ESTIMATION AND MODEL SELECTION FOR BLOCK CLUSTERING WITH MIXTURES: A COMPOSITE LIKELIHOOD APPROACH

A Dissertation in
Statistics
by
Prabhaní Kuruppumullage Don

© 2014 Prabhaní Kuruppumullage Don

Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

August 2014
The dissertation of Prabhani Kuruppumullage Don was reviewed and approved* by the following:

Bruce G. Lindsay
Willaman Professor of Statistics
Dissertation Co-adviser, Co-chair of Committee

Francesca Chiaromonte
Professor of Statistics and Public Health Sciences
Dissertation Co-advisor, Co-chair of Committee

David R. Hunter
Professor of Statistics and Department Head

Kateryna Markova
Professor of Biology

Aleksandra Slavkovic
Associate Professor of Statistics
Chair of Graduate Program

*Signatures are on file in the Graduate School.
Abstract

Clustering is the task of finding useful and meaningful groups in data, in a way that members within a group are more similar to each other than to members of other groups. There are many well established statistical methods that are used for clustering; among these mixture-based approaches have several advantages and have become increasingly popular.

In this thesis, we introduce a mixture-based approach for block clustering (i.e. simultaneous clustering of rows and columns of a data matrix). We discuss the computational challenges that prevent the use of traditional likelihood approaches in this setting, and provide an alternative. We build a composite likelihood that overcomes the computational burden, and devise a nested Expectation-Maximization (EM) algorithm to estimate the block mixture model.

Moreover, we develop two useful tools for model selection in block clustering. These can be used, in particular, to determine the number of row and column groups. We discuss how the gradient function can be used to assess the lack of fit of a block mixture model and provide an EM gradient search algorithm to progress towards better fitting models. Further, we develop a composite likelihood ratio test for comparing two block mixture models and incorporate it into a forward model selection method.

We then use our methods for two human genomics applications. In one we simultaneously cluster loci of enhanced microsatellite mutability and a large array of genomic features characterizing their environment. In the second, we do the same type of analysis for so-called common fragile sites in the genome.

Finally, we list some of the limitations of our methods, identifying challenges and discussing avenues for future developments.
# Table of Contents

List of Figures vii  
List of Tables viii  
Acknowledgments x  

## Chapter 1  Background and Motivation  
1. Early Bi-clustering Methods  
   1.1 Direct Clustering Algorithm  
   1.1.1 Formulation of the mixture model in one-way clustering  
   1.1.2 CEM construction for the block mixture model  
   1.1.3 Other developments  
1.2 Block Mixture Model  
1.3 Existing Methods to Estimate a Block Mixture Model  
   1.3.1 Block Classification EM (BCEM)  
   1.3.1.1 Variational approximation  
   1.3.1.2 Generalized EM algorithm  
   1.3.1.3 GEM construction for the block mixture model  
   1.3.2 Block EM (BEM)  
   1.3.2.1 Variational approximation  
   1.3.2.2 Generalized EM algorithm  
   1.3.2.3 GEM construction for the block mixture model  
   1.3.3 Block clustering with continuous data  
   1.3.4 Collapsed latent block models  

## Chapter 2  Composite Likelihood  
2. Definition and notation  
2.1 Early developments  
2.2 Other developments  
2.3 Asymptotic theory  
2.3.1 Standard error of \( \hat{\theta}_{CL} \)  
2.3.2 Composite likelihood EM algorithm  
2.3.3 Strategies in selecting a composite likelihood  

iv
6.2.2 Stepwise selection ......................................................... 87
6.3 Other directions .............................................................. 87
  6.3.1 Combining blocks for parsimony .................................... 87
  6.3.2 Multi-way arrays ....................................................... 88
  6.3.3 Other data types ....................................................... 88

Appendix A Maximum composite likelihood estimates of the block mixture model via nested EM .................................................. 89

Appendix B Maximum composite likelihood estimates via nested EM: the Gaussian data case ......................................................... 94
  B.1 Construction with missing data ........................................ 97

Appendix C Gradient function for the block mixture model .................. 99
  C.1 EM on block mixture gradient function ............................. 103

Bibliography ........................................................................... 108
List of Figures

1.1 Illustration of the data and the results of clustering .......................... 2
1.2 The block mixture model setting .................................................. 7

3.1 Variation of $LCL$ with every 5th EM iteration for example 1. The red curve in panel (a) shows the $LCL_t(\infty)$ and the black curve the values of $LCL_t$. The blue curve in panel (b) shows the variation of estimated asymptotic convergence rate $c_t$ with every 5th EM iterations for example 1. ........................................ 36
3.2 Variation of $LCL$ with every 5th EM iteration for example 2. In panel (a), the red curve shows the values of $LCL_t(\infty)$ and the black curve the values of $LCL_t$. The blue curve in panel (b) shows the variation of the estimated asymptotic convergence rate $c_t$ with every 5th EM iterations for example 2. .......................... 38
3.3 The block mean patterns used in the numerical study. ......................... 45
3.4 The block prevalence scenarios used in the numerical study. ................. 45
3.5 Sample block mean structure ...................................................... 46

4.1 Heatmaps for the simulated data .................................................. 62

5.1 Dendograms for clustering windows of enhanced microsatellite mutability (rows) and genomic landscape features (columns) using hierarchical clustering with complete linkage. ......................................................... 71
5.2 Heatmaps of the original and rearranged microsatellite data .................. 73
5.3 Enhanced microsatellite mutability windows along the human genome, color coded according to row clusters in the block clustering solution with $K_1 = 7$ and $K_2 = 4$. 75
5.4 Dendograms for clustering common fragile sites (rows) and genomic landscape features (columns) using hierarchical clustering with complete linkage. .................. 77
5.5 Heatmaps for the CFS data with 24 blocks ...................................... 79
5.6 Common fragile sites along the human genome, color coded according to row clusters in the block clustering solution with $K_1 = 8$ and $K_2 = 3$. ............. 81
5.7 Profiles of CFS clusters obtained using the block clustering solution with $K_1 = 8$ and $K_2 = 3$ .............................................................. 82
## List of Tables

3.1 Summary structure (i.e. true block means and block sizes) in example 1 ........... 35
3.2 Rearranged true mean structure for example 1 ........................................ 36
3.3 Rearranged estimated mean structure for example 1 ................................. 36
3.4 Summary structure in example 2 ................................................................. 37
3.5 Rearranged true mean structure for example 2 ........................................... 38
3.6 Rearranged estimated mean structure for example 2 ................................... 38
3.7 Contingency table of overlap between two cluster solutions ............................ 41
3.8 Different simulation designs ....................................................................... 46
3.9 Performance of our approach in the four basic simulation designs with $\sigma = 2$ ... 48
3.10 Estimated block means in the four basic simulation designs with $\sigma = 2$ ....... 48
3.11 Performance of our approach in the four basic simulation designs with $\sigma = 5$ ... 49
3.12 Estimated block means in the four basic simulation designs with $\sigma = 5$ ........ 49
3.13 Reorganized estimated std. errors for block means estimates in designs $(p_1, s_2)$ and $(p_2, s_1)$ with $\sigma = 5$ .......................................................... 49
3.14 Performance of our approach in the four basic simulation designs with $\sigma = 10$ ... 50
3.15 Estimated block means in the four basic simulation designs with $\sigma = 10$ ....... 50
3.16 Reorganized estimated std. errors for block means estimates in designs $(p_1, s_2)$ and $(p_2, s_1)$ with $\sigma = 10$ .......................................................... 51
3.17 Average adjusted RAND index in the four simulation designs with $\sigma = 2$, $\sigma = 5$ and $\sigma = 10$ ................................................................. 52
3.18 Comparing the performance of our labeling rule to the MAP rule in the four simulation designs, with $\sigma = 2$, $\sigma = 5$ and $\sigma = 10$ ................................. 52
3.19 Mean square error for block means estimates in the four simulation designs, with $\sigma = 2$, $\sigma = 5$, and $\sigma = 10$ ......................................................... 53

4.1 Rearranged true block mean structure for example 1 - same as Table 3.2 ........... 59
4.2 Rearranged estimated block mean structure for example 1 with $K_1 = 3$ and $K_2 = 3$ 59
4.3 Candidate initial values for the EM gradient search to assess the possibility of adding a column cluster to the solution with $K_1 = 3$ and $K_2 = 3$ in example 3.3.3.1 60
4.4 Rearranged true block mean structure for example 1 - same as Table 3.2 and 4.1 60
4.5 Rearranged estimated block mean structure for example 1 with $K_1 = 3$ and $K_2 = 4$ 60
4.6 True mean block structure (block sizes in parentheses) ............................... 62
4.7 Estimated mean block structure (standard errors in parentheses) for the model with $K_1 = 3$ and $K_2 = 2$ ......................................................... 62
4.8 Results of forward model selection using the CLRT ........................................ 63
4.9 Results for forward model selection using the CLRT for design $(p_1, s_1)$ when $\sigma = 2$ 64
4.10 Results for forward model selection using the CLRT for design $(p_1, s_1)$ when $\sigma = 5$ 64
4.11 Results for forward model selection using the CLRT for design \((p1, s1)\) when \(\sigma = 10\)
4.12 Results for forward model selection using the CLRT for design \((p1, s2)\) when \(\sigma = 2\)
4.13 Results for forward model selection using the CLRT for design \((p1, s2)\) when \(\sigma = 5\)
4.14 Results for forward model selection using the CLRT for design \((p1, s2)\) when \(\sigma = 10\)
4.15 Results for forward model selection using the CLRT for design \((p2, s1)\) when \(\sigma = 2\)
4.16 Results for forward model selection using the CLRT for design \((p2, s1)\) when \(\sigma = 5\)
4.17 Results for forward model selection using the CLRT for design \((p2, s1)\) when \(\sigma = 10\)
4.18 Results for forward model selection using the CLRT for design \((p2, s2)\) when \(\sigma = 2\)
4.19 Estimated block means in the design \((p2, s1)\) with \(\sigma = 10\)
4.20 Results for forward model selection using the CLRT for design \((p2, s2)\) when \(\sigma = 5\)
4.21 Results for forward model selection using the CLRT for design \((p2, s2)\) when \(\sigma = 10\)

5.1 Description of the 33 genomic landscape features in the microsatellite data
5.2 Results of CLR testing for the microsatellite data
5.3 Prevalences of window (row) clusters in the microsatellite data \((K_1 = 7)\)
5.4 Prevalences and memberships of feature (column) clusters in the microsatellite data \((K_2 = 4)\)
5.5 Description of the 49 genomic landscape features in the common fragile site data
5.6 Results of CLR testing for the common fragile sites data
5.7 Prevalences and memberships of feature (column) clusters in the common fragile sites data \((K_2 = 3)\)
5.8 Prevalences and memberships of site (row) clusters in the common fragile sites data \((K_1 = 8)\)

6.1 The increase in time to calculate CMLEs with increasing data size
6.2 Comparison of standard errors of MLEs and MCLEs
Acknowledgments

This doctoral thesis would never have been possible without the support and encouragement of the people around me.

First and foremost, I am forever indebted to my advisers Dr. Bruce G. Lindsay and Dr. Francesca Chiaromonte, for their advice, guidance, and enduring patience. Their direction and advice were key to the completion of this thesis and words cannot explain how fortunate I am to have them as mentors in my life. Despite their busy schedules, they always had time for me, patiently listened to my sometimes anxious self, and persuaded me to reach levels I never thought possible. They are indeed the wind beneath my wings and I can only hope that one day I can inspire someone half as much as they have done for me.

Next, I owe my deepest gratitude to Dr. Kateryna Makova for her continued guidance and support, Dr. David Hunter for his invaluable insights and expertise, Dr. Micheal Akritas and Dr. Murali Haran for their kind words and encouragement, the rest of the faculty at Penn State Statistics department who were always available to help throughout the last few years, and to Jenn, Melissa, Kathy and other staff members for their willing assistance and support.

I am also grateful to Malika Kumarasiri, Vindhya Panagoda, Primalie and Mahlon Rambukkange, Dr. Sarath Fernando, Sasie Fernando, and all other Sri Lankan families and friends around State College, for being there for me the during last few years, for constantly feeding me, and for making my stay in State College enjoyable and pleasant. I am undoubtedly thankful to Mr. George Henning, Mrs. Sue Henning, all my friends at Penn State; specially, Wendy, Guru, Chai and Lucia, for their constant help, and to many others for all the unforgettable memories.

Lastly, I am indebted to my parents Raveendra and Sumana Kuruppumullage, my loving brothers Rasantha and Nadun, and my best friend Priyangi for their unconditional love, endless support, and for being there for me at every single step of my life.
Background and Motivation

Cluster analysis is the identification of natural groups in data. Most clustering done in practice has focused on one-way clustering using \( \{x_1, \ldots, x_n\} \); that is grouping individuals \( \{i\} \) that are similar to each other based on vector measurements \( x_i \). One can think of clustering as creating a permutation of the index \( i \) so that observations naturally fall into a set of similar bins. Most of the early methods were ad hoc, with no principled way of comparing performances and clustering results, or to decide the number of clusters. Clustering based on probability models provides a solid alternative to these methods.

In model based clustering, the data \( \{x_1, \ldots, x_n\} \) are assumed to be generated from a mixture of underlying distributions in which each mixture component represents a different cluster [25, 3]. The two most common ways to formulate the problem are to use either the classification likelihood or the mixture likelihood. Given the data \( x = \{x_1, \ldots, x_n\} \), the classification likelihood approach maximizes

\[
L_C(\theta, \gamma|x) = \prod_{i=1}^{n} f_{\gamma_i}(x_i|\theta_{\gamma_i})
\]

where \( f_k(x_i|\theta_k) \) is the density of observation \( x_i \) from the \( k^{th} \) component, \( \theta = (\theta_1, \ldots, \theta_K) \) are the parameters of the \( K \) components, and \( \gamma = (\gamma_1, \ldots, \gamma_n) \) are the labels of the \( n \) observations. The mixture likelihood approach maximizes

\[
L_M(\theta, \pi|x) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_k(x_i|\theta_k)
\]

where \( \pi_k \geq 0 \) is the probability that an observation belongs to the \( k^{th} \) component, and \( \sum_{k=1}^{K} \pi_k = 1 \). More details on these approaches are introduced in sections 1.3.1 and 1.3.2.
In some instances clustering is also used as a feature selection tool in which measured features are grouped based on their similarity across a given set of observations ([27, 37]). However, with large amount of data arising from applications such as gene expression studies and text mining studies, there has been renewed interest in bi-clustering (also known as co-clustering) methods that are designed to group observations and features simultaneously ([15, 16, 24, 34, 45, 61, 74]).

Define a data matrix $X = \{x_{ij}\}$, where the index $i$ denotes the $i^{th}$ row (or unit) and the index $j$ denotes the $j^{th}$ column (or feature) measured. We could perform a one-way clustering in which we group together either the row vectors $x_i$ or the column vectors $x_j$. In bi-clustering we find groups of observations and associated features simultaneously. In other words, we make permutations of both rows and columns simultaneously to obtain a correspondence structure (see Figure 1.1).

The earliest bi-clustering method was proposed by Hartigan [35]; ever since, different ap-
proaches have been proposed by various scientists, many appearing in the data mining literature [16, 74, 20]. We discuss some of these methods briefly in section 1.1. However, many of the proposed methods are heuristically motivated; using probability models as we do in this thesis can offer a solid alternative with greater statistical power.

It is important to notice that while some existing methods allow overlaps of blocks (see for instance Busygin et al. [13]), in this thesis we restrict our focus to the case in which we seek “checkerboard” patterns in a two-way array. Specifically, we cluster the rows of a two-way array, viewed as units, into groups of similar rows, and columns, viewed as features, into groups of similar columns. We then sort the rows into row clusters and columns into column clusters, to obtain a two-way array of observations in blocks, where each block corresponds to a fixed row cluster and a fixed column cluster. We refer to this arrangement as “block clustering”, a special case of bi-clustering.

In one-way clustering, it is well known that model-based clustering by maximization of the mixture likelihood provides very good performance. Modern computing tools like the Expectation-Maximization (EM) algorithm [19] have made this approach computationally feasible. In addition, model-based clustering also provides a measure of uncertainty about a clustering solution [25]. Though one would be naturally interested in employing maximum likelihood also for block clustering, it can be shown that evaluating the likelihood is computationally infeasible for tables of any realistic magnitude. Approximations are therefore essential (see section 1.2).

There have been several attempts to use mixture models for block clustering [31, 32, 53, 33]. Govaert and Nadif [31] propose two main options to tackle the block clustering problem using the mixture model and EM algorithm; one is to maximize the classification (or complete) likelihood [10, 52]. The other is to use a variational approximation [10] of the likelihood. However, in both these approaches the authors employ alternating EM maximization; i.e. they fix the parameters corresponding to clustering one-way (say, rows) and maximize the likelihood with respect to the parameters corresponding to clustering the other-way (columns), and vice-versa (see sections 1.3.1 and 1.3.2).

In this thesis, we propose a different approach to tackle the block clustering problem using the mixture model and the EM algorithm. We introduce an alternative objective function based on composite likelihood theory [49, 66] and derive the necessary steps of a new two layer EM algorithm. We study the convergence properties of the algorithm using well-known techniques. We further develop two promising model selection tools that enhance the usability of mixture models for block clustering.

In the remainder of this chapter we review some of the existing bi-clustering methods, introduce the general formulation of the mixture model in the context of block clustering and explain
the difficulty in estimating the parameters using the EM algorithm. We then review the existing
techniques to tackle the problem.

1.1 Early Bi-clustering Methods

The challenges in summarizing and extracting information from enormous volumes of data has
ignited the recent interest in bi-clustering methods. In this section we briefly summarize two of
the early methods which were the basis for later developments. Busygin et al. [13] provides a
good survey on some of bi-clustering methods and shows how they are related to singular value
decomposition.

1.1.1 Direct Clustering Algorithm

This algorithm was proposed by Hartigan in 1972 [35], and utilizes the deviation of an observed
data matrix $X = \{x_{ij}\}_{R \times C}$ from the ideal model on a particular block structure $\{P_1, P_2, \ldots, P_K\}$. This deviation is measured by the sums of squares

$$SSQ = \sum_{k=1}^{K} \sum_{i,j \in P_k} (X_{ij} - \mu_k)^2,$$

where $\mu_k$ is the average value of $x_{ij}$ in the block $P_k$. The contribution from each block $P_k$
measures the quality of the block; the smaller the contribution the better the block. Notice that
when $K$ is the size of the data matrix (i.e. when $K = R \times C$) then $SSQ$ will be trivially zero.
To restrict estimation to non-trivial cases, Hartigan proposed to minimize $SSQ$ for a fixed value
of $K < (R \times C)$.

To reach a reasonable solution in practice, the author introduces a splitting algorithm that
considers a split of a block $P_k$ into two by either rows or columns; at the $K^{th}$ splitting step the
solution changes from $\{P_1, P_2, \ldots, P_k, \ldots, P_K\}$ to $\{P_1, P_2, \ldots, P_k', P_k'', \ldots, P_K\}$. At each step
the algorithm chooses the split that maximizes the SSQ reduction, and stops when the observed
SSQ reduction is less than expected. Hartigan (1972) [35] provides explicit formulas to calculate
the expected SSQ reduction.

1.1.2 Node-deletion Algorithm

This algorithm was introduced by Cheng and Church in 2000 [16] as a tool for studying gene
expression data. The algorithm finds partitions in data that have a low mean squared residue.
This is a measure of the coherence of the genes (or rows) and conditions (or columns) in the
partition. Unlike the previous method this algorithm allows blocks to overlap, which authors
believe is desirable in expression data analysis.
If \( X = \{x_{ij}\}_{R \times C} \) is a data matrix, and \( I \subset R \) and \( J \subset C \) are subsets of rows and columns, then the pair \((I, J)\) specifies a submatrix \( X_{IJ} \) where the mean squared residue score is defined as

\[
H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (x_{ij} - x_{iJ} - x_{Ij} + x_{IJ})^2
\]

where \( x_{iJ} = \frac{1}{|J|} \sum_{j \in J} x_{ij} \), \( x_{Ij} = \frac{1}{|I|} \sum_{i \in I} x_{ij} \), and \( x_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} x_{ij} \).

A submatrix \( X_{IJ} \) is called a \( \delta \)-bicluseter if \( H(I, J) \leq \delta \) for some \( \delta \geq 0 \). Notice that the above definition ensures that any single element is a “perfect” bi-cluster with zero \( H(I, J) \), which is uninteresting. Interesting biclusters hence should have large size and a low \( H \) score.

To find such bi-clusters, the authors introduce a node-deletion algorithm that starts with \( X_{IJ} = X \) and deletes rows and columns for which

\[
d(i) = \frac{1}{|I|} \sum_{j \in J} (x_{ij} - x_{iJ} - x_{Ij} + x_{IJ})^2 \quad \text{and} \quad d(j) = \frac{1}{|J|} \sum_{i \in I} (x_{ij} - x_{iJ} - x_{Ij} + x_{IJ})^2
\]

are above a certain threshold. They show that every such deletion decreases the mean squared residue, terminating when a \( \delta \)-bicluseter is obtained.

To ensure maximal \( \delta \)-bicusters, this phase is followed by a node-addition algorithm, which finds the rows and columns that can be added without increasing the mean squared residue.

### 1.1.3 Other developments

In addition to the methods described above (and their extensions), there are some other interesting approaches to the bi-clustering problem. Notably, some of these approaches are probabilistic. We summarize a few that are most closely related to our model in section 1.3. Here we summarize two other interesting probabilistic approaches.

Dhillion et al. [20] proposed a method that views a contingency table as an empirical joint probability distribution of two discrete random variables. They define the optimal co-clustering (a pair of maps from rows to row cluster and from columns to column clusters) to maximize the mutual information of the clustered random variables, subject to constraints on the number of row and column clusters. In other words, optimal co-clustering minimizes the difference between the mutual information of the original random variables and the mutual information of the clustered random variables.

Let \( A \) and \( B \) be two discrete random variables that take values in the sets \( \{a_1, a_2, \ldots, a_R\} \) and \( \{b_1, b_2, \ldots, b_C\} \) respectively, and let \( p(A, B) \) denote the joint probability distribution of \( A \) and \( B \). The authors think of \( p(A, B) \) as a \( R \times C \) matrix. The problem of interest is to cluster \( A \) into \( K_1 \) disjoint clusters and \( B \) into \( K_2 \) disjoint clusters, and the authors use mutual information \( I(A, B) \) [17] to measure the quality of the co-clusters.
The mutual information is the amount of information shared by the two variables and can be defined as
\[ I(A; B) = \sum_a \sum_b p(a, b) \log \frac{p(a, b)}{p(a)p(b)}. \]
The method therefore minimizes the loss in mutual information, i.e. \( \min_{\hat{A}, \hat{B}} I(A; B) - I(\hat{A}; \hat{B}) \).

Sheng et al. [62] tackle the bi-clustering problem in a Bayesian framework. In their work, they present a bi-clustering strategy that utilizes a simple frequency model for the expression pattern of a bicluster in discretized microarray data and adapts Gibbs sampling for parameter estimation. The authors show that their method not only unveils the biclusters but also represents the pattern of a bicluster as a probabilistic model described by the posterior frequencies.

Consider a dataset that contains \( R \) rows (or genes) and \( C \) columns (or conditions). The authors start by assuming that there is only a single bicluster in the data. The two vectors \( r = [r_1 \ r_2 \ldots \ r_R] \) and \( c = [c_1 \ c_2 \ldots \ c_C] \), where \( r_i \) and \( c_j \) are Bernoulli random variables, indicate whether the \( i^{th} \) row and the \( j^{th} \) column belong to the bicluster. They are called label vectors. Since the data are discretized, the authors use multinomial distributions to model it. They assume that the “background data” (the part of the data that does not belong to the bicluster) is generated by one single multinomial distribution. Further, they assume that the expression level is consistent across the genes of the bicluster for each selected condition, but may be different across conditions. Thus, they use a multinomial distribution to model the data under every condition (column) in the bicluster, assuming that multinomial distributions for different conditions are independent. They introduce prior distributions for the parameters in the model and work with Gibbs sampling to estimate them. Because the assumption of a single bi-cluster in the data is unrealistic, the authors then extend their work to enable detecting multiple multi-clusters in a data set.

### 1.2 Block Mixture Model

In this section we formulate the block mixture model that allows us to utilize mixtures in the block clustering setting.

Let the data matrix be defined by \( X = \{x_{ij} : i \in I \text{ and } j \in J\} \), where \( I \) is the set of \( R \) row indices and \( J \) is the set of \( C \) column indices. The goal of block clustering is to group the rows into \( K_1 \) clusters and the columns into \( K_2 \) clusters. In our block mixture model we will suppose that each row has a label \( A_i = a_i \) that is drawn independently from the density \( p(a) \) on the set \( \{1, 2, \ldots, K_1\} \). The columns will have labels \( B_j = b_j \) drawn independently from the density \( q(b) \) on \( \{1, 2, \ldots, K_2\} \). The rows having the same hidden label “a” will be viewed as a row cluster. Columns with a common value of “b” will be a column cluster. Our goal will therefore be to
infer the true labels.

If we assume that the row and column labels (i.e. \(a_i\)s and \(b_j\)s) are sampled from the joint density \(Pr(a, b) = Pr(a) \times Pr(b)\), the density of the matrix \(X\) is

\[
f(X) = \sum_a \sum_b f(X|a, b)Pr(a, b)
\]

**Figure 1.2.** The block mixture model setting

Conditional on the true labels \(a\) and \(b\), we propose that the observations in the array are independent with the \(ij^{th}\) observation having a density of the form \(k(x_{ij}; \mu(a_i, b_j))\). We will focus on the case where \(k(x_{ij}; \mu(a_i, b_j))\) is normal. In this case we can think of \(x_{ij}\) as being data from a two way analysis of variance (ANOVA), where \(a_i\) identifies the row factor and \(b_j\) the column factor. With \(a_i\) and \(b_j\) unknown, we have a hidden ANOVA. We use \(\Theta = \{\mu(m, n), p(m), q(n) : m = 1, \ldots, K_1 \text{ and } n = 1, \ldots, K_2\}\) to denote the vector of all parameters.

The unconditional density of \(X\) is

\[
f(X) = \sum_{a_1}^{K_1} \cdots \sum_{a_R}^{K_1} \sum_{b_1}^{K_2} \cdots \sum_{b_C}^{K_2} \left( \prod_{i=1}^{R} \prod_{j=1}^{C} k(x_{ij}; \mu(a_i, b_j)) \right) \left( \prod_{i=1}^{R} p(a_i) \right) \left( \prod_{j=1}^{C} q(b_j) \right). \tag{1.2.1}
\]

Notice that \(\mu = \{\mu(m, n) : m = 1, \ldots, K_1 \text{ and } n = 1, \ldots, K_2\}\) can be viewed as a summary of
the data matrix $X$ via the means of the hidden two-way ANOVA.

The problem with the density in equation (1.2.1) is the number of mathematical operations required to compute it. As it is given, we must calculate $K_1^R K_2^C$ separate summands, each requiring $(RC + R + C + 1)$ multiplications, then add them together. If any of $K_1, K_2, R,$ or $C$ are large, the computation will be enormous. This means that standard likelihood methodology, including the EM algorithm, cannot be applied directly.

Remark: The number of calculations in (1.2.1) can actually be reduced to \( \min(K_1^R, K_2^C) \) summands. We are therefore assuming that this number represents an intractable number of calculations.

Govaert and Nadif [31, 32, 33, 53] considered this same model. They proposed several strategies to tackle the computational problem. In one of their early methods they propose maximizing the complete data log-likelihood associated with the block mixture model, in effect treating the labels $a$ and $b$ as parameters. Later in 2005, they developed a variational approximation [10] of the likelihood in equation (1.2.1) ([32]). In both these approaches, they use an alternating EM. Section 1.3 will discuss their methods in more detail.

In this thesis, we propose a composite likelihood approximation [49, 66] to (1.2.1). From this we can build a single nested EM algorithm to maximize the composite likelihood. There are several advantages to using a composite likelihood approach. Chapter 2 presents a comprehensive discussion about composite likelihood theory.

1.3 Existing Methods to Estimate a Block Mixture Model

Here we summarize some of the existing techniques to tackle the computational issues in block mixture models.

1.3.1 Block Classification EM (BCEM)

Govaert and Nadif [31] proposed an EM algorithm that optimizes the classification likelihood to estimate the parameters in the block mixture model. Before reviewing the BCEM approach, we introduce some terminology and notations by looking at the one-way clustering problem.

1.3.1.1 Formulation of the mixture model in one-way clustering

Usually, we assume that $X = (X_1, \ldots, X_n)$ comes from a probability distribution with density

$$f(X_i; \alpha) = \sum_{k=1}^{g} p_k \varphi_k(X_i; \alpha_k),$$
where \( g \) is the number of mixture components. Then the density of the observed data \( X \) is

\[
f(X; \alpha) = \prod_{i=1}^{R} \sum_{k=1}^{g} p_k \varphi_k(X_i; \alpha_k).
\]

Govaert and Nadif (2003) [31] rewrite the density of the observed data as

\[
f(X; \alpha) = \sum_{z \in \mathcal{Z}} p(z) f(X|z; \alpha),
\]

where \( p(z) = \prod_{i=1}^{R} p_{z_i} \) and

\[
f(X|z; \alpha) = \prod_{i=1}^{R} \varphi_{z_i}(X_i; \alpha_{z_i}).
\]

Here, \( z = (z_1, \ldots, z_R) \) and \( z_i = (z_{i1}, \ldots, z_{ig}) \). Thus, \( z_i = k \Leftrightarrow z_{ik} = 1 \) and \( z_{is} = 0, \forall s \neq k \).

Let \( \mathcal{Z} \) denote the set of all the partitions of \( R \) observations into \( g \) clusters. The data \((X, z)\) = \(((X_1, z_1), \ldots, (X_R, z_R))\), which treats \( z_i \) as known, is called complete data.

Clustering with a mixture model can be viewed with two approaches: the mixture likelihood approach, and the classification (or complete-data) likelihood approach [31, 52].

The first approach estimates the parameters of the mixture and uses maximum posterior probabilities to derive partitions. It optimizes the log-likelihood of the observed data

\[
L(\alpha; X) = \sum_{i=1}^{R} \log \left( \sum_{k=1}^{g} p_k \varphi_k(X_i; \alpha_k) \right)
\]

using the EM algorithm, which iteratively maximizes the conditional expectation of the complete-data log-likelihood:

\[
L(\alpha; X, z) = \sum_{i=1}^{R} \sum_{k=1}^{g} z_{ik} \log(p_k \varphi_k(X_i; \alpha_k)).
\]

The conditional expectation of the complete-data log-likelihood given the current estimate of \( \alpha \), say \( \alpha^{(c)} \), is

\[
Q(\alpha; \alpha^{(c)}) = \sum_{i=1}^{R} \sum_{k=1}^{g} t_{ik}^{(c)} \{ \log p_k + \log \varphi_k(X_i; \alpha_k) \}
\]

where \( t_{ik}^{(c)} = \frac{p_k^{(c)} \varphi_k(X_i; \alpha_k^{(c)})}{\sum_{k'=1}^{g} p_{k'}^{(c)} \varphi_{k'}(X_i; \alpha_{k'}^{(c)})} \). The E-step of the algorithm will evaluate \( t_{ik}^{(c)} \) and the M-step will find \( \alpha \) by maximizing \( Q(\alpha; \alpha^{(c)}) \), at each iteration \( c \).

An alternative formulation was proposed by Symons (1981) [65], who treats the \( t_{ik} \)'s as unknown parameters. This corresponds to a likelihood, hereafter referred to as the classification likelihood. The classification likelihood approach estimates the partitions (labels) as additional parameters in the EM construction. The classification EM (CEM) converts the posterior proba-
bilities \( t_{ik} \) to a discrete classification before the M-step. So CEM proceeds by evaluating \( t_{ik}^{(c)} \) at each E-step, obtaining a discrete classification \( z_{ik}^{(c+1)} = \arg \max_{k=1 \ldots g} t_{ik}^{(c)} \), and computing the maximum likelihood estimates of \( \alpha \) at each M-step.

### 1.3.1.2 CEM construction for the block mixture model

In this approach, Govaert and Nadif use the classification log-likelihood associated with the block mixture model to construct an estimation algorithm to tackle the bi-clustering problem. The classification log-likelihood associated with the block mixture model given in equation (1.2.1) is

\[
\ell_{\text{class}}(\Theta, a, b) = \log \left\{ \left( \prod_{i=1}^{R} \prod_{j=1}^{C} k(x_{ij}; \mu(a_i, b_j)) \right) \left( \prod_{i=1}^{R} p(a_i) \right) \left( \prod_{j=1}^{C} q(b_j) \right) \right\} \tag{1.3.1}
\]

By letting \( a_{im} = 1 \) if the \( i^{th} \) observation belongs to cluster \( m \) and zero otherwise, and \( b_{jn} = 1 \) if the \( j^{th} \) feature belongs to cluster \( n \) and zero otherwise, we can re-write the classification log-likelihood in equation (1.3.1) as follows

\[
\ell_{\text{class}}(\Theta, a, b) = \sum_{i=1}^{R} \sum_{m=1}^{K_1} a_{im} \log p_m + \sum_{j=1}^{C} \sum_{n=1}^{K_2} b_{jn} \log q_n
\]

\[
+ \sum_{i=1}^{R} \sum_{m=1}^{K_1} \sum_{j=1}^{C} \sum_{n=1}^{K_2} a_{im} b_{jn} \log k(x_{ij}; \mu(m, n)).
\]

To maximize \( \ell_{\text{class}}(\Theta) \), the authors propose to maximize the classification log-likelihood in an alternating fashion; with \( b \) and \( q \) fixed and then with \( a \) and \( p \) fixed.

