi_g^{-1} \Lambda \beta_g^{(k)} S_g^{(k+1)} \right) - \frac{1}{2} \text{tr} \left[ \Lambda' \Psi_g^{-1} \Lambda \Theta_g^{(k+1)/2} \right] \right\}$$

$$\Psi_g^{(k+1)} = \text{diag} \left( S_g^{(k+1)} - \Lambda \beta_g^{(k)} S_g^{(k+1)} \right)$$
A.5 Model CCU

Constraints: \( \Lambda_g = \Lambda, \Psi_g = \Psi \)

\[
l_2(\theta) = C + \sum_{g=1}^{G} \left\{ -\frac{n_g}{2} \log |\Psi| - \frac{n_g}{2} \text{tr} \left( \Psi^{-1} S^{(k+1)} \right) + \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) \left( x_i - \mu_g^{(k+1)} \right) \right. \\
&\quad \left. - \frac{1}{2} \text{tr} \left[ \Lambda' \Psi^{-1} \Lambda \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) u_i u_i' \right] \right\}
\]

\[
\beta^{(k)} = \Lambda^{(k)} \left( \Lambda^{(k)} \Lambda^{(k)'} + \Psi^{(k)} \right)^{-1}
\]

\[
S^{(k+1)} = \sum_{g=1}^{G} \hat{\pi}_g S_g^{(k+1)}
\]

\[
Q_2(\theta_2) = C + \sum_{g=1}^{G} n_g^{(k+1)/2} \left\{ \frac{1}{2} \log |\Psi| - \frac{1}{2} \text{tr} \left( \Psi^{-1} S^{(k+1)} \right) + \text{tr} \left( \Psi^{-1} \Lambda \beta^{(k)} S^{(k+1)} \right) \right. \\
&\quad \left. - \frac{1}{2} \text{tr} \left[ \Lambda' \Psi^{-1} \Lambda \Theta^{(k+1/2)} \right] \right\}
\]

\[
\Theta^{(k+1/2)} = I_q - \beta^{(k)} \Lambda + \beta^{(k)} S^{(k+1)} \beta^{(k)}
\]

\[
\Psi^{(k+1)} = \sum_{g=1}^{G} \hat{\pi}_g \text{diag} \left\{ S^{(k+1)} - \Lambda \beta^{(k)} S^{(k+1)} \right\}
\]
A.6 Model CUC

Constraints: $\Lambda_g = \Lambda, \Psi_g = \psi_g I_d$

\[ l_{2c}(\vartheta) = C + \sum_{g=1}^{G} \left\{ -\frac{n_g}{2} d\log\psi_g - \frac{n_g}{2} \psi^{-1}_g \text{tr}\left( S_g^{(k+1)} \right) ight. \\
+ \psi^{-1}_g \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) \left( x_i - \mu^{(k+1)}_g \right)' \Lambda u_{ig} \\
- \frac{1}{2} \psi^{-1}_g \text{tr} \left[ \Lambda' \Lambda \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) u_{ig} u_{ig}' \right] \right\} \]

\[ \beta_g^{(k)} = \Lambda^{(k)'} \left( \Lambda^{(k)} \Lambda^{(k)'} + \psi_g I_d^{(k)} \right)^{-1} \]

\[ Q_2(\vartheta_2) = C + \sum_{g=1}^{G} n_g^{(k+1)/2} \left\{ d\log\left( \psi^{-1}_g \right) - \frac{1}{2} \psi^{-1}_g \text{tr}\left( S_g^{(k+1)} \right) ight. \\
+ \psi^{-1}_g \text{tr}\left( \Lambda \beta_g^{(k)} S_g^{(k+1)} \right) - \frac{1}{2} \psi^{-1}_g \text{tr}\left[ \Lambda' \Lambda \Theta_g^{(k+1)/2} \right] \right\} \]

\[ \psi_g^{(k+1)} = \frac{1}{d} \text{tr} \left\{ S_g^{(k+1)} - \Lambda \beta_g^{(k)} S_g^{(k+1)} \right\} \]
A.7 Model CCC

Constraints: $\Lambda_g = \Lambda, \Psi = \psi I_d$

$$l_{2c}(\varphi) = C + \sum_{g=1}^{G} \left\{ -\frac{n_g}{2} d \log \psi - \frac{n_g}{2} \psi^{-1} \text{tr} \left( S^{(k+1)} \right) ight. $$

$$+ \psi^{-1} \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) \left( x_i - \mu_g^{(k+1)} \right)^t \Lambda u_{ig}$$

$$\left. - \frac{1}{2} \psi^{-1} \text{tr} \left[ \Lambda' \Lambda \sum_{i=1}^{n} z_{ig} \left( v_{ig} + \frac{1 - v_{ig}}{\eta_g^{(k+1)}} \right) u_{ig} u_{ig}^t \right] \right\}$$

$$\beta^{(k)} = \Lambda^{(k)} \left( \Lambda^{(k)} \Lambda^{(k)}' + \psi I_d^{(k)} \right)^{-1}$$

$$S^{(k+1)} = \sum_{g=1}^{G} \hat{\pi}_g S_g^{(k+1)}$$

$$Q_2(\varphi_2) = C + \sum_{g=1}^{G} n_g^{(k+1/2)} \left\{ \frac{d}{2} \log\left( \psi^{-1} \right) - \frac{1}{2} \psi^{-1} \text{tr} \left( S^{(k+1)} \right) ight. $$

$$+ \psi^{-1} \text{tr} \left( \Lambda \beta^{(k)} S^{(k+1)} \right) - \frac{1}{2} \psi^{-1} \text{tr} \left[ \Lambda' \Lambda \Theta^{(k+1/2)} \right] \right\}$$

$$\Theta^{(k+1/2)} = I_q - \beta^{(k)} \Lambda + \beta^{(k)} S^{(k+1)} \beta^{(k)}$$

$$\psi^{(k+1)} = \frac{1}{d} \sum_{g=1}^{G} \hat{\pi}_g \text{tr} \left\{ S^{(k+1)} - \Lambda \beta^{(k)} S^{(k+1)} \right\}$$
Bibliography


Bohning, D., Dietz, E., Schaub, R., Schlattmann, P., and Lindsay, B. G. (1994). The distribution of the likelihood ratio for mixtures of densities from the one-parameter


Fraley, C., Raftery, A. E., Murphy, T. B., and Scrucca, L. (2012). *mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation.*