When \( b \) and \( q \) are fixed at \( b^{(0)} \) and \( q^{(0)} \), the classification log-likelihood can be written as

\[
\ell_{\text{class}}(\Theta, a, b) = \sum_{i=1}^{R} \sum_{m=1}^{K_1} a_{im} \log p_m + \sum_{j=1}^{C} \sum_{n=1}^{K_2} b_{jn}^{(0)} \log q_n^{(0)}
\]

\[
+ \sum_{i=1}^{R} \sum_{m=1}^{K_1} \sum_{n=1}^{K_2} \sum_{\{j : b_{jn}^{(0)} = 1\}} \log k(x_{ij}; \mu(m, n)).
\]

Assume that \( k(x_{ij}; \mu(m, n)) \) has a real sufficient statistic \( u_{in} = T(x_{ij}; \{j : b_{jn}^{(0)} = 1\}) \). For example, when \( k(x_{ij}; \mu(m, n)) = N(\mu(m, n), \sigma^2) \) with known \( \sigma^2 \) the sufficient statistic is \( u_{in} = \sum_{j=1}^{C} b_{jn}^{(0)} x_{ij} \), with density \( g(u_{in}; \mu(m, n)) \). Then

\[
\ell_{\text{class}}(\Theta, a, b) = \sum_{i=1}^{R} \sum_{m=1}^{K_1} a_{im} \log p_m + \sum_{j=1}^{C} \sum_{n=1}^{K_2} b_{jn}^{(0)} \log q_n^{(0)}
\]

\[
+ \sum_{i=1}^{R} \sum_{m=1}^{K_1} \sum_{n=1}^{K_2} \log(h(x_{ij}) g(u_{in}; \mu(m, n))).
\]
which can be re-arranged to have the form

$$\ell_{\text{class}}(\Theta, a) = \sum_{r=1}^{R} \sum_{m=1}^{K_1} a_{rm} \log(p_m g(u_i; \mu_m)) + g_0(X, q, b)$$

where $u_i = (u_{i1}, \ldots, u_{iK_2})$ and $\mu_m = (\mu(m, 1), \ldots, \mu(m, K_2))$. Therefore to maximize $\ell_{\text{class}}(\Theta, a)$.

we can maximize $\sum_{r=1}^{R} \sum_{m=1}^{K_1} a_{rm} \log(p_m g(u_i; \mu_m))$. Similarly, we can show that maximizing $\sum_{j=1}^{C} \sum_{n=1}^{K_2} b_{jn} \log(q_n g(v_j; \mu_n))$ is equivalent to maximizing $\ell_{\text{class}}(\Theta, b_{jn}(\cdot))$ when $a$ and $p$ are fixed, where $v_{jm} = T(x_{ij}; \{i : a_{im} = 1\})$ is a real sufficient statistic and $g(v_{jm}; \mu(m, n))$ is the density of $v_{jm}$.

Using the classification log-likelihood, the authors proposed a BCEM algorithm to perform block clustering [31, 33] that iterates the following steps until convergence:

- compute $a^{(c+1)}$ and $p^{(c+1)}$ using the CEM algorithm on the data $u_1, \ldots, u_R$
- compute $b^{(c+1)}$, $q^{(c+1)}$ and $\mu^{(c+1)}$ using the CEM algorithm on the data $v_1, \ldots, v_C$.

1.3.2 Block EM (BEM)

In 2005, Govaert and Nadif [32] introduced another approach, where they propose a generalized EM (GEM) algorithm using a variational approximation [10] to estimate the parameters of the block mixture model. In this approach they use the original mixture likelihood.

It is well known that the CEM does not converge to the maximum likelihood estimates of the parameters and yields inconsistent estimates. The differences are especially large when the mixture components are overlapping or have very different proportions [32]. Thus, in this alternative approach the authors use the GEM of Dempster et al. [19] to maximize a variational approximation of the likelihood. We will briefly review the ideas of variational approximation and GEM in the next sections.

1.3.2.1 Variational approximation

Variational methods have their roots in variational calculus [10, 58]. These methods do not provide any essential approximations, but are useful in finding approximate solutions by optimizing through a restricted range of functions. Even though not popular among statisticians, this is a useful tool in likelihood based inferences. In an EM algorithm, variational approximations are useful for approximating posterior densities by other more tractable densities; the approach relies on the Kullback-Leibler (KL) divergence [46].
To explain the idea of variational approximation, consider a simple mixture of Gaussians

\[ f(X; \Theta) = \sum_{k=1}^{g} p_k \mathcal{N}(X; \mu_k, \Sigma_k). \]

Let \( z = (z_1, \ldots, z_g) \) where \( z_k = \mathbb{I}(X \in k) \) and \( \sum_{k=1}^{g} z_k = 1 \). Then we have

\[ p(X, Z; \Theta) = p(Z; \Theta)p(X|Z, \Theta) \]

where \( p(z) = \prod_{k=1}^{K} \pi_k^{z_k} \) and \( p(X|z) = \prod_{k=1}^{K} [\mathcal{N}(X; \mu_k, \Sigma_k)]^{z_k} \). Also,

\[ p(X; \Theta) = \sum_{z} p(X, z; \Theta) = \sum_{z} p(z; \Theta)p(X|z, \Theta). \]

Given an arbitrary density \( q(z) \) we can write,

\[ \ell(\Theta; X) = \log p(X|\Theta) = \log \sum_{z} p(X, z; \Theta) = \mathcal{L}(q, \Theta) + \text{KL}(q||p) \]

where \( \mathcal{L}(q, \Theta) = \sum_{z} q(z) \log \left\{ \frac{p(X, z; \Theta)}{q(z)} \right\} \) and \( \text{KL}(q||p) = -\sum_{z} q(z) \log \left\{ \frac{p(z|X, \Theta)}{q(z)} \right\} \). Using non-negativity of the KL divergence we see that \( \mathcal{L}(q, \Theta) \) is a lower bound for \( \log p(X|\Theta) \). If we allow any possible choice of \( q(z) \) then the maximum lower bound occurs when \( q(z) = p(z|X, \Theta) \). If we can find a good surrogate density \( q \) we consider maximizing \( \mathcal{L}(q, \Theta) \) instead of \( \ell(\Theta; X) \).

We are interested in the case where the true posterior distribution is not tractable. Therefore we need to consider a restricted family of distributions \( q(z) \) and then find the form of \( q(z) \) that minimizes the KL divergence. One way to restrict the family of distributions is to assume that \( q(z) \) factorizes with respect to a partition in \( m \) disjoint groups within \( z \) \cite{10}; that is

\[ q(z) = \prod_{i=1}^{m} q_i(z_i). \]

This factorized form corresponds to the so-called mean field theory approximation in physics \cite{10, 43, 44}. This approximation is the basis for the construction in section 1.3.2.3

### 1.3.2.2 Generalized EM algorithm

In practice, the solution to the M-step often exists in closed form \cite{52}. But in instances where it does not, optimizing the conditional expectation of the complete-data log-likelihood \( \mathcal{Q}(\Theta, \Theta^{old}) \) over all \( \Theta \) can be infeasible. Dempster et al. \cite{19} proposed a GEM algorithm that can be used in such situations. The M-step finds \( \Theta^{new} \) such that

\[ \mathcal{Q}(\Theta^{new}, \Theta^{old}) \geq \mathcal{Q}(\Theta^{old}, \Theta^{old}) \]
It can be shown that the above condition is sufficient to ensure that

\[ L(\Theta^{new}, \Theta^{old}) \geq L(\Theta^{old}, \Theta^{old}). \]

### 1.3.2.3 GEM construction for the block mixture model

The complete-data log-likelihood for the block mixture model is given by

\[ \ell_{\text{complete}}(\Theta) = \sum_{i=1}^{R} \sum_{m=1}^{K_1} a_{im} \log p_m + \sum_{j=1}^{C} \sum_{n=1}^{K_2} b_{jn} \log q_n + \sum_{i=1}^{R} \sum_{m=1}^{C} \sum_{j=1}^{K_1} \sum_{n=1}^{K_2} a_{im} b_{jn} \log k(x_{ij}; \mu(m, n)). \]

Hence, the conditional expectation of the complete-data log-likelihood given the previous parameters is given by

\[ Q(\Theta, \Theta^{old}) = \sum_{i=1}^{R} \sum_{m=1}^{K_1} P(a_{im} = 1|x, \Theta^{old}) \log p_m + \sum_{j=1}^{C} \sum_{n=1}^{K_2} P(b_{jn} = 1|x, \Theta^{old}) \log q_n \]

\[ + \sum_{i=1}^{R} \sum_{m=1}^{C} \sum_{j=1}^{K_1} \sum_{n=1}^{K_2} P(a_{im} b_{jn} = 1|x, \Theta^{old}) \log k(x_{ij}; \mu(m, n)). \]

To avoid complications arising in determining \( P(a_{im} b_{jn} = 1|x, \Theta^{old}) \), Govaert and Nadif propose to use a variational approximation and replace exact computation of \( P(a_{im} b_{jn} = 1|x, \Theta^{old}) \) by \( P(a_{im} = 1|x, \Theta^{old}) \times P(b_{jn} = 1|x, \Theta^{old}) \). Therefore,

\[ Q'(\Theta, \Theta^{old}) = \sum_{i=1}^{R} \sum_{m=1}^{K_1} P(a_{im} = 1|x, \Theta^{old}) \log p_m + \sum_{j=1}^{C} \sum_{n=1}^{K_2} P(b_{jn} = 1|x, \Theta^{old}) \log q_n \]

\[ + \sum_{i=1}^{R} \sum_{m=1}^{C} \sum_{j=1}^{K_1} \sum_{n=1}^{K_2} P(a_{im} = 1|x, \Theta^{old}) P(b_{jn} = 1|x, \Theta^{old}) \log k(x_{ij}; \mu(m, n)). \]

Even though this approximation reduces the computational cost of the E-step dramatically, the M-step still does not provide closed from solutions. Therefore, the authors employ the GEM proposed by Dempster et al [19], in which they consider increasing \( Q'(\Theta, \Theta^{old}) \) at each step rather than maximizing over all \( \Theta \).

Now, assume the density function \( k(x_{ij}; \mu(m, n)) \) has a real sufficient statistic \( u_{im} \). If \( k(x_{ij}; \mu(m, n)) = N(\mu(m, n), \sigma^2) \) with known \( \sigma^2 \), then \( u_{im} = \sum_{j=1}^{C} P(b_{jn} = 1|x, \Theta^{old}) x_{ij} \). As a consequence, \( Q'(\Theta, \Theta^{old}) \) can be decomposed as follows:

\[ Q'(\Theta, \Theta^{old}) = Q'(\Theta, \Theta^{old}|d) + \sum_{j=1}^{C} \sum_{n=1}^{K_2} P(b_{jn} = 1|x, \Theta^{old}) \log q_n + h(u, d, q) \]

where \( d \) is the matrix defined by \( P(b_{jn} = 1|x, \Theta^{old}) \) and \( Q'(\Theta, \Theta^{old}|d) = \sum_{i} \sum_{m} P(a_{im} = 1|x, \Theta^{old}) \log \{p_{m} g_k(u; \alpha, d)\} \). \( g_k(u; \alpha, d) \) is the density of \( h(u, d, q) \) and does not depend on
the \( P(a_{im} = 1|x, \Theta^{old}) \)'s. Therefore, maximizing \( Q' (\Theta, \Theta^{old}|d) \) will guarantee an increase in \( Q' (\Theta, \Theta^{old}|d) \) when \( d \) is fixed. We can rearrange the terms similarly when \( c \), the matrix defined by \( P(a_{im} = 1|x, \Theta^{old}) \), is fixed.

The resulting alternating BEM algorithm iterates the following steps until convergence:

- compute \( c^{(t+1)}, p^{(t+1)}, \alpha^{(t+1/2)} \) using the data \( u_1, \ldots, u_R \), starting with \( c^{(t)}, p^{(t)}, \alpha^{(t)} \)
- compute \( d^{(t+1)}, q^{(t+1)}, \alpha^{(t+1)} \) using the data \( v_1, \ldots, v_C \), starting with \( d^{(t)}, q^{(t)}, \alpha^{(t+1/2)} \).

### 1.3.3 Block clustering with continuous data

Refer back to the density given in equation (1.2.1), i.e. the block mixture model by Govaert and Nadif. As explained in sections 1.3.1 and 1.3.2, the authors proposed two EM algorithms to obtain likelihood based estimates for this problem. The authors also illustrated the performance of their algorithms by applying them to problems with binary data and contingency tables.

The block mixture model and the proposed EM algorithms (BCEM and BEM) can be used also in the context of continuous data. However, in 2010 Govaert and Nadif proposed an alternative to the block mixture model in (1.2.1) for continuous data [54]. They referred to it as the parsimonious mixture model for co-clustering. In this section we review the details of this model.

Recall the notations introduced in section 1.2, where \( a \) denotes a partition of the observations \( (I) \) into \( K_1 \) clusters, and \( b \) a partition of the variables \( (J) \) into \( K_2 \) clusters. Consider the mixture model in which the \( b \)-labels are included as parameters, but not the \( a \)-labels. Then the density function is

\[
f(x_i; \Theta) = \sum_{m=1}^{K_1} p_m \varphi(x_i; b, \alpha).
\]

Assuming Gaussian data, we have

\[
\varphi(x_i; b, \alpha) = \prod_{j=1}^C \prod_{n=1}^{K_2} \left( \frac{1}{\sqrt{2\pi\sigma_{mn}^2}} e^{-\frac{1}{2\sigma_{mn}^2} (x_{ij} - \mu_{mn})^2} \right)^{b_{jn}}.
\]

Nadif and Govaert (2010) explain that the above model can be viewed as a block mixture model with Gaussian data and constraints on the \( K_1 \) mean vectors and \( K_1 \) variance matrices. For each component \( m \), the \( (C \times 1) \) mean vector \( \mu_m \) takes the form \( (\mu_{m1}, \ldots, \mu_{m1}, \ldots, \mu_{mK_2}, \ldots, \mu_{mK_2})^T \), where each \( \mu_{mn} \) is repeated \( b_n \) times. Recall that \( b_n = \sum_j b_{jn} \), the number of columns belonging to the \( n^{th} \) column cluster. Similarly, the variance matrix \( \Sigma_m \) will be a diagonal matrix of size \( (C \times C) \) given by \( \text{Diag}(\sigma_{m1}^2, \ldots, \sigma_{m1}^2, \ldots, \sigma_{mK_2}^2, \ldots, \sigma_{mK_2}^2) \), where each variance \( \sigma_{mn}^2 \) is repeated \( b_n \) times.
The row-classified log-likelihood associate with the observed data will be
\[
\sum_{i=1}^{R} \log \left\{ \sum_{m=1}^{K_i} p_m \varphi(x_i; b, \alpha) \right\}
\]
and the corresponding complete-data log-likelihood will be
\[
\sum_{i=1}^{R} \sum_{m=1}^{K_i} a_{im} \log \{ p_m \varphi(x_i; b, \alpha) \}.
\]

With Gaussian data the complete-data log-likelihood, up to the constant \(-\frac{RC}{2} \log(2\pi)\), takes the form
\[
\sum_{i=1}^{R} \sum_{m=1}^{K_i} a_{im} \log p_m - \frac{1}{2} \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{m=1}^{K_1} \sum_{n=1}^{K_2} a_{im} b_{jn} \left( \log \sigma_{mn}^2 + \frac{(x_{ij} - \mu_{mn})^2}{\sigma_{mn}^2} \right)
\]
and the generalized EM algorithm (see section 1.3.2.2) can be used to estimate its unknown parameters.

### 1.3.4 Collapsed latent block models

Wyse and Friel (2012) [73] extended the block mixture model proposed by Govaert and Nadif (2003) [31] to a Bayesian block mixture model. Van Dijk et al. [22] has also considered this idea, but a possible advantage of the method proposed by Wyse and Friel is that the number of row and column clusters do not need to be specified in advance.

Recall the notation used in section 1.3.1.2. We consider \(X = \{x_{ij} : i \in I \text{ and } j \in J\}\), with \(I = \{1, \ldots, R\}\) and \(J = \{1, \ldots, C\}\), and we aim to decompose the set \(I\) into \(K_1\) clusters and set \(J\) into \(K_2\) clusters. To obtain the classification likelihood Govaert and Nadif [31] introduced the indicators \(a_{im} = 1\) if the \(i^{th}\) row belongs to cluster \(m\), and \(b_{jn} = 1\) if the \(j^{th}\) column belongs to cluster \(n\). The classification log likelihood is
\[
\ell_{\text{class}}(\Theta, a, b) = \sum_{i=1}^{R} \sum_{m=1}^{K_i} a_{im} \log p_m + \sum_{j=1}^{C} \sum_{n=1}^{K_1} b_{jn} \log q_n
\]
\[
+ \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{m=1}^{K_2} \sum_{n=1}^{K_1} a_{im} b_{jn} \log k(x_{ij}; \mu(m, n)).
\]

The Baysian block mixture model is formed by taking prior densities on \(K_1, K_2, \mu, p, \text{ and } q\). Then the posterior of the number of clusters and latent cluster allocations will be
\[
\pi(K_1, K_2, a, b, p, q, \mu | X) \propto f(a|K_1, p) f(b|K_2, q) f(X|K_1, K_2, a, b, \mu) f(\mu|K_1, K_2)
\]
\[
\times f(p|K_1) f(q|K_2) f(K_1) f(K_2).
\]
The authors integrate out \( p, q \) and each \( \mu(a, b) \) analytically from the posterior by using conjugate priors. They refer to this as “collapsing”. Collapsing allows the authors to obtain the explicit marginal posterior \( \pi(K_1, K_2, a, b|X) \). They devise a Markov chain Monte Carlo (MCMC) sampler to sample from this posterior.

Wyse and Friel choose standard conjugate priors \( p|K_1 \sim \text{Dirichlet}(\alpha, \ldots, \alpha) \) and \( q|K_2 \sim \text{Dirichlet}(\beta, \ldots, \beta) \) and point out that the prior for \( \mu(m, n) \) would depend on the distribution assumed for the data. But the \( \mu(m, n) \)s are assumed to be independent. This gives the following posterior:

\[
\pi(K_1, K_2, a, b|X) \propto \pi(K_1) \pi(K_2) \Gamma\{\alpha K_1\} \prod_{m=1}^{K_1} \Gamma\{|m| + \alpha\} \frac{\Gamma\{\beta K_2\} \prod_{n=1}^{K_2} \Gamma\{|n| + \beta\}}{\Gamma\{\beta\}^{K_2} \Gamma\{C + \beta K_2\}} \prod_{m=1}^{K_1} \prod_{n=1}^{K_2} M_{mn}
\]

where \(|m|\) is the number of rows in cluster \( m \), \(|n|\) is the number of columns in cluster \( n \), and \( M_{mn} \) has following form:

\[
M_{mn} = \int \pi(\mu(m, n)) \prod_{i:a_i=m} \prod_{j:b_j=n} f(x_{ij}|\mu(m, n)) \, d\mu(m, n).
\]

The priors for the number of clusters, \( \pi(K_1) \) and \( \pi(K_2) \) are taken to be truncated Poisson(1) over the ranges \( 1, \ldots, K_1 \) and \( 1, \ldots, K_2 \). For Gaussian data, the authors assume

\[
x_{ij}|a_i = m, b_j = n \sim N(\mu(m, n), \sigma^2(m, n))
\]

and take the priors to be \( \mu(m, n) \sim N(\xi, \tau^2 \sigma^2(m, n)) \) where \( \sigma^2(m, n) \sim \text{Inverse-Gamma}(\delta/2, \gamma/2) \).

The proposed MCMC sampler comprises 3 different steps: a Gibbs update for the row and column label; reallocation of collections of rows and columns; and cluster addition/removal.

Notably, in this approach, one can observe the so-called label switching problem [64] independently for rows and columns. Out of the many methods that have been proposed to overcome the label switching problem, the authors adopt a method proposed by Nobile and Fearnside (2007) [57].
Chapter 2

Composite Likelihood

Maximum likelihood estimation (MLE) is a useful statistical approach that provides estimators with powerful properties in many situations. However, in some practical instances, computing MLEs is computationally infeasible, either due to the structure of the model or to the dimensionality of the data. The model introduced in section 1.2 is one such instance. In these cases, various likelihood modifications have been developed by several authors. In this chapter we introduce composite likelihood, one such likelihood modification proposed by Lindsay (1988) [49].

A composite likelihood is a likelihood-type object obtained by multiplying a collection of component conditional or marginal likelihoods (i.e. adding component log-likelihoods) [49, 66]. Varin et al. (2011) [66] explain that the main motivation for using any version of composite likelihood is usually computational. In section 1.2 we explained why computing the true full likelihood in the mixture based block clustering setup could be computationally infeasible. The main goal of the first part of this thesis is to propose a composite likelihood modification to the true full likelihood to overcome the computational burden.

In this chapter we introduce composite likelihood in general and review some of the existing work based on composite likelihood.

2.1 Definition and notation

Consider a \( d \)-dimensional random vector \( Y = (Y_1, \ldots, Y_d)^T \), with probability density function \( f(y; \theta) \) depending on some unknown \( p \)-dimensional real parameter vector \( \theta \in \Theta \). Let \( L(\theta; y) = f(y; \theta) \) be the likelihood function, \( \ell(\theta) \) the log-likelihood function, \( S(\theta) = \nabla\ell \) the associated score function, and \( H(\theta) = \nabla^2\ell \) the associated hessian matrix, where \( \nabla \) denotes differentiation with respect to \( \theta \).
If \( \{A_1, \ldots, A_K \} \) denotes a set of marginal or conditional events with likelihoods \( L_k(\theta; y) \propto f(y \in A_k; \theta) \), then the composite likelihood is

\[
CL(\theta) = \prod_{k=1}^{K} L_k(\theta; y)
\]

Accordingly, the composite log-likelihood is

\[
c\ell(\theta) = \sum_{k=1}^{K} \log L_k(\theta; y)
\]

and the composite score function is

\[
CS(\theta) = \nabla c\ell(\theta) = \sum_{k=1}^{K} S_k(\theta)
\]

In the next few sections we review some of the early and recent work on composite likelihood methods. We start by reviewing the work of Lindsay (1988).

### 2.2 Early developments

Lindsay (1988) [49] proposed the concept of composite likelihood as a generalization of the concept of pseudolikelihood proposed by Besag (1974) [6] and the concept of partial likelihood proposed by Cox (1975) [18]. Even though the main focus of Lindsays’ work is on the efficiency of the estimates, the author also demonstrates the flexibility and the strength of the composite likelihood approach in constructing consistent estimates in complicated situations.

Consider an unbiased estimating function, \( g \); that is, a \( p \)-dimensional random vector \( g(\theta; y) = (g_1(\theta), \ldots, g_p(\theta))^T \) with a finite second moment \( E[gg^T] \) that satisfies \( E(g(\theta_0)) = 0 \) for the true parameter \( \theta_0 \). The Godambe information [30] in \( g \) is defined by

\[
I_g(\theta) = E[\nabla g(\theta)]\{Var[g(\theta)]\}^{-1}\{E[\nabla g(\theta)]\}^T
\]

where \( \nabla g(\theta) \) is a \( p \times p \) matrix \( [\nabla g_1(\theta), \ldots, \nabla g_p(\theta)] \). Lindsay [49] produced an alternative form for the information in \( g \), namely

\[
I_g(\theta) = E[S_g^T]V_g^{-1}E[gS^T].
\]

where \( V_g \) is the covariance matrix of \( g \) and \( S = \nabla \log f(y; \theta) \), and showed that attaining full information (or Fisher information) can be associated with a linear relationship between \( S \) and \( g \).
We know that the Kullback Leibler information [46] inequality holds for each component log likelihood and therefore also for the composite log likelihood; that is

\[ E_{\theta_0}[\ell_k(\theta)] \leq E_{\theta_0}[\ell_k(\theta_0)] \Rightarrow \sup_{\theta} E_{\theta_0}[c\ell(\theta)] \leq E_{\theta_0}[c\ell(\theta_0)]. \]

This leads to the conclusion that maximizing the composite log-likelihood should lead to a consistent method of estimation. Recall that the component likelihood \( L_k(\theta; y) \) is a true likelihood (conditional or marginal) and under regularity, \( E_{\theta}[S_k(\theta)] = 0 \). This gives \( E_{\theta}[CS(\theta)] = 0 \), and hence the composite score function is an unbiased estimating function. Therefore the maximum composite likelihood estimator \( \hat{\theta}_{CL} \) would be consistent for the same \( \theta \) as the one we would obtain by using the complete model.

Moreover, information in \( S_k(\theta) \) satisfies \( I = \text{Var}[S_k(\theta)] = E[-\nabla S_k(\theta)] \) and therefore the information in \( CS \) satisfies,

\[ I_{opt} = \left( \sum_{k=1}^{K} \text{Var}[S_k(\theta)] \right)^{-1} \left( \sum_{k=1}^{K} \text{Var}[S_k(\theta)] \right). \]

Notice that here the scores \( S_k(\theta) \) can be correlated with each other. This distinguishes the approach from the partial likelihood approach proposed by Cox (1975) [18].

Lindsay (1988) demonstrates that composite likelihood methods are fully efficient only at restricted parameter values. However at parameter values where the \( CS \) is not fully efficient, it may be possible to improve the efficiency by using a weighting schemes for the component scores.

Consider the problem of maximizing information over the class of estimating functions \( \{ \omega : \omega = \sum_{k=1}^{K} \omega_k S_k(\theta) \} \). This is equivalent to the least squares problem

\[ \min_{\omega} E_{\theta} \{(S(\theta) - \omega^T U)^2\} \]

where \( \omega = (\omega_1(\theta), \ldots, \omega_K(\theta))^T \) and \( U = (S_1(\theta), \ldots, S_K(\theta))^T \). Solving this, the best weights are

\[ \omega_{opt} = \{\text{Var} U\}^{-1} E[U] \]

and the information achieved under optimal weighting is,

\[ I_{opt} = v \{\text{Var} U\}^{-1} v \]

where \( v \) is the vector with \( k^{th} \) coordinate \( E[S_k^2(\theta)] = E[S(\theta) S_k(\theta)] \). In the ideal case, where no
weighting needs to be done, we have, \( \omega_{\text{opt}} = m \mathbf{1} \) for some constant \( m \). This implies

\[
m \sum_{j=1}^{K} \frac{\text{cov}[S_k(\theta), S_j(\theta)]}{\text{Var}[S_k(\theta)]} = 1 \quad \text{for all } k.
\]

Comparison of information matrices over a class of linear combinations of estimating functions leads \( I_{\text{opt}} \) to be strongly optimal by dominating every other information matrix in the sense of positive definiteness. However, finding optimal weights may not be computationally feasible, even in the scalar parameter case. The work of Lindsay [49] led to a more general definition of composite likelihood given by

\[
CL_{\omega}(\theta) = \prod_{k=1}^{K} L_{\omega_k}(\theta; y)
\]

where \( \omega_k, k = 1, \ldots, K \) are arbitrary, fixed, non-negative weights. When all \( \omega_k = 1 \), we obtain \( CL(\theta) \).

### 2.3 Other developments

Varin et al. (2011) [66] summarize many ideas related to composite likelihoods, with a wide range of applications. The authors explain that a composite likelihood can be seen as a misspecified likelihood, because of the working independence assumption among its components. Since the second Bartlett identity \([4, 5], E(\nabla^2 \ell) + (E(\nabla \ell))^2 = 0\), does not hold, it is important for one to distinguish between the \textit{sensitivity matrix}

\[
H(\theta) = E_\theta\{-\nabla_\theta CS(\theta)\}
\]

and the \textit{variability matrix}

\[
J(\theta) = \text{var}_\theta\{CS(\theta)\}.
\]

Therefore the \textit{Fisher information} matrix needs to be substituted by the \textit{Godambe information} matrix (also called the \textit{sandwich information} matrix) defined as:

\[
G(\theta) = H(\theta)J(\theta)^{-1}H^T(\theta).
\]

If \( c\ell(\theta) \) was a true log-likelihood, we would have \( G = H = J \).
2.3.1 Asymptotic theory

Considering \( n \) independent and identically distributed observations \( Y_1, \ldots, Y_n \) from \( f(y; \theta) \) on \( \mathbb{R}^d \) we have

\[
CL(\theta; y) = \prod_{i=1}^{n} CL(\theta; y_i) \quad \text{and} \quad c\ell(\theta; y) = \sum_{i=1}^{n} c\ell(\theta; y_i)
\]

Under regularity conditions on the composite log-likelihood and with large \( n \), the maximum composite likelihood estimator \( \hat{\theta}_{CL} \) is asymptotically normally distributed:

\[
\sqrt{n}(\hat{\theta}_{CL} - \theta) \overset{d}{\rightarrow} N_d\left(0, G^{-1}(\theta)\right)
\]

where \( N_d(\cdot, \cdot) \) is the \( d \)-dimensional normal distribution and \( G(\theta) \) is the Godambe information matrix in a single observation.

The ratio of \( G(\theta) \) to the expected Fisher information, \( I(\theta) \) determines the asymptotic efficiency of \( \hat{\theta}_{CL} \) relative to the maximum likelihood estimator from the full model.

Even though we do not present the details here, Varin et al. (2011) [66] summarize the composite likelihood versions of Wald, score and likelihood ratio statistics together with their advantages and disadvantages.

2.3.2 Model selection

Composite likelihood versions of the Akaike information criterion (\( AIC_{\ell} \)) [67] and the Bayesian information criterion (\( BIC_{\ell} \)) [28] have been derived and have the forms,

\[
AIC_{\ell} = -2c\ell(\hat{\theta}_{CL}; y) + 2\text{dim}(\theta)
\]

and

\[
BIC_{\ell} = -2c\ell(\hat{\theta}_{CL}; y) + \text{dim}(\theta) \log(n)
\]

where \( \text{dim}(\theta) = \text{tr}\{H(\theta)G^{-1}(\theta)\} \), the effective number of parameters. Like their counterparts for standard likelihood, these criteria can provide useful guidance in model selection.

2.3.3 Computational aspects

In this section we present some of the useful developments of composite likelihood theory related to computation.
2.3.3.1 Standard error of $\hat{\theta}_{CL}$

In their survey, Varin et al. (2011) [66] discuss the issue of standard error calculations. Consider $n$ independent and identically distributed observations $Y_1, \ldots, Y_n$ from $f(y; \theta)$ on $\mathbb{R}^d$. With fixed $d$ and large $n$, the sample estimate of the sensitivity matrix is given by

$$\hat{H}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \nabla S(\hat{\theta}_{CL}; y_i)$$

where $S(\theta; y_i) = \nabla \ell(\theta; y_i)$. Since each component is a valid likelihood, the second Bartlett identity will hold and the sample estimate of the variability matrix is

$$\hat{J}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} S(\hat{\theta}_{CL}; y_i) S(\hat{\theta}_{CL}; y_i)^T.$$

If the sample size $n$ is not sufficiently large relative to $d$, the dimension of $\theta$, the above empirical estimates might not be precise. A more robust estimate of the covariance matrix of $\hat{\theta}_{CL}$ is obtained by the jackknife method and is given by

$$\text{Var}_{\text{jack}}(\hat{\theta}_{CL}) = \frac{n-1}{n} \sum_{i=1}^{n} \left( \hat{\theta}_{CL}^{(-i)} - \hat{\theta}_{CL} \right) \left( \hat{\theta}_{CL}^{(-i)} - \hat{\theta}_{CL} \right)^T,$$

where $\hat{\theta}_{CL}^{(-i)}$ is the maximum composite likelihood estimator of $\theta$ with $y_i$ deleted.

2.3.3.2 Composite likelihood EM algorithm

Varin et al. (2011) [66] discuss how the EM algorithm can be extended to composite likelihoods and how it will be useful in problems with models where expectation involves high-dimensional integration. The pairwise EM algorithm can be devised as follows.

Let $x_1, \ldots, x_n$ be the complete data and $y_1, \ldots, y_n$ be the observed data, then for each iteration $h$ the pairwise EM algorithm finds $\theta^{(h+1)}$ such that,

$$Q \left( \theta^{(h+1)} | \theta^{(h)} \right) \geq Q \left( \theta | \theta^{(h)} \right) \text{ for any } \theta \in \Theta$$

where

$$Q \left( \theta | \theta^{(h)} \right) = \sum_{r=1}^{n-1} \sum_{s=r+1}^{n} E \left\{ \log f(x_r, x_s; \theta) | y_r, y_s; \theta^{(h-1)} \right\}.$$  

Gao and Song (2011) [29] proved the ascent property, algorithmic convergence, and convergence rate for this EM construction.
2.3.4 Strategies in selecting a composite likelihood

We have already seen that, although composite likelihood methods possess some important computational and statistical features, they can be less statistically efficient compared to traditional likelihood methods. Lindsay et al. (2011) [48] argued that, there is no universal rule for constructing a composite likelihood that is both computationally convenient and statistically appealing. The authors proposed some ideas that can be useful in balancing the trade-off between the efficiency and computational cost when formulating a composite likelihood.

Recall that the composite likelihood is

$$CL(\theta) = \prod_{k=1}^{K} L_k(\theta; y),$$

where $$L_k(\theta; y) \propto f(y \in A_k; \theta)$$ for some marginal or conditional events $$\{A_1, \ldots, A_K\}$$. The computational cost largely depends on the number of factors in $$CL(\theta)$$, $$K$$, and the cost of calculating each $$L_k(\theta; y)$$. Let $$D_k$$ be the dimension of each $$L_k(\theta; y)$$ and $$N_{ops}(D)$$ be the number of computer operations needed to calculate a distribution of $$D$$ variables, then the overall number of computations needed to calculate the $$CL(\theta)$$ is

$$K \times N_{ops}(D^*)$$

where $$D^*$$ is the largest data dimension among the $$A_k$$. Lindsay et al. [48] suggest that in designing composite likelihood setups, one faces a clear trade-off in controlling $$D^*$$ and $$K$$. To study the design issue, the authors mainly compare composite likelihoods based on the following events:

- one-wise marginals (i.e. $$A_k$$ is a single variable $$y_i$$)
- pairwise marginals (i.e. $$A_k = (y_i, y_j)$$)
- pairwise conditionals (i.e. $$A_k = y_i | y_j$$).

The authors also show that the theory of optimally weighted estimating equations described in section 2.2 has a limited usefulness in balancing the trade-off between computational cost and statistical efficiency of composite likelihood methods.

To help build a better strategy the authors introduce a more general framework; additive estimating functions. They use Hoeffding additive scores [48, 38] to find the optimal additive estimating function. They propose a strategy that works well for parameter values that are closer to the independence parameter values (i.e. any value of parameters for which all variable components $$Y_i$$ are independent). To provide a general guideline, the authors move the focus from optimality to reliability, and propose adding information unbiased scores as a practical strategy - that is, combining estimating functions, $$g_i$$, that satisfy $$E(g_i g_i^T) \geq 0$$ to construct a composite likelihood.
Estimating the Block Mixture Model

We have seen that evaluating the mixture likelihood for the block clustering problem is computationally infeasible, and as discussed in chapter 1 there are existing approaches to tackle this problem. In this thesis we propose an alternative procedure based on composite likelihood theory. In particular, this chapter is devoted to creating a composite likelihood as an alternative to the full likelihood, and devising an EM algorithm to estimate the model parameters.

3.1 Composite likelihood for the block mixture model

In this section we explain the construction of the composite likelihood for the block mixture model. First, from (1.2.1) notice that each row vector of $x_{i*} = (x_{i1}, \ldots, x_{iC})$ of $X$ has the same marginal density. In particular, let $g(x_{ij}; a, Q)$ be the the density of the $(i, j)^{th}$ cell with known row label $a$, then

$$g(x_{ij}; a, Q) = \sum_{b=1}^{K_2} k(x_{ij}; \mu(a, b))q(b).$$

This density has the same computational difficulty as a standard mixture density. The marginal density of the $i^{th}$ row with known row label $a$ will then be

$$g(x_{i*}; a, Q) = \prod_{j=1}^{C} g(x_{ij}; a, Q)$$

which has the structure of an i.i.d likelihood from a mixture. Finally, we can obtain the marginal density of the $i^{th}$ row as

$$g(x_{i*}; P, Q) = \sum_{a=1}^{K_1} p(a)g(x_{i*}; a, Q)$$
which is a simple sum over \( a \) from the previous step. In all, the computations are similar to a standard mixture model.

Now to create a composite likelihood alternative to the full likelihood, we pretend the rows are independent and obtain the row composite likelihood,

\[
L_R(\Theta) = \prod_{i=1}^{R} g(x_i^*; P, Q) = \prod_{i=1}^{R} \left[ \sum_{a=1}^{K_1} p(a) \left( \prod_{j=1}^{C} \sum_{b=1}^{K_2} q(b) k(x_{ij}; \mu(a, b)) \right) \right].
\]

This composite likelihood treats the rows of the data as if they were independent. Clearly this method of construction throws away information about the parameters that might be available in the dependence relationships among rows (these are generated by having common column labels). To compensate for this, we also create a column composite likelihood

\[
L_C(\Theta) = \prod_{j=1}^{C} \left[ \sum_{b=1}^{K_2} q(b) \left( \prod_{i=1}^{R} \sum_{a=1}^{K_1} p(a) k(x_{ij}; \mu(a, b)) \right) \right].
\]

At this point we could combine the two composite likelihoods using weights. However as a start we simply combine the row and column composite likelihoods without any weighting, to obtain the full composite likelihood

\[
L_{RC}(\Theta) = L_R(\Theta) L_C(\Theta).
\]

A possible extension of the results here would be to consider a weighting system. The log full composite likelihood is

\[
\log L_{RC}(\Theta) = \sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} p(a) \left( \prod_{j=1}^{C} \sum_{b=1}^{K_2} q(b) k(x_{ij}; \mu(a, b), \sigma^2) \right) \right\} + \sum_{j=1}^{C} \log \left\{ \sum_{b=1}^{K_2} q(b) \left( \prod_{i=1}^{R} \sum_{a=1}^{K_1} p(a) k(x_{ij}; \mu(a, b), \sigma^2) \right) \right\} \quad (3.1.1)
\]

Notice that the computations no longer increase exponentially with the dimensions of \( X \) (i.e. \( R \) and \( C \)), but rather vary multiplicatively.

Now we have a composite likelihood that can be calculated quickly even on very large two-way data arrays. The remainder of this thesis focuses on resolving the usual statistical issues such as estimation, labeling, and model selection in the model based block clustering setting.
3.2 EM algorithm

The EM algorithm is generally used to obtain maximum likelihood estimates (MLEs) in incomplete data problems [52, 19]. It is an iterative process with two steps: an expectation step (E-step), and a maximization step (M-step).

3.2.1 The traditional view

Consider a random vector $Y$ corresponding to the observed data $y$ from $g(y; \theta)$ where $\theta = (\theta_1, \ldots, \theta_p)^T$ is a vector of unknown parameters. In the EM-context $y$ is considered as incomplete data. Let $z$ denotes the missing data and $x = (y, z)$ be the complete data. If $x$ were fully observed, then to obtain the MLE of $\theta$ one would use the complete-data log likelihood function $\log L_c(\theta) = \log (g_c(x; \theta))$. But since $z$ is unobservable, an EM algorithm iteratively replaces $\log L_c(\theta)$ by its conditional expectation given $y$ at the current value of $\theta$. So for some initial value $\theta^{(0)}$ the E-step of the first iteration calculates

$$Q(\theta; \theta^{(0)}) = E_{\theta^{(0)}}[\log L_c(\theta)|y].$$

The M-step maximizes $Q(\theta; \theta^{(0)})$ with respect to $\theta$ and selects $\theta^{(1)}$ such that $Q(\theta; \theta^{(1)}) \geq Q(\theta; \theta^{(0)})$. One repeats the E-step and M-step until convergence; commonly defined by the difference of $\ell(\theta^{(t+1)}) - \ell(\theta^{(t)})$, where $\ell(\theta^{(t)}) = \log g(y; \theta^{(t)})$, becoming satisfactorily small.

Dempster et al. [19] showed that the incomplete data likelihood $L(\theta) = g(y; \theta)$ is guaranteed to not decrease after an EM iteration; that is

$$L(\theta^{(k+1)}) \geq L(\theta^{(k)}).$$

This ensures the monotonicity of an EM algorithm [52]. In section 1.3.2.2 we briefly introduced the idea of generalized EM proposed by Dempster et al. [19], an extension of EM in which the solution to the M-step does not exist in closed form. It is important to notice that the generalized EM inherits the monotonicity property.

3.2.2 A novel view

The traditional view of EM traps it usage in a maximum likelihood with missing data paradigm [41]. In a working paper, Hunter et al. [41] provide a new insight to EM that broadens its scope to maximizing any real valued objective function with a product-of-sums structure. The key to their work is to recognize that EM is a special case of Minorization-Maximization (MM) algorithm [40], a general purpose tool for optimization.
Consider a real valued objective function $S(\theta)$ of the form

$$S(\theta) = \prod_{a=1}^{A} S_a(\theta),$$

in which each $S_a(\theta)$ can be written as $S_a(\theta) = \sum_{b=1}^{B_a} s_{ab}(\theta)$, where $s_{ab}(\theta)$ is a non negative function for all $a$ and $b$. Notice that $S(\theta)$ is of the form product-of-sums.

Now consider optimizing $s(\theta)$ where

$$s(\theta) = \log S(\theta) = \sum_{a=1}^{A} \log \left( \sum_{b=1}^{B_a} s_{ab}(\theta) \right).$$

$s(\theta)$ is of the form sum-of-log-of-sums. The EM construction of Hunter et al. proceeds by focusing on the expression $\log S_a(\theta)$ and creating a surrogate function utilizing the Jensen’s inequality as follows:

$$\log S_a(\theta) - \log S_a(\theta^{(t)}) = \log \left( \sum_{b=1}^{B_a} s_{ab}(\theta) \right) - \log \left( \sum_{b=1}^{B_a} s_{ab}(\theta^{(t)}) \right)$$

$$= \log \left( \frac{\sum_{b=1}^{B_a} s_{ab}(\theta)}{\sum_{b=1}^{B_a} s_{ab}(\theta^{(t)})} \right)$$

$$= \log \left( \frac{\sum_{b=1}^{B_a} s_{ab}(\theta)}{\sum_{b=1}^{B_a} s_{ab}(\theta^{(t)})} \right)$$

$$= \log \left( \frac{\sum_{b=1}^{B_a} \omega_{ab}^{(t)} s_{ab}(\theta)}{\sum_{b=1}^{B_a} \omega_{ab}^{(t)} s_{ab}(\theta^{(t)})} \right)$$

where

$$\omega_{ab}^{(t)} = \frac{s_{ab}(\theta^{(t)})}{\sum_{b=1}^{B_a} s_{ab}(\theta^{(t)})}$$

and $\theta^{(t)}$ is the value of $\theta$ at the $t^{th}$ iteration.

Notice that since the non-negative weights $\omega_{ab}^{(t)}$ satisfy $\sum_{b=1}^{B_a} \omega_{ab}^{(t)} = 1$, we can use Jensen’s inequality to obtain

$$\log S_a(\theta) - \log S_a(\theta^{(t)}) \geq \sum_{b=1}^{B_a} \omega_{ab}^{(t)} \log \left( \frac{s_{ab}(\theta)}{s_{ab}(\theta^{(t)})} \right).$$

The authors [41] conclude their representation by obtaining the EM likelihood, $Q(\theta; \theta^{(t)})$, as

$$Q(\theta; \theta^{(t)}) = \sum_{a=1}^{A} \sum_{b=1}^{B_a} \omega_{ab}^{(t)} \log S_{ab}(\theta),$$
and showing that \( s(\theta) - s(\theta(t)) \geq Q(\theta; \theta(t)) - Q(\theta(t); \theta(t)) \). The authors point out that if \( S(\theta) \) is a likelihood, then calculating \( Q(\theta; \theta(t)) \) is the same as obtaining the conditional expectation. Thus, the E-step in this novel representation is equivalent to calculating the weights in \( Q(\theta; \theta(t)) \), whereas maximizing \( Q(\theta; \theta(t)) \) gives the M-step.

### 3.2.3 Convergence

EM algorithm is very popular for its numerical stability, computational simplicity and guaranteed monotone ascent property. The main weakness of the algorithm is that it converges linearly (i.e. the order of convergence is linear) and often with a very slow rate [12, 59].

**Definition 1.** (Order of Convergence) A sequence \( \{\theta^p\}_{p=0,1,...} \) in \( \mathcal{R}^d \) that converges to \( \hat{\theta} \in \mathcal{R}^d \) is said to exhibit convergence of order \( \beta \) if

\[
\lim_{p \to \infty} \frac{||\theta^{p+1} - \hat{\theta}||}{||\theta^p - \hat{\theta}||^\beta} = c
\]

for some constant \( 0 \leq c < 1 \). The sequence is said to converge linearly if \( \beta = 1 \), quadratically if \( \beta = 2 \), and superlinearly if \( \beta = 1 \) and \( c = 0 \). Here \( || \cdot || \) is a norm in \( \mathcal{R}^d \).

**Definition 2.** (Rate of Convergence) The rate of convergence of a sequence \( \{\theta^p\}_{p=0,1,...} \) that converges linearly to \( \hat{\theta} \) is measured by a \( q \)-factor or \( \tau \)-factor defined as

\[
q(\{\theta^p\}) = \limsup_{p \to \infty} \frac{||\theta^{p+1} - \hat{\theta}||}{||\theta^p - \hat{\theta}||^\beta}
\]

and

\[
\tau(\{\theta^p\}) = \limsup_{p \to \infty} ||\theta^{p+1} - \hat{\theta}||^{frac{1}{\beta}}
\]

respectively.

As a consequence of slow linear convergence, the EM algorithm can require many iterations to estimate parameters with reasonable accuracy, and accelerating techniques are useful. Also, defining a stopping rule that guarantees reasonable accuracy is challenging. The most common stopping rule is \( |\ell_{t+1} - \ell_t| \leq \epsilon \), where \( \ell_t \) is the EM likelihood calculated at \( t^{th} \) iteration and \( \epsilon \) is a small constant. This stopping rule captures the idea of lack of progress rather than actual numerical accuracy.

Aitken acceleration is a commonly used technique for accelerating EM algorithms [59]. In 1994, Böhning et al. [1, 12, 60] discussed how Aitken acceleration can be used on the log-likelihood estimates and also proposed a useful stopping rule for the EM algorithm using Aitken acceleration.
3.2.4 Aitken $\delta^2$ Method

In this section we briefly discuss the idea of the Aitken acceleration method. Recall definition 1, and that the $\{\theta^p\}_{p=0,1,\ldots}$ has a order of convergence $\beta$ with asymptotic rate $c$. We know that if $\beta = 1$ and $0 < c < 1$ then the sequence is said to be linearly convergent. Notice that larger values of $\beta$ imply faster convergence.

Suppose the sequence $\{e_p\} = (\theta^p - \hat{\theta})$ is converging linearly to $\hat{\theta}$ such that

$$[e_p - \hat{\theta}] \cong \lambda[e_{p-1} - \hat{\theta}]$$

or equivalently

$$[e_p - e_{p-1}] \cong (1 - \lambda)[\hat{\theta} - e_{p-1}]$$

where $\cong$ means $\lim_{p \to \infty} \frac{|e_p|}{|e_{p-1}|^{\beta}} = c$.

If the constant $c$ can be estimated by $\tilde{c}_p$, when $p$ is sufficiently large, one can predict $\hat{\theta}$ as,

$$\hat{\theta} \cong e_{p-1} + \frac{1}{1 - \tilde{c}_p}[e_p - e_{p-1}]$$

for $p > 2$.

Now define the backward difference $\Delta e_p = (e_p - e_{p-1})$ for $p > 1$. Let higher powers be recursively defined by $\Delta^k e_p = \Delta^{k-1}(\Delta e_p)$ for $k > 2$. Then, $\Delta^2 e_p = \Delta(\Delta e_p - \Delta e_{p-1}) = (e_p - 2e_{p-1} + e_{p-2})$ for $p > 2$, and we can estimate $\hat{\theta}_p$ as

$$\hat{\theta}_p = \frac{\Delta e_p}{\Delta e_{p-1}}$$

for $p > 2$.

The Aitken $\delta^2$ estimator of $\hat{\theta}$ can be obtained as follows:

$$\hat{\theta}^\delta_p = e_{p-1} - \frac{\Delta e_p \Delta e_{p-1}}{\Delta^2 e_p}$$

for $p > 2$.

According to the Aitken Acceleration Theorem [60], the Aitken sequence $\hat{\theta}^\delta_p$ will converge to $\hat{\theta}$ faster (in a smaller number of iterations) than $e_p$.

3.2.5 Aitken acceleration on log-likelihood estimates

As mentioned earlier, Böhning et al. [12] explored the possibility of using Aitken acceleration on the log-likelihood estimates, and developed a technique to predict the value of the log likelihood at the maximum likelihood solution. They showed that their approach enables them to predict the final maximum likelihood solution with fewer EM iterations. In this section we will briefly review this approach.
Suppose we have an arbitrary log-likelihood sequence $\ell_i$ converging linearly to $\hat{\ell}$. Then we have

$$\ell_{i+1} - \hat{\ell} \cong c(\ell_i - \hat{\ell}) \text{ for all } i \text{'s and some } c, 0 < c < 1.$$

(3.2.1)

By re-arranging terms we get

$$\ell_{i+1} - \ell_i \cong (1 - c)(\hat{\ell} - \ell_i).$$

From equation (3.2.1) we have $\ell_{i+1} - \ell_i \cong c(\ell_i - \ell_{i-1})$ for all $i$ and hence

$$\ell_{i+1} - \ell_i \cong c^i(\ell_1 - \ell_0).$$

Therefore we obtain the geometric series

$$\hat{\ell} = \lim_{i \to \infty} \ell_i = \ell_0 + \left(\sum_{i=0}^{\infty} c^i\right)(\ell_1 - \ell_0) = \ell_0 + \frac{1}{1 - c}(\ell_1 - \ell_0).$$

(3.2.2)

where the unknown $c$ can be estimated by

$$c_i = \frac{\ell_{i+1} - \ell_i}{\ell_i - \ell_{i-1}}.$$

(3.2.3)

The Aitken accelerated estimate of $\hat{\ell}$ is

$$\ell_i^\infty = \ell_{i-1} + \frac{1}{1 - c_i}(\ell_i - \ell_{i-1}).$$

(3.2.4)

Notice that their acceleration device is applicable to any log-likelihood sequence with linear convergence.

Lindsay [50] explained how the predicted final log-likelihood can be used to construct a stopping rule for the EM; one that, unlike the commonly used stopping rule introduced in section 3.2.3, captures the idea of numerical accuracy. The new stopping rule is

Stop EM if $0 < (\ell_i^\infty - \ell_i) < \epsilon$,

where $\epsilon$ is a more meaningful measure of the actual numerical accuracy attained. The author further analyzed the tolerance value ($\epsilon$) and explained that if one sets a smaller tolerance than 0.005, one is pursuing a numerical accuracy in the parameter estimates that is minor relative to the magnitude of their likelihood confidence intervals [50].
3.3 Estimation via “nested” EM

To obtain the parameter estimates for the block mixture model through composite likelihood (see section 3.1), we need to maximize \( \log L_{RC}(\Theta) \). In this section we discuss how we devise an EM algorithm to achieve this goal. The objective function to be optimized, the log full composite likelihood (see equation (3.1.1)) is

\[
\log L_{RC}(\Theta) = \sum_{i=1}^{R} \log \left( \sum_{a=1}^{K_1} \omega_i(a) \frac{p(a)g(x_{i*}; a, Q)}{p_0(a)g(x_{i*}; a, Q_0)} \right) + \sum_{j=1}^{C} \log \left( \sum_{b=1}^{K_2} q(b)k(x_{ij}; \mu(a, b), \sigma^2) \right) \]

Following the novel view of EM by Hunter et al. ([41]), we think of the EM algorithm as a general optimization tool that replaces any “log of sums” in an objective function (e.g. \( \log \left( \sum_{b=1}^{B} S_b \right) \)) with “sums of log” to create a simpler objective function (e.g. \( \sum_{b=1}^{B} \omega_b \log S_b \)). We do this using Jensens’ inequality with special weights \( \omega_b = \frac{S_0^b}{\sum_{b=1}^{B} S_0^b} \), where \( S_0^b \) is the current value of \( S_b \).

In our case for simplicity, consider \( \log L_R(\Theta) \); there are two summations to “slip the logarithm inside” and we do this in two steps. Let \( \Theta_0 \) be the current value of \( \Theta \). The Jensens’ trick on \( \log \left\{ \frac{L_R(\Theta)}{L_R(\Theta_0)} \right\} \) gives

\[
\sum_{i=1}^{R} \log \left( \sum_{a=1}^{K_1} \omega_i(a) \frac{p(a)g(x_{i*}; a, Q)}{p_0(a)g(x_{i*}; a, Q_0)} \right) \geq \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_i(a) \log \left( \frac{p(a)g(x_{i*}; a, Q)}{p_0(a)g(x_{i*}; a, Q_0)} \right)
\]

where

\[
g(x_{i*}; a, Q) = \prod_{j=1}^{C} \sum_{b=1}^{K_2} k(x_{ij}; \mu(a, b))q(b) \quad \text{and} \quad \omega_i(a) = \frac{p_0(a)g(x_{i*}; a, Q_0)}{\sum_{a=1}^{K_1} p_0(a)g(x_{i*}; a, Q_0)}.
\]

Maximizing the right hand side over \( \Theta \) necessarily gives a non-negative number, and a \( \Theta \) that gives a positive number will also cause \( L_R(\Theta) > L_R(\Theta_0) \). Therefore the right hand side of (3.3.1) gives an EM likelihood, which we call the outer EM:

\[
\ell_{R, em} = \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_i(a) \log p(a) + \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_i(a) \left\{ \sum_{b=1}^{K_2} q(b)k(x_{ij}; \mu(a, b)) \right\}.
\]
If we find a value of $\Theta$ that maximizes $\ell_{R,em_1}$, we know that the same $\Theta$ increases the likelihood over $\Theta_0$. Notice that the parameters $p(a)$ separate out, and we can obtain an explicit EM step for $p(\cdot)$. However, this outer EM is still complicated to maximize for $q(\cdot)$ and $\mu(\cdot, \cdot)$’s. Therefore, we propose to add an *inner EM* step to decompose the expression $\ell_2(\Theta)$ in $\ell_{R,em_1}$, utilizing another standard EM step on $\ell_2(\Theta)$. By combining the outer and *inner EM* steps we obtain a “nested” EM likelihood, $\ell_{R,em_2}$, that maximizes $\log L_R(\Theta)$:

$$\ell_{R,em_2} = \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_i(a) \log p(a) + \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \omega_{ij}^*(a, b) \log [k(x_{ij}; \mu(a, b))q(b)]$$

where

$$\omega_{ij}(a, b) = \frac{k(x_{ij}; \mu_0(a, b))q_0(b)}{\sum_{b=1}^{K_2} k(x_{ij}; \mu_0(a, b))q_0(b)}.$$

In a similar fashion, we can create a nested EM representation for the column composite likelihood, $L_C(\Theta)$ and obtain an EM likelihood $\ell_{C,em_2}$:

$$\ell_{C,em_2} = \sum_{j=1}^{C} \sum_{b=1}^{K_2} \psi_j(b) \log q(b) + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \sum_{a=1}^{K_1} \psi_{ij}^*(a, b) \log [k(x_{ij}; \mu(a, b))p(a)]$$

where

$$\psi_j(b) = \left\{ \frac{q_0(b)h_0(x_{ij}, [b: \Theta_0])}{\sum_{a=1}^{K_1} q_0(b)h_0(x_{ij}, [b: \Theta_0])} \right\}, \quad \psi_{ij}(a, b) = \left\{ \frac{k(x_{ij}; \mu_0(a, b))p_0(a)}{\sum_{a=1}^{K_1} k(x_{ij}; \mu_0(a, b))p_0(a)} \right\}, \quad \text{and} \quad \psi_{ij}^*(a, b) = \psi_{ij}(a, b)$$

Finally, to maximize the full composite likelihood $L_{RC}(\Theta)$, we sequentially maximize

$$\ell_{RC,em_2} = \ell_{R,em_2} + \ell_{C,em_2}. \quad (3.3.2)$$

**Proposition 3.3.1.** If $\Theta$ gives a positive value to $\ell_{RC,em_2}$, then $L_{RC}(\Theta) > L_{RC}(\Theta_0)$.

Notice that so far we have not assumed any form for $k(x_{ij}; \mu(a, b))$; in section 3.3.2 we show that assuming $k(x_{ij}; \mu(a, b)) = \mathcal{N}(x_{ij}; \mu(a, b), \sigma^2)$ we can obtain closed form solutions for $p(\cdot)$, $q(\cdot)$, $\mu(\cdot, \cdot)$, and $\sigma^2$. The complete proof of the construction is given in appendix A.

Assuming that the number of row clusters ($K_1$) and the number of column clusters ($K_2$) are known, the general nested EM algorithm can be summarized as follows.

**Pre-step:** To create initial values for the algorithm, perform separate clustering of row and columns with pre-specified $K_1$ and $K_2$ to obtain row labels and column labels. This can be done, for example, by using K-means [36]. This gives each $x_{ij}$ a label $(a_i, b_j)$. 
Next define the pseudo EM weights to be as:

\[
\omega_{ij}^{(0)}(a, b) = \begin{cases} 
1 & \text{if } (a_i = a \text{ and } b_j = b) \\
0 & \text{o/w}
\end{cases}
\]

\[
\omega_i^{(0)}(a) = \begin{cases} 
1 & \text{if } (a_i = a) \\
0 & \text{o/w}
\end{cases}
\]

\[
\psi_{ij}^{(0)}(a, b) = \begin{cases} 
1 & \text{if } (a_i = a \text{ and } b_j = b) \\
0 & \text{o/w}
\end{cases}
\]

\[
\psi_j^{(0)}(b) = \begin{cases} 
1 & \text{if } (b_j = b) \\
0 & \text{o/w}
\end{cases}
\]

Step 1: Using these weights and the EM formula (3.3.2), obtain \( p^{(0)}(a), q^{(0)}(b), \) and \( \mu^{(0)}(a, b) \) for \( a = 1, \ldots, K_1 \) and \( b = 1, \ldots, K_2 \).

Step 2: Calculate the log composite likelihood, \( LCL_0 = \log L_{RC}(\Theta^{(0)}) \).

Step 3: Obtain new EM weights, \( \omega_{ij}^{(t)}(a, b), \psi_{ij}^{(t)}(a, b), \omega_i^{(t)}(a), \) and \( \psi_j^{(t)}(b) \), for the rows and columns.

Step 4: Obtain \( p^{(t)}(a), q^{(t)}(b), \) and \( \mu^{(t)}(a, b) \), and calculate \( LCL_t = \log L_{RC}(\Theta^{(t)}) \).

Step 5: Iterate steps 3 and 4 for \( t = 1, 2, \ldots \) until convergence (see below).

### 3.3.1 Convergence of the nested EM algorithm

As discussed in section 3.2.3, one of the weaknesses of EM algorithm is its often slow linear convergence. An EM can require many iterations to estimate parameters with reasonable accuracy, and acceleration techniques are useful. Also, ensuring that one has reached reasonable accuracy is difficult [12, 59].

Following Böhning et al. [12] we use Aitken acceleration on the logs of the composite likelihoods. Let \( LCL_1, LCL_2, \ldots, LCL_t \) be the log composite likelihood values. We predict the final log composite likelihood value \( LCL_t(\infty) \) based on these values and the geometric convergence of the series:

\[
LCL_t(\infty) = LCL_{t-1} + \frac{1}{1-c_t}(LCL_t - LCL_{t-1})
\]

(3.3.3)

where \( c_t = \frac{LCL_{t+1} - LCL_t}{LCL_t - LCL_{t-1}} \). This method is most reliable after the algorithm has settled into stable iterations with nearly constant values of the estimated rate constant \( c_t \).

Following Lindsay (1995) [50], we then employ the following stopping criterion with \( \epsilon = 0.005 \):

\[
\text{Stop EM if } 0 < LCL_t(\infty) - LCL_t \leq \epsilon.
\]
3.3.2 Gaussian data

The nested EM discussed in section 3.3 is applicable for estimating a block mixture model irrespective of the form of \(k(x_{ij}; \mu(a, b))\), the density of data conditional on true row and columns labels. In this section, and throughout the remainder of this thesis, we will assume Gaussian data; that is

\[
k(x_{ij}; \mu(a, b)) = \mathcal{N}(x_{ij}; \mu(a, b), \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} (x_{ij} - \mu(a, b))^2 \right\}.
\]

We perform the M-step of the nested EM by maximizing \(\ell_{RC,em,2}\), and with Gaussian data we obtain the following closed form solutions to update the parameter estimates of the block mixture model:

\[
p^{(t)}(a) = \frac{\sum_{i=1}^{R} \omega^{(t)}_i (a) + \sum_{j=1}^{C} \sum_{k=1}^{K_2} \psi^{(t)}_{ij} (a, b)}{R + RC} \tag{3.3.4}
\]

\[
q^{(t)}(b) = \frac{\sum_{j=1}^{C} \psi^{(t)}_j (b) + \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \omega^{(t)}_{ij} (a, b)}{C + RC} \tag{3.3.5}
\]

\[
\mu^{(t)}(a, b) = \frac{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \sum_{b=1}^{K_2} \left\{ \omega^{(t)}_{ij} (a, b) + \psi^{(t)}_{ij} (a, b) \right\} x_{ij}}{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \sum_{b=1}^{K_2} \left\{ \omega^{(t)}_{ij} (a, b) + \psi^{(t)}_{ij} (a, b) \right\}} \tag{3.3.6}
\]

The complete derivations are given in appendix B. In addition, when \(k(x_{ij}; \mu(a, b)) = \mathcal{N}(x_{ij}; \mu(a, b), \sigma^2)\) we update the estimate of the common variance \(\sigma^2\) as follows:

\[
\sigma^{2(t)} = \frac{\sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \left\{ \omega^{(t)}_{ij} (a, b) + \psi^{(t)}_{ij} (a, b) \right\} (x_{ij} - \mu(a, b))^2}{2RC} \tag{3.3.7}
\]

Recall that \(\omega^{(t)}_{ij} (a, b) = \omega^{(t)}_i (a) \times \omega^{(t)}_{ij} (a, b)\) and \(\psi^{(t)}_{ij} (a, b) = \psi^{(t)}_i (b) \times \psi^{(t)}_{ij} (a, b)\) where

\[
\omega^{(t)}_i (a) = \frac{p^{(t-1)}_0 (a) g_0(x_{ia} | \Theta_0^{(t-1)})}{\sum_{a=1}^{K_1} p^{(t-1)}_0 (a) g_0(x_{ia} | \Theta_0^{(t-1)})} \quad \text{and} \quad \omega^{(t)}_{ij} (a, b) = \frac{k(x_{ij}; \mu^{(t-1)}_0 (a, b)) q^{(t-1)}_0 (b)}{\sum_{b=1}^{K_2} k(x_{ij}; \mu^{(t-1)}_0 (a, b)) q^{(t-1)}_0 (b)}
\]

\[
\psi^{(t)}_j (b) = \left\{ \frac{q^{(t-1)}_0 (b) h_0(x_{ja} | \Theta_0^{(t-1)})}{\sum_{b=1}^{K_2} q^{(t-1)}_0 (b) h_0(x_{ja} | \Theta_0^{(t-1)})} \right\} \quad \text{and} \quad \psi^{(t)}_{ij} (a, b) = \left\{ \frac{k(x_{ij}; \mu^{(t-1)}_0 (a, b)) p^{(t-1)}_0 (a)}{\sum_{a=1}^{K_1} k(x_{ij}; \mu^{(t-1)}_0 (a, b)) p^{(t-1)}_0 (a)} \right\}.
\]
With Gaussian data, the nested EM would start with calculating the initial EM weights as explained in section 3.3 (i.e. the pre-step) and then proceed as follows:

Step 1: Update $p^{(t)}(a), q^{(t)}(b), \mu^{(t)}(a, b)$, and $\sigma^2(t)$ for $t = 0, 1, \ldots$.

Step 2: Calculate $LCL_t = \log L_{RC}(\Theta^{(t)})$ for $t = 0, 1, \ldots$.

Step 3: Update the EM weights $\omega^{(t)}_{ij}(a, b), \psi^{(t)}_{ij}(a, b), \omega^{(t)}_i(a, b)$, and $\psi^{(t)}_i(a, b)$ for $t = 1, \ldots$.

Step 4: Repeat steps 1-3 until convergence, with convergence defined by stopping at the first step $t = 1, 2, \ldots$ such that

$$0 < LCL_t(\infty) - LCL_t \leq 0.005.$$

### 3.3.3 Nested EM with Gaussian data

In this section we present two simulated examples that illustrate how the nested EM performs in estimating the block mixture model with Gaussian data. In both examples, we assume we know the true number of row and column clusters (i.e. $K_1$ and $K_2$, respectively, are fixed and known).

#### 3.3.3.1 Example 1

In this example, the data matrix comprise 110 rows and 11 columns. These are generated from 12 block means with 3 row clusters and 4 column clusters. More specifically, the data are simulated from $\mathcal{N}(\mu(a, b), \sigma^2 = 16)$. The cells in Table 3.1 show the means of (e.g. $\mu(1, 1) = 18$) together with the size of each block (e.g. 30 rows and 2 columns in block 1).

<table>
<thead>
<tr>
<th>$\mu(1, 1) = 18$</th>
<th>$\mu(1, 2) = 22$</th>
<th>$\mu(1, 3) = 30$</th>
<th>$\mu(1, 4) = 27$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(30 \times 2)$</td>
<td>$(30 \times 3)$</td>
<td>$(30 \times 2)$</td>
<td>$(30 \times 4)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mu(2, 1) = 40$</th>
<th>$\mu(2, 2) = 37$</th>
<th>$\mu(2, 3) = 46$</th>
<th>$\mu(2, 4) = 33$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(45 \times 2)$</td>
<td>$(45 \times 3)$</td>
<td>$(45 \times 2)$</td>
<td>$(45 \times 4)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mu(3, 1) = 14$</th>
<th>$\mu(3, 2) = 19$</th>
<th>$\mu(3, 3) = 10$</th>
<th>$\mu(3, 4) = 24$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(35 \times 2)$</td>
<td>$(35 \times 3)$</td>
<td>$(35 \times 2)$</td>
<td>$(35 \times 4)$</td>
</tr>
</tbody>
</table>

As mentioned earlier we assume $K_1 = 3$ and $K_2 = 4$ are known. We use the algorithm described in section 3.3.2. To initialize the algorithm, we use k-means [36] on each direction of the data matrix separately, generating initial row and columns labels. We proceed with the
nested EM algorithm until convergence; as a smoothing technique, the convergence criterion was evaluated only at every 5th EM iteration. Figure 3.1(a) shows how two series, the log composite likelihood $LCL_t$ and its predicted convergence point $LCL_t(\infty)$, change with each 5th EM iteration.

![Figure 3.1](image)

**Figure 3.1.** Variation of $LCL$ with every 5th EM iteration for example 1. The red curve in panel (a) shows the $LCL_t(\infty)$ and the black curve the values of $LCL_t$. The blue curve in panel (b) shows the variation of estimated asymptotic convergence rate $c_t$ with every 5th EM iterations for example 1.

Recall that the accuracy of our stopping rule relies on the quality of the geometric series approximation obtained in equation (3.2.2) where $0 < c < 1$ is constant [12]. To understand the reliability of the estimated $LCL_t(\infty)$, we studied the behavior of the estimated asymptotic convergence rate $c_t$. Figure 3.1(b) shows the value of $c_t$ at every 5th iteration. It appears to behave as expected, i.e. to approach 0.8 as we approach the stopping point of the nested EM, suggesting it is a reliable stopping criterion.

<table>
<thead>
<tr>
<th>Table 3.2. Rearranged true mean structure for example 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3.3. Rearranged estimated mean structure for example 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.098</td>
</tr>
<tr>
<td>17.672</td>
</tr>
<tr>
<td>32.793</td>
</tr>
</tbody>
</table>

Next, we compare the estimated block means to the true block means to evaluate the performance of the nested EM. We rearrange the mean structure shown in Table 3.1 as shown in Table 3.2, for ease of comparisons. Table 3.3 shows the “rearranged” estimated mean structure. It
appears our algorithm works well in estimating the mean structure of the block mixture model. This is further confirmed by comparing the estimated row and column proportions, (31.34%, 40.97% and 27.68%) and (35.38%, 17.38%, 28.78% and 18.46%), to the true row and column proportions, (31.82%, 40.91%, and 27.27%) and (36.36%, 18.18%, 27.27% and 18.18%). Also the estimated common standard deviation (3.937) is close to the true value 4.

### 3.3.3.2 Example 2

In this example, we generate 12 blocks using $\mathcal{N}(\mu(a, b), \sigma^2 = 25)$. There are 170 rows and 10 columns in the data matrix. The true summary structure by design has 4 row clusters and 3 column clusters with block means and sizes shown in Table 3.4.

**Table 3.4. Summary structure in example 2**

<table>
<thead>
<tr>
<th>$\mu(1, 1) = 25$</th>
<th>$\mu(1, 2) = 29$</th>
<th>$\mu(1, 3) = 19$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(35 × 3)</td>
<td>(35 × 4)</td>
<td>(35 × 3)</td>
</tr>
<tr>
<td>$\mu(2, 1) = 23$</td>
<td>$\mu(2, 2) = 17$</td>
<td>$\mu(2, 3) = 21$</td>
</tr>
<tr>
<td>(45 × 3)</td>
<td>(45 × 4)</td>
<td>(45 × 3)</td>
</tr>
<tr>
<td>$\mu(3, 1) = 31$</td>
<td>$\mu(3, 2) = 28$</td>
<td>$\mu(3, 3) = 24$</td>
</tr>
<tr>
<td>(40 × 3)</td>
<td>(40 × 4)</td>
<td>(40 × 3)</td>
</tr>
<tr>
<td>$\mu(4, 1) = 20$</td>
<td>$\mu(4, 2) = 26$</td>
<td>$\mu(4, 3) = 30$</td>
</tr>
<tr>
<td>(50 × 3)</td>
<td>(50 × 4)</td>
<td>(50 × 3)</td>
</tr>
</tbody>
</table>

With fixed and known number of row and column clusters, $K_1 = 4$ and $K_2 = 3$, we obtain the initial labels using k-means separately on each direction of the data matrix. We then proceed with the nested EM algorithm described in section 3.3.2.

Figure 3.2(a) shows how $LCL_t$ and $LCL_t(\infty)$ varied with each 5th EM iteration. Figure 3.2(b) shows how the estimated asymptotic convergence rate varied with each 5th EM iteration. Note that here we have considered a larger number of iterations than in Figure 3.2(a).

Our nested EM took 165 (33 × 5) iterations to achieve convergence as defined in section 3.3.1. However, it is clear that the estimated asymptotic convergence rate has not reached its asymptotic value of over 0.9 at this point. At $t = 165$ (33 × 5) the estimated $c_t$ is about 0.8 which means that $LCL(\infty)$ is underestimated, and premature stopping might occur. Figure 3.2(b) suggests
that it took about 500 (100 × 5) EM iterations to achieve a stable convergence rate. Since \( c_t \) is decreasing near \( t = 165(33 × 5) \), the algorithm is actually speeding up in this region. As a check on whether \( t = 33 × 5 \) was a good stopping point we ran the nested EM to \( t = 500 (100 × 5) \) to see if this would significantly reduce errors.

Interestingly, even though in this example we started with 12 blocks (\( K_1 = 4 \) and \( K_2 = 3 \)), block clustering results using the nested EM provides only 6 blocks (with 3 row groups and 2 column groups). In other words, two columns and two rows of the estimated mean structure were the same. To understand the reason, we rearrange the true and estimated mean structures (see Table 3.5 and 3.6).

### Table 3.5. Rearranged true mean structure for example 2

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>21</td>
<td>23</td>
</tr>
<tr>
<td>19</td>
<td>25</td>
<td>29</td>
</tr>
<tr>
<td>20</td>
<td>26</td>
<td>30</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>31</td>
</tr>
</tbody>
</table>

### Table 3.6. Rearranged estimated mean structure for example 2

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16.604</td>
<td>22.430</td>
</tr>
<tr>
<td>21.318</td>
<td>29.354</td>
</tr>
<tr>
<td>23.981</td>
<td>29.661</td>
</tr>
</tbody>
</table>

Recall that the true sampling error in this example is \( \sigma = 5 \). This results in a higher degree of statistical overlap in the true block densities. In a simple normal mixture model, the total variance in \( X \) is \( \sigma^2 + \tau_Q^2 \), where \( \tau_Q^2 \) is the variance of the means in the mixing distribution and
\( \sigma^2 \) is the constant variance within the components. There is therefore a “competition” between \( \sigma^2 \) and the mixing distribution to explain the variance in the data. It is therefore not surprising that our algorithm collapses the number of blocks, as we use the CMLE of \( \sigma^2 \), which is \( \hat{\sigma}^2 \). This is an illustration of a common situation in which one tries to fit too many groups relative to those that are actually distinguishable in the data. In this type of situation, the EM automatically collapses the number of clusters in the solution. Notice that one could “force” a larger number of groups by forcing \( \hat{\sigma}^2 \) to be a smaller value.

### 3.4 Accuracy of the parameter estimates

As in any estimation process, it is important to assess the reliability of the parameter estimates. For a fixed model (say, \( Q \)), if the true parameter values (say \( T_Q \)) are known, then one can estimate the co-variance matrix of the parameter estimates (say, \( \Sigma \)) using the following simulation approach. For \( b = 1, \ldots, B \) independently

- simulate a data matrix \( S^{(b)} \) from \( Q \)
- estimate \( T_Q \) on \( S^{(b)} \), say \( T^{(b)} \)
- with the resulting \( T^{(b)} \) values, construct the estimated co-variance of the parameter estimates as

\[
\hat{\Sigma} = \frac{1}{B} \sum_{b=1}^{B} (T^{(b)} - T_Q)(T^{(b)} - T_Q)^T.
\]

Here \( \sqrt{\text{diag}(\Sigma)} \) gives the standard errors for the parameter estimates.

One could implement this simulation methodology by using the parameter estimates as the truth in the simulation. However, these parameter estimates are not \textit{not robust} to the assumption that the model is correct. As an alternative, we used the following nonparametric cross validation approach. For \( b = 1, \ldots, B \) independently

- obtain half samples by randomly partitioning \( X \) into two equal size sets, say, \( S_1^{(b)} \) and \( S_2^{(b)} \)
- estimate \( T_Q \) for both halves using \( S_1^{(b)} \) (say, \( T_1^{(b)} \)) and \( S_2^{(b)} \) (say, \( T_2^{(b)} \))
- with the resulting \( T_1^{(b)} \) and \( T_2^{(b)} \) values, construct the estimated co-variance matrix of the parameter estimates for the half samples as

\[
\hat{\Sigma}_2 = \frac{1}{B} \sum_{b=1}^{B} (T_1^{(b)} - T_2^{(b)})(T_1^{(b)} - T_2^{(b)})^T.
\]
Next, estimate the co-variance matrix for the full sample by extrapolation as $\hat{\Sigma} = \frac{\hat{\Sigma}_1 + \hat{\Sigma}_2}{2}$, and use $\sqrt{\text{diag}(\hat{\Sigma})}$ for the standard errors for the parameter estimates [71, 72].

### 3.5 Labeling blocks

To estimate the true cluster labels of the rows and columns, we would ideally use the estimated posterior probabilities coming from the mixture model. The posterior probability of the $i^{th}$ row having row label $a$ is $P(A_i = a | X)$. The ideal rule to use for labeling is arg max $P(A_i = a | X)$. However to estimate these posterior probabilities we would need to be able calculate the joint density of the data and we have assumed this is infeasible in our setting.

Our proposed strategy to solve this problem comes from the EM algorithm. We observe that the EM updates for the weight parameters, $p(a) = P(A_i = a)$, can be written in terms of “fitted posterior probabilities” as

$$\hat{p}(a) = \frac{1}{R} \sum_{i=1}^{R} p_{ai}^*$$

(3.5.1)

where $p_{ai}^*$ represents the “vote” of row $i$ for label $a$. If we had used the full likelihood, then (3.5.1) would hold with $p_{ai}^* = P(A_i = a | X)$.

With Gaussian data, we have already seen that $\hat{p}(a)$ for row label $a$ is given by (equation 3.3.4)

$$\hat{p}(a) = \frac{\sum_{i=1}^{R} \omega_i(a) + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \psi_{ij}(a, b)}{R + RC}.$$

Hence the vote for row $i$ for label $a$ is

$$p_{ai}^* = \frac{\omega_i(a) + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \psi_{ij}(a, b)}{1 + C}.$$  

Once we obtain $p_{ai}^*$ for all $i = 1, \ldots, R$ and $a = 1, \ldots, K_1$, we use the following technique to assign rows to each of the row groups:

Step 1: Scan $\hat{p}(\cdot)$ and find the smallest row group (say $a^*$); i.e. $a^*$ such that $\hat{p}(a^*) = \min_a \hat{p}(a)$

Step 2: Determine the number of rows that should be assigned to $a^*$ (say $|a^*|$); i.e. $|a^*| = R \times \hat{p}(a^*)$.

Step 3: Sort the matrix $p_{ai}^*$ in the ascending order of vector $p_{ai}^*$ and assign the first $|a^*|$ rows to row group $a^*$.

Step 4: Update $\hat{p}(\cdot)$ and $p_{ai}^*$ by taking out $a^*$. 
Step 5: Repeat steps 1-4 until all rows are assigned to row groups.

This strategy ensures that we maintain the cluster weights as estimated by the EM and that each cluster consists of the rows that most “contributed” to it. Note that by starting with the smallest clusters we attempt to privilege the accurate identification of small groups of rows and columns. In many applications the strongest and most interesting signals may indeed concern such groups - although this is not a completely general approach.

We can obtain the vote of column $j$ for label $b$ in a similar fashion, and use the same technique to assign the column labels.

### 3.6 Comparing cluster solutions

There are various measurements that have been developed to compare two cluster solutions. In this section we introduce two such measurements: the popular *adjusted RAND index* [39] and a new measure based on the $\chi^2$ test of independence.

Let $X = \{x_1, x_2, \ldots, x_N\}$ to be a set of $N$ data points and $A = \{A_1, A_2, \ldots, A_{K_1}\}$ and $B = \{B_1, B_2, \ldots, B_{K_2}\}$ to be two partitions of $X$ with $K_1$ clusters and $K_2$ clusters respectively. Notice that $A_i \cap A_j = \emptyset$ for $i, j = 1, \ldots, K_1$ and $\bigcup_{i=1}^{K_1} A_i = X$. Also $B_i \cap B_j = \emptyset$ for $i, j = 1, \ldots, K_2$ and $\bigcup_{i=1}^{K_2} B_i = X$. The two cluster solutions can be summarized in a contingency table as in Table 3.7.

<table>
<thead>
<tr>
<th></th>
<th>$B_1$</th>
<th>$B_2$</th>
<th>$\cdots$</th>
<th>$B_{K_2}$</th>
<th>$A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$n(A_1 \cap B_1)$</td>
<td>$n(A_1 \cap B_2)$</td>
<td>$\cdots$</td>
<td>$n(A_1 \cap B_{K_2})$</td>
<td>$n(A_1)$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>$n(A_2 \cap B_1)$</td>
<td>$n(A_2 \cap B_2)$</td>
<td>$\cdots$</td>
<td>$n(A_2 \cap B_{K_2})$</td>
<td>$n(A_2)$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$A_{K_1}$</td>
<td>$n(A_{K_1} \cap B_1)$</td>
<td>$n(A_{K_1} \cap B_2)$</td>
<td>$\cdots$</td>
<td>$n(A_{K_1} \cap B_{K_2})$</td>
<td>$n(A_{K_1})$</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>$n(B_1)$</td>
<td>$n(B_2)$</td>
<td>$\cdots$</td>
<td>$n(B_{K_2})$</td>
<td>$N$</td>
</tr>
</tbody>
</table>

In this table we might suppose that partition $A$ is based on the true labels and partition $B$ is based on the estimated labels, and that we wish to measure how well the estimates reproduce the true labels. Or, if $A$ and $B$ are both estimated labels, we might wish to assess how well the two estimates agree.

#### 3.6.1 Adjusted RAND index

Hubert and Arabie (1985) proposed the RAND index [39] as a tool to assess the *agreement* of two cluster solutions. The RAND index summarizes the agreement with two numbers: $N_{11} =$ the number of pairs $\{x_i, x_j\}$ that are in the same cluster in both cluster solutions, and $N_{00} =$ the
number of pairs that are in different clusters in both cluster solutions. Note that \((N_2 - N_{00} - N_{11})\) is the number of pairs \(\{x_i, x_j\}\) where the solutions disagree - in one scheme the two points are in one cluster, whereas in the other they are in separate clusters.

The RAND index (RI) for the two cluster solutions \(A\) and \(B\) is

\[
RI(A, B) = \frac{N_{00} + N_{11}}{\binom{N}{2}}
\]

After realizing that the expected value of \(RI\) for two random partitions is not constant, the authors proposed a corrected version, the Adjusted RAND index \([39]\) as:

\[
ARI(A, B) = \frac{RI - E(RI)}{\max(RI) - E(RI)}.
\]

The adjusted RAND index (ARI) can be rewritten in terms of the elements of table 3.7 as

\[
ARI(A, B) = \frac{\sum_{i,j} \binom{n(A_i \cap B_j)}{2} - \left[ \sum_i \binom{n(A_i)}{2} \sum_j \binom{n(B_j)}{2} \right] / \binom{N}{2}}{\frac{1}{2} \left[ \sum_i \binom{n(A_i)}{2} + \sum_j \binom{n(B_j)}{2} \right] - \left[ \sum_i \binom{n(A_i)}{2} \sum_j \binom{n(B_j)}{2} \right] / \binom{N}{2}}.
\]

Here the expected value of the index can be obtained using a generalized hypergeometric distribution \([51]\) to describe the randomness. The \(ARI\) lies between 0 and 1; the higher its value, the stronger the agreement between the two cluster solutions. Vinh et al discuss some properties of the \(ARI\) and compare it with some other useful measurements \([69, 70]\).

### 3.6.2 \(\chi^2\) test of independence

While the adjusted RAND index provides a measure of the agreement between two cluster solutions, it does not provide a measure of statistical evidence for such agreement. This is true for most of the existing measurements, and in this section we discuss a new approach that offers a probabilistic framework.

Here, we start by associating each cluster solution with a projection matrix. We let the \(uv^{th}\) element of the projection matrix of solution \(A\), \(P_A\), be

\[
(P_A)_{uv} = \sum_{i=1}^{K} \frac{1}{n(A_i)} I(u \in A_i) I(v \in A_i) \text{ where } u, v = 1, \ldots, N.
\]

Notice that \(P_A (X_1, \ldots X_N)^T\) gives a vector where each \(X_i\) is replaced by the mean of the corresponding cluster. We obtain the projection matrix for cluster solution \(B\) in a similar fashion, say \(P_B\). To assess the similarity between the two cluster solution, we calculate the difference
between the projection matrices using the trace distance [55], 
\[ tr(P_A - P_B)^2. \]

Notice that \( tr(P_A) = K_1 \), because

\[
tr(P_A) = \sum_{u=1}^{N} \sum_{i=1}^{K_1} \frac{1}{n(A_i)} I(u \in A_i) I(u \in A_i) \text{ where } u = 1, \ldots, N
= \sum_{i=1}^{K_1} \sum_{n(A_i)}^{N} \sum_{u=1}^{1} \frac{1}{n(A_i)} I(u \in A_i) I(u \in A_i) = \sum_{i=1}^{K_1} \frac{1}{n(A_i)} n(A_i) = K_1.
\]

In the same fashion \( tr(P_B) = K_2 \). Next rewrite

\[
(P_A P_B)_{uv} = \sum_{w=1}^{N} \sum_{i=1}^{K_1} \frac{K_2}{n(A_i)} I(u \in A_i) I(w \in A_i) I(w \in B_j) I(v \in B_j)
= \sum_{i=1}^{K_1} \sum_{j=1}^{K_2} \sum_{w=1}^{N} \left[ \frac{n(A_i \cap B_j) - n(A_i) n(B_j)}{n(A_i) n(B_j)} \right]^{2}
= \sum_{i=1}^{K_1} \sum_{j=1}^{K_2} \sum_{w=1}^{N} \frac{n(A_i \cap B_j)}{n(A_i) n(B_j)} n(A_i) n(B_j).
\]

It follows that \( tr(P_A P_B) \), the sum of diagonal elements of \( P_A P_B \), is

\[
tr(P_A P_B) = \sum_{i=1}^{K_1} \sum_{j=1}^{K_2} n^2(A_i \cap B_j). \tag{3.6.1}
\]

Therefore the trace distance takes values between 0 and \( K_1 + K_2 \), and one could divide it by \( K_1 + K_2 \) to obtain a number between 0 and 1. However, we think the following analysis gives greater statistical insight into the trace distance.

Consider the \( \chi^2 \) test for independence for Table 3.7; that is, compute the statistic

\[
\chi^2 = \sum_{i=1}^{K_1} \sum_{j=1}^{K_2} \frac{[\text{observed}_{ij} - \text{expected}_{ij}]^2}{\text{expected}_{ij}}
= \sum_{i=1}^{K_1} \sum_{j=1}^{K_2} \frac{n(A_i \cap B_j) - n(A_i) n(B_j) / N}{n(A_i) n(B_j) / N} \tag{3.6.2}
\]

where \( \text{expected}_{ij} = N \left( \frac{n(A_i)}{N} \right) \left( \frac{n(B_j)}{N} \right) \). Equations (3.6.2) and (3.6.1) give the relationship

\[
\chi^2 = N (tr(P_A P_B) - 1)
\]
so that
\[ tr(P_A - P_B)^2 = K_1 + K_2 - 2 \left[ 1 + \frac{\chi^2}{N} \right]. \]

We see that similar cluster solutions will have higher values for the \( \chi^2 \) statistics, and smaller trace distances. Therefore to assess the similarity of two cluster solutions we could calculate the \( \chi^2 \) test statistic; the higher the value the more similar the two cluster solutions. Using the properties of the \( \chi^2 \) distribution, we could measure the statistical evidence against the independence of two cluster solutions by computing
\[ T_{AB} = \frac{(\chi^2 - df)}{\sqrt{2df}} \]
where the degrees of freedom of \( \chi^2 \) is \( df = (K_1 - 1)(K_2 - 1) \) and \( T_{AB} \sim N(0,1) \). The larger the value of \( T_{AB} \), the stronger the evidence for the agreement of the two cluster solutions.

### 3.7 A numerical study of the nested EM with Gaussian data

In this section we present the results of a numerical investigation of the performance of our approach in estimating a block mixture model with Gaussian data (i.e. \( \mathcal{N}(x_{ij}; \mu(a,b), \sigma^2) \)). The study was designed to assess:

- how effective our approach is with large sampling errors (or noisy data)
- how well our approach detects small blocks
- how our approach performs in detecting different summary patterns (i.e. different patterns in the block means).

Throughout the study, the overall size of the two way table was held constant (i.e. \( R = 110 \) and \( C = 80 \)). Moreover, it was assumed that the number of row and column clusters are fixed and known (i.e. \( K_1 = 3 \) and \( K_2 = 2 \)).

#### 3.7.1 Preliminary simulation study

Here we considered two arrangements of block means. In the first, the block means monotonically increase within each row or column cluster (pattern \( p_1 \) in Figure 3.3). In the second pattern (\( p_2 \) in Figure 3.3) the block means in the first column are increasing but those in the second column are decreasing. Notice that the set of means of the six blocks in both \( p_1 \) and \( p_2 \) are the same, but they are arranged differently. Our intuition is that \( p_2 \) makes the columns more different and so any method should be more effective in \( p_2 \) than in \( p_1 \).
In constructing our numerical study we could have built the data $X$ using randomly selected labels for the rows and columns based on the multinomial distribution. Instead we removed this source of variability by fixing the number of labels for each group in advance, and sampled $X$ conditionally.

We also considered two different block prevalence scenarios; one with relatively evenly balanced blocks (see $s_1$ in Figure 3.4) and the other that included a block with a small number of both columns and rows (see $s_2$ in Figure 3.4). The six blocks in $s_1$ are roughly of the same size in each dimension. These two scenarios allow us to study how well our methods work at detecting smaller blocks in data.

In addition to the two block mean patterns ($p_1$ and $p_2$) and the two block prevalence scenarios ($s_1$ and $s_2$), in order to understand the performance of our methods when the sampling errors are large, we also considered three different sampling errors: $\sigma = 2$, $\sigma = 5$, and $\sigma = 10$. With all the different options we considered, we have a $2 \times 2 \times 3$ (i.e. pattern $\times$ prevalence $\times$ error) factorial study design. In Table 3.8 we show the four basic (pattern $\times$ prevalence) designs. Each of the four basic designs is then sampled at the three levels of error.

To generate a data set from one element of the basic designs, we fixed the row and column labels first and then sampled the data within each block using a Gaussian distribution with the corresponding sampling error.
Table 3.8. Different simulation designs

<table>
<thead>
<tr>
<th>Design</th>
<th>Pattern</th>
<th>Prevalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ((p_1, s_1))</td>
<td>(p_1)</td>
<td>(s_1)</td>
</tr>
<tr>
<td>2 ((p_1, s_2))</td>
<td>(p_1)</td>
<td>(s_2)</td>
</tr>
<tr>
<td>3 ((p_2, s_1))</td>
<td>(p_2)</td>
<td>(s_1)</td>
</tr>
<tr>
<td>4 ((p_2, s_2))</td>
<td>(p_2)</td>
<td>(s_2)</td>
</tr>
</tbody>
</table>

Collapsibility score

We created a measure of cluster separation called the **collapsibility score**. Our goal is to create a single summary number of the separability of the cluster groups. To introduce the collapsibility score consider a block mean structure, \(\mu = \{\mu(a, b) : a = 1, \ldots, K_1, b = 1, \ldots, K_2\}\) as shown in Figure 3.5. Consider comparing the means of two column clusters \(u\) and \(v\), that is, \(\mu(\cdot, u)\) and \(\mu(\cdot, v)\). Let \(p(a), a = 1, \ldots, K_1\) and \(q(b), b = 1, \ldots, K_2\) be the row and column proportions for the blocks, viewed as fixed as in Figure 3.4. If \(N = R \times C\), where \(R\) is number of rows in \(X\) and \(C\) is the number of columns in \(X\), then \(Np(a)q(b)\) is the size of the block \((a, b)\).

![Sample block mean structure](image)

We define the collapsibility score for comparing the two column clusters \(u\) and \(v\) in \(\mu\) as

\[
\Delta_{uv}^2 = \sum_{a=1}^{K_1} \left[\frac{n(a, u) - \mu(a, v))^2}{\frac{1}{Np(a)q(u)} + \frac{1}{Np(a)q(v)}}\right] = \sum_{a=1}^{K_1} \left(\frac{Np(a)q(u)q(v)}{\sigma^2(u) + q(v)}\right)[\mu(a, u) - \mu(a, v)]^2.
\]

This measure has the following motivation: suppose we knew the true row labels. Then the comparison of the means in the two set of blocks in each column is similar to an ANOVA problem, where we let \(\bar{x}_1, \ldots, \bar{x}_{K_1}\) be the means in column \(u\) and \(\bar{y}_1, \ldots, \bar{y}_{K_1}\) be the means in column \(v\). In such a setting the statistical noncentrality parameter for testing the difference in
column means is

$$\Delta^2_{uv} = (\mu_1 - \mu_2) \left[\text{Var}(\bar{x} - \bar{y})\right]^{-1} (\mu_1 - \mu_2).$$

If we use the independence of the block sample means to calculate $\text{Var}(\bar{x} - \bar{y})$, along with noting that the block size for block $(a, u)$ is $N_{au} = N p(a) q(u)$, then we arrive at our formula. Note that a $\chi^2$-test for the equality of the means has $K_1$ degrees of freedom, so one might wish to use $\frac{\Delta_{uv}}{\sqrt{K_1}}$ instead.

Of course this measure ignores the missing labels; that is, when comparing $\mu(\cdot, u)$ to $\mu(\cdot, v)$ we treat the estimated labels as true, and $\Delta^2_{uv}$ relies largely on how well the labels are estimated. When $\sigma^2$ is relatively large, separating the two groups in the mixture and reproducing the correct labels is hard, and hence $\Delta^2_{uv}$ poorly measures the ability to distinguish the two columns. Nonetheless, when $\sigma^2$ is relatively small, so that labels are mostly correctly estimated, then this seems a good start for understanding the collapsibility of two columns.

We obtain a collapsibility score for comparing two rows $m$ and $n$ in the same fashion. Thus, we can define the overall collapsibility score for $\mu$ as

$$\Delta_\mu = \min \left\{ \left( \frac{\Delta_{uu}}{\sqrt{K_1}} : u, v = 1, \ldots, K_1 \right), \left( \frac{\Delta_{mn}}{\sqrt{K_2}} : m, n = 1, \ldots, K_2 \right) \right\}. \quad (3.7.1)$$

Notice that the larger the collapsibility score for a design, the more distinct all the blocks are. We note that if $\Delta_\mu < 1$ then there exist a pair of clusters (row or column) that are not very distinct, even when we know the labels exactly. In such a situation, we are not likely to do a good job in estimation.

**Results**

We simulated a single dataset $X$ from each design setting with $N = 8800(110 \times 80)$ observations. We then used our proposed approach to estimate the models. Here, we assumed that the true number of row and column clusters are known ($K_1 = 3$ and $K_2 = 2$). The method explained in section 3.5 was then used to estimate the labels for the rows and columns in each $X$. In addition, for each case, we also calculated the standard errors of the parameter estimates using the nonparametric cross validation approach described in section 3.4.

We start with the four basic designs in Table 3.8 at a low level of sampling error. For each of the four settings we draw a single $X$ with a true sampling error of $\sigma = 2$. Table 3.9 shows a summary of the results.

The collapsibility score in column 2 of Table 3.9 is our measurement of overall cluster separation in (3.7.1). The higher the collapsibility score, the higher the overall cluster separation,
Table 3.9. Performance of our approach in the four basic simulation designs with $\sigma = 2$

<table>
<thead>
<tr>
<th>Design</th>
<th>Collapsibility score</th>
<th>ARI</th>
<th>$\chi^2$-test</th>
<th>$\hat{\sigma}$ (std. error)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\chi^2/\chi^2_{max}$</td>
<td>p-value</td>
</tr>
<tr>
<td>$(p_1, s_1)$</td>
<td>22.913 1.00 1.00 1.000 0.000</td>
<td>1.9930 (0.0255)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(p_1, s_2)$</td>
<td>7.906 1.00 1.00 1.000 0.000</td>
<td>1.9887 (0.0289)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(p_2, s_1)$</td>
<td>29.122 1.00 1.00 1.000 0.000</td>
<td>1.9830 (0.0274)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(p_2, s_2)$</td>
<td>9.712 1.00 1.00 1.000 0.000</td>
<td>1.9754 (0.0289)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The $p$-value shown in the table for $\chi^2$ test is using $T_{AB}$ with $df = 25$ (see section 3.6.2).

and so the easier the clustering problem. We see here that the overall cluster separation in the designs with prevalence $s_2$ (small blocks) is poor compared to the corresponding designs with prevalence $s_1$ (balanced blocks). If we hold prevalence fixed, then pattern $p_2$ has more separation than $p_1$.

We see that both the adjusted RAND index (ARI) and the $\chi^2$ measure of classification error are at their maximum values, indicating that at the given level of $\sigma$, we made no classification error. The high quality of our inferences can be further confirmed by comparing the estimated $\sigma$ in Table 3.9 and the estimated block means in Table 3.10 to the true values in these designs.

Table 3.10. Estimated block means in the four basic simulation designs with $\sigma = 2$

<table>
<thead>
<tr>
<th>Design</th>
<th>True Block Means</th>
<th>Estimated Block Means (std. errors)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(p_1, s_1)$</td>
<td>10 15</td>
<td>10.017 (0.0884) 15.064 (0.1076)</td>
</tr>
<tr>
<td></td>
<td>14 20</td>
<td>14.011 (0.0722) 19.987 (0.0966)</td>
</tr>
<tr>
<td></td>
<td>19 28</td>
<td>19.031 (0.0744) 28.107 (0.0970)</td>
</tr>
<tr>
<td>$(p_1, s_2)$</td>
<td>10 15</td>
<td>9.973 (0.1167) 14.889 (0.3230)</td>
</tr>
<tr>
<td></td>
<td>14 20</td>
<td>14.011 (0.0495) 19.947 (0.1985)</td>
</tr>
<tr>
<td></td>
<td>19 28</td>
<td>19.067 (0.0518) 27.909 (0.2127)</td>
</tr>
<tr>
<td>$(p_2, s_1)$</td>
<td>10 28</td>
<td>9.995 (0.0668) 28.073 (0.0999)</td>
</tr>
<tr>
<td></td>
<td>15 20</td>
<td>15.029 (0.0948) 20.167 (0.1208)</td>
</tr>
<tr>
<td></td>
<td>19 14</td>
<td>18.999 (0.0875) 14.057 (0.0908)</td>
</tr>
<tr>
<td>$(p_2, s_2)$</td>
<td>10 28</td>
<td>9.943 (0.1055) 27.834 (0.3388)</td>
</tr>
<tr>
<td></td>
<td>15 20</td>
<td>15.073 (0.0601) 19.939 (0.2486)</td>
</tr>
<tr>
<td></td>
<td>19 14</td>
<td>19.006 (0.0511) 14.009 (0.1848)</td>
</tr>
</tbody>
</table>

Note that the estimated standard errors in Tables 3.9 and 3.10 are not from a simulation study over multiple samples, but from our cross-validation of point estimators described in section 3.4.

If we increase the standard deviation of the data to $\sigma = 5$, some block means in Figure 3.3 are within one standard error of each other. Table 3.11 corresponds to Table 3.9 for this noisier setting. The collapsibility scores are indeed $(2/5)$ times the scores in Table 3.9, so that the relative conclusions about the separability of four basic designs remain the same. Table 3.11 shows that our methods perform well also in the noisier setting, although we now have some misclassification
errors in pattern $p_2$. Note that we still identified the labels in pattern $p_1$ with 100% accuracy. We were somewhat surprised by this, particularly since design $(p_2, s_1)$ had a much larger relative collapsibility score than $(p_1, s_2)$. We notice that the standard error estimates for the block mean estimates (see Table 3.12) are, as expected, relatively higher when $\sigma = 5$ compared to $\sigma = 2$ in Table 3.10. Surprisingly, if we reorganize the results by the block means as in Table 3.13, $(p_1, s_2)$ seems to yield much better mean estimates for block means 14 and 19 than $(p_2, s_1)$ does. This suggests that collapsibility scores are not a good proxy for the difficulty of the problem.

### Table 3.11. Performance of our approach in the four basic simulation designs with $\sigma = 5$

<table>
<thead>
<tr>
<th>Design</th>
<th>Collapsibility score</th>
<th>ARI</th>
<th>$\chi^2$-test $\chi^2 / \chi^{2}_{max}$</th>
<th>p-value</th>
<th># Misclassified</th>
<th>$\hat{\sigma}$ (std. error)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(p_1, s_1)$</td>
<td>9.165</td>
<td>1.00</td>
<td>1.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$(p_1, s_2)$</td>
<td>3.162</td>
<td>1.00</td>
<td>1.000</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$(p_2, s_1)$</td>
<td>11.649</td>
<td>0.93</td>
<td>0.943</td>
<td>0.000</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>$(p_2, s_2)$</td>
<td>3.885</td>
<td>0.97</td>
<td>0.984</td>
<td>0.000</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

p value shown in the table for $\chi^2$ test is using $T_{AB}$ with $df = 25$ (see section 3.6.2)

### Table 3.12. Estimated block means in the four basic simulation designs with $\sigma = 5$

<table>
<thead>
<tr>
<th>Design</th>
<th>True Block Means</th>
<th>Estimated Block Means (std. Errors)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(p_1, s_1)$</td>
<td>10 15</td>
<td>9.951 (0.2348) 15.132 (0.3033)</td>
</tr>
<tr>
<td></td>
<td>14 20</td>
<td>14.012 (0.1896) 19.954 (0.2512)</td>
</tr>
<tr>
<td></td>
<td>19 28</td>
<td>19.021 (0.2234) 28.212 (0.3118)</td>
</tr>
<tr>
<td>$(p_1, s_2)$</td>
<td>10 15</td>
<td>9.842 (0.2849) 15.662 (1.4212)</td>
</tr>
<tr>
<td></td>
<td>14 20</td>
<td>14.029 (0.1270) 19.683 (0.5739)</td>
</tr>
<tr>
<td></td>
<td>19 28</td>
<td>19.188 (0.1373) 28.000 (0.7249)</td>
</tr>
<tr>
<td>$(p_2, s_1)$</td>
<td>10 28</td>
<td>10.077 (0.1895) 28.325 (0.3034)</td>
</tr>
<tr>
<td></td>
<td>15 20</td>
<td>14.899 (0.3483) 20.494 (0.4630)</td>
</tr>
<tr>
<td></td>
<td>19 14</td>
<td>18.758 (0.1111) 14.338 (0.4355)</td>
</tr>
<tr>
<td>$(p_2, s_2)$</td>
<td>10 28</td>
<td>9.863 (0.2871) 27.288 (1.5775)</td>
</tr>
<tr>
<td></td>
<td>15 20</td>
<td>15.142 (0.1826) 19.754 (1.1468)</td>
</tr>
<tr>
<td></td>
<td>19 14</td>
<td>19.019 (0.1626) 13.918 (0.6482)</td>
</tr>
</tbody>
</table>

### Table 3.13. Reorganized estimated std. errors for block means estimates in designs $(p_1, s_2)$ and $(p_2, s_1)$ with $\sigma = 5$

<table>
<thead>
<tr>
<th>True Mean</th>
<th>10</th>
<th>14</th>
<th>15</th>
<th>19</th>
<th>20</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated std. error for estimated mean</td>
<td>$(p_1, s_2)$</td>
<td>0.28</td>
<td>0.13</td>
<td>1.42</td>
<td>0.14</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>$(p_2, s_1)$</td>
<td>0.19</td>
<td>0.43</td>
<td>0.35</td>
<td>0.41</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Finally, we increased the sampling error to $\sigma = 10$, creating a very noisy setting. Table 3.14 shows the corresponding results for each $X$ obtained from the four basic designs. The collapsi-
bility scores are again reduced relative to Table 3.9 by a factor of $(2/10)$.

As noted earlier, it appears that our methods worked better in pattern $p_1$ than in $p_2$, though we expected an opposite behavior. Table 3.14 shows that even with very noisy data, our methods perform well in pattern $p_1$, since we have only very few misclassified rows and columns. At this level of noisiness, in design $p_2$, we observe a relatively higher number of misclassifications.

<table>
<thead>
<tr>
<th>Design</th>
<th>Collapsibility score</th>
<th>ARI</th>
<th>$\chi^2$-test $\chi^2/\chi^2_{max}$ p-value</th>
<th># Misclassified Rows</th>
<th>Columns</th>
<th>$\hat{\sigma}$ (std. error)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(p_1, s_1)$</td>
<td>4.582</td>
<td>0.93</td>
<td>0.929</td>
<td>0.000</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$(p_1, s_2)$</td>
<td>1.581</td>
<td>0.96</td>
<td>0.921</td>
<td>0.000</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>$(p_2, s_1)$</td>
<td>5.824</td>
<td>0.50</td>
<td>0.492</td>
<td>0.000</td>
<td>46</td>
<td>3</td>
</tr>
<tr>
<td>$(p_2, s_2)$</td>
<td>1.942</td>
<td>0.64</td>
<td>0.594</td>
<td>0.000</td>
<td>12</td>
<td>2</td>
</tr>
</tbody>
</table>

p value shown in the table for $\chi^2$ test is using $T_{AB}$ with $df = 25$ (see section 3.6.2)

Table 3.15 shows the estimated block means for each $X$ obtained from the four basic designs with $\sigma = 10$. We see that the estimated standard errors for the block mean estimates are larger when $\sigma = 10$ compared to settings with less noise ($\sigma = 2$ and $\sigma = 5$). If we compare designs $(p_1, s_2)$ and $(p_2, s_1)$, we see again that in the more collapsible $(p_1, s_2)$ we do better at estimating four of six block means.

<table>
<thead>
<tr>
<th>Design</th>
<th>True Block Means</th>
<th>Estimated Block Means (std. errors)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(p_1, s_1)$</td>
<td>10 15</td>
<td>9.776 (0.6132) 15.494 (0.9044)</td>
</tr>
<tr>
<td></td>
<td>14 20</td>
<td>14.204 (0.4989) 19.650 (0.8372)</td>
</tr>
<tr>
<td></td>
<td>19 28</td>
<td>18.896 (0.4699) 28.539 (0.5551)</td>
</tr>
<tr>
<td>$(p_1, s_2)$</td>
<td>10 15</td>
<td>9.820 (0.8281) 17.978 (4.1159)</td>
</tr>
<tr>
<td></td>
<td>14 20</td>
<td>14.177 (0.3667) 18.712 (1.6008)</td>
</tr>
<tr>
<td></td>
<td>19 28</td>
<td>19.336 (0.3349) 28.204 (1.8634)</td>
</tr>
<tr>
<td>$(p_2, s_1)$</td>
<td>10 28</td>
<td>9.842 (1.2727) 29.532 (1.2288)</td>
</tr>
<tr>
<td></td>
<td>15 20</td>
<td>11.842 (1.2165) 27.156 (2.6664)</td>
</tr>
<tr>
<td></td>
<td>19 14</td>
<td>17.443 (0.9689) 16.345 (1.3753)</td>
</tr>
<tr>
<td>$(p_2, s_2)$</td>
<td>10 28</td>
<td>10.092 (2.0936) 30.874 (8.6556)</td>
</tr>
<tr>
<td></td>
<td>15 20</td>
<td>15.882 (1.4769) 13.452 (2.7473)</td>
</tr>
<tr>
<td></td>
<td>19 14</td>
<td>18.808 (1.2426) 19.512 (3.7876)</td>
</tr>
</tbody>
</table>

The results of this study suggest that when we have less noisy data (or well separated blocks) our methods work very well in identifying the correct block labels. As expected, when the level of noisiness increase the performance of our methods deteriorates. Surprisingly, our methods work better in designs with pattern $p_1$, though our expectation was to observe the contrary.
Table 3.16. Reorganized estimated std. errors for block means estimates in designs \((p1, s2)\) and \((p2, s1)\) with \(\sigma = 10\)

<table>
<thead>
<tr>
<th>True Mean</th>
<th>10</th>
<th>14</th>
<th>15</th>
<th>19</th>
<th>20</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated std. error for estimated mean ((p1, s2))</td>
<td>0.83</td>
<td>0.37</td>
<td>4.12</td>
<td>0.33</td>
<td>1.60</td>
<td>1.86</td>
</tr>
<tr>
<td>Estimated std. error ((p2, s1))</td>
<td>0.19</td>
<td>0.43</td>
<td>0.35</td>
<td>0.41</td>
<td>0.46</td>
<td>0.30</td>
</tr>
</tbody>
</table>

3.7.2 Extended simulation study

Next, we extended our study to compare the performance of our methods with some existing methods. Here, we compare our methods to BEM (see section 1.3.2) and BCEM (see section 1.3.1) proposed by Govaert and Nadif [31, 32]. To estimate the models using BEM and BCEM we used the \textit{R} library \textit{blockcluster} [8, 7]. In addition, we also used k-means and hierarchical clustering with complete linkage on rows and columns \textit{separately} (referred to as two-way k-means and two-way hierarchical hereafter).

For each of the basic designs, we generated 25 data sets at each of the 3 levels of the sampling error \((\sigma)\). We then estimated the row and column labels for each case using the five methods under comparison. Notice that here the true number of row and column clusters are assumed to be fixed and known.

Table 3.17 shows the average ARI calculated over the 25 datasets generated from each of the four basic designs at the 3 levels of \(\sigma\). The results indicate that all methods work well when the true sampling error is small (\(\sigma = 2\) and \(\sigma = 5\)). As \(\sigma\) increases the performance of all methods is reduced. Surprisingly, all methods that rely on the block mixture model seem to work better with the block arrangement \(p1\) than with \(p2\). If we hold \(\sigma\) and block pattern fixed, our approach seems to estimate the labels better in the presence of a smaller block \((s1)\) whereas all other methods under consideration seem to estimate the labels better with balanced blocks \((s2)\).

It appears that the BEM and BCEM outperform our methodology, and this is more evident when the sampling error is large \((\sigma = 10)\). It should be noted that the average RAND index shown in Table 3.17 depends largely on how the rows and columns are labeled using each method. Recall that in our approach, we use a labeling technique based on the EM weights (see section 3.5), whereas the library \textit{blockcluster} in \textit{R} uses the maximum a posteriori (MAP) rule with different estimates of posterior probabilities.

In order to compare the performance of our labeling technique with the MAP rule, we applied our labeling approach to the BEM posterior output to estimate the row and column labels. In Table 3.18 we show the ARI values obtained by comparing these block labels to the true labels, for each of the designs. From the results it is clear that our labeling technique worked as well as the MAP rule used by [8, 7], so this difference does not explain the relative performance in
Table 3.17. Average adjusted RAND index in the four simulation designs with $\sigma = 2, \sigma = 5$ and $\sigma = 10$

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>Design</th>
<th>Average ARI (SD)</th>
<th>Nested EM</th>
<th>BEM</th>
<th>BEM</th>
<th>Two-way k-means</th>
<th>Two-way hierarchical</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$(p1, s1)$</td>
<td>0.989 (0.018)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(p1, s2)$</td>
<td>0.997 (0.012)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>0.974 (0.090)</td>
<td>1.000 (0.000)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(p2, s1)$</td>
<td>0.989 (0.020)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(p2, s2)$</td>
<td>0.990 (0.015)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$(p1, s1)$</td>
<td>0.941 (0.059)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>0.997 (0.008)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(p1, s2)$</td>
<td>0.960 (0.041)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>0.974 (0.090)</td>
<td>0.995 (0.010)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(p2, s1)$</td>
<td>0.927 (0.042)</td>
<td>0.999 (0.005)</td>
<td>0.999 (0.005)</td>
<td>0.999 (0.005)</td>
<td>0.998 (0.006)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(p2, s2)$</td>
<td>0.971 (0.024)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>1.000 (0.000)</td>
<td>0.979 (0.046)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>$(p1, s1)$</td>
<td>0.899 (0.063)</td>
<td>0.961 (0.028)</td>
<td>0.964 (0.025)</td>
<td>0.921 (0.051)</td>
<td>0.625 (0.077)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(p1, s2)$</td>
<td>0.910 (0.056)</td>
<td>0.935 (0.045)</td>
<td>0.935 (0.045)</td>
<td>0.563 (0.140)</td>
<td>0.475 (0.133)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(p2, s1)$</td>
<td>0.676 (0.108)</td>
<td>0.961 (0.037)</td>
<td>0.963 (0.034)</td>
<td>0.946 (0.044)</td>
<td>0.682 (0.112)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(p2, s2)$</td>
<td>0.692 (0.135)</td>
<td>0.886 (0.054)</td>
<td>0.882 (0.058)</td>
<td>0.645 (0.140)</td>
<td>0.386 (0.096)</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.18. Comparing the performance of our labeling rule to the MAP rule in the four simulation designs, with $\sigma = 2, \sigma = 5$ and $\sigma = 10$

<table>
<thead>
<tr>
<th>Design</th>
<th>Average ARI for BEM over 25 data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma = 2$</td>
</tr>
<tr>
<td></td>
<td>our rule</td>
</tr>
<tr>
<td>$(p1, s1)$</td>
<td>1.000</td>
</tr>
<tr>
<td>$(p1, s2)$</td>
<td>1.000</td>
</tr>
<tr>
<td>$(p2, s1)$</td>
<td>1.000</td>
</tr>
<tr>
<td>$(p2, s2)$</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Next restricting our attention to comparisons with BEM, we investigated how our methods perform in terms of estimating the block means. We calculated the mean square error ($MSE$).
for the block mean estimates as:

$$MSE = \frac{1}{(B \times M)} \sum_{b=1}^{B} \sum_{m=1}^{M} (\hat{\mu}_m^{(b)} - \mu_m)^2$$

Here, $M$ is the number of blocks ($M = 6$ in our study) and $B$ is the number of simulations ($B = 25$ in our study). Notice that the smaller the $MSE$ the better the estimation. To calculate the MSE, we first sorted the true block means and the estimated block means in ascending order, and then calculated the difference $\left(\hat{\mu}_m^{(b)} - \mu_m\right)$. Table 3.19 shows the results for the four basic simulation designs at the 3 levels of $\sigma$.

Table 3.19. Mean square error for block means estimates in the four simulation designs, with $\sigma = 2$, $\sigma = 5$, and $\sigma = 10$

<table>
<thead>
<tr>
<th>Design</th>
<th>$\sigma = 2$ MSE</th>
<th>$\sigma = 5$ MSE</th>
<th>$\sigma = 10$ MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nested EM BEM</td>
<td>nested EM BEM</td>
<td>nested EM BEM</td>
</tr>
<tr>
<td>$(p1, s1)$</td>
<td>0.00081 0.00027</td>
<td>0.01777 0.00169</td>
<td>0.04364 0.01301</td>
</tr>
<tr>
<td>$(p1, s2)$</td>
<td>0.00338 0.00227</td>
<td>0.05432 0.01417</td>
<td>0.41844 0.08289</td>
</tr>
<tr>
<td>$(p2, s1)$</td>
<td>0.00045 0.00227</td>
<td>0.00383 0.00170</td>
<td>0.06684 0.00670</td>
</tr>
<tr>
<td>$(p2, s2)$</td>
<td>0.00304 0.00227</td>
<td>0.04924 0.01418</td>
<td>1.16618 0.05393</td>
</tr>
</tbody>
</table>

As noted earlier, when the data gets noisier ($\sigma$ increases) the performance of both methods decreases. In general both methods work well in estimating the underlying block mixture model, though BEM seems to outperform our approach.

The fact that our methodology does not perform better then BEM in this simulation study does not necessarily reduce its potential practical applications. However, it does point us towards interesting questions that we plan on exploring more exhaustively in the near future. For instance, we intend to investigate in detail whether the performance of our approach can be significantly improved with changes in the implementation of the nested EM algorithm (see section 6).
Chapter 4

Model Selection for Block Clustering

The methods described in chapter 3 assume that $K_1$ and $K_2$ are fixed - though in a typical exploratory analysis they are often unknown. In this chapter we describe two tools we developed in order to help select $K_1$ and $K_2$ in a block mixture model.

4.1 The Gradient Function

The block mixture model provides a challenging likelihood surface due to the interaction of row and column effects. We start by providing a tool that enables one to search, through the high dimensional spaces involved, for lack of fit of the current model. This tool is the gradient function [50].

In its original formulation, the gradient function is used to test whether a given latent distribution (say $Q_0$) was the nonparametric maximum likelihood estimator in an infinite dimensional space of mixing distributions [50]. In the space of distribution functions, define a path from $Q_0$ to any other distribution ($Q_1$) as $Q_\alpha = (1 - \alpha)Q_0 + \alpha Q_1$. Notice that for every $\alpha$ this generates an intermediate distribution. Let $L^*(\alpha) = L(Q_\alpha)$ be the likelihood along the above path. Then the derivative of $\ln(L^*(\alpha))$ at $\alpha = 0$ is the directional derivative corresponding to the path from $Q_0$ to $Q_1$ and has the form,

$$D_{Q_0}(Q_1) = \sum_{i=1}^{D} n(i) \left( \frac{L_i(Q_1)}{L_i(Q_0)} - 1 \right).$$

The gradient function is defined as a special case of the directional derivative with degenerate $Q_1$ at $\phi$ and has the form,

$$D_{Q_0}(\phi) := D_{Q_0}(\Delta \phi) = \sum_{i=1}^{D} n(i) \left( \frac{L_i(\phi)}{L_i(Q_0)} - 1 \right).$$
Lindsay (1995) [50] also shows that any $Q$ is a MLE if and only if

$$D_Q(\phi) \geq 0 \forall \phi.$$ 

Therefore if this gradient inequality fails for any candidate $Q$ at some $\phi_0$ then we know that we are not at the maximum (i.e. $Q$ is not the MLE) and by moving some mass to $\phi_0$ we can increase the likelihood.

### 4.2 Gradient function for the block mixture model with Gaussian data

Extending the idea in section 4.1 to block clustering with Gaussian data is fairly simple. Given any fitted row and column model, we ask if we can improve the mixture fit by increasing $K_2$ to $K_2 + 1$, which requires adding one column weight $q(K_2 + 1)$ and a new column mean vector $\mu_{\text{new}} = [\mu(1, K_2 + 1), \mu(2, K_2 + 1), \ldots] = \{\mu(a, K_2 + 1); a = 1, \ldots, K_1\}$.

Given a proposed new mean vector $\mu_{\text{new}}$ we construct the log composite likelihood $L_\pi$ defined along a path with only a scalar parameter $\pi$, where the original columns get masses $(1 - \pi)q(b)$ and the new column with mean $\mu_{\text{new}}$ gets mass $\pi$. The log composite likelihood in equation (3.1.1) is therefore updated to:

$$L_\pi = \sum_{i=1}^{R} \log \left( \sum_{a=1}^{K_1} p(a) \left( \prod_{j=1}^{C} \left( 1 - \pi \sum_{b=1}^{K_2} q(b) k(x_{ij}; \mu(a, b)) + \pi k(x_{ij}; \mu(a, K_2 + 1)) \right) \right) \right) + \sum_{j=1}^{C} \log \left( (1 - \pi) \sum_{b=1}^{K_2} q(b) \left( \prod_{i=1}^{R} \prod_{a=1}^{K_1} p(a) g(x_{i. | a}) k^*(x_{ij}) \right) + \pi \left( \prod_{i=1}^{R} \prod_{a=1}^{K_1} p(a) g(x_{i. | a}) k^*(x_{ij}) \right) \right)$$

Let $D(\mu_{\text{new}}) = \frac{\partial}{\partial \pi} L_\pi |_{\pi=0}$ indicate the gradient function. We have

$$D(\mu_{\text{new}}) = \frac{\partial}{\partial \pi} L_\pi |_{\pi=0} = \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_{ia} \sum_{j=1}^{C} \frac{k^*(x_{ij})}{g(x_{i. | a})} + \sum_{j=1}^{C} \omega_j h^*(x_{.j | K_2 + 1}) - C(R + 1)$$

where

$$k^*(x_{ij}) = k \left( x_{ij}; \mu_{\text{new}}, \sigma^2 \right)$$

$$g(x_{i. | a}) = \prod_{j=1}^{C} \left( \sum_{b=1}^{K_2} q(b) k(x_{ij}) \right)$$

$$h^*(x_{.j | K_2 + 1}) = \prod_{i=1}^{R} \left( \sum_{a=1}^{K_1} p(a) k^*(x_{ij}) \right)$$
and
\[
\omega_j = \frac{1}{\sum_{b=1}^{K_2} q(b) h(x_j|b)} \quad \text{where} \quad h(x_j|b) = \prod_{i=1}^{R} \left( \sum_{a=1}^{K_1} p(a) k(x_{ij}; \mu(a, b), \sigma^2) \right)
\]
\[
\omega_{ia} = \frac{p_a g(x_i|a)}{\sum_{a=1}^{K_1} p_a g(x_i|a)}.
\]

The detailed derivation of \(D(\mu_{\text{new}})\) is given in appendix C.

The same theory used in Lindsay (1995) [50] can be used to show that if \(D(\mu_{\text{new}}) > 0\), then a higher likelihood can be obtained by taking some mass from the original column weights and giving it to the new column component. We will call \(\mu_{\text{new}}\) with \(D(\mu_{\text{new}}) > 0\) a gradient violator. Large violations are evidence that the model fit can be considerably improved.

This leads to a new computational problem, namely to find those \(\mu_{\text{new}}\) in a \(K_1\) dimensional space that are the worst violators. To address this we will apply the generalized EM algorithm to the gradient (see section 4.3). If indeed this leads to a positive maximum value of \(D(\mu_{\text{new}})\), then to obtain a new solution with \(K_1\) row groups and \(K_2 + 1\) column groups we re-run the block clustering algorithm with the estimated mean structure as initial values.

For this approach to work, we also need good initial values for the EM gradient search to find the major violators. To this end, we look for a column vector in the data that is poorly fit by the current model. Using the estimated row labels from the current solution, for each column \(X_j\) in the data matrix, we cluster the values (by row label) and compute cluster means \(\mu_j\). We then select the \(\mu_j\) that gives the largest positive gradient as the initial value in our EM gradient search to find the optimal \(\mu_{\text{new}}\).

Notice that exactly the same approach can be used to explore the addition of a row group, i.e. increasing \(K_1\) to \(K_1 + 1\).

### 4.3 EM on the block mixture gradient function

As mentioned in the previous section, large gradient violators (\(\mu_{\text{new}}\) such that \(D(\mu_{\text{new}}) > 0\)) indicate that we can improve the model fit considerably by shifting some weight to \(\mu_{\text{new}}\). In this section, we explain how we apply the generalized EM to the block mixture gradient to find the worst gradient violators for a given \(Q_0\).

Suppose \(Q_0\) has \(K_1\) row groups and \(K_2\) column groups, and we want to explore whether we can improve the mixture fit by increasing the number of column groups to \(K_2 + 1\). To find the worst gradient violators, we need to maximize the gradient function in equation (4.2.1). Therefore our objective function is
\[
D(\mu_{\text{new}}) = \sum_{i=1}^{R} \sum_{a=1}^{K_i} \omega_{ia} \sum_{j=1}^{C} k^*(x_{ij}) + \omega_j h^*(x_{.j}|K_2 + 1) - C(R + 1).
\]

Notice that \(g(x_{.i}|a)\) is independent of \(\mu_{\text{new}}\) and we can write
\[
D(\mu_{\text{new}}) = \sum_{i=1}^{R} \sum_{a=1}^{K_i} \omega_{ia} \sum_{j=1}^{C} k^*(x_{ij}) + \omega_j h^* \left( \sum_{a=1}^{K_i} p(a) k^*(x_{.a}) \right) - C(R + 1).
\]

where \(\omega_{ija} = \frac{1}{g(x_{.i}|a)}\) and \(h^* \left( x_{.j}|K_2 + 1 \right) = \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k^*(x_{ij}) \right)\). Optimizing \(D(\mu_{\text{new}})\) with respect to \(\mu_{\text{new}}\) is same as optimizing \(D^*(\mu_{\text{new}})\) with respect to \(\mu_{\text{new}}\).

Let \(\mu_{\text{new}}^{(0)}\) be the current value of a possible gradient violator, then we have
\[
\log \left\{ \frac{D^*(\mu_{\text{new}})}{D^*(\mu_{\text{new}}^{(0)})} \right\} = \log \left\{ \frac{\sum_{i=1}^{R} \sum_{a=1}^{K_i} \omega_{ia} \sum_{j=1}^{C} k^*(x_{ij}) + \sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k^*(x_{ij}) \right)}{\sum_{i=1}^{R} \sum_{a=1}^{K_i} \omega_{ia} k^*_{(0)}(x_{ij}) + \prod_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k^*_{(0)}(x_{ij}) \right)} \right\}
\]

where \(k^*(x_{ij}) = k \left( x_{ij}; \mu_{\text{new}}, \sigma^2 \right)\) and \(k^*_{(0)}(x_{ij}) = k \left( x_{ij}; \mu_{\text{new}}^{(0)}, \sigma^2 \right)\).

Using the techniques described in section 3.2.2, the E-step of the EM on \(D(\mu_{\text{new}})\) is
\[
\ell_D = \phi^{(0)} \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \alpha_{ija}^{* (0)} \log k^*(x_{ij}) + \left( 1 - \phi^{(0)} \right) \sum_{j=1}^{C} \sum_{a=1}^{K_i} \beta_{j}^{* (0)} \delta_{ija}^{* (0)} \log k^*(x_{ij})
\]

where
\[
\phi^{(0)} = \frac{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \omega_{ia} \omega_{ija} k^*_{(0)}(x_{ij})}{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \omega_{ia} \sum_{j=1}^{C} \omega_{ija} k^*_{(0)}(x_{ij}) + \sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k^*_{(0)}(x_{ij}) \right)}
\]
\[
\alpha_{ija}^{* (0)} = \frac{\omega_{ija} k^*_{(0)}(x_{ij})}{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \omega_{ia} \omega_{ija} k^*_{(0)}(x_{ij})}
\]
\[
\beta_{j}^{* (0)} = \frac{\omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k^*_{(0)}(x_{ij}) \right)}{\sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k^*_{(0)}(x_{ij}) \right)}
\]
\[
\delta_{ija}^{* (0)} = \frac{p(a) k^*_{(0)}(x_{ij})}{\sum_{a=1}^{K_i} p(a) k^*_{(0)}(x_{ij})}
\]
To obtain the M-step, let 

$$
\epsilon_{ija}^* = \phi^* \alpha_{ija}^* \text{ and } \gamma_{ija}^* = (1 - \phi^*) \beta_j^* \delta_{ija}^*.
$$

Then

$$
\ell_D = \sum_{i=1}^R \sum_{j=1}^C \sum_{a=1}^{K_1} \left( \epsilon_{ija}^{*(0)} + \gamma_{ija}^{*(0)} \right) \log k^* (x_{ij})
$$

and the M-step of the EM on \( D(\mu_{\text{new}}) \) calculates

$$
\hat{\mu}_{a(K_2+1)} = \frac{\sum_{i=1}^R \sum_{j=1}^C \sum_{a=1}^{K_1} \left( \epsilon_{ija}^{*(0)} + \gamma_{ija}^{*(0)} \right) x_{ij}}{\sum_{i=1}^R \sum_{j=1}^C \sum_{a=1}^{K_1} \left( \epsilon_{ija}^{*(0)} + \gamma_{ija}^{*(0)} \right)} \forall a = 1, \ldots, K_1.
$$

See appendix C for more details. With a fixed initial value \( \mu_{\text{new}}^{(0)} \), the EM algorithm to find the optimum \( \mu_{\text{new}}^{*} \) is thus:

**Step 1:** Calculate the weights \( \epsilon_{ija}^{*(t)} \) and \( \gamma_{ija}^{*(t)} \) for \( t = 0, 1, 2, \ldots \).

**Step 2:** Calculate \( \ell_D^{(t)} \) for \( t = 0, 1, 2, \ldots \).

**Step 3:** Update \( \mu_{\text{new}}^{(t)} \) for \( t = 1, 2, \ldots \).

**Step 4:** Repeat steps 1-3 until convergence,

i.e. until \( 0 < \left( \ell_D^{(t)}(\infty) - \ell_D^{(t)} \right) < 0.005 \) where \( \ell_D^{(t)}(\infty) = \ell_D^{(t-1)} + \frac{1}{\epsilon_t} \left( \ell_D^{(t)} - \ell_D^{(t-1)} \right) \).

Once we have optimum \( \mu_{\text{new}}^{\ast} \) (say, \( \mu_{\text{new}}^{\ast} \)) with \( D(\mu_{\text{new}}^{\ast}) > 0 \), to estimate the new solution with \( K_1 \) row groups and \( K_2 + 1 \) column groups, we re-run the block clustering algorithm with \( \left[ \mu_1^{(c)}, \ldots, \mu_{K_2}^{(c)}, \mu_{\text{new}}^{(c)} \right] \) as initial values, where \( \left[ \mu_1^{(c)}, \ldots, \mu_{K_2}^{(c)} \right] \) represents the mean vectors of the existing solution \( Q_0 \) with \( K_1 \) row groups and \( K_2 \) column groups.

One major issue in using the gradient approach to assess model fit is to identify good initial values for the EM gradient search to find the major violators (i.e. to identify good \( \mu_{\text{new}}^{(0)} \)’s). We suggest using the following scheme:

- For each column \( X_{\ast j} \) in the data matrix, cluster its values using the estimated row labels from the current solution \( Q_0 \). This gives a vector of cluster means \( \mu_j \):

$$
\mu_j = \begin{pmatrix}
\mu_{1j} \\
\mu_{2j} \\
\vdots \\
\mu_{K_{2j}} 
\end{pmatrix}
$$

- Calculate \( D(\mu_j) \) for all \( j = 1, \ldots, C \).
• Find \( \mu_{new}^{(0)} \) such that

\[
D(\mu_{new}^{(0)}) = \max_j D(\mu_j)
\]

• Run the EM on the gradient function starting at \( \mu_{new}^{(0)} \) and find the optimum, say \( \mu_{new}^* \)

• Re-run the nested EM on the data starting at \( [\mu_{c_1}^{(c)}, \ldots, \mu_{c_{K_2}}^{(c)}, \mu_{new}^*] \), and estimate the new solution with \( K_1 \) row groups and \( K_2 + 1 \) column groups.

Once again, the same approach can be used when adding a new row group.

### 4.3.1 Illustrative numerical example

In this section we revisit one of the simulated example (example 3.3.3.1 in section 3.3.3) to illustrate how the EM gradient search is used for assessing model fit. The data comprises 110 rows and 11 columns generated using a Gaussian model with a 12 block mean structure (shown in Table 3.1) and variance \( \sigma^2 = 16 \).

The true number of row and column clusters here are \( K_1 = 3 \) and \( K_2 = 4 \). For illustration, we estimate a model with 3 row clusters and 3 column clusters; the estimated block mean structure is shown in Table 4.2.

**Table 4.1.** Rearranged true block mean structure for example 1 - same as Table 3.2

<table>
<thead>
<tr>
<th>10</th>
<th>14</th>
<th>19</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>22</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td>33</td>
<td>37</td>
<td>40</td>
<td>46</td>
</tr>
</tbody>
</table>

| 12.932 | 20.677 | 24.549 |
| 19.968 | 26.287 | 28.460 |
| 32.437 | 36.784 | 43.216 |

Next, to explore the possibility of adding a new column group, we employ the approach described in section 4.3. For each of the 11 columns in the data matrix, we cluster values using the estimated row labels from the solution with 3 row clusters and 3 columns clusters. This produces 11 mean vectors shown in Table 4.3.

<table>
<thead>
<tr>
<th>10</th>
<th>14</th>
<th>19</th>
<th>24</th>
<th>12.932</th>
<th>20.677</th>
<th>24.549</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>22</td>
<td>27</td>
<td>30</td>
<td>19.968</td>
<td>26.287</td>
<td>28.460</td>
</tr>
<tr>
<td>33</td>
<td>37</td>
<td>40</td>
<td>46</td>
<td>32.437</td>
<td>36.784</td>
<td>43.216</td>
</tr>
</tbody>
</table>

All \( \mu_j, j = 1, \ldots, 11 \) have \( D(\mu_j) > 0 \), so by taking any of the mean vectors in table 4.3 as initial values for our EM gradient search we could reach a better solution than the one with \( K_1 = 3 \) and \( K_2 = 3 \). Interestingly, initializing the EM gradient search with any of the mean vectors in Table 4.3 leads to the same \( \mu_{new}^* = [15.826, 41.160, 15.826]^T \). The solution we reach by
Table 4.3. Candidate initial values for the EM gradient search to assess the possibility of adding a column cluster to the solution with $K_1 = 3$ and $K_2 = 3$ in example 3.3.3.1

<table>
<thead>
<tr>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\mu_3$</th>
<th>$\mu_4$</th>
<th>$\mu_5$</th>
<th>$\mu_6$</th>
<th>$\mu_7$</th>
<th>$\mu_8$</th>
<th>$\mu_9$</th>
<th>$\mu_{10}$</th>
<th>$\mu_{11}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.407</td>
<td>41.202</td>
<td>37.305</td>
<td>36.239</td>
<td>45.538</td>
<td>46.104</td>
<td>33.153</td>
<td>31.618</td>
<td>32.943</td>
<td>37.729</td>
<td>45.538</td>
</tr>
</tbody>
</table>

re-running the nested EM starting at $[\mu_1^{(c)}, \mu_2^{(c)}, \mu_3^{(c)}, \mu_{\text{new}}]$ is shown in Table 4.5. Notably, this solution classifies all points with 100% accuracy and provides excellent estimates of the block means.

Table 4.4. Rearranged true block mean structure for example 1 - same as Table 3.2 and 4.1

<table>
<thead>
<tr>
<th>10</th>
<th>14</th>
<th>19</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>22</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td>33</td>
<td>37</td>
<td>40</td>
<td>46</td>
</tr>
</tbody>
</table>

Table 4.5. Rearranged estimated block mean structure for example 1 with $K_1 = 3$ and $K_2 = 4$

<table>
<thead>
<tr>
<th>10.098</th>
<th>13.997</th>
<th>19.603</th>
<th>23.762</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.673</td>
<td>21.397</td>
<td>26.604</td>
<td>30.117</td>
</tr>
<tr>
<td>32.793</td>
<td>37.180</td>
<td>40.609</td>
<td>45.820</td>
</tr>
</tbody>
</table>

4.4 Composite Likelihood Ratio Test (CLRT)

In the previous section, we saw how the gradient methodology can be used to guide us towards progressing larger models, with more row and/or column clusters. However, it should be noted that if $\sigma^2$ is unknown, the likelihood will always increase as the $K$’s increase, up to a perfect fit with $\sigma^2 = 0$. We therefore build a tool to identify suitable $K_1$ and $K_2$, using a composite likelihood ratio test statistic along with a parametric bootstrap critical value.

Suppose we want to test the hypothesis. $H_0 : K_2 = c$ vs $H_0 : K_2 = c + 1$. Here the null model ($Q_0$) is the one with $K_1 = r$ and $K_2 = c$, and the alternative model ($Q_1$) is the one with $K_1 = r$ and $K_2 = c + 1$. We start by estimating $Q_0$ and $Q_1$ on the data, and obtaining the corresponding log composite likelihoods and the composite likelihood ratio $CLR(X) = LCL(Q_1(X)) - LCL(Q_0(X))$.

Since the null distribution for the composite likelihood ratio is ambiguous, we simulate it using a parametric bootstrap approach. We simulate data matrices $X_b$, $b = 1 \ldots B$ from $Q_0(X)$, then estimate both models $Q_0$ and $Q_1$ and compute the composite likelihood ratio $CLR(X_b)$.
on each simulated data set. This allows us to produce a bootstrap p value to compare the two models (see below).

Our gradient methodology is critical for constructing this test. As in many mixture models, there are potentially many local likelihood maxima. If we choose initial values for the EM algorithm to estimate \( Q_0 \) and \( Q_1 \) at random, there would be no guarantee that the alternative likelihood value would be larger than the null value; that is, \( LCL (\hat{Q}_1(X)) > LCL (\hat{Q}_0(X)) \).

However, if we choose an alternative \( \hat{Q}_1(X) \) by applying our gradient approach to \( \hat{Q}_0(X) \), an increase is guaranteed. Hence our proposed strategy is,

Step 1: Estimate \( Q_0 \) on the data \( X \), and obtain \( LCL (\hat{Q}_0(X)) \)

Step 2: Use the EM gradient search (with our proposed data-based initialization) to estimate \( Q_1 \) on the data \( X \) and obtain \( LCL (\hat{Q}_1(X)) \)

Step 3: Calculate \( CLR(X) = LCL (\hat{Q}_1(X)) - LCL (\hat{Q}_0(X)) \)

Step 4: For \( b = 1, \ldots, B \)

(a) Simulate \( X_b \) from \( \hat{Q}_0(X) \), estimate \( Q_0 \) on \( X_b \) and obtain \( LCL (\hat{Q}_0(X_b)) \)

(b) Use the EM gradient search to estimate \( Q_1 \) on \( X_b \) and obtain \( LCL (\hat{Q}_1(X_b)) \)

(c) Calculate \( CLR(X_b) = LCL (\hat{Q}_1(X_b)) - LCL (\hat{Q}_0(X_b)) \)

Step 5: Compute the bootstrap p value for comparing \( Q_0 \) and \( Q_1 \) as

\[
\frac{1}{B} \sum_{b=1}^{B} I( CLR(X) > CLR(X_b) )
\]

Using this strategy, we can formulate a forward model selection scheme as follows:

- Start with \( Q_0 \equiv \{ \text{the model with } K_1 = 1 \text{ and } K_2 = 1 \} \). Use the CLRT to test adding a column cluster or a row cluster to \( Q_0 \):

  \( H_0 : Q_0 \) vs \( H_1 : Q_1^{(a)} \) \( \equiv \{ \text{the model with } K_1 = 1 \text{ and } K_2 = 2 \} \)

  \( H_0 : Q_0 \) vs \( H_1 : Q_1^{(b)} \) \( \equiv \{ \text{the model with } K_1 = 2 \text{ and } K_2 = 1 \} \)

  (a) If neither \( Q_1^{(a)} \) or \( Q_1^{(b)} \) is statistically preferable over \( Q_0 \), then set \( Q_0^{(new)} = Q_0 \).

  (b) If only \( Q_1^{(a)} \) is statistically preferable over \( Q_0 \), then add a column cluster to \( Q_0 \) and set \( Q_0^{(new)} = Q_1^{(a)} \).

  (c) If only \( Q_1^{(b)} \) is statistically preferable over \( Q_0 \), then add a row cluster to \( Q_0 \) and set \( Q_0^{(new)} = Q_1^{(b)} \).
(d) If both $Q^{(a)}$ and $Q^{(b)}$ are statistically preferable over $Q_0$, then adding either a column group or row group is preferable over $Q_0$; then choose the option with the most statistically significant gain (smallest p-value or largest test statistic) and set it as $Q_0^{(new)}$.

- Formulate the new systems of hypotheses and repeat steps (a) to (d) until $Q_0^{(new)} = Q_0$.

4.4.1 Illustrative numerical example

In this section we illustrate how the CLRT forward model selection scheme described in section 4.4 can be used for block mixture models. We simulate data comprising 100 rows and 70 columns using a Gaussian model with block mean structure shown in table 4.6 and variance $\sigma^2 = 4$. Each cell in Table 4.6 shows the true block mean together with the block size. For example, the data in block 1 is generated from a $N(10, 4)$ and consists of 30 rows and 30 columns. Figure 4.1(a) shows the heatmap of the simulated data prior to clustering.

<table>
<thead>
<tr>
<th>Table 4.6. True mean block structure (block sizes in parentheses)</th>
<th>Table 4.7. Estimated mean block structure (standard errors in parentheses) for the model with $K_1 = 3$ and $K_2 = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>(30 × 30)</td>
<td>(30 × 40)</td>
</tr>
<tr>
<td>15</td>
<td>21</td>
</tr>
<tr>
<td>(40 × 30)</td>
<td>(40 × 40)</td>
</tr>
<tr>
<td>27</td>
<td>12</td>
</tr>
<tr>
<td>(30 × 30)</td>
<td>(30 × 40)</td>
</tr>
<tr>
<td>10.008</td>
<td>5.988</td>
</tr>
<tr>
<td>(0.1024)</td>
<td>(0.0701)</td>
</tr>
<tr>
<td>14.950</td>
<td>20.967</td>
</tr>
<tr>
<td>(0.1049)</td>
<td>(0.0987)</td>
</tr>
<tr>
<td>26.940</td>
<td>12.006</td>
</tr>
<tr>
<td>(0.0907)</td>
<td>(0.0719)</td>
</tr>
</tbody>
</table>

Figure 4.1. Heatmaps for the simulated data

We start with the simplest model where $K_1 = 1$ and $K_2 = 1$. We use the composite likelihood
ratio test to compare this with the models with $K_1 = 1$ and $K_2 = 2$, and $K_1 = 2$ and $K_2 = 1$. The estimated composite likelihood ratios for the two comparisons are 147.30 and 2189.36 respectively (see Table 4.8). The parametric bootstrap p-values suggest there is enough statistical evidence to support both adding a column cluster or a row cluster (both p-values are 0 to the second decimal approximation). We choose to add a row cluster because the statistical support for this is stronger (larger CLR). Our new model is thus the one with $K_1 = 2$ and $K_2 = 1$. We then consider the possibility of adding a row cluster or a column cluster to the model with $K_1 = 2$ and $K_2 = 1$. As before, there is enough statistical evidence to support either adding a column cluster or a row cluster, but adding a column group has stronger support (see Table 4.8).

Table 4.8. Results of forward model selection using the CLRT

<table>
<thead>
<tr>
<th>$K_1$</th>
<th>$K_2$</th>
<th>$Q_0$</th>
<th>$Q_1$</th>
<th>$CLR(X)$</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2189.36</td>
<td>0.00</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>147.30</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>113.54</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2179.55</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2785.77</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>207.66</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1.25</td>
<td>0.35</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2.35</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Table 4.8 shows how proceeding sequentially in this fashion does lead to the identification of the correct model; adding a row or a column cluster to $K_1 = 3$ and $K_2 = 2$ leads to p-values of 0.35 and 0.38 respectively.

Table 4.7 shows the estimated block means for the correct model ($K_1 = 3$, $K_2 = 2$) together with their estimated standard errors (e.g. the estimated mean for block 1 is 10.008 and its standard error is 0.1024). Figure 4.1(b) shows the heatmap of the clustered data using the correct model. Here we were able to identify the block labels with 100% accuracy.

4.5 Numerical Study on model selection for block clustering

In section 3.7 we investigated the performance of our methods in estimating a block mixture model using a numerical study. We considered the four basic designs shown in table 3.8, and generated a single $X$ for each design at 3 different levels of sampling errors (see section 3.7.1). Throughout section 3.7.1 it was assumed that $K_1 = 3$ and $K_2 = 2$ were fixed and known.

In this section we use the same simulated data to study the performance of the model selection tools we developed. For each of the $X$ generated in section 3.7.1, we adopt the forward model
selection scheme described earlier, which uses a sequence of composite likelihood ratio tests “guided” by EM gradient searches, to see whether we are capable of identifying the correct model.

**Design** \((p1, s1)\)

The block structure in design \((p1, s1)\) has blocks with relatively even sizes, arranged so that block means increase monotonically within row and column clusters. We apply our forward model selection scheme to each \(X\) generated from \((p1, s1)\) with \(\sigma = 2, \sigma = 5,\) and \(\sigma = 10.\)

Results for the case when \(\sigma = 2\) are shown in Table 4.9. We start with the simplest model where \(K_1 = 1\) and \(K_2 = 1,\) and consider the possibility of adding a row group or a column group; that is, we test the two sets of hypotheses

\[
H_0 : (K_1 = 1 \text{ and } K_2 = 1) \text{ vs } H_1 : (K_1 = 2 \text{ and } K_2 = 1)
\]

and

\[
H_0 : (K_1 = 1 \text{ and } K_2 = 1) \text{ vs } H_0 : (K_1 = 1 \text{ and } K_2 = 2).
\]

Here the CLRT provides evidence for either operation, but adding a row group is preferable (larger \(CLR(X)\)). Hence we proceed forward with the model with \(K_1 = 2\) and \(K_2 = 1.\)

| Table 4.9. Results for forward model selection using the CLRT for design \((p1, s1)\) when \(\sigma = 2\) |
|---|---|---|---|
| \(Q_0\) | \(Q_1\) | \(CLR(X)\) | \(p\) value |
| \(K_1\) | \(K_2\) | \(K_1\) | \(K_2\) | |
| 1 | 1 | 2 | 1 | 2604.41 | 0.00 |
| 1 | 2 | 1 | 1 | 1851.02 | 0.00 |
| 2 | 1 | 3 | 1 | 682.96 | 0.00 |
| 2 | 2 | 2 | 1 | 2767.39 | 0.00 |
| 2 | 2 | 3 | 2 | 1636.45 | 0.00 |
| 2 | 3 | 2 | 1 | 17.33 | 0.00 |
| 3 | 2 | 4 | 2 | 0.05 | 0.89 |
| 3 | 3 | 2 | 3 | 2.36 | 0.20 |

In the next step, we test the two sets of hypotheses:

\[
H_0 : (K_1 = 2 \text{ and } K_2 = 1) \text{ vs } H_1 : (K_1 = 3 \text{ and } K_2 = 1)
\]

and

\[
H_0 : (K_1 = 2 \text{ and } K_2 = 1) \text{ vs } H_0 : (K_1 = 2 \text{ and } K_2 = 2)
\]

and proceed to choose the model with \(K_1 = 2\) and \(K_2 = 2.\) We continue the forward model selection in a similar fashion. Table 4.9 shows that we were able to detect the correct model
(i.e. the model with $K_1 = 3$ and $K_2 = 2$) in design $(p_1,s_1)$ with $\sigma = 2$; adding a row or column cluster to $K_1 = 3$ and $K_2 = 2$ leads to p-values of 0.89 and 0.20 respectively.

We repeat the process with $\sigma = 5$. We start with the simplest model ($K_1 = 1$ and $K_2 = 1$) and use the CLRT for forward model selection. The results in Table 4.10 show that also in this noisier setting we were able to identify the correct model.

Finally, we consider the case with very noisy data ($\sigma = 10$). Again, we use our forward model selection starting with $K_1 = 1$ and $K_2 = 1$. The results in Table 4.11 suggest that our scheme identifies the correct model even with $\sigma = 10$. Notably, at all 3 levels of $\sigma$, the model selection path was the same for design $(p_1,s_1)$.

### Table 4.11. Results for forward model selection using the CLRT for design $(p_1,s_1)$ when $\sigma = 10$

<table>
<thead>
<tr>
<th>$Q_0$</th>
<th>$Q_1$</th>
<th>CLR($X$)</th>
<th>$p$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>2 1</td>
<td>454.63</td>
<td>1.00</td>
</tr>
<tr>
<td>1 2</td>
<td>1 1</td>
<td>351.51</td>
<td>0.00</td>
</tr>
<tr>
<td>2 1</td>
<td>3 1</td>
<td>73.07</td>
<td>0.00</td>
</tr>
<tr>
<td>2 2</td>
<td>3 2</td>
<td>360.06</td>
<td>0.00</td>
</tr>
<tr>
<td>2 2</td>
<td>4 2</td>
<td>26.24</td>
<td>0.00</td>
</tr>
<tr>
<td>3 2</td>
<td>3 3</td>
<td>1.19</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Table 4.12. Results for forward model selection using the CLRT for design $(p_1,s_2)$ when $\sigma = 2$

<table>
<thead>
<tr>
<th>$Q_0$</th>
<th>$Q_1$</th>
<th>CLR($X$)</th>
<th>$p$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>2 1</td>
<td>938.06</td>
<td>0.00</td>
</tr>
<tr>
<td>2 1</td>
<td>3 1</td>
<td>663.18</td>
<td>0.00</td>
</tr>
<tr>
<td>2 2</td>
<td>3 2</td>
<td>1191.84</td>
<td>0.00</td>
</tr>
<tr>
<td>3 2</td>
<td>4 2</td>
<td>2.38</td>
<td>0.00</td>
</tr>
<tr>
<td>3 3</td>
<td>3 3</td>
<td>0.93</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Table 4.13. Results for forward model selection using the CLRT for design $(p_1,s_2)$ when $\sigma = 5$

<table>
<thead>
<tr>
<th>$Q_0$</th>
<th>$Q_1$</th>
<th>CLR($X$)</th>
<th>$p$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>2 1</td>
<td>1082.25</td>
<td>0.00</td>
</tr>
<tr>
<td>1 2</td>
<td>3 1</td>
<td>329.78</td>
<td>0.00</td>
</tr>
<tr>
<td>2 1</td>
<td>2 2</td>
<td>177.95</td>
<td>0.00</td>
</tr>
<tr>
<td>2 2</td>
<td>3 2</td>
<td>197.16</td>
<td>0.00</td>
</tr>
<tr>
<td>3 2</td>
<td>4 2</td>
<td>0.86</td>
<td>0.00</td>
</tr>
<tr>
<td>3 3</td>
<td>3 3</td>
<td>2.24</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Table 4.14. Results for forward model selection using the CLRT for design $(p_1,s_2)$ when $\sigma = 10$

<table>
<thead>
<tr>
<th>$Q_0$</th>
<th>$Q_1$</th>
<th>CLR($X$)</th>
<th>$p$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>2 1</td>
<td>306.37</td>
<td>0.00</td>
</tr>
<tr>
<td>2 1</td>
<td>3 1</td>
<td>36.36</td>
<td>0.00</td>
</tr>
<tr>
<td>2 2</td>
<td>3 2</td>
<td>37.75</td>
<td>0.00</td>
</tr>
<tr>
<td>3 2</td>
<td>4 2</td>
<td>0.09</td>
<td>0.00</td>
</tr>
<tr>
<td>3 3</td>
<td>3 3</td>
<td>3.65</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Design $(p_1,s_2)$

The block means monotonically increase within row and column clusters also in design $(p_1,s_2)$. However, this design comprises a relatively smaller block with few number of rows and columns. Tables 4.12 and 4.13 show the results of forward model selection using CLRT for design $(p_1,s_2)$,
when \( \sigma = 2 \) and \( \sigma = 5 \), respectively. In both cases we are able to detect the correct model.

Table 4.14 shows that our scheme identifies the correct model in design \((p_1, s_2)\) even with very noisy data \((\sigma = 10)\). Again, the paths taken to reach the correct model is the same at all levels of \( \sigma \).

**Design \((p_2, s_1)\)**

In this design the blocks are of relatively even size and the block means in the first column are arranged increasingly whereas those in the second column are arranged in the decreasing order. Intuitively, in this pattern the columns are more different than in pattern \(p_1\). In both the designs with \(p_1\), in the first step, our scheme led to adding a row group to the model with \(K_1 = 1\) and \(K_2 = 1\). But the results for design \((p_2, s_1)\) in Tables 4.15, 4.16, and 4.17 clearly suggest that the first step is prefer adding a column group.

Table 4.15. Results for forward model selection using the CLRT for design \((p_2, s_1)\) when \(\sigma = 2\)

<table>
<thead>
<tr>
<th>(Q_0)</th>
<th>(Q_1)</th>
<th>(CLR(X))</th>
<th>(p) value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(K_1)</td>
<td>(K_2)</td>
<td>(K_1)</td>
<td>(K_2)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>762.70</td>
<td>0.00</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>866.18</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>16.67</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1.35</td>
<td>0.29</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.69</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Table 4.16. Results for forward model selection using the CLRT for design \((p_2, s_1)\) when \(\sigma = 5\)

<table>
<thead>
<tr>
<th>(Q_0)</th>
<th>(Q_1)</th>
<th>(CLR(X))</th>
<th>(p) value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(K_1)</td>
<td>(K_2)</td>
<td>(K_1)</td>
<td>(K_2)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>405.21</td>
<td>0.00</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.00</td>
<td>0.71</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>25.32</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1.17</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Our scheme identifies the correct model in design \((p_2, s_1)\) when data is less noisy \((\sigma = 2\) and \(\sigma = 5\)). When \(\sigma = 10\), there is no statistical evidence for adding either a row group or a column group to the model with \(K_1 = 2\) and \(K_2 = 2\); here, our methods suggest that there are only 4
blocks in data. As a check, let us consider again the estimated block means obtained in section 3.7.1:

Table 4.19. Estimated block means in the design \((p2, s1)\) with \(\sigma = 10\)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9.842</td>
<td>29.532</td>
<td></td>
</tr>
<tr>
<td>11.842</td>
<td>27.156</td>
<td></td>
</tr>
<tr>
<td>17.443</td>
<td>16.345</td>
<td></td>
</tr>
</tbody>
</table>

The estimated means do indeed capture the \(p2\) pattern (increasing in the first column and decreasing in the second column). However, the estimated row means \(\hat{\mu}(1, \cdot)\) and \(\hat{\mu}(2, \cdot)\) are very similar relative to estimated sampling error \(\hat{\sigma} = 10.07\). Therefore, the outcome of our model selection scheme (4 blocks) in this case is not surprising.

**Design \((p2, s2)\)**

The block means increase monotonically within the first column and decrease within the second column also in design \((p2, s2)\). However, this design also includes a relatively smaller block. Tables 4.18, 4.20, and 4.21 show the results of forward model selection using the CLRT in design \((p2, s2)\), when \(\sigma = 2, \sigma = 5\) and \(\sigma = 10\), respectively.

Table 4.20. Results for forward model selection using the CLRT for design \((p2, s2)\) when \(\sigma = 5\)

<table>
<thead>
<tr>
<th>(Q_0)</th>
<th>(Q_1)</th>
<th>(CLR(X))</th>
<th>(p) value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

These tables show that our scheme detects the correct model at all 3 level of \(\sigma\).

The results of this numerical study clearly suggest that the model selection scheme we developed performs well in identifying the correct block mixture model and is effective even when the sampling error underlying the data is large.
We applied the methods introduced in the previous chapters to two genomic datasets. Here we present the results of our analyses.

5.1 Genomic features and sites of enhanced mono-nucleotide microsatellite mutability

It is well known that the rates of different mutagenic processes tend to vary and co-vary along the human genome, in a way that can be associated to local features of the genomic landscape in which these changes occur (see [2], and references therein). Kuruppumullage Don et al. [23] used Multivariate Gaussian Hidden Markov Models (MG-HMM; see [9]) to segment the human genome on the basis of the joint behavior of four mutation rates, namely:

- the rate of single nucleotide substitution
- the rate of small (≤ 30-bp – base pair) insertions
- the rate of small (≤ 30-bp) deletions
- the rate at which mono-nucleotide microsatellite loci alter their repeat number.

In more detail, the authors subdivided the 22 autosomes and the X chromosome of the human genome into non-overlapping 1-Mb (mega base) windows. In each window, they computed the four rates using primate alignments, restricting attention to putatively neutral (i.e. non-functional) portions of the window – this was done to quantify the underlying mutagenic mechanisms without the effects of selection. Further, using a number of publicly available data sets and annotations, the authors computed values for 37 genomic landscape features in each window. We note here that these 37 features have themselves a marked interdependence structure, reflected in strong linear and non-linear co-variation along the genome.
Table 5.1. Description of the 33 genomic landscape features in the microsatellite data

<table>
<thead>
<tr>
<th>Notation</th>
<th>Variable name</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC</td>
<td>GC content</td>
<td>Replication</td>
</tr>
<tr>
<td>SNPd</td>
<td>SNP density</td>
<td></td>
</tr>
<tr>
<td>RepT</td>
<td>Replication timing in human ES cells</td>
<td></td>
</tr>
<tr>
<td>SINE</td>
<td>Number of SINE elements</td>
<td></td>
</tr>
<tr>
<td>LINE</td>
<td>Number of LINE elements</td>
<td></td>
</tr>
<tr>
<td>dna_trans</td>
<td>Number of DNA transposons per window</td>
<td>Transposition</td>
</tr>
<tr>
<td>mir</td>
<td>Number of mammalian interspersed repetitive elements</td>
<td></td>
</tr>
<tr>
<td>alu</td>
<td>Number of Alu elements</td>
<td></td>
</tr>
<tr>
<td>mir</td>
<td>Number of mammalian DNA transposons</td>
<td></td>
</tr>
<tr>
<td>l1</td>
<td>Number of L1-elements</td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>Number of L2-elements</td>
<td></td>
</tr>
<tr>
<td>l1target</td>
<td>Number of L1 target sites</td>
<td></td>
</tr>
<tr>
<td>fRec</td>
<td>Female recombination rates</td>
<td>Recombination</td>
</tr>
<tr>
<td>mRec</td>
<td>Male recombination rates</td>
<td></td>
</tr>
<tr>
<td>NLp</td>
<td>Number of nuclear lamina associated regions</td>
<td></td>
</tr>
<tr>
<td>H3K4me1</td>
<td>Number of H3K4me1 sites per window</td>
<td>Chromatin structure</td>
</tr>
<tr>
<td>pol2</td>
<td>Number of pol-II binding sites per win</td>
<td></td>
</tr>
<tr>
<td>h3K14ac</td>
<td>Number of Histone H3K14 acetylation sites</td>
<td></td>
</tr>
<tr>
<td>mirRNA</td>
<td>Number of miRNA</td>
<td></td>
</tr>
<tr>
<td>triplex</td>
<td>Number of triplex motifs</td>
<td></td>
</tr>
<tr>
<td>inverted</td>
<td>Number of inverted repeats</td>
<td></td>
</tr>
<tr>
<td>quadruplex</td>
<td>Number of G-Quadruplex structure forming motifs</td>
<td></td>
</tr>
<tr>
<td>dna1</td>
<td>Number of DNA-1 hypersensitive sites per window</td>
<td></td>
</tr>
<tr>
<td>CExon</td>
<td>Coverage by coding exons per window</td>
<td>Transcription</td>
</tr>
<tr>
<td>CpG</td>
<td>Number of CpG islands</td>
<td></td>
</tr>
<tr>
<td>nCGm</td>
<td>Number of non-CpG methyl-cytosines</td>
<td>Methylation</td>
</tr>
<tr>
<td>X5hmMc</td>
<td>Number of 5-hydroxymethylcytosines per window</td>
<td></td>
</tr>
<tr>
<td>meth_level</td>
<td>Average value of DNA methylation level for the window</td>
<td></td>
</tr>
<tr>
<td>MS</td>
<td>Number of microsatellites (≤ 8 – bp)</td>
<td>Slippage</td>
</tr>
<tr>
<td>SR</td>
<td>Number of TRs (&lt; 8 – bp)</td>
<td></td>
</tr>
<tr>
<td>telo</td>
<td>Distance to the telomere</td>
<td>Repair</td>
</tr>
<tr>
<td>telomerase_hex</td>
<td>Number of telomerase containing hexamers per win</td>
<td></td>
</tr>
<tr>
<td>centro</td>
<td>Distance to centromere (in bp)</td>
<td></td>
</tr>
</tbody>
</table>

The results in [23] suggested that the windows could be categorized into six “states of neutral variation”. One of these states was characterized by having strongly enhanced mono-nucleotide microsatellite mutability – with “background” levels for the rates of substitutions, insertions and deletions. Only 138 out of a total of 2,558 1Mb windows were allocated to this state. Unlike windows allocated to other states, these windows were interspersed across the genome with no clear preference in terms of chromosomal location, and did not appear to be strongly associated with genome landscape features.
Our goal here is to use block clustering to investigate in more detail the potential relationships between these 138 windows with enhanced microsatellite mutability and their genomic landscape features. In particular, we would like to ascertain whether the 138 windows (which can be treated as reasonably independent observational units since they are far removed from each other) can be clustered into distinct groups based on the genomic landscape features, while simultaneously studying how the pattern of the landscape measurements might characterize the clusters of windows. We will restrict our attention to 33 of the genomic features in [23], which are described in Table 5.1.

For this data analysis, we transformed the raw data by taking normal scores [11, 63] of each of the 33 features separately. This creates sets of values that are matched to the ordering of the original values, but behave as draws from a standard normal distribution. The 4 features that were omitted from our analysis were too bi-modal in nature to be meaningfully transformed using normal scores. This is a very strong “whitening” transformation, but it eliminates any predictive relationships that are strictly univariate from our analysis.

5.1.1 Model selection

As noted earlier, our block clustering approach requires that the number of row and column clusters be fixed at the outset. We developed two tools that are useful in making a decision on these values. As an initial exploration, here we use hierarchical clustering with complete linkage (i.e. \( D_{X,Y} = \max(d_{x,y}), \forall x \in X \text{ and } \forall y \in Y \) [42]) separately on the row and column vectors, to understand the marginal grouping of our data. Figure 5.1 shows the dendograms, which clearly suggest some grouping structure in the windows as well as in the features. In particular they suggest the existence of 3 groups of genomic features and 2 groups of windows.

Next we apply our model selection tools to this data. Table 5.2 summarizes the composite likelihood ratio (CLR) test results, which suggest several solutions one might want to investigate further; these have 15 blocks \((K_1 = 5 \text{ and } K_2 = 3)\), 18 blocks \((K_1 = 6 \text{ and } K_2 = 3)\), and 28 blocks \((K_1 = 7 \text{ and } K_2 = 4)\). Note that here we adopt the forward model selection approach described in section 4.4. In the first step, we test two sets of hypotheses; namely:

\[
H_0 : (K_1 = 1 \text{ and } K_2 = 1) \text{ vs } H_1 : (K_1 = 2 \text{ and } K_2 = 1)
\]

and

\[
H_0 : (K_1 = 1 \text{ and } K_2 = 1) \text{ vs } H_0 : (K_1 = 1 \text{ and } K_2 = 2).
\]

The p-values indicate that the model with two row clusters and one column clusters \((K_1 = 2 \text{ and } K_2 = 1)\) is preferred over the model with one row cluster and one column cluster \((K_1 = 1 \text{ and } K_2 = 1)\), whereas there is no significant difference between the model with one row cluster and two column clusters \((K_1 = 1 \text{ and } K_2 = 2)\) and the model with one row cluster and one
column cluster ($K_1 = 1$ and $K_2 = 1$). Therefore we proceed forward with the model with two row clusters and one column cluster ($K_1 = 1$ and $K_2 = 1$). In the second step, we consider the possibility of adding a row group or a column group to the model with $K_1 = 2$ and $K_2 = 1$, and use the CLR test to evaluate the following two sets of hypotheses:

$$H_0 : (K_1 = 2 \text{ and } K_2 = 1) \text{ vs } H_1 : (K_1 = 3 \text{ and } K_2 = 1)$$

and

$$H_0 : (K_1 = 2 \text{ and } K_2 = 1) \text{ vs } H_0 : (K_1 = 2 \text{ and } K_2 = 2)$$

The p-values indicate that both adding a row group and adding a column group to the model with $K_1 = 2$ and $K_2 = 1$ is statistically significant; we proceed with the model with $K_1 = 2$ and $K_2 = 2$ because it has the highest increase in composite likelihood. We move forward in a similar fashion for the third step, etc.

At the step where we consider adding a row or a column group to the model with $K_1 = 5$ and $K_2 = 3$, the p-values in Table 5.2 do not provide enough statistical evidence to move forward at
Table 5.2. Results of CLR testing for the microsatellite data

<table>
<thead>
<tr>
<th>$Q_0$</th>
<th>$Q_1$</th>
<th>CLR($X$)</th>
<th>$p$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>$K_2$</td>
<td>$K_1$</td>
<td>$K_2$</td>
</tr>
<tr>
<td>1 1</td>
<td>2 1</td>
<td>172.4600</td>
<td>0.00</td>
</tr>
<tr>
<td>1 2</td>
<td></td>
<td>0.0000</td>
<td>1.00</td>
</tr>
<tr>
<td>2 1</td>
<td>3 1</td>
<td>21.3707</td>
<td>0.00</td>
</tr>
<tr>
<td>2 2</td>
<td>0.0000</td>
<td>26.7825</td>
<td>0.00</td>
</tr>
<tr>
<td>2 3</td>
<td>0.0001</td>
<td>96.8163</td>
<td>0.00</td>
</tr>
<tr>
<td>3 2</td>
<td>4 2</td>
<td>33.5559</td>
<td>0.00</td>
</tr>
<tr>
<td>3 3</td>
<td>7.7899</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td>5 2</td>
<td>11.1100</td>
<td>0.06</td>
</tr>
<tr>
<td>4 3</td>
<td>21.1234</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>4 4</td>
<td>5.5781</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td>5 3</td>
<td>6 3</td>
<td>10.5633</td>
<td>0.10</td>
</tr>
<tr>
<td>5 4</td>
<td>7.2119</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>6 3</td>
<td>7 3</td>
<td>3.3011</td>
<td>0.39</td>
</tr>
<tr>
<td>6 4</td>
<td>7.5266</td>
<td>0.21</td>
<td></td>
</tr>
<tr>
<td>5 4</td>
<td>6 4</td>
<td>10.8779</td>
<td>0.01</td>
</tr>
<tr>
<td>5 5</td>
<td>4.7112</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td>6 4</td>
<td>7 4</td>
<td>5.9579</td>
<td>0.11</td>
</tr>
<tr>
<td>6 5</td>
<td>1.5140</td>
<td>0.77</td>
<td></td>
</tr>
<tr>
<td>7 4</td>
<td>8 4</td>
<td>2.8940</td>
<td>0.55</td>
</tr>
<tr>
<td>7 5</td>
<td>2.0320</td>
<td>0.72</td>
<td></td>
</tr>
</tbody>
</table>

5% significance level. So we choose the model with $K_1 = 5$ and $K_2 = 3$ as a reasonable solution for our data.

Obviously, this is not the only reasonable choice. We could keep stepping forward along two avenues: (i) adding a row or column group to the model with $K_1 = 6$ and $K_2 = 3$, or (ii) adding a row or a column group to the model with $K_1 = 5$ and $K_2 = 4$. At 20% significance level there is no evidence to support adding a row or a column group to the model with $K_1 = 6$ and $K_2 = 3$, but at this level of significance we see that it is preferred over the model with $K_1 = 5$ and $K_2 = 3$. So, using avenue (i), the model with $K_1 = 6$ and $K_2 = 3$ would be another reasonable solution for our data. Table 5.2 shows that avenue (ii) leads us to the model with $K_1 = 7$ and $K_2 = 4$ at 20% level of significance; another reasonable solution.

Considering block clusters produced by each of the 3 reasonable solutions, we found the one with 28 blocks ($K_1 = 7$ and $K_2 = 4$) more biologically interpretable. Below we provide some more detail and interpretation for this solution.
5.1.2 Block clustering results

Figure 5.2 shows the heatmaps of the original data and of the data rearranged using the block clustering solution with 28 blocks (7 row clusters and 4 column clusters). The red lines in the heatmap of the rearranged data mark the borders of the 28 blocks, and the numbers shown on the left edge and bottom are the row and column cluster numbers.

In terms of groups of windows, row cluster 4 resembles “background noise” because the values in this cluster are very close to zero across all landscape feature (column) clusters. Interestingly though, the windows in row clusters 1, 2, and 3 show an opposite profile along the landscape features (yellow to blue when moving from left to right) compared to the windows that are in row clusters 6 and 7 (blue to yellow moving from left to right). These patterns are more marked for row clusters 1 and 7, which are perhaps the most interesting clusters of windows.

In terms of groups of features, column cluster 2 appears to be relatively “nondescript” across windows with both yellow and blue in all window (row) clusters. However, the features in column cluster 1 shows an opposite profile along the windows (yellow to blue when moving from top to bottom) compared to the features in column clusters 3 and 4 (blue to yellow moving from top to bottom).
Tables 5.3 and 5.4 show prevalences for the row and column clusters. Interestingly, most features in column cluster 1 (l1, l1target, and LINE) are proxies for transposition, and most features in column cluster 2 (telo, centro, and telomerase_hex) are proxies for repair (see Table 5.1). Also, female and male recombination rates (fRec and mRec) are grouped together in column cluster 3.

The 21 windows in row cluster 1 may represent locations where enhanced microsatellite mutability goes together with enhanced transposition activity and reduced recombination, while the opposite holds for the 15 windows in row cluster 7.

Figure 5.3 shows the 138 windows along the human genome color coded according to their row cluster. The genomic location of the windows does not seem to have a strong association with the 7 window clusters in the block clustering solution. Notably, this is consistent with the fact that distance to telomere (telo) and distance to centromere (centro) belong to column cluster 2, which behaves in a “nondescript” way across all windows (see above).

### 5.2 Genomic features and common fragile sites

As explained by Fungtammasan et al. (2012) [27], chromosomal fragile sites (CFSs) are unstable genomic regions that break under replication stress and are involved in structural variation.
Increased chromosomal instability has been observed at CFSs at early stages of human cancer development and following some genetic diseases [26]. Although these sites are biologically and medically relevant, their molecular characterization is still not well studied. However, recent availability of genome-wide data has made it possible to predict these sites computationally and to associate their instability to their local genomic contexts.

Fungtammasan et al. [27] studied the relationship between CFSs that are specifically induced by cellular treatment with aphidicolin (called APH-induced CFSs) and genomic contexts. Using standard and logistic multivariate regression, the authors modeled the fragility of well-characterized autosomal APH-induced CFSs as a function of their genomic landscape, and contrasted them with nonfragile regions. The authors worked with 76 well characterized APH-induced CFSs and 54 genomic landscape features.

Here, we use the data from Fungtammasan et al. [27] to see whether block clustering can shed further light on the relationships between APH-induced CFSs and genomic landscape features. We employ 49 genomic features (out of the 54 in [27]), which are described in Table 5.5 (for more details see supplementary material of [27]). Two of the features used in [27] (distance to telomere and distance to centromere) were kept aside from block clustering so that they can be used for cluster interpretations, and 3 of the features were removed from our analysis due to the bi-modal nature of the data. We then transformed the 49 remaining features by taking normal scores.
Table 5.5. Description of the 49 genomic landscape features in the common fragile site data

<table>
<thead>
<tr>
<th>Notation</th>
<th>Variable name</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purine</td>
<td>Purine percent</td>
<td>Global genome organization</td>
</tr>
<tr>
<td>GC</td>
<td>GC content</td>
<td></td>
</tr>
<tr>
<td>knownGene</td>
<td>Gene region</td>
<td>Gene expression/chromatin structure</td>
</tr>
<tr>
<td>CpGIsland</td>
<td>CpG island</td>
<td></td>
</tr>
<tr>
<td>TSS</td>
<td>Transcription start sites</td>
<td></td>
</tr>
<tr>
<td>lamin</td>
<td>Nuclear lamina binding sites</td>
<td></td>
</tr>
<tr>
<td>H3K4me1</td>
<td>H3K4me1 sites</td>
<td></td>
</tr>
<tr>
<td>H3K9Ac</td>
<td>H3K9Ac sites</td>
<td></td>
</tr>
<tr>
<td>H3K14Ac</td>
<td>H3K14Ac sites</td>
<td></td>
</tr>
<tr>
<td>pol2</td>
<td>RNA polymerase II binding sites</td>
<td></td>
</tr>
<tr>
<td>mRNA</td>
<td>mRNA sites</td>
<td></td>
</tr>
<tr>
<td>alu</td>
<td>Alu repeats</td>
<td></td>
</tr>
<tr>
<td>MIR</td>
<td>MIR (type of SINE) repeats</td>
<td>DNA sequence elements</td>
</tr>
<tr>
<td>TTTAAA</td>
<td>LINE1 repeats</td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>LINE2 repeats</td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>L1 endonuclease cleavage sites</td>
<td></td>
</tr>
<tr>
<td>DNA</td>
<td>DNA transposon</td>
<td></td>
</tr>
<tr>
<td>MER</td>
<td>MER (type of DNA transposon) repeats</td>
<td></td>
</tr>
<tr>
<td>Twist</td>
<td>Twist value calculated from sequence</td>
<td></td>
</tr>
<tr>
<td>LowComAT</td>
<td>Low complexity A/T rich regions</td>
<td></td>
</tr>
<tr>
<td>mono</td>
<td>Mononucleotide microsatellites (&gt; 9 repeats)</td>
<td></td>
</tr>
<tr>
<td>di</td>
<td>Dinucleotide microsatellites (&gt; 5 repeats)</td>
<td></td>
</tr>
<tr>
<td>tri</td>
<td>Trinucleotide microsatellites (&gt; 4 repeats)</td>
<td></td>
</tr>
<tr>
<td>tetra</td>
<td>Tetranucleotide microsatellites (&gt; 4 repeats)</td>
<td></td>
</tr>
<tr>
<td>allAT</td>
<td>All A/T motif microsatellites</td>
<td></td>
</tr>
<tr>
<td>GQ</td>
<td>G-Quadruplex forming repeats</td>
<td></td>
</tr>
<tr>
<td>ZDNA</td>
<td>Z-DNA motifs</td>
<td></td>
</tr>
<tr>
<td>invert</td>
<td>Inverted repeats</td>
<td></td>
</tr>
<tr>
<td>cruciform</td>
<td>Cruciform motifs</td>
<td></td>
</tr>
<tr>
<td>direct</td>
<td>Directed repeats</td>
<td></td>
</tr>
<tr>
<td>Slip</td>
<td>Slipped motifs</td>
<td></td>
</tr>
<tr>
<td>mirror</td>
<td>Mirror repeats</td>
<td></td>
</tr>
<tr>
<td>triplex</td>
<td>Triplex motifs</td>
<td></td>
</tr>
<tr>
<td>APzase</td>
<td>A-Phased repeats</td>
<td></td>
</tr>
<tr>
<td>RepRate</td>
<td>Replication timing (Woodfine)</td>
<td>DNA replication</td>
</tr>
<tr>
<td>rep3</td>
<td>Replication timing (Ryba)</td>
<td></td>
</tr>
<tr>
<td>f-Lymph_early_rep</td>
<td>Early replicated regions in lymphocyte cell line</td>
<td></td>
</tr>
<tr>
<td>ori_skew</td>
<td>Origins of replication computationally predicted from base skew</td>
<td>DNA replication</td>
</tr>
<tr>
<td>Topo1_CAT</td>
<td>Topoisomerase 1 motif (CAT)</td>
<td></td>
</tr>
<tr>
<td>emphTopol1_CTY</td>
<td>Topoisomerase 1 motif (CTY)</td>
<td></td>
</tr>
<tr>
<td>Topo1_GTY</td>
<td>Topoisomerase 1 motif (GTY)</td>
<td></td>
</tr>
<tr>
<td>Topo1_RAK</td>
<td>Topoisomerase 1 motif (RAK)</td>
<td></td>
</tr>
<tr>
<td>Topo1_YCCTT</td>
<td>Topoisomerase 1 motif (YCCCTT)</td>
<td></td>
</tr>
<tr>
<td>Topo1_YTA</td>
<td>Topoisomerase 1 motif (YTA)</td>
<td></td>
</tr>
<tr>
<td>RecombRate</td>
<td>Sex-averaged recombination rates</td>
<td></td>
</tr>
<tr>
<td>HotSpotRegion</td>
<td>Hotspots of recombination</td>
<td></td>
</tr>
<tr>
<td>hotSpotMotif</td>
<td>Hotspot of recombination motif</td>
<td></td>
</tr>
<tr>
<td>EBR</td>
<td>Evolutionary breakpoint regions</td>
<td></td>
</tr>
<tr>
<td>HSB</td>
<td>homologous synteny blocks</td>
<td></td>
</tr>
</tbody>
</table>

Recombination and mutational pathways
5.2.1 Model selection

As a preliminary exploration we used hierarchical clustering with complete linkage to obtain the dendograms for features and CFSs separately. These are shown in Figure 5.4, and suggest there may be two clusters of CFSs and two clusters of features.

![Dendograms for clustering common fragile sites (rows) and genomic landscape features (columns) using hierarchical clustering with complete linkage.](image)

**Figure 5.4.** Dendograms for clustering common fragile sites (rows) and genomic landscape features (columns) using hierarchical clustering with complete linkage.

Next we apply our model selection tools to the data. Table 5.6 summarizes the composite likelihood ratio (CLR) test results.

As usual we start the forward model selection process with the simplest model ($K_1 = 1$ and $K_2 = 1$), and consider the possibility of adding a row group or a column group. In the first step, we test the following sets of hypotheses:

$$H_0 : (K_1 = 1 \text{ and } K_2 = 1) \text{ vs } H_1 : (K_1 = 2 \text{ and } K_2 = 1)$$

and

$$H_0 : (K_1 = 1 \text{ and } K_2 = 1) \text{ vs } H_0 : (K_1 = 1 \text{ and } K_2 = 2).$$
The p-values indicate that there is sufficient statistical evidence to add a row group to the model ($K_1 = 1$ and $K_2 = 1$), whereas there is no evidence for adding a column group. Therefore, we select the model with $K_1 = 2$ and $K_2 = 1$, and then use CLR testing to evaluate the following two sets of hypotheses:

$$H_0 : (K_1 = 2 \text{ and } K_2 = 1) \text{ vs } H_1 : (K_1 = 3 \text{ and } K_2 = 1)$$

and

$$H_0 : (K_1 = 2 \text{ and } K_2 = 1) \text{ vs } H_0 : (K_1 = 2 \text{ and } K_2 = 2).$$

Now the p-values support adding a column group to the model ($K_1 = 2$ and $K_2 = 1$), but not a row group. Hence we select the model with $K_1 = 2$ and $K_2 = 2$, and proceed with our forward model selection approach in the same fashion. At 5% significance level CLR testing suggests that there are 8 blocks in the data with $K_1 = 4$ and $K_2 = 2$, since adding a row group or column group to the model ($K_1 = 4$ and $K_2 = 2$) is not significant at 5%. One could stop the model selection process here, but we decided to consider also 20% significance level and move forward. At 20%, CLR testing suggests that there are 24 blocks in the data with $K_1 = 8$ and $K_2 = 3$. Below we provide some more details and interpretations for this solution.

### Table 5.6. Results of CLR testing for the common fragile sites data

| $Q_0$ $Q_1$ $CLR(X)$ $p$ value |
|---|---|---|---|
| $K_1$ $K_2$ $K_1$ $K_2$ | 1 2 | 1 14.8535 | 0.00 |
| 1 2 | 0.0000 | 1.00 |
| 2 1 | 3 1 | 0.3219 | 0.21 |
| 2 2 | 245.3162 | 0.00 |
| 2 2 | 3 2 | 113.4147 | 0.00 |
| 2 3 | 0.0000 | 1.00 |
| 3 2 | 4 2 | 28.2137 | 0.01 |
| 3 3 | 3.2719 | 0.09 |
| 4 2 | 5 2 | 18.3705 | 0.09 |
| 4 3 | 9.8629 | 0.12 |
| 5 2 | 6 2 | 3.3563 | 0.28 |
| 5 3 | 13.5038 | 0.11 |
| 5 3 | 6 3 | 19.9944 | 0.03 |
| 5 4 | 11.0498 | 0.13 |
| 6 3 | 7 3 | 11.0676 | 0.20 |
| 6 4 | 4.7057 | 0.53 |
| 7 3 | 8 3 | 12.4317 | 0.10 |
| 7 4 | 6.2520 | 0.44 |
| 8 3 | 9 3 | 3.9405 | 0.52 |
| 8 4 | 7.9066 | 0.38 |
### 5.2.2 Block clustering results

Figure 5.5 shows the heatmaps of the original data and of the rearranged data using the block clustering solution with 24 blocks (8 row clusters and 3 column clusters). The red lines in the heatmap of the rearranged data mark the borders of the 24 blocks. The cluster labels for features (columns) and CFSs (rows) are shown in the boxes to the left and bottom of the panel.

The heatmap of the rearranged data shows that the CFSs in row cluster 1 and 2 are elevated for the features in column clusters 1 and 2 and depressed for the features in column cluster 3, whereas CFSs in row clusters 7 and 8 mostly show the opposite pattern. CFSs in row cluster 3 behave similarly to CFS in row clusters 1 and 2 relative to column clusters 1 and 2, but not relative column cluster 3. CFSs in row clusters 4, 5, and 6 behave more like “background noise” relative to almost all the features.

![Heatmaps for the CFS data with 24 blocks](image)

**Figure 5.5.** Heatmaps for the CFS data with 24 blocks

More generally, even though there are some exceptions to this pattern, if we move from left to right in the heatmap of the rearranged data, we can see that sites (rows) where the features in column cluster 1 and 2 are elevated tend to have depressed features in column cluster 3, and in vice versa.
Table 5.7 shows prevalences and memberships of the column clusters for our solution. Interestingly, column cluster 1 groups several proxies for transcription (i.e. \textit{GQ}, \textit{pol2}, \textit{TSS}, \textit{CpGIsland}, \textit{H3K9AC}, and \textit{GC}) and DNA replication (i.e. \textit{RepRate}, \textit{rep3}, \textit{f\_Lymph\_early\_rep}, and \textit{ori\_skew}). CFSs in row clusters 1, 2 and 3 are elevated for these features whereas CFSs in row cluster 7 and 8 mostly are depressed. Column cluster 2 groups \textit{slip}, \textit{direct}, and \textit{triplex}, which are proxies for secondary DNA structure, as well as some features that proxy transcription (\textit{H3K14AC}, \textit{H3K4me1}, and \textit{knownGene}). Column cluster 3 groups four features that represent the DNA replication due to different Topoisomerase motifs (\textit{TopoI\_CAT}, \textit{TopoI\_RAK}, \textit{TopoI\_YTA}, and \textit{TopoI\_GTY}). Column cluster 3 also includes RNA transposons (\textit{L1} and \textit{L2}), features that measure low complexity with high AT (\textit{Twist}, \textit{LowComAT}, and \textit{TTTTAA}), non-BDNA abundance (\textit{mirror}, \textit{APhase}, \textit{cruciform}, and \textit{di}), and DNA transposons (\textit{MER}, \textit{DNA}).

Table 5.7. Prevalences and memberships of feature (column) clusters in the common fragile sites data \((K_2 = 3)\)

<table>
<thead>
<tr>
<th>Cluster No</th>
<th>Number of members</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9</td>
<td>\textit{H3K14AC}, \textit{H3K4me1}, \textit{knownGene} \textit{slip}, \textit{direct}, \textit{triplex} \textit{HotSpotRegion}, allAT, miRNA</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>\textit{TopoI_CAT}, \textit{TopoI_RAK}, \textit{TopoI_YTA}, \textit{TopoI_GTY} \textit{L1}, \textit{L2} \textit{mirror}, \textit{APhase}, \textit{cruciform}, \textit{di} \textit{Twist}, \textit{LowComAT}, \textit{TTTTAA} \textit{MER}, \textit{DNA} \textit{EBR}, \textit{HSB} \textit{lamin}</td>
</tr>
</tbody>
</table>

see Table 5.5 for a description of the features.

Table 5.8 shows prevalences and memberships of the row clusters for our solution, and Figure 5.6 shows the CFSs along the human genome color coded according to their cluster. Three out of the 6 CFSs that are in cluster 2 (shown in red in Figure 5.6) are located at the very tips of the respective chromosomes and the other three are is still located away from the centromeres. In addition, we see that the row cluster 7 is located away from the telomere. It is hard to observe any other marked associations of CFS clusters and genomic locations in Figure 5.6.

To give a closer look, we produced the profiles of the eight CFS clusters in terms of distance to telomere and distance to centromere (recall these two genomic features were not used in block clustering), which are shown in Figure 5.7. In addition, Figure 5.7 shows the profiles of the eight
Table 5.8. Prevalences and memberships of site (row) clusters in the common fragile sites data ($K_1 = 8$)

<table>
<thead>
<tr>
<th>Cluster No</th>
<th>Number of members</th>
<th>Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>1a, 10g, 11f, 14c, 17b</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>2j, 5f, 7i, 8d, 16c, 22b</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>1d, 1e, 2c, 3d, 4d, 7d, 7e, 9d, 11e, 12c, 15a, 16d</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>1b, 6b, 6g, 7c, 7g, 11c, 20b</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>1f, 1g, 1h, 2f, 2g, 2r, 3c, 4a, 6c, 6e, 6f, 7h, 8c, 9b, 10d, 10e, 11h, 13d, 14b, 18b</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>8b, 10f, 18a</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>1i, 2c, 2d, 3a, 3b, 4f, 5d, 5e, 7f, 7j, 13a, xc</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>1k, 2h, 4c, 5c, 7b, 11d, 11g, 12b, 13c, xc, xd</td>
</tr>
</tbody>
</table>

**Figure 5.6.** Common fragile sites along the human genome, color coded according to row clusters in the block clustering solution with $K_1 = 8$ and $K_2 = 3$.

CFS clusters in terms of two characteristics of the fragile sites themselves; namely, breakage frequency (a quantification of the fragility) and sequence length (a quantification of the size).

We see that the CFSs in row clusters 1 and 2 have similar breakage frequencies, but CFSs in cluster 1 have shorter sequences and are located away from centromere relative to the CFSs in cluster 2 (row clusters 1 and 2 are shown in blue and red, respectively, in Figure 5.6). CFSs in row cluster 7 have the highest breakage frequency. They also have longer sequences and are located closer to centromeres relative to the other row clusters (row clusters 7 is shown in purple in Figure 5.6).
The sites in row clusters 2 and 7 appear to have fairly similar sequence lengths. Sites in row cluster 2 have elevated transcription and replication, depleted transposition, and tend to be close to telomeres. They also have low fragility (low breakage frequency). In contrast, sites in row cluster 7 have depleted transcription and replication, elevated transposition, and are located away from telomeres. Moreover, they have high fragility. Notably, in both row clusters 2 and 7 there is one “odd” site (site 5f in cluster 2 and site 5e in cluster 7), with patterns different from the rest of the sites in the cluster.

CFSs in row clusters 3 and 8 have the smallest median breakage frequency and their median sequence lengths are similar. Most of the CFSs in cluster 8 (in pink in Figure 5.6) are located further from centromeres relative to CFSs in cluster 3 (in yellow in Figure 5.6).

The results of this study might be useful in designing follow up experiments. In practice,
a researcher cannot experimentally validate fragility for all 76 sites, but only a handful. The results of this analysis can be used to select a few sites for experimental validation.
Chapter 6

Discussion and Future Work

The main focus of this thesis has been the development of model-based methods for block clustering. In this chapter, we discuss some interesting observations raised by our work and some open issues that we plan to address in the future. This discussion focuses both on estimation and model selection. We also briefly describe other avenues for future work.

6.1 Estimation

6.1.1 Computation time

In chapter 1, we discussed the computational difficulties that arise when one tries to perform block clustering using mixture models. In particular, we saw that calculating the full likelihood for block clustering requires an intractable number of calculations. This number increases exponentially with the dimensions of data when $K_1$ and $K_2$ fixed. To overcome this issue we proposed an alternative estimation method based on a composite likelihood; if $K_1$ and $K_2$ are fixed, the number of calculations required to assess the composite likelihood increases proportionally to the size of the data, as does the amount of time required to obtain maximum composite likelihood estimates (MCLEs). To illustrate this, we used a simple numerical example.

Consider an experiment with

$$K_1 = 3, K_2 = 2, \sigma = 2, \text{ and } \mu = \begin{bmatrix} 10 & 28 \\ 19 & 14 \\ 15 & 20 \end{bmatrix}.$$ 

Table 6.1 shows the time taken to find MCLEs, using our nested EM approach, for different data sizes. As expected, we see that this time increases proportionally with the dimensions of the data.
Table 6.1. The increase in time to calculate CMLEs with increasing data size

<table>
<thead>
<tr>
<th>Data Size</th>
<th>Time</th>
<th>Time/1000 data points</th>
</tr>
</thead>
<tbody>
<tr>
<td>R  C N = R × C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100 80 8,000</td>
<td>2.8 sec</td>
<td>0.35</td>
</tr>
<tr>
<td>500 320 160,000</td>
<td>69.4 sec ≈ 1 min</td>
<td>0.43</td>
</tr>
<tr>
<td>1000 800 800,000</td>
<td>376.3 sec ≈ 6.3 mins</td>
<td>0.47</td>
</tr>
<tr>
<td>5000 3500 16,500,000</td>
<td>10,239.8 sec ≈ 2.8 hrs</td>
<td>0.62</td>
</tr>
</tbody>
</table>

6.1.2 Statistical efficiency

We observed that using composite likelihood enabled us to perform block clustering with mixture models in practice, even with considerably large datasets. But it is important that we study how effective and accurate our approach is in producing the estimates. We used a numerical example to compare MCLEs to the maximum likelihood estimates (MLEs); in order to obtain MLEs, we had to keep the dimension of the data small (small R and small C). In this example, we generated \( X \) with 10 rows and 6 columns using \( \mu = \begin{bmatrix} 10 & 28 \\ 19 & 15 \end{bmatrix} \) and \( \sigma = 2 \) and computed the MLEs as well as MCLEs. We repeated the exercise with 20 simulated data arrays; table 6.2 shows the estimated standard errors of MLEs and MCLEs.

Table 6.2. Comparison of standard errors of MLEs and MCLEs

<table>
<thead>
<tr>
<th>True Parameters</th>
<th>( p(\cdot) )</th>
<th>( q(\cdot) )</th>
<th>( \mu(\cdot, \cdot) )</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>MLE</td>
<td>CMLE</td>
<td>0.5 0.5</td>
<td>0.5 0.5</td>
</tr>
<tr>
<td></td>
<td>0.000 0.000</td>
<td>0.000 0.000</td>
<td>0.408 0.249</td>
<td>0.305 0.252</td>
</tr>
<tr>
<td></td>
<td>0.000 0.000</td>
<td>0.000 0.000</td>
<td>0.379 0.283</td>
<td>0.304 0.288</td>
</tr>
</tbody>
</table>

We see that the values are very close, indicating that the composite likelihood approach worked well in estimating the block mixture model.

6.1.3 Comparison with BEM

Recall that the numerical study in section 3.7.2 indicated that BEM outperforms our method in estimating the block mixture model. It is of importance to understand the reason behind this observation. Our thinking is that the two methods are very similar in structure, so if we can better understand how the differences arise, we can hope to improve the composite likelihood method in this and other settings. We have investigated several possibilities to date.

First, as mentioned in section 3.7.2, we used the R library `blockcluster` to obtain BEM estimates. As a result, it is not straightforward to diagnose possible sources of differences. It is our
understanding that *blockcluster* implements an extensive search for starting values separately on rows and columns. It is commonly known that multimodality of the likelihood surface greatly affects the performance of model-based approaches, and the solutions of the likelihood equations depend on initialization of the EM [50]. Naturally, our first guess was that the difference in performance between our method and BEM was due to starting values. We also used multiple starting values to initialize our nested EM, but BEM could be doing a better job on this front. To check, we used the BEM estimates as initial values in our nested EM, and observed that the nested algorithm still moved from these to its own, different estimates. Thus, the better performance of BEM seems to be due to factors other than multimodality and initialization.

There are other possible implementation issues that could affect the comparison of the two methods. The stopping rule we use in our nested EM relies on accuracy of the estimated asymptotic convergence rate ($\hat{c}_t$). In section 3.3.3.2 we observed that, even in situations in which $\hat{c}_t$ had not reached its asymptotic value, our methods could work well in estimating the block mixture model. Just the same, our stopping rule has an ad hoc element, as the target accuracy of the final composite likelihood is based on a likelihood criterion (i.e. choosing $\epsilon = 0.005$), and is not tied to the accuracy of the composite likelihood. We could be terminating the EM prematurely. It would be interesting to assess whether in such situations one could obtain better performance by running the algorithm longer.

### 6.1.4 Further enhancements

In constructing the *full* composite likelihood, we combined row and column composite likelihoods without any weighting (see section 3.1). Lindsay et al [48] presented an optimal weighting strategy for composite likelihoods; though adopting such a strategy might complicate the computations in our setting, it might be worth exploring whether it improves estimation performance. In the long run we would like to tackle the efficiency question with new flexible tools.

We also think it may be promising to replace or complement the composite likelihood approach with a Monte Carlo likelihood approach [21, 56]. With such approach one constructs a simulated likelihood by sampling the hidden labels from $P(a,b|X,\Theta)$, and uses it in place of the composite likelihood we utilized.

### 6.2 Model selection

One of the major current obstacles in using composite likelihood methods is the need for more model building and selection tools. To ameliorate this, in this thesis we proposed two useful tools: a gradient approach for finding good models and a “bootstrap-based” composite likelihood ratio test for selection. But developing more tools, including graphical diagnostics where possible, would complement our proposals and help enhancing the usability and popularity of composite
likelihood approximations.

6.2.1 Cross validation

Following Wang and Lindsay [71, 72], we investigated the possibility of using a likelihood cross validation approach for model selection and carried out a numerical study (not presented in this thesis). This provided valuable insights about the potential of likelihood cross validation in model selection. In particular, it led us to the idea, currently under development, of a predicted residual sum of squares ($R^2$) which compares estimation performance for alternative models of interest.

6.2.2 Stepwise selection

One disadvantage of our forward model selection approach is that it may skip past some interesting solutions. To avoid this, one could compare all possible models (i.e. combinations of $K_1$ and $K_2$), but this would require a large amount of computation. Another possibility would be to devise and implement a stepwise model selection approach. Recall that our gradient tool proved to be useful in finding EM initial values when moving forward from a smaller model to a larger model (see section 4.4). In moving backward, we need to collapse two column/row groups, and the collapsibility score could be a useful tool in determining the “best two groups” to collapse.

One other interesting avenue we wish to pursue is to develop a model selection approach that guarantees “hierarchical block clustering”. We anticipate that searching for the modes [47] in row and column marginal densities could be a potential way to develop such a tool.

6.3 Other directions

In addition to the observations and plans above, here we briefly list some other ideas that we wish to pursue in the future.

6.3.1 Combining blocks for parsimony

The block mixture model described in this thesis is applicable only for two-way analyses where one seeks “checker-board” arrangements in the data. In practice though, more flexible models are often needed to characterize other interesting two-way patterns ([14, 68, 73]). Much further work is required to solve the computational complexities arising in such models. One avenue could be to pursue constrained maximum likelihood estimation: we would first use our methods to find checker-board patterns in the data, and then improve the partitioning outcomes by looking at the resulting blocks and introducing further constraints to the block mean structure.
6.3.2 Muti-way arrays

Also, our methods can be extended to detect *multi-way structures*. Our specific interest is to develop methodology that would allow the detection of three-way structures, as well as the simultaneous handling of order-less dimensions (where clustering is required) and order-sensitive dimensions (where segmentation is required). This methodology would find immediate application in genomics. For example, following up on analyses we already published [23], it could be used to investigate the interdependencies among different biochemical processes and genomic landscape features on different chromosomes and DNA regions - while taking into account the contiguous nature of the genome.

6.3.3 Other data types

Finally, we wish to extend the methods developed here to allow *non-Gaussian component densities* in the model, especially to integrate binary and categorical data. For instance, a popular example in the block clustering literature concerns US congressional voting records. This is a publicly available dataset comprising voting records for 435 members of the 98th congress on 16 different key issues. The votes were recorded in one of the four categories: yes, no, abstain, or absent ([73]). A block mixture model would indeed allow one to discover patterns by party and issue in this example. It would also be applicable in a variety of social sciences and genome sciences setting where binary and categorical encodings of the features of interest are routinely used. However, it will take substantial work to extend the methods proposed in this thesis to categorical variables.
Appendix A

Maximum composite likelihood estimates of the block mixture model via nested EM

In chapter 3 we derive the composite likelihood for the block mixture model. Here we provide the details of the nested EM algorithm we use to obtain the maximum composite likelihood estimates (MCLE) for this model.

The log (joint) composite likelihood for the block mixture model is (as in equation 3.1.1)

$$\log L_{RC}(\Theta) = \sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} p(a) \left( \prod_{j=1}^{C} \sum_{b=1}^{K_2} q(b) k(x_{ij}; \mu(a,b)) \right) \right\} + \log L_R(\Theta) + \sum_{j=1}^{C} \log \left\{ \sum_{b=1}^{K_2} q(b) \left( \prod_{i=1}^{R} \sum_{a=1}^{K_1} p(a) k(x_{ij}; \mu(a,b)) \right) \right\} \right\}$$

To obtain the MCLEs, we devise an EM algorithm based on the methods described in section 3.2.2. As explained in section 3.2.2, we think of the EM algorithm as a general optimization tool that replaces any “log of sums” in the objective function with a “sum of logs”. We start by looking at $\log L_R(\Theta)$.

$$\log L_R(\Theta) = \sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} p(a) q(x_{i*}; a, Q) \right\}$$
where \( g(x_{is}; a, Q) = \prod_{j=1}^{C} \sum_{b=1}^{K_2} q(b)k(x_{ij}; \mu(a, b)) \). For any \( \Theta_0 \) (a current value of \( \Theta \)) we have

\[
\log L_R(\Theta) - \log L_R(\Theta_0) = \sum_{a=1}^{R} \log \left\{ \sum_{i=1}^{K_1} p(a)g(x_{is}; a, Q) \right\} - \sum_{a=1}^{R} \log \left\{ \sum_{i=1}^{K_1} p_0(a)g(x_{is}; a, Q_0) \right\}
\]

\[
= \sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} \frac{p(a)g(x_{is}; a, Q)}{\sum_{a=1}^{K_1} p_0(a)g(x_{is}; a, Q_0)} \right\}
\]

\[
= \sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} \frac{p(a)g(x_{is}; a, Q)}{\sum_{a=1}^{K_1} p_0(a)g(x_{is}; a, Q_0)} \right\} \frac{p_0(a)g(x_{is}; a, Q_0)}{\sum_{a=1}^{K_1} p_0(a)g(x_{is}; a, Q_0)}
\]

Now let

\[
\omega_i(a) = \frac{p_0(a)g(x_{is}; a, Q_0)}{\sum_{a=1}^{K_1} p_0(a)g(x_{is}; a, Q_0)},
\]

then we have

\[
\log L_R(\Theta) - \log L_R(\Theta_0) = \sum_{a=1}^{R} \log \left\{ \sum_{i=1}^{K_1} \omega_i(a) \frac{p(a)g(x_{is}; a, Q)}{p_0(a)g(x_{is}; a, Q_0)} \right\}
\]

where the \( \omega_i(a) \) depend on \( \Theta \) only through the current value \( \Theta_0 \), and are such that \( \omega_i(a) \geq 0 \) and \( \sum_{a=1}^{K_1} \omega_i(a) = 1 \). Using Jensens’ inequality with weights \( \omega_i(a) \) we have

\[
\sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} \omega_i(a) \frac{p(a)g(x_{is}; a, Q)}{p_0(a)g(x_{is}; a, Q_0)} \right\} \geq \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_i(a) \log \left\{ \frac{p(a)g(x_{is}; a, Q)}{p_0(a)g(x_{is}; a, Q_0)} \right\}
\]

Maximizing the right hand side of the above equation over \( \Theta \) results in \( L_R(\Theta) \geq L_R(\Theta_0) \). Therefore the right hand side of the above equation represent an EM likelihood which we indicate with \( \ell_{R,em1} \), and call the outer EM:

\[
\ell_{R,em1} = \sum_{a=1}^{K_1} \sum_{i=1}^{R} \omega_i(a) \log p(a) + \sum_{a=1}^{K_1} \sum_{i=1}^{R} \omega_i(a)p(a)g(x_{is}; a, Q)
\]

\[
= \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_i(a) \log p(a) + \sum_{a=1}^{K_1} \sum_{i=1}^{R} \omega_i(a) \left\{ \sum_{j=1}^{C} \log \sum_{b=1}^{K_2} k(x_{ij}; \mu(a, b))q(b) \right\} \right\} \ell_{z}(\Theta)
\]

Notice that the parameters \( p(a) \) separate out, and we can obtain an explicit M-step update for \( p(\cdot) \) as shown below:

\[
\ell_1 = \sum_{a=1}^{K_1} \sum_{i=1}^{R} \omega_i(a) \log p(a) + \sum_{a=1}^{K_1} \sum_{i=1}^{R} \omega_i(a)p(a)g(x_{is}; a, Q) + \lambda \left( 1 - \sum_{a=1}^{K_1} p(a) \right)
\]
As a consequence, we have

\[ \frac{\partial \ell_1}{\partial p(a)} = 0 \Rightarrow \sum_{i=1}^{R} \frac{\omega_i(a)}{p(a)} - \lambda = 0 \Rightarrow \sum_{i=1}^{R} \frac{\omega_i(a)}{p(a)} = \lambda \Rightarrow \sum_{i=1}^{R} \frac{\omega_i(a)}{\lambda} = p(a) \]

\[ \frac{\partial \ell_1}{\partial \lambda} = 0 \Rightarrow 1 - \sum_{a=1}^{K_i} p(a) = 0 \Rightarrow \sum_{a=1}^{K_i} p(a) = 1 \Rightarrow \sum_{a=1}^{K_i} \sum_{i=1}^{R} \frac{\omega_i(a)}{\lambda} = 1 \Rightarrow \sum_{a=1}^{K_i} \sum_{i=1}^{R} \omega_i(a) = \lambda \]

\[ \lambda = \sum_{i=1}^{R} \sum_{a=1}^{K_i} \frac{p_0(a)g(x_{i*}; a, Q_0)}{\sum_{a=1}^{K_i} p_0(a)g(x_{i*}; a, Q_0)} = \sum_{i=1}^{R} 1 = R \Rightarrow p(a) = R^{-1} \sum_{i=1}^{R} \omega_i(a) \]

\[ p(a) = R^{-1} \sum_{i=1}^{R} \frac{p_0(a)g(x_{i*}; a; Q_0)}{\sum_{a=1}^{K_i} p_0(a)g(x_{i*}; a, Q_0)}. \]

This is the classic M-step update for mixture weights. However, this outer EM is still difficult to maximize for \( q(\cdot) \) and \( \mu(\cdot, \cdot) \)'s. We therefore add an inner EM step to decompose the term \( \ell_2(\Theta) \) in \( \ell_{R,em_1} \). Noting that \( g(x_{ij}; a, Q) \) is itself a mixture density, we obtain a nested EM representation that can be explicitly maximized over \( q \) and \( \mu \):

\[ \ell_{R,em_1} = \sum_{i=1}^{R} \sum_{a=1}^{K_i} \omega_i(a) \log p(a) + \sum_{i=1}^{R} \sum_{a=1}^{K_i} \omega_i(a) \left\{ \sum_{j=1}^{C} \log \left[ \sum_{b=1}^{K_2} k(x_{ij}; \mu(a, b))q(b) / k(x_{ij}; \mu(a, b))q_0(b) \right] \right\} \]

\[ \ell_2(\Theta) - \ell_2(\Theta_0) \]

\[ \omega_{ij}(a, b) = \frac{k(x_{ij}; \mu_0(a, b))q_0(b)}{\sum_{b=1}^{K_2} k(x_{ij}; \mu_0(a, b))q_0(b)} \]

with \( \omega_{ij}(a, b) \geq 0 \) and \( \sum_{b=1}^{K_2} \omega_{ij}(a, b) = 1 \). Using Jensens'
inequality we have

\[ \ell_2(\Theta) - \ell_2(\Theta_0) \geq \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_i(a) \left\{ \sum_{j=1}^{C} \sum_{b=1}^{K_2} \omega_{ij}(a, b) \log \left[ \frac{k(x_{ij}; \mu(a, b))q(b)}{k(x_{ij}; \mu_0(a, b))q_0(b)} \right] \right\} \]

\[ = \sum_{i=1}^{R} \sum_{a=1}^{K_1} C \sum_{j=1}^{K_2} \omega_i(a) \omega_{ij}(a, b) \log \left[ \frac{k(x_{ij}; \mu(a, b))q(b)}{k(x_{ij}; \mu_0(a, b))q_0(b)} \right] \quad (A.0.2) \]

Notice that a parameter \( \Theta \) that maximizes the right hand side will increase \( \ell_2(\Theta) \), which will in turn ensure an increase in \( L_R(\Theta) \). Now consider

\[ \omega_i(a)\omega_{ij}(a, b) = \frac{p_0(a)g(x_{ij}; a; Q_0)}{\sum_{a=1}^{K_1} p_0(a)g(x_{ij}; a, Q_0)} \frac{k(x_{ij}; \mu_0(a, b))q_0(b)}{\sum_{b=1}^{K_2} k(x_{ij}; \mu_0(a, b))q_0(b)} \]

\[ = p_0(a)q_0(b) \frac{g(x_{ij}; a, Q_0)}{\sum_{a=1}^{K_1} p_0(a)g(x_{ij}; a, Q_0)} \frac{k(x_{ij}; \mu_0(a, b))q_0(b)}{\sum_{b=1}^{K_2} k(x_{ij}; \mu_0(a, b))q_0(b)} \]

\[ = p_0(a)q_0(b) \right] \quad (A.0.3) \]

Combining (A.0.1) and (A.0.2) we obtain a new EM likelihood which we indicate with \( \ell_{R, em_2} \) and call inner EM:

\[ \ell_{R, em_2} = \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_i(a) \log p(a) + \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \omega_{ij}(a, b) \log \left[ k(x_{ij}; \mu(a, b))q(b) \right] \]

Notice that increasing \( \ell_{R, em_2} \) guarantees increasing \( \ell_{R, em_1} \) which in turn guarantees an increase in the likelihood.

In a similar fashion, we can create a nested EM representation for \( L_C(\Theta) \). We will have

\[ \ell_{C, em_2} = \sum_{j=1}^{C} \sum_{b=1}^{K_2} \psi_j(b) \log q(b) + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{C} \sum_{a=1}^{K_1} \psi_{ij}(a, b) \log \left[ k(x_{ij}; \mu(a, b))p(a) \right] \]

where

\[ \psi_{ij}(a, b) = \psi_j(b)\psi_{ij}(a, b) \]

\[ = \left\{ \frac{q_0(b)h(x_{ij}; a, P_0)}{\sum_{a=1}^{K_1} q_0(b)h(x_{ij}; a, P_0)} \right\} \left\{ \frac{k(x_{ij}; \mu_0(a, b))p_0(a)}{\sum_{a=1}^{K_1} k(x_{ij}; \mu_0(a, b))p_0(a)} \right\} \]

\[ = p_0(a)q_0(b) \frac{h(x_{ij}; a, P_0)k(x_{ij}; \mu_0(a, b))}{h(x_{ij}; a, P_0)h(x_{ij}; a, P_0)}. \]
Finally, to devise an EM algorithm to obtain the maximum composite likelihood estimates for the block mixture model, we combine the EM likelihoods $\ell_{R,em_2}$ and $\ell_{C,em_2}$ as

$$\ell_{RC,em_2} = \ell_{R,em_2} + \ell_{C,em_2}.$$  \hspace{1cm} (A.0.4)

The E-step of the nested EM evaluates $\ell_{RC,em_2}$, and to obtain the M-step we maximize $\ell_{RC,em_2}$ with respect to $p(\cdot), q(\cdot)$ and $\mu(\cdot, cdot)s^\prime$. 
Appendix B

Maximum composite likelihood estimates via nested EM: the Gaussian data case

Appendix A provides the details concerning the E-step of the nested EM algorithm to obtain the MCLEs of the block mixture model. Here we present the derivation for the M-step of the nested EM with Gaussian data; that is where \( k(x_{ij}; \mu(a, b)) = \mathcal{N}(x_{ij}; \mu(a, b), \sigma^2) \). Consider

\[
\ell_3 = \ell_{R, em_2} + \ell_{C, em_2} + \lambda_1 \left( 1 - \sum_{a=1}^{K_1} p(a) \right) + \lambda_2 \left( 1 - \sum_{b=1}^{K_2} q(b) \right)
\]

\[
= \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_i(a) \log p(a) + \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \omega^*_{ij}(a, b) \log k(x_{ij}; \mu(a, b)) +
\]

\[
\sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \sum_{a=1}^{K_1} \psi^*_j(b) \log q(b) + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \sum_{a=1}^{K_1} \psi^*_j(b) \log q(b) +
\]

\[
\sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \sum_{a=1}^{K_1} \psi^*_j(b) \log k(x_{ij}; \mu(a, b)) + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \sum_{a=1}^{K_1} \psi^*_j(b) \log p(a) +
\]

\[
\lambda_1 \left( 1 - \sum_{a=1}^{K_1} p(a) \right) + \lambda_2 \left( 1 - \sum_{b=1}^{K_2} q(b) \right).
\]
Now,

\[ \frac{\partial \ell_3}{\partial p(a)} = 0 \Rightarrow \sum_{i=1}^{R} \frac{\omega_i(a)}{p(a)} + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \frac{\psi_{ij}^*(a,b)}{p(a)} - \lambda_1 = 0 \]

\[ p(a) = \frac{\sum_{i=1}^{R} \omega_i(a) + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \psi_{ij}^*(a,b)}{\lambda_1} \]

\[ \frac{\partial \ell_3}{\partial \lambda_1} = 0 \Rightarrow \sum_{a=1}^{K_1} \sum_{i=1}^{R} p(a) - 1 = 0 \Rightarrow \sum_{a=1}^{K_1} p(a) = 1 \]

by substituting for \( p(a) \) we obtain

\[ \lambda_1 = \sum_{a=1}^{K_1} \sum_{i=1}^{R} \omega_i(a) + \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \psi_{ij}^*(a,b). \]

We know that \( \sum_{a=1}^{K_1} \sum_{i=1}^{R} \omega_i(a) = R \) and

\[ \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \psi_{ij}^*(a,b) = \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \psi_{ij}(a,b) = \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \psi_{ij}(a,b) = \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \psi_{ij}(a,b) \]

\[ = R \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \psi_{ij}(a,b) = \frac{q_0(b)h(x_s; bP_0)}{\sum_{b=1}^{K_2} q_0(b)h(x_s; b, P_0)} = RC. \]

Therefore we have \( \lambda_1 = R + RC \), and hence

\[ p(a) = \frac{\sum_{i=1}^{R} \omega_i(a) + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \psi_{ij}^*(a,b)}{R + RC}. \]

Also

\[ \frac{\partial \ell_3}{\partial q(b)} = 0 \Rightarrow \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \omega_{ij}^*(a,b) q(b) + \sum_{j=1}^{C} \psi_j(b) q(b) - \lambda_2 = 0 \]

\[ q(b) = \frac{\sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \omega_{ij}(a,b) + \sum_{j=1}^{C} \psi_j(b)}{\lambda_2} \]

\[ \frac{\partial \ell_3}{\partial \lambda_2} = 0 \Rightarrow \sum_{b=1}^{K_2} q(b) - 1 = 0 \Rightarrow \sum_{b=1}^{K_2} q(b) = 1 \]

by substituting for \( q(b) \) we obtain

\[ \lambda_2 = \sum_{b=1}^{K_2} \sum_{j=1}^{C} \psi_j(b) + \sum_{b=1}^{K_2} \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \omega_{ij}^*(a,b). \]
Similarly, we know that \( \sum_{b=1}^{K_2} \sum_{j=1}^{C} \omega^*_{ij}(a, b) = C \) and

\[
\sum_{b=1}^{K_2} \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega^*_{ij}(a, b) = \sum_{b=1}^{K_2} \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \omega_{ij}(a, b) = R \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \omega_{ij}(a, b)
\]

\[
= \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \omega_{ij}(a, b) \frac{k(x_{ij}; \mu_0(a, b)) q_0(b)}{\sum_{b=1}^{K_2} k(x_{ij}; \mu_0(a, b)) q_0(b)}
\]

\[
= C \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \omega_{ij}(a, b) \frac{p_0(a) g(x_{ij}; a, Q_0)}{\sum_{a=1}^{K_1} \sum_{i=1}^{R} \sum_{j=1}^{C} p_0(a) g(x_{ij}; a, Q_0)}
\]

\[
\sum_{b=1}^{K_2} \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \omega^*_{ij}(a, b) = R C.
\]

Therefore we have \( \lambda_2 = C + RC \), and hence

\[
q(b) = \frac{\sum_{j=1}^{C} \psi_j(b)}{C + RC} + \frac{\sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \omega^*_{ij}(a, b)}{C + RC}.
\] (B.0.2)

Also

\[
\frac{\partial \ell_3}{\partial \mu(a, b)} = \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{\omega^*_{ij}(a, b)}{k(x_{ij}; \mu(a, b))} k'(x_{ij}; \mu(a, b)) + \sum_{j=1}^{C} \sum_{i=1}^{R} \frac{\psi^*_{ij}(a, b)}{k(x_{ij}; \mu(a, b))} k'(x_{ij}; \mu(a, b))
\]

\[
\frac{\partial \ell_3}{\partial \mu(a, b)} = 0 \Rightarrow \sum_{i=1}^{R} \sum_{j=1}^{C} \left( \frac{\omega^*_{ij}(a, b) + \psi^*_{ij}(a, b)}{k(x_{ij}; \mu(a, b))} \right) \frac{k'(x_{ij}; \mu(a, b))}{k(x_{ij}; \mu(a, b))} = 0.
\] (B.0.3)

Since

\[
k(x_{ij}; \mu(a, b)) = \frac{1}{\sqrt{2\pi \sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} (x_{ij} - \mu(a, b))^2 \right\}
\]

we have

\[
k'(x_{ij}; \mu(a, b)) = \frac{1}{\sqrt{2\pi \sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} (x_{ij} - \mu(a, b))^2 \right\} \left\{ -\frac{1}{2\sigma^2} 2(x_{ij} - \mu(a, b))(1) \right\}
\]

\[
k'(x_{ij}; \mu(a, b)) = \frac{k(x_{ij}; \mu(a, b))}{\sigma^2} (x_{ij} - \mu(a, b)).
\]
Using (B.0.3) we obtain
\[
\sum_{i=1}^{R} \sum_{j=1}^{C} \left\{ \omega_{ij}^*(a, b) + \psi_{ij}^*(a, b) \right\} \frac{k(x_{ij}; \mu(a, b))(x_{ij} - \mu(a, b))}{\sigma^2 k(x_{ij}; \mu(a, b))} = 0
\]

\[
\mu(a, b) = \frac{\sum_{i=1}^{R} \sum_{j=1}^{C} \left\{ \omega_{ij}^*(a, b) + \psi_{ij}^*(a, b) \right\} x_{ij}}{\sum_{i=1}^{R} \sum_{j=1}^{C} \left\{ \omega_{ij}^*(a, b) + \psi_{ij}^*(a, b) \right\}}.
\] (B.0.4)

Finally
\[
\frac{\partial \ell_3}{\partial \sigma^2} = \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \omega_{ij}^*(a, b) \frac{\partial k'}{\partial (x_{ij}; \mu(a, b))} + \sum_{j=1}^{C} \sum_{a=1}^{K_1} \sum_{b=1}^{K_2} \frac{\partial k'}{\partial (x_{ij}; \mu(a, b))} k'_{\sigma}(x_{ij}; \mu(a, b))
\]
\[
= \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \left\{ \omega_{ij}^*(a, b) + \psi_{ij}^*(a, b) \right\} \frac{k(x_{ij}; \mu(a, b))}{k'_{\sigma}(x_{ij}; \mu(a, b))}
\]

where
\[
k'_{\sigma}(x_{ij}; \mu(a, b)) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} (x_{ij} - \mu(a, b))^2 \right\} \left\{ -\frac{1}{2\sigma^2} + \frac{(x_{ij} - \mu(a, b))^2}{2(\sigma^2)^2} \right\}
\]
\[
= \left( \frac{1}{2\sigma^2} \right) k(x_{ij}; \mu(a, b)) \left\{ \frac{(x_{ij} - \mu(a, b))^2}{\sigma^2} - 1 \right\}
\]

and therefore
\[
\frac{\partial \ell_3}{\partial \sigma^2} = 0 \Rightarrow \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \left\{ \omega_{ij}^*(a, b) + \psi_{ij}^*(a, b) \right\} \left( \frac{1}{2\sigma^2} \right) \left\{ \frac{(x_{ij} - \mu(a, b))^2}{\sigma^2} - 1 \right\} = 0
\]

\[
\sigma^2 = \frac{\sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \left\{ \omega_{ij}^*(a, b) + \psi_{ij}^*(a, b) \right\} (x_{ij} - \mu(a, b))^2}{\sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \left\{ \omega_{ij}^*(a, b) + \psi_{ij}^*(a, b) \right\}}
\]

\[
\sigma^2 = \frac{\sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \left\{ \omega_{ij}^*(a, b) + \psi_{ij}^*(a, b) \right\} (x_{ij} - \mu(a, b))^2}{2RC}
\] (B.0.5)

The M-step updates for the nested EM with Gaussian data are given in (B.0.1), (B.0.2), (B.0.4), and (B.0.5).

### B.1 Construction with missing data

As with any EM, the above nested EM construction for estimating MCLEs works also with missing data, as long as they are missing at random. In this section we present the E-step and M-step updates when the data matrix \( X \) is not completely observed and some values in \( X \) are missing at random.
Assume that for each row $i$ in the matrix we observe $C_i$ ($C_i \leq C$) values, and for each column $j$ we observe $R_j$ ($R_j \leq R$) values. The E-step of the nested EM will calculate

$$
\ell_{RC,em_2} = \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_i(a) \log p(a) + \sum_{i=1}^{R} \sum_{a=1}^{K_1} \sum_{j=1}^{C} \sum_{b=1}^{K_2} \omega^*_ij(a, b) \log [k(x_{ij}; \mu(a, b)) q(b)]
$$

$$
\quad + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \psi_j(b) \log q(b) + \sum_{j=1}^{C} \sum_{b=1}^{K_2} \sum_{i=1}^{R} \sum_{a=1}^{K_1} \psi^*_ij(a, b) \log [k(x_{ij}; \mu(a, b)) p(a)]
$$

where

$$
\omega_i(a) = \frac{p_0(a) g(x_{i*}; a, Q_0)}{\sum_{a=1}^{K_1} p_0(a) g(x_{i*}; a, Q_0)} \quad \text{with} \quad g(x_{i*}; a, Q_0) = \prod_{b=1}^{K_2} q_0(b) k(x_{ij}; \mu(a, b), \sigma^2)
$$

$$
\omega^*_ij(a, b) = \omega_i(a) \omega_{ij}(a, b) \quad \text{with} \quad \omega_{ij}(a, b) = \frac{k(x_{ij}; \mu(a, b), \sigma^2_0) q_0(b)}{\sum_{b=1}^{K_2} k(x_{ij}; \mu(a, b), \sigma^2_0) q_0(b)}
$$

$$
\psi_j(b) = \frac{q_0(b) h(x_{*j}; b, P_0)}{\sum_{b=1}^{K_2} q_0(b) h(x_{*j}; b, P_0)} \quad \text{with} \quad h(x_{*j}; b, P_0) = \prod_{i=1}^{R} \sum_{a=1}^{K_1} p_0(a) k(x_{ij}; \mu(a, b), \sigma^2)
$$

$$
\psi^*_ij(a, b) = \psi_j(b) \psi_{ij}(a, b) \quad \text{with} \quad \psi_{ij}(a, b) = \frac{k(x_{ij}; \mu(a, b), \sigma^2_0) p_0(a)}{\sum_{a=1}^{K_1} k(x_{ij}; \mu(a, b), \sigma^2_0) p_0(a)}.
$$
Appendix C

Gradient function for the block mixture model

Here we present the details of deriving the gradient function for the block mixture model with Gaussian data and explain how we devise an EM algorithm to optimize it.

Consider a block mixture model with \( K_1 \) row clusters and \( K_2 \) columns cluster. Recall that the log composite likelihood for this model is

\[
\log L_{RC}(Q) = \sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} p(a) \left[ \prod_{j=1}^{C} \sum_{b=1}^{K_2} q_b k(x_{ij}; \mu(a,b)) \right] \right\} + \sum_{j=1}^{C} \log \left\{ \sum_{b=1}^{K_2} q_b \left[ \prod_{i=1}^{R} \sum_{a=1}^{K_1} p(a) k(x_{ij}; \mu(a,b)) \right] \right\},
\]

where \( k(x_{ij}; \mu(a,b)) = N(x_{ij}; \mu(a,b), \sigma^2) \).

To obtain the gradient function, we ask if we can improve the mixture fit by increasing \( K_2 \) to \( K_2 + 1 \), which requires adding one column weight \( q(K_2 + 1) \) and new column means \( \mu_{\text{new}} = \{ \mu(1, K_2 + 1), \mu(2, K_2 + 1), \ldots \} = \{ \mu(a, K_2 + 1) ; a = 1, \ldots, K_1 \} \). Given a proposed new mean vector \( \mu_{\text{new}} \), we construct the log composite log likelihood \( L_\pi \) defined along a path with only a scalar parameter \( \pi \), where the original columns have masses \( (1 - \pi) q(b) \) and the new column with mean vector \( \mu_{\text{new}} \) has mass \( \pi \). By correspondingly updating the log composite likelihood for the model with \( K_1 \) and \( K_2 \), we obtain:

\[
L_\pi = \sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} p(a) \left[ \prod_{j=1}^{C} \left( (1 - \pi) \sum_{b=1}^{K_2} q(b) k(x_{ij}; \mu(a,b)) + \pi k(x_{ij}; \mu(a,K_2 + 1)) \right) \right] \right\} + \sum_{j=1}^{C} \log \left\{ (1 - \pi) \sum_{b=1}^{K_2} q_b \left( \prod_{i=1}^{R} \sum_{a=1}^{K_1} p(a) k(x_{ij}; \mu(a,b)) \right) + \pi \left( \prod_{i=1}^{R} \sum_{a=1}^{K_1} p(a) k(x_{ij}; \mu(a,K_2 + 1)) \right) \right\}.
\]
Now let

\[ k^*(x_{ij}) = k \left( x_{ij}; \mu_{new}, \sigma^2 \right) \]

\[ g(x_{ij}|a) = \sum_{b=1}^{K_2} q(b) k \left( x_{ij}; \mu(a, b) \right) \forall a = 1, \ldots, K_1 \]

\[ g(x_i|a) = \prod_{j=1}^{K_1} \left( \sum_{b=1}^{K_2} q(b) k \left( x_{ij}; \mu(a, b) \right) \right) \forall a = 1, \ldots, K_1 \]

\[ h(x_{ij}|b) = \sum_{a=1}^{K_1} p(a) k \left( x_{ij}; \mu(a, b) \right) \forall b = 1, \ldots, K_2 \]

\[ h(x_i|b) = \prod_{j=1}^{R} \left( \sum_{a=1}^{K_1} p(a) k \left( x_{ij}; \mu(a, b) \right) \right) \forall b = 1, \ldots, K_2 \]

\[ h^*(x_{ij}|K_2 + 1) = \sum_{a=1}^{K_1} p(a) k^*(x_{ij}) \]

\[ h^*(x_i|K_2 + 1) = \prod_{j=1}^{R} \left( \sum_{a=1}^{K_1} p(a) k^*(x_{ij}) \right). \]

Then we have

\[ L_x = \sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} p_a \left[ \prod_{j=1}^{C} ((1 - \pi) g(x_{ij}|a) + \pi k^*(x_{ij})) \right] \right\} + \sum_{j=1}^{C} \log \left\{ (1 - \pi) \sum_{b=1}^{K_2} q(b) h(x_j|b) + \pi h^*(x_j|K_2 + 1) \right\} \]

We define the gradient function as \( D(\mu_{new}) = \frac{\partial}{\partial \pi} L_x \big|_{\pi=0} \). Note that

\[ \frac{\partial}{\partial \pi} L_x \big|_{\pi=0} = \frac{\partial D_1}{\partial \pi} \big|_{\pi=0} + \frac{\partial D_2}{\partial \pi} \big|_{\pi=0}. \]  

(C.0.1)

First consider

\[ \frac{\partial D_2}{\partial \pi} = \sum_{j=1}^{C} \frac{h^*(x_j|K_2 + 1) - \sum_{b=1}^{K_2} q(b) h(x_j|b)}{(1 - \pi) \sum_{b=1}^{K_2} q(b) h(x_j|b) + \pi h^*(x_j|K_2 + 1)} \]

\[ \frac{\partial D_2}{\partial \pi} \big|_{\pi=0} = \sum_{j=1}^{C} \frac{h^*(x_j|K_2 + 1) - \sum_{b=1}^{K_2} q(b) h(x_j|b)}{\sum_{b=1}^{K_2} q(b) h(x_j|b)} \]

\[ \frac{\partial D_2}{\partial \pi} \big|_{\pi=0} = \sum_{j=1}^{C} \left\{ \frac{h^*(x_j|K_2 + 1)}{\sum_{b=1}^{K_2} q(b) h(x_j|b)} - 1 \right\} \]

\[ \frac{\partial D_2}{\partial \pi} \big|_{\pi=0} = \sum_{j=1}^{C} \frac{h^*(x_j|K_2 + 1)}{\sum_{b=1}^{K_2} q(b) h(x_j|b)} - C. \]
If we let \( \omega_j = \frac{1}{\sum_{k=1}^{K_2} q(b) h(x_j | b)} \), then we have
\[
\frac{\partial \ell_D_2}{\partial \pi} \big|_{\pi=0} = \sum_{j=1}^{C} \omega_j h^*(x_j | K_2 + 1) - C. \tag{C.0.2}
\]

Next consider
\[
\ell_{D_1} = \sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} p(a) \left( \prod_{j=1}^{C} ((1 - \pi)g(x_{ij} | a) + \pi k^*(x_{ij})) \right) \right\} = \sum_{i=1}^{R} \log \left\{ \sum_{a=1}^{K_1} p(a) \ell_{D_3} \right\}
\]
\[
\frac{\partial \ell_{D_1}}{\partial \pi} = \sum_{i=1}^{R} \sum_{a=1}^{K_1} p(a) \frac{\partial \ell_{D_3}}{\partial \pi} \tag{C.0.3}
\]
where
\[
\frac{\partial \ell_{D_3}}{\partial \pi} = \sum_{j=1}^{C} \left\{ \prod_{m=1; m \neq j}^{C} ((1 - \pi)g(x_{im} | a) + \pi k^*(x_{im})) \right\} \left[ k^*(x_{ij}) - g(x_{ij} | a) \right]
\]
\[
= \sum_{j=1}^{C} \frac{k^*(x_{ij}) - g(x_{ij} | a)}{(1 - \pi)g(x_{ij} | a) + \pi k^*(x_{ij})}
\]
\[
\frac{\partial \ell_{D_3}}{\partial \pi} \big|_{\pi=0} = \ell_{D_3} \big|_{\pi=0} \sum_{j=1}^{C} \left\{ \frac{k^*(x_{ij}) - g(x_{ij} | a)}{g(x_{ij} | a)} \right\} \tag{C.0.4}
\]
and
\[
\ell_{D_3} \big|_{\pi=0} = \prod_{j=1}^{C} g(x_{ij} | a) = g(x_i | a). \tag{C.0.5}
\]

Combining equations (C.0.4) and (C.0.5) gives
\[
\frac{\partial \ell_{D_3}}{\partial \pi} \big|_{\pi=0} = g(x_i | a) \sum_{j=1}^{C} \left\{ \frac{k^*(x_{ij}) - g(x_{ij} | a)}{g(x_{ij} | a)} \right\}. \tag{C.0.6}
\]
Combining (C.0.1), (C.0.2) and (C.0.7) we have

\[
\begin{align*}
\frac{\partial \ell_{D_\pi}}{\partial \pi} |_{x=0} &= \sum_{i=1}^{R} \sum_{a=1}^{K_1} \frac{p(a)g(x_i,|a) \sum_{j=1}^{C} \left\{ \frac{k^*(x_{ij}) - g(x_{ij}|a)}{g(x_{ij}|a)} \right\}}{\sum_{a=1}^{K_1} p(a)g(x_i,|a)} \\
\frac{\partial \ell_{D_\pi}}{\partial \pi} |_{x=0} &= \sum_{i=1}^{R} \sum_{a=1}^{K_1} p(a)g(x_i,|a) \left\{ \sum_{j=1}^{C} \frac{k^*(x_{ij})}{g(x_{ij}|a)} - 1 \right\} \\
\frac{\partial \ell_{D_\pi}}{\partial \pi} |_{x=0} &= \sum_{i=1}^{R} \left\{ \sum_{a=1}^{K_1} \frac{p(a)g(x_i,|a) \sum_{j=1}^{C} k^*(x_{ij})}{\sum_{a=1}^{K_1} p(a)g(x_i,|a)} - C \right\} \\
\frac{\partial \ell_{D_\pi}}{\partial \pi} |_{x=0} &= \sum_{i=1}^{R} \sum_{a=1}^{K_1} p(a)g(x_i,|a) \sum_{j=1}^{C} \frac{k^*(x_{ij})}{g(x_{ij}|a)} - RC.
\end{align*}
\]

If we let \(\omega_{ia} = \frac{p(\pi|g(x_i,|a))}{\sum_{a=1}^{K_1} p(a|g(x_i,|a))}\) and rewrite

\[
\frac{\partial \ell_{D_\pi}}{\partial \pi} |_{x=0} = \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_{ia} \sum_{j=1}^{C} \frac{k^*(x_{ij})}{g(x_i,|a)} - RC. \tag{C.0.7}
\]

Combining (C.0.1), (C.0.2) and (C.0.7) we have

\[
\frac{\partial}{\partial \pi} L_\pi |_{x=0} = \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_{ia} \sum_{j=1}^{C} \frac{k^*(x_{ij})}{g(x_i,|a)} - RC + \sum_{j=1}^{C} \omega_j h^*(x_j|K_2 + 1) - C. \tag{C.0.8}
\]

Hence the gradient function, \(D(\mu_{new})\) is given by

\[
D(\mu_{new}) = \frac{\partial}{\partial \pi} L_\pi |_{x=0} = \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_{ia} \sum_{j=1}^{C} \frac{k^*(x_{ij})}{g(x_i,|a)} + \sum_{j=1}^{C} \omega_j h^*(x_j|K_2 + 1) - C(R + 1).
\]

If we let \(\omega_{ija} = \frac{1}{g(x_i,|a)}\) then we can also rewrite this as

\[
\begin{align*}
D(\mu_{new}) &= \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_{ia} \sum_{j=1}^{C} \omega_{ija} k^*(x_{ij}) + \sum_{j=1}^{C} \omega_j h^*(x_j|K_2 + 1) - C(R + 1) \\
D(\mu_{new}) &= \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_{ia} \sum_{j=1}^{C} \omega_{ija} k^*(x_{ij}) + \sum_{j=1}^{C} \omega_j \prod_{i=1}^{K_1} \left( \sum_{a=1}^{K_1} p(a)k^*(x_{ij}) \right) - C(R + 1) \\
D(\mu_{new}) &= \sum_{j=1}^{C} \left\{ \sum_{i=1}^{R} \sum_{a=1}^{K_1} \omega_{ia} \omega_{ija} k^*(x_{ij}) + \omega_j \prod_{i=1}^{K_1} \left( \sum_{a=1}^{K_1} p(a)k^*(x_{ij}) \right) \right\} - C(R + 1)
\end{align*}
\]
C.1 EM on block mixture gradient function

The gradient function for the block mixture model with Gaussian data is

\[
D(\mu_{\text{new}}) = \sum_{j=1}^{C} \left\{ \sum_{i=1}^{R} \sum_{a=1}^{K_i} \omega_{ia} \omega_{ija} k^*(x_{ij}) + \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p_a k^*(x_{ij}) \right) \right\} - C(R + 1).
\]

Notice that maximizing \( D(\mu_{\text{new}}) \) with respect to \( \mu_{\text{new}} \) is same as maximizing \( D^*(\mu_{\text{new}}) \) with respect to \( \mu_{\text{new}} \). We implement the EM on \( D^*(\mu_{\text{new}}) \) using the methods explained in section 3.2.2. Let \( D^*(\mu_{\text{new}}^{(0)}) \) be the value of \( D^*(\cdot) \) evaluated at \( \mu_{\text{new}}^{(0)} \), the current value of \( \mu_{\text{new}} \). Then

\[
\log \left( \frac{D^*(\mu_{\text{new}})}{D^*(\mu_{\text{new}}^{(0)})} \right) = \log \left\{ \frac{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \omega_{ia} \omega_{ija} k^*(x_{ij}) + \sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k^*(x_{ij}) \right)}{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \omega_{ia} \omega_{ija} k_{(0)}^*(x_{ij}) + \sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k_{(0)}^*(x_{ij}) \right)} \right\}.
\]

where \( k^*(x_{ij}) = k \left( x_{ij}; \mu_{\text{new}}, \sigma^2 \right) \) and \( k_{(0)}(x_{ij}) = k \left( x_{ij}; \mu_{\text{new}}^{(0)}, \sigma^2 \right) \), and therefore

\[
\log \left( \frac{D^*(\mu_{\text{new}})}{D^*(\mu_{\text{new}}^{(0)})} \right) = \log \left\{ \frac{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \omega_{ia} \omega_{ija} k_{(0)}^*(x_{ij}) + \sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k_{(0)}^*(x_{ij}) \right)}{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \omega_{ia} \omega_{ija} k_{(0)}^*(x_{ij}) + \sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k_{(0)}^*(x_{ij}) \right)} \right\}.
\]

Now let

\[
\phi^{(0)} = \frac{\sum_{i=1}^{R} \sum_{j=1}^{C} \omega_{ia} \omega_{ija} k_{(0)}^*(x_{ij})}{\sum_{i=1}^{R} \sum_{j=1}^{C} \omega_{ia} \omega_{ija} k_{(0)}^*(x_{ij}) + \sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k_{(0)}^*(x_{ij}) \right)}
\]

then

\[
1 - \phi^{(0)} = \frac{\sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k_{(0)}^*(x_{ij}) \right)}{\sum_{i=1}^{R} \sum_{j=1}^{C} \omega_{ia} \omega_{ija} k_{(0)}^*(x_{ij}) + \sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k_{(0)}^*(x_{ij}) \right)}
\]
and we can rewrite

\[
\log \left( \frac{D^* (\mu_{\text{new}})}{D^* (\mu_{\text{old}})} \right) = \log \left\{ \phi^*(0) \left( \frac{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \omega_{1a} \omega_j x^a k^*(x_{ij})}{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \omega_{1a} \omega_j x^a k^*_0(x_{ij})} \right) +
\right\}
\]

\[
\left( 1 - \phi^*(0) \right) \left\{ \sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_1} p(a) k^*_0(x_{ij}) \right) \right\}
\]

\[
\log \left( \frac{D^* (\mu_{\text{new}})}{D^* (\mu_{\text{old}})} \right) \geq \phi^*(0) \log \left\{ \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \omega_{1a} \omega_j x^a k^*_0(x_{ij}) \right\}
\]

\[
\left( 1 - \phi^*(0) \right) \log \left\{ \sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_1} p(a) k^*(x_{ij}) \right) \right\}
\]

Next, let

\[
\alpha^*_{ija} = \frac{\omega_{1a} \omega_j x^a k^*(x_{ij})}{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \omega_{1a} \omega_j x^a k^*_0(x_{ij})}
\]

and

\[
\beta^*_j = \frac{\omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_1} p(a) k^*_0(x_{ij}) \right)}{\sum_{j=1}^{C} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_1} p(a) k^*_0(x_{ij}) \right)}
\]

Then we have

\[
\log \left( \frac{D^* (\mu_{\text{new}})}{D^* (\mu_{\text{old}})} \right) \geq \phi^*(0) \log \left\{ \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \alpha^*_{ija} \left( \frac{k^*(x_{ij})}{k^*_0(x_{ij})} \right) \right\}
\]

\[
\left( 1 - \phi^*(0) \right) \log \left\{ \sum_{j=1}^{C} \beta^*_j \left( \frac{\prod_{i=1}^{R} \sum_{a=1}^{K_1} p(a) k^*_0(x_{ij})}{\prod_{i=1}^{R} \sum_{a=1}^{K_1} p(a) k^*_0(x_{ij})} \right) \right\}
\]

\[
\log \left( \frac{D^* (\mu_{\text{new}})}{D^* (\mu_{\text{old}})} \right) \geq \phi^*(0) \log \left\{ \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \alpha^*_{ija} \left( \frac{k^*(x_{ij})}{k^*_0(x_{ij})} \right) \right\}
\]

\[
\left( 1 - \phi^*(0) \right) \log \left\{ \sum_{j=1}^{C} \beta^*_j \left( \frac{\prod_{i=1}^{R} \sum_{a=1}^{K_1} p(a) k^*_0(x_{ij})}{\prod_{i=1}^{R} \sum_{a=1}^{K_1} p(a) k^*_0(x_{ij})} \right) \right\}
\]
Finally, let

\[
\log \left( \frac{D^*(\mu_{\text{new}})}{D^*(\mu_{\text{old}})} \right) \geq \phi^{(0)} \log \left\{ \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \alpha_{ija}^{(0)} \left( \frac{k^*(x_{ij})}{k_{(0)}^*(x_{ij})} \right) \right\} + \\
(1 - \phi^{(0)}) \log \left\{ \sum_{j=1}^{C} \beta_j^{(0)} \left( \frac{\prod_{i=1}^{R} \sum_{a=1}^{K_i} p(a)k^*(x_{ij})}{\prod_{i=1}^{R} \sum_{a=1}^{K_i} p(a)k_{(0)}^*(x_{ij})} \right) \right\}
\]

\[
\geq \phi^{(0)} \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \alpha_{ija}^{(0)} \log \left( \frac{k^*(x_{ij})}{k_{(0)}^*(x_{ij})} \right) + \\
(1 - \phi^{(0)}) \sum_{j=1}^{C} \beta_j^{(0)} \log \left( \frac{\sum_{a=1}^{K_i} p(a)k^*(x_{ij})}{\sum_{a=1}^{K_i} p(a)k_{(0)}^*(x_{ij})} \right)
\]

Finally, let

\[
\delta_{ija}^{(0)} = \frac{p(a)k_{(0)}^*(x_{ij})}{\sum_{a=1}^{K_i} p(a)k_{(0)}^*(x_{ij})}.
\]

Then we have

\[
\log \left( \frac{D^*(\mu_{\text{new}})}{D^*(\mu_{\text{old}})} \right) \geq \phi^{(0)} \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \alpha_{ija}^{(0)} \log \left( \frac{k^*(x_{ij})}{k_{(0)}^*(x_{ij})} \right) + (1 - \phi^{(0)}) \sum_{j=1}^{C} \sum_{a=1}^{K_i} \delta_{ija}^{(0)} \log \left( \frac{k^*(x_{ij})}{k_{(0)}^*(x_{ij})} \right)
\]

Hence the E-step of the EM to maximize \( D(\mu_{\text{new}}) \) is given by

\[
\ell_D = \phi^{(0)} \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \alpha_{ija}^{(0)} \log k^*(x_{ij}) + (1 - \phi^{(0)}) \sum_{j=1}^{C} \sum_{a=1}^{K_i} \beta_j^{(0)} \delta_{ija}^{(0)} \log k^*(x_{ij})
\]
To obtain the M-step we maximize $\ell_D$ with respect to $\mu_{\text{new}}$.

Rewrite

\[
\phi^{*}(0)\alpha_{ija}^{*}(0)
= \frac{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \omega_a \omega_{ija} k_{ij}^{*}(0) (x_{ij})}{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \omega_a \omega_{ija} k_{ij}^{*}(0) (x_{ij})}
\]

and

\[
\beta_j^{*}(0)
= \frac{\sum_{i=1}^{R} \omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k_{ij}^{*}(0) (x_{ij}) \right)}{\omega_j \prod_{i=1}^{R} \left( \sum_{a=1}^{K_i} p(a) k_{ij}^{*}(0) (x_{ij}) \right)}
\]

It follows that

\[
\ell_D = \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_i} \left( \epsilon_{ija}^{*} + \gamma_{ija}^{*} \right) \log k^{*}(x_{ij})
\]

Also, since

\[
k^{*}(x_{ij}) = \mathcal{N}(\mu_{\text{new}}^{*}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2} \left( \frac{x_{ij} - \mu_{\text{new}}^{*}}{\sigma} \right)^2 \right\}
\]
we have

\[
\frac{\partial k^*(x_{ij})}{\partial \mu_{new}} = k^*(x_{ij}) \left( \frac{x_{ij} - \mu_{new}}{\sigma^2} \right).
\]

From equation (C.1.1) we have

\[
\frac{\partial \ell}{\partial \mu_{new}} = R \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \left( \epsilon_{ij}^{(0)} + \gamma_{ij}^{(0)} \right) \frac{1}{k^*(x_{ij})} \frac{\partial k^*(x_{ij})}{\partial \mu_{new}} = R \sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \left( \epsilon_{ij}^{(0)} + \gamma_{ij}^{(0)} \right) \left( x_{ij} - \mu_{new} \right) \left( x_{ij} - \mu_{new} \right).
\]

Hence the M-step of the EM to maximize \(D(\mu_{new})\) with Gaussian data is given by

\[
\hat{\mu}_{new} = \frac{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \left( \epsilon_{ij}^{(0)} + \gamma_{ij}^{(0)} \right) x_{ij}}{\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{a=1}^{K_1} \left( \epsilon_{ij}^{(0)} + \gamma_{ij}^{(0)} \right)}.
\]
Bibliography


[41] D. R. Hunter, B.G. Lindsay, and P. Kuruppumullage Don. Em is more than likelihood. To be submitted.


Vita

Prabhani Kuruppumullage Don

Research Interests: Mixture models, Model-based clustering and segmentation approaches with applications to Genomics, Statistical computing, Statistical Genomics.

Education:

- **Doctorate in Statistics**, Department of Statistics, The Pennsylvania State University, August 2014
  Advisors: Bruce G. Lindsay, Willaman Professor of Statistics, and Francesca Chiaromonte, Professor of Statistics and Public Health Sciences

- **Master of Statistics**, Department of Statistics, The Pennsylvania State University, May 2011
  Advisor: Francesca Chiaromonte, Professor of Statistics and Public Health Sciences

- **Bachelor of Science in Statistics**, Department of Statistics, University of Colombo, Sri Lanka, October 2005

Honors and Awards:

- Finalist, ASA Nonparametric Section Student Paper Competition, Joint Statistical Meeting (2014).

- Travel support by The Department of Statistics, University of Wisconsin-Madison, to attend the conference on Nonparametric Statistics for Big Data (2014).

- Scholarship and travel award by The Department of Biostatistics, University of Washington, to attend the workshop on 19th Summer Institute in Statistical Genetics (2014).

- Travel award by Institute for Pure and Applied Mathematics, University of California Los Angeles, to attend the workshop on Coancestry, Association, and Population Genomics (2011).


- Received the Certificate in Teaching given by the Staff and Educational Development Association (SEDA) for Higher Education in the UK (2008), for the successful completion of Certificate in Teaching in Higher Education course organized by University of Colombo, Sri Lanka.

- Award for best Sri Lankan undergraduate research project of the Sri Lanka Association for the Advancement of Science (SLAAS), Physical Science and Statistics (2006).

- Gold Medal for best undergraduate research project, University of Colombo, Department of Statistics, Sri Lanka (2005).

Publications:
