BAYESIAN LATENT CLASS MODELS

BY

JIANFENG XU

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Statistics in the Graduate College of the University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:

Professor Feng Liang, Chair
Professor Douglas G. Simpson
Professor John I. Marden
Professor Yuan Ji
Professor Yuguo Chen
Abstract

The latent class model (LCM) is a statistical method that introduces a set of latent categorical variables. The main advantage of LCM is that conditional on latent variables, the manifest variables are mutually independent of each other. In some scenarios, the LCM makes the modeling or computation feasible. In some other scenarios, the latent variables themselves are key. In the past a few decades, LCM has been widely applied to many areas such as Engineering, Medicine, Biology and Marketing.

In this paper, several LCMs are developed in Bayesian framework to address new challenges in different applications. The first work is about the MR image segmentation. For MR images, we usually need to simultaneously segment multiple images, which are believed to have similar segmentation results. In our co-segmentation model, a Markov random field prior is utilized to encourage the information sharing. Clustering is usually regarded as an unsupervised problem. In our second work, we extend the clustering into supervised setting. This supervised clustering is evaluated in the application of market segmentation. In our third work, we relax the all-feature-in and all-object-in assumptions of the existing clustering approaches and propose a novel model called Multiple Partition Process (MPP) to obtain multiple clustering structures from the data. This MPP model is applied into the clustering of the breast cancer microarray data. In the last part of this paper, our future work is represented.
To my family.
Table of Contents

List of Tables ................................................. vi
List of Figures ............................................... vii

Chapter 1  Introduction ........................................ 1
  1.1 Latent Class Models ...................................... 1
  1.2 Co-segmentation .......................................... 4
    1.2.1 Finite Mixture Model ................................ 4
    1.2.2 Multiple Image Co-segmentation ...................... 5
  1.3 Bayesian Collaborative Model ............................. 6
    1.3.1 Supervised Model-based Clustering ................. 6
    1.3.2 Market Segmentation ................................. 6
  1.4 Multiple Partition Process ................................ 8
    1.4.1 Traditional Statistical Clustering ................ 8
    1.4.2 Mosaic Type Clustering ................................ 8
    1.4.3 Multiple-Structure Clustering ...................... 9
  1.5 Multiple Partition for Latent Features ................... 10
    1.5.1 Two-stage Clustering Approach ...................... 10
    1.5.2 Indian Buffet Process ................................ 10
    1.5.3 Future Work ........................................ 11

Chapter 2  Bayesian Co-segmentation of Multiple MR Images ............... 12
  2.1 MR Images Segmentation .................................. 12
  2.2 Method .................................................. 14
    2.2.1 Gaussian Mixture Models ............................ 14
    2.2.2 Co-segmentation Priors .............................. 15
  2.3 MCMC Algorithms ......................................... 17
    2.3.1 A Single-site Gibbs Sampler ....................... 17
    2.3.2 A Swendsen-Wang Gibbs Sampler ................... 18
    2.3.3 Posterior Inference and Choice of Hyper-parameters 20
  2.4 Experiments .............................................. 20
    2.4.1 Synthetic Contaminated Brain Image Pair .......... 22
    2.4.2 Multiple Corpus Callosum Slices ................... 24
    2.4.3 Sequential Liver Images ............................ 25
    2.4.4 4D cardiac data .................................. 25

Chapter 3  A Bayesian Approach for Collaborative Inference ............... 28
  3.1 Marketing Segmentation ................................... 28
  3.2 Collaborative Inference with Structured Sparsity ........... 30
  3.3 Proposed Bayesian Approach .............................. 33
    3.3.1 Model and Priors .................................. 33
    3.3.2 Computation ...................................... 35
List of Tables

2.1 Evaluation of Individual Segmentation and Co-segmentation .......................... 23

3.1 Comparison of variable selection for BCM, gLasso, and Independent in Scenario I and II for simulation study with regression models. .......................... 37

3.2 Comparison of variable selection for BCM, gLasso, and Independent in Scenario I and II for simulation study with probit models. .......................... 40

3.3 AIC (BIC) scores ($\times 10^4$) for the online shopping data with prior (3.8), where “NA” indicates the outcome has size 0 segment, so it is not consistent with the pre-specified $K$ value. ................................................................. 43

3.4 Estimated Segment Coefficients using Algorithm I ........................................ 43

3.5 AIC (BIC) scores ($\times 10^4$) for the online shopping data with prior (3.9), where “NA” indicates the outcome has size 0 segment, so it is not consistent with the pre-specified $K$ value. ................................................................. 44

3.6 Estimated Segment Coefficients using Algorithm II ....................................... 44

4.1 $s=3$: Results of Structure Indicators ............................................................ 56

4.2 $s=3$: Clustering for Structure 2 (comparing with true $C_2$) ............................ 56

4.3 $s=3$: Clustering for Structure 3 (comparing with true $C_3$) ............................ 56

4.4 Proportions of Different Structures in $\hat{S}$ with $s=3$ ................................. 58
List of Figures

2.1 The top row are the original sequential MR images; the second row are their closed contours, which form super-vertices; the bottom plot is the final super-vertices for the three images. .................................................. 22
2.2 Data and results for the contaminated brain image pair: original images (top row), results from individual segmentation (middle row), and results from co-segmentation (bottom row). ........................................... 23
2.3 Data and results for CC slices: original images (top row), results from individual segmentation (middle row), and results from co-segmentation (bottom row). ......... 24
2.4 The plot of AIC and BIC versus various $K$ for CC slices (top panel) and sequential liver (lower panel) images. .......................................................... 25
2.5 Data and results for sequential liver images: original images (top row), results from individual segmentation (middle row), and results from co-segmentation (bottom row). 26
2.6 Data and results for the cardiac data: one slice of the 3D image is presented at 3 time points (top panel, from left to right), and results from co-segmentation (lower panel). 27
2.7 Convergence comparison between SSGS and SWGS. The left plot is the convergence rates of SSGS and SWGS which indicates SWGS needs only 1/10 iterations steps to be convergent. The right plot shows the speed in time for convergency which evaluates the SWGS is around 25 time faster than SSGS. ................................. 27

3.1 Compare the PMSEs among $BCM$, $Oracle$, $gLasso$ and $Independent$ in Scenario I for simulation study with regression models. The y-coordinate indicates the percentage of PMSE relative to the PMSE from $Independent$ on the same data set. .............. 38
3.2 Compare the PMSEs among $BCM$, $Oracle$, $gLasso$, and $Independent$ in Scenario II for simulation study with regression models. The y-coordinate indicates the percentage of PMSE relative to the PMSE from $Independent$ on the same data set. ................. 39
3.3 Compare the PMSE's among $BCM$, $Oracle$, $gLasso$, and $Independent$ in Scenario I for simulation study with probit models. The y-coordinate indicates the percentage of PMSE relative to the PMSE from $Independent$ on the same data set. .............. 40
3.4 Compare the PMSEs among $BCM$, $Oracle$, $gLasso$, and $Independent$ in Scenario II for simulation study with probit models. The y-coordinate indicates the percentage of PMSE relative to the PMSE from $Independent$ on the same data set. ................. 41
3.5 Demonstration of one pair of websites offering the 22-inch Samsung LCD monitor. Totally 15 web features are considered which include 6 three-level features and 9 two-level features. .......................................................... 42
4.1 Different clustering scenarios for the matrix data. From left to right panel, the plots are for: the traditional consistent clustering over all \( p \) features; the mosaic clustering with 3 partitions and the clustering scenario with mixed partitions over each feature. Top panel: the original data with three scenarios from left to right; Middle panel: rows of the data matrices have been re-arranged for a better visual effect for the first partition. Bottom panel: The re-arranged matrices for the second partition if applicable. For better visualization of clustering effect of the third scenario, only related entries are plotted as seen the most right plots in the middle and bottom panels.

4.2 The data has 120 subjects and 20 features. The left panel is the data in the generated order (by row); the middle one is the data shuffled according to the memberships of structure 2; the right panel is shuffled according to the memberships of structure 3.

4.3 Select \( s \) by ALPML. We set \( \hat{s} = 3 \), because (1) after \( s \geq 3 \), the ALPML values are very close; and (2) the \( S \)'s for cases \( s = 4 \) and \( 5 \) just contain 3 different structures.

4.4 The histogram of the cluster sizes in structure 2 and 3: \( \hat{c}_2 = 3 \) and \( \hat{c}_3 = 2 \).

4.5 The breast cancer data set contains 517 patients and 9 features (left). The left panel is the data with objects ordered by original order; the middle panel is ordered by the cluster membership in structure 2; the right panel is ordered by the cluster membership in structure 3.

4.6 Select \( s \) by ALPML for the breast cancer data. We have \( \hat{s} = 3 \), because ALPML reaches the maximum when \( s = 3 \).

5.1 The aCGH data set from six different lung cancer patients from three different lung cancer subtypes: “TN,” “HER2+” and “ER2+.”

5.2 The first row are four objects generated by the four features shown in the second row; the third row are the first 4 most frequent features in MCMC.

5.3 The estimation of parameters.

5.4 The histogram of \( K^+ \) with mode \( K^+ = 4 \).

5.5 The feature frequencies over all MCMC steps.
Chapter 1

Introduction

1.1 Latent Class Models

The latent variable model is a statistical method that introduces a set of latent variables. The introduced variables are called latent variables because they cannot be observed. The main advantage of latent variables model is that conditional on latent variables the manifest variables are mutually independent of each other. The latent variable models could be grouped by whether the latent variables are continuous or categorical. When the latent variables are categorical, the latent variable model is called latent class model (LCM).

The LCM was initially proposed by Lazarsfeld and Henry (1968) and then was improved by many other researchers (Goodman, 1974; McCutcheon, 1987; Lindsay et al., 1991). In LCM, each object is assumed to belong to one of the unobserved classes or groups. Take the finite mixture model for example. The mixture model assumes that the data come from a source with several subpopulations each of which is modeled separately. The overall data is a mixture of these subpopulations, so the resulting model is a mixture of a finite components. The general form of a mixture model with $K$ components is

$$f(X_i) = \sum_{k=1}^{K} p_k f_k(X_i|\theta_k), \quad (1.1)$$

where $X_i$, $i \in 1 : n$, is the $i$-th observation, $p_k$ is the mixing proportion in the $k$-th group and $f_k(\cdot|\theta_k)$ is density function of the $k$-th group with parameter $\theta_k$. If $f_k$’s are normal densities, the mixture model is Gaussian Mixture Model (GMM). The distribution of model 1.1 is equivalent to the distribution of the model that introduces a latent variable for each object,

$$f(X_i|Z_i = k) = f_k(X_i|\theta_k), \quad (1.2)$$

where $Z_i$ is the introduced latent variable to indicate which component of the mixtures the $i$-th object belongs to.
In some scenarios, the LCM makes the modeling or computation feasible. As in the finite mixture model above, the introduction of the latent variables makes the Expectation-Maximization (EM) algorithm possible. In other scenarios, the latent variables themselves are key. For instance, in the model-based clustering, the latent variables refer to the cluster membership.

In our work, the LCMs are extended in several different applications. In the multiple MR image co-segmentation, each image is modeled as a finite mixture model. Because of the similarity among the multiple images, different mixture models are encouraged to share the information. In order to achieve the information sharing, Potts priors are employed for the latent variables. In the market segmentation, a supervised model-based clustering is developed. In traditional clustering, we cluster the observations based on the similarity of the covariates \( X_{n \times p} \), i.e., based on the similarity of the \( P(X_i) \) where \( X_i \) is the \( i \)-th row of \( X \) representing the \( i \)-th observation. This traditional clustering is also called unsupervised learning. In supervised setting, the \( i \)-th observation is \((X_i, Y_i)\) where \( X_i \) is an \( m \times p \) matrix and \( Y_i \) is the response vector with dimension \( m \) (\( m \geq 1 \)). The clustering of \( n \) observations in supervised setting is based on how \( Y_i \) depends on \( X_i \), i.e., based on the similarity of the conditional distribution \( P(Y_i | X_i) \). For example, if \((X_i, Y_i)\) fits a linear regression model: \( Y_i = X_i \beta_i + \epsilon \), then the supervised clustering could be corresponding to the clustering of the coefficients \( \beta = \{ \beta_1, \cdots, \beta_n \} \). In the market segmentation, a Bayesian Collaborative Model (BCM) is developed to extend the model-based clustering to the supervised setting. In the traditional clustering, there exists one and only one clustering structure and each object possesses only one membership. However, some applications, such as the clustering of the breast cancer microarray data, involves multiple clustering structures which motivates us to propose Multiple Partition Process (MPP). In MPP, for the \( l \)-th clustering structure, a latent variable \( Z_{n \times l} = \{ Z_{1l}, \cdots, Z_{nl} \} \) is used to indicate the cluster membership. Meanwhile, for cell \((i, j)\) in the data matrix \( X \), \( X_{ij} \), a latent variable \( S_{ij} \) is introduced to indicate the membership of the clustering structure. Given \( S_{ij} = l \), \( X_{ij} \) follows a finite mixture model. In the finite mixture model as in 1.1, the number of components \( K \) is pre-specified. The value of \( K \) could be decided by some specific background, or selected by some criteria, such as Akaike information criterion (AIC) and Bayesian information criterion (BIC). In MPP, we employ a nonparametric Bayes, Dirichlet Process (DP), which assumes infinite number of components. MPP is evaluated on a breast cancer microarray data set. When the feature dimension \( p \) is large, the MPP is inefficient. For example, in the clustering of the array Comparative Genomic Hybridization (CGH) data, the feature number \( p \) is usually in the scale of tens of thousands. One natural way to overcome this large-dimensionality problem is to perform
dimension reduction before applying the MPP. Due to the properties of the aCGH data, Indian Buffet Process (IBP) is utilized to extract the latent features. In IBP, each object is represented by a vector of latent feature values: \( \mathbf{X} = \mathbf{ZF} + \epsilon \), where \( \mathbf{F} \) is the feature matrix each row of which presents a feature and \( \mathbf{Z} \) is a matrix with \( Z_{ik} = 1 \) if object \( i \) has feature \( k \) and 0 otherwise. A further clustering by MPP on matrix \( \mathbf{Z} \) is performed. This two-stage procedure is called Multiple Partition for Latent Features (MPLF). This two-stage approach separates the feature identification and multiple-structure clustering. Apparently, it is not the optimal. As our future work, we will develop a integrated model which is able to identify the latent features and cluster the objects simultaneously.

In this thesis, we focus on the Bayesian approaches for LCMs. Bayesian approaches treat the parameter \( \theta \) as being random. This is different from the frequentist approaches that consider the parameter fixed but unknown. In the Bayesian approaches, the inference is the posterior, because, given the data \( \mathbf{Y} \), the posterior \( P(\theta|\mathbf{Y}) \propto P(\mathbf{Y}|\theta)P(\theta) \) has all the information about the parameter \( \theta \). There are several benefits of the Bayesian approach to statistics (O’Hagan, 2003):

1. The computing tools now available for Bayesian statistics allow us to tackle enormously more complex problems;

2. Bayesian methods make use of all available information. This is simply a reference to the fact that the Bayesian approach includes the prior information. Although the data will dominate when the data size increase, the prior information is helpful especially when the data size is limited;

3. Bayesian techniques are particularly well suited for decision-making. What makes decisions hard is uncertainty. There is uncertainty about the consequences of any given decision, due to lack of knowledge about some relevant facts or parameters. Bayesian methods can quantify those uncertainties using personal probability. This quantification of the uncertainties in a decision is a crucial component of rational, evidence-based decision-making;

4. Bayesian statistics provides more meaningful inferences. For example, a frequentist \( p \)-value has a convoluted interpretation that does not actually say how likely the null hypothesis is on the basis of the evidence. A Bayesian analysis gives the more direct and meaningful statement of the probability that the hypothesis is true.

The thesis is organized as follows. In this chapter, we briefly discuss the four proposed LCM’s. In chapter 2, 3, 4, we describe the details of the models, computation and experiments of Co-
segmentation, BCM and MPP respectively. As an extension of MPP and our future work, the idea of MPLE and some simulations are represented in chapter 5.

1.2 Co-segmentation

1.2.1 Finite Mixture Model

The general form of a mixture model with $K$ groups is in 1.1. Given the latent variable, we have the model 1.2. Expectation maximization (EM) (Dempster et al., 1977) is a popular technique to estimate the finite mixture model. EM is iterative algorithm with two steps: an expectation step and a maximization step. As an alternative to the EM algorithm, the mixture model parameters can be deduced in a Bayesian framework. In Beyesian finite mixture model, Gibbs Sampler (Liu, 2002) is employed to sample the posterior. Gibbs Sampler is a special Monte Carol Markov Chain (MCMC) scheme of which the idea is that a distribution can be estimated by a sequence of conditional distributions.

In Bayesian finite mixture model, a so-called label switching problem arises (Stephens, 2000). The term of label-switching was used by Render and Walker (1984) to describe the invariance of likelihood under relabeling of the mixture components. for instance, switch the label of the first and second component or cluster (“first” and “second” are just labels), then the original “first” component is named second and the original “second” is named as first. Under the switching, the likelihood remains unchanged. In Bayesian finite mixture model, this invariance could result in the posterior distribution being symmetric and multimodal and further make it difficult to summarize. To overcome this labeling switching issue, in our Gibbs sampler, a reordering step is added: after each iteration of Gibbs sampler, the labels are reordered such that the means are decreasing.

Another issue in Bayesian finite mixture model is the choice of the cluster number, that is $K$. For a certain class of clustering algorithms, like K-mean, the value of $K$ is prefixed. Increasing $K$ without penalty will always reduce the amount of error in the resulting clustering, to the extreme case of zero error if each data point is considered its own cluster. In practical, smaller $K$ has a better interpretation. So, the optimal choice of $K$ will strike a balance between maximum compression of the data using a single cluster, and maximum accuracy by assigning each data point to its own cluster. Typically, the number of clusters $K$ is choosen by some information criteria, such as AIC, BIC and the Deviance information criterion (DIC).
1.2.2 Multiple Image Co-segmentation

A basic problem in magnetic resonance imaging (MRI) is to precisely segment regions of interest (ROIs) from the image data, which is a crucial part of diagnosis, surgery, therapy guidance, and other medical research and applications (Kass et al., 1988; Zou et al., 2001; Frangi et al., 2001). These existing segmentation algorithms can be roughly divided into three categories (Lakare and Kaufman, 2000): the algorithms based on geometric or topological structure models, including Snakes (Kass et al., 1988), Level Set (Yang et al., 2004), and Watershed (Beucher and Meyer, 1993), the algorithms based on statistical models, including the thresholding method (Weszka, 1978), K-means clustering (Wells et al., 1996), Markov random fields based methods (Winkler, 2006), and classification methods, and the algorithms based on hybrid approaches. Despite such a wide array of literature, MRI segmentation still remains a challenging problem due to the complex structure between ROI and the neighboring parts and imaging artifacts such as noise, motion, contrast, etc.

In our work, we are motivated to incorporate the information from similar images to improve the accuracy of segmentation. We consider the problem of simultaneously segmenting multiple MR images, which, for example, can be a series of MR images scanned over time such as images of liver perfusion or dynamic cardiac motion, spatial slices of a volume, or images of symmetrical tissues such as lung and hippocampus. Due to their similarities, it is beneficial to share the image information with each other when executing segmentation, which we refer to as co-segmentation. In our co-segmentation model, each image is modeled as a finite Gaussian mixture, and segmentation of each image is equivalent to partition vertices in an image into different clusters, based on the homogeneity of their intensity measures. So, the co-segmentation is a simultaneously clustering based on multiple finite mixture model. In traditional Bayesian finite mixture models, the latent variables, or membership, are modeled as i.i.d. discrete random variables. However, such an independent model does not work well with image data since the spatial dependence. Thus, it is natural to assume that neighboring vertices are likely to belong to the same segment. In addition, in order to utilize similarity across images, we further extend the neighborhood on the same image to multiple images (e.g., image pairs, or adjacent images if images are obtained over time). Then, Potts prior is assigned to the latent variables in order to encourage the information sharing among the neighboring vertices. This Potts prior is able to lead to information sharing over multiple images and meanwhile preserving the spatial structure within each individual image which has been destroyed by vectorization. The detail of the Potts prior can be found in section 2.2.2.
1.3 Bayesian Collaborative Model

1.3.1 Supervised Model-based Clustering

In the traditional clustering, we cluster the observations based on the similarity of the covariates $X$, i.e. based on the similarity of the $P(X_i)$. In supervised setting, the $i$-th observation is $(X_i, Y_i)$ where $X_i$ is a $m \times p$ matrix and $Y_i$ is a $m$-dimensional vector. Usually $m \geq 1$. The clustering of $n$ observations in supervised setting is based on how $Y_i$ depends on $X_i$, i.e, based on the similarity of the conditional distribution $P(Y_i|X_i)$. For instance, suppose that there are $n$ objects and each object fits a model, such as linear regression:

$$Y_i = X_i \beta_i + \epsilon,$$

where $\beta_i$ is the coefficient of the $i$-th object. The clustering is based on the unknown parameters $\beta = \{\beta_1, \cdots, \beta_n\}$ rather than the data $(X_i, Y_i)$, so this kind of clustering called Supervised Model-based Clustering. In this work, we discuss this supervised model-based clustering in a Bayesian framework.

1.3.2 Market Segmentation

Market segmentation is an essential element in marketing theory and practice (Wedel and Kamakura, 1999). The concept of market segmentation was originally introduced by Smith (1956), “... viewing a heterogeneous market as a number of smaller homogeneous markets, in response to different preferences, attributable to the desires of consumers for more precise satisfaction of their varying wants.” The variables or criteria used to segment a market, or in other words, to group customers, are called the bases for market segmentation. Choices of bases include demographical, cultural, geographical variables, and personality or life-style, which are variables not depending on products. On the other hand, we can also segment a market based on product-specific variables such as usage frequency, store loyalty, and product benefit. Specifically “benefit segmentation” refers to segmenting markets based on the way consumers respond to product features in their decision making, which are widely used in marketing research and practice for better advertising and distribution of new products (Haley, 1968; Calantone and Sawyer, 1978; Wind, 1978). Conjoint analysis is a common approach for benefit segmentation, and regarding statistical methodology, mixture models are often used.
Although mixture models and conjoint analysis have been successfully applied on many real cases, the modern marketing environment imposes some new challenges. The first challenge is the increasing number of product features. Companies are adding more and more features into a single product such as a cellphone or a laptop. This new trend restricts the use of some traditional methods, for example, conjoint analysis is recommended not to handle more than six features (Green and Srinivasan, 1978). In responding to this new challenge, researchers in marketing science have proposed some recent alternatives, including hybrid conjoint techniques using self-explication (Johnson, 1987) that relies on subjects in the experiment to tell the research what the important product features are, and dimension reduction techniques such as multidimensional scaling (DeSarbo et al., 2008). These new approaches, however, either do not reflect the real-life choice scenario or are difficult to interpret. From the managerial standpoint, of major interest is the relationship between a single feature and consumer’s utility, so managers can focus their limited resources on a few important features, instead of spread over all features. From the consumers’ perspective, given such a long list of product features, they usually make their choice only based on a subset of important features, due to convenience, cost of thinking or lack of expertise about some features (Gilbride et al., 2006), in contrast to the assumption held in traditional segmentation methods that consumers consider every feature for their product choice. Further, consumers’ heterogeneity in feature selection should be incorporated into marketing models as a new index for market segmentation.

Another challenge comes from the restriction that the number of observations researchers collect from each subject, cannot be large, before the subject gets bored or fatigued. So when the dimension gets large, the model for each individual falls into the typical “high dimension low sample-size” paradigm, which makes inference on feature selection for each individual model challenging.

In this work, a model-based clustering model is employed to perform market segmentation. More precisely, each object possess a probit model and the model coefficients divide all the objects into different segments. The clustering structure imposed make the information sharing among all customers. In order to overcome the first challenge above, feature/variable selection is embedded the Bayesian model-based clustering model. In addition, a similarity graph is constructed based on the objects based on social networks or pairwise correlation calculated using auxiliary psychological, cultural, or demographic data. Additional similarity weight might be assigned to each edge of the graph. We will incorporate this graph structure into prior specification with the rationale that similar consumers may respond to product features in a similar way, therefore similar consumers share their information for statistical inference.
1.4 Multiple Partition Process

Clustering methods are widely used in many fields, such as biology, medicine, engineering and marketing. The traditional clustering methods assume the consistency of the clustering structure over all features. However, in many applications, this assumption is violated and multiple clustering structures might exist. In this part, mosaic type clustering is discussed and then a more flexible multiple-structure clustering, Multiple Partition Process (MPP), is proposed.

1.4.1 Traditional Statistical Clustering

Suppose that there are \( n \) observations \( \{X_1, \ldots, X_n\} \) each of which is a \( p \)-dimensional vector. In the traditional approaches, various clustering algorithms are applied on the \( p \)-dimensional vectors. Taking the K-means clustering for example, the basic idea is to cluster objects to a pre-specified number \( K \) of groups over all \( p \) features in such a way as to minimize the within-cluster sum of squares

\[
\arg \min_{Z} = \sum_{k=1}^{K} \sum_{i: Z_i = k} ||X_i - \mu_k||^2,
\]

where \( Z = \{Z_1, \ldots, Z_n\} \) is the cluster membership set and \( \mu_k \), a \( p \)-dimensional vector, is the \( k \)-th cluster mean.

Another commonly used clustering is hierarchical clustering. In hierarchical clustering (Johnson, 1967) the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place, which may run from a single cluster containing all objects to \( n \) clusters each containing a single object. The hierarchical clustering can be divided to agglomerative method, which proceed by series of fusion of \( n \) objects to groups, and divisive method, which successively separate a group of \( n \) objects into finer groups.

The traditional clustering approaches just output one and only one clustering structure, or partition. In other words, the clustering structure is consistent in each of the \( p \) features.

1.4.2 Mosaic Type Clustering

Nowadays many applications violate the assumption of partition sharing among all \( p \) features and the data may exhibit different partitions when being associated with different sets of features. In the microarray analysis example, the data structure might exhibit like this: some features divide the objects into two clusters, some divide the objects into three clusters, and there is no clustering structure for the remaining features, or in other words, the \( n \) objects form just one cluster with
A natural extension of the traditional all-features-in approaches is the mosaic type clustering. The mosaic type clustering in this paper is defined as multiple clustering structures, or partitions, in different non-overlap sets of features. Biclustering (Hartigan, 1972) is such a mosaic type clustering technique which simultaneously clusters the rows and columns of a data matrix. The biclustering always generates biclusters each of which is a subset of rows which exhibit similar behavior across a subset of columns, or vice versa. A variety of biclustering algorithms can be found in the literatures. In Hartigan (1972), the author proposes a method so called direct clustering which begins with the entire data as a single block and then iteratively finds the row and column split of every block into two pieces. Cheng and Church (2000) constructs one bicluster at a time using some criteria. Once a bicluster is created, its entries are replaced by random numbers, and the procedure is repeated iteratively. A nonparametric Bayesian biclustering is discussed in Meeds et al. (2007). Two independent DP priors are introduced over row and column clusters separately.

1.4.3 Multiple-Structure Clustering

Although the biclustering relaxes the all-feature-in assumption and can retrieve multiple partitions simultaneously, it might cause undesired fragments which will be detailed in chapter 4. This unexpected clustering result is caused by the improper assumption that each feature can only join in one and only one clustering structure. In order to avoid this fragmentation issue, we model the multiple partitions in a more flexible manner: each of the $p$ features can belong to multiple rather than single partition. More precisely, in each feature, different objects might belong to different partitions. This flexibility makes each clustering structure consider the overall features and leads to a novel clustering method, Multiple Partition Process, abbreviated as MPP.

Recently, Dunson (2009) proposed Local Partition Process (LPP) that is constructed through a locally-weighted mixture of global and local clustering structures. The local clustering in LPP is a partition over single feature while the global clustering is overall partition. In LPP, each object is associated with one global cluster and $p$ local clusters. The information borrowing in LPP can be induced across subjects through both global and local clustering. The LPP can be considered a special case of MPP with $p + 1$ partitions each of which is associated with one single feature.
1.5 Multiple Partition for Latent Features

1.5.1 Two-stage Clustering Approach

The Multiple Partition Model (MPP) is efficient when the feature number $p$ is relatively small. In some applications with large $p$, however, MPP suffers. For example, the aCGH data matrix $X_{n \times p}$ involves $n$ cancer patients (objects) and each patient has millions of measurements ($p$ is huge) over all chromosomes. It is difficult for MPP to converge on the data sets with high dimensions. A natural way to overcome the large-$p$ problem is to perform dimension reduction before applying MPP. In this chapter, a two-stage clustering approach is proposed: 1) perform dimension reduction; 2) apply MPP to the reduced “new” data matrix, say $Z_{n \times p'}$ where $p'$ is relatively small.

One important property of the aCGH data is that all the objects can be better captured by representing each one as possessing multiple latent features. Thus, for aCGH data, we could reduce the dimension in this way: each “new” dimension refers to a latent features and the “new” data matrix is a feature possession matrix entry $(i,j)$ takes 0/1 values to indicate whether the $i$-th object possesses the $j$-th latent feature or not. Usually the “new” data matrix has relatively small dimension. There are two advantages of this dimension reduction: the important patterns critical to clustering can be identified and the the original dimensions are preserved which is important to the cancer treatment. IBP can be adopted as such a dimension reduction tool.

1.5.2 Indian Buffet Process

The typical clustering algorithms represent data in terms of which cluster each object belongs to. Clustering models are restrictive, because they do not have distributed representations. For example, we can describe a person as “student”, “female”, “Asian”, “married” and so on. These features are latent and the number of these potential features are always unlimited. Meanwhile, each object could be represented as possessing multiple latent features. Several methods exist for representing objects in terms of latent features, such as Blei et al. (2003) and Ueda and Saito (2003). These methods, however, still could not solve one critical question: how many latent features are needed to express the latent structure responsible for the observed data. In Griffiths and Ghahramani (2005), the authors “take the idea of defining priors over infinite combinatorial structures from nonparametric Bayesian statistics, and use it to develop methods for unsupervised learning in which each object is represented by a sparse subset of an unbounded number of features”. In IBP, a distribution over $Z$ is designed to be used as a prior in probabilistic models that represent objects using a potentially
infinite array of features. This probability distribution can be derived from a simple stochastic process, called exchangeable IBP (Griffiths and Ghahramani, 2005).

In Gibbs Sampler, we can derive the conditional distribution from the exchangeable IBP:

\[ P(Z_{ik} = 1|\mathbf{Z}_{-i,k}) = \frac{m_{-i,k}}{n}, \]  

(1.3)

where \( \mathbf{Z}_{-i,k} \) is the set of assignments of other objects, excluding the \( i \)-th object for the \( k \)-th feature, and \( m_{-i,k} \) is the number of objects possessing feature \( k \) excluding the \( i \)-th object. From this conditional distribution, we can find IBP contains a rich get richer phenomenon similar to Chinese Restaurant Process.

### 1.5.3 Future Work

This two-stage approach, however, is not optimal, so we are working on an integrated model which could simultaneously exact the latent features and perform multiple-structure clustering over objects. In other words, our goal is to integrate the IBP and DP priors in a multiple clustering model.
Chapter 2

Bayesian Co-segmentation of Multiple MR Images

Segmentation is one of the basic problems in magnetic resonance (MR) image analysis. We consider the problem of simultaneously segmenting multiple MR images, which, for example, can be a series of 2D/3D images of the same tissue scanned over time, different slices of a volume image, or images of symmetric parts. These multiple MR images share common structure information and hence they can assist each other in the segmentation procedure. We propose a Bayesian co-segmentation algorithm where the shared information across multiple images is utilized via a Markov random field prior. An efficient algorithm based on the Swendsen-Wang method is employed for posterior sampling, which is more efficient than the single-site Gibbs sampler. Because our co-segmentation algorithm pulls all the image information into consideration, it provides more accurate and robust results than individual segmentation, as supported by our experimental studies with real examples.

2.1 MR Images Segmentation

A basic problem in magnetic resonance imaging (MRI) is to precisely segment regions of interest (ROIs) from the image data, which is a crucial part of diagnosis, surgery, therapy guidance, and other medical research and applications (Kass et al., 1988; Zou et al., 2001; Frangi et al., 2001). Many segmentation algorithms have been introduced in literature. These existing algorithms can be roughly divided into three categories (Lakare and Kaufman, 2000): the algorithms based on geometric or topological structure models, including Snakes (Kass et al., 1988), Level Set (Yang et al., 2004), and Watershed (Beucher and Meyer, 1993), the algorithms based on statistical models, including the thresholding method (Weszka, 1978), K-means clustering (Wells et al., 1996), Markov random fields based methods (Winkler, 2006), and classification methods, and the algorithms based on hybrid approaches.

Despite such a wide array of literature, MRI segmentation still remains a challenging problem due to the complex structure between ROI and the neighboring parts and imaging artifacts such
as noise, motion, contrast, etc. To address these challenges, various methods have been proposed to incorporate more information, either from prior knowledge or from other sources, to improve the accuracy of segmentation. For instance, Cootes et al. (1995) introduced the shape prior into segmentation to keep the deformation of the contour consistent with statistical models from the PCA analysis. Following Cootes' method, several papers have focused on the incorporation of the shape prior information into traditional methods (Yang et al., 2004; Leventon et al., 2000; Tsai et al., 2003). In many situations, however, the use of shape prior models is limited because there are not enough training images to build the prior model. Further, it is not suitable for applications involving a large number of MR images that call for automatic computer-assisted segmentation procedures. Recently, Younis et al. (2007) proposed to combine the information from MRI and MR spectroscopy imaging (MRSI) for segmentation, with the advantage that MRI segmentation can be further corrected or enhanced by MRSI. Such a method, however, is not accessible for single-modality MRI analysis.

In this paper, we propose to incorporate the information from similar images to improve the accuracy of segmentation, instead of relying on information from prior knowledge on the shape of ROI or from other sources (i.e., modalities). We consider the problem of simultaneously segmenting multiple MR images, which, for example, can be a series of MR images scanned over time such as images of liver perfusion or dynamic cardiac motion, spatial slices of a volume, or images of symmetrical tissues such as lung and hippocampus. Due to their similarities, it is beneficial to share the image information with each other when executing segmentation, which we refer to as co-segmentation. Similar problems have been considered by Cheng and Figueiredo (2007). However they treated each image independently and only utilized the spatial information within each image for segmentation.

The term “co-segmentation” was used by others before, but with a slightly different meaning. For example, in Younis et al. (2007), co-segmentation refers to segmenting the brain region based on two modalities, MR and MRSI images. In Rother et al. (2006), co-segmentation refers to extracting a common part (e.g., the foreground) from an image pair, in which what is shared is not the segmentation structure across images but the model for the foreground segment. In other words, some pixels in these two images are assumed to be generated by the same statistical model, but how these pixels are located and how other pixels are generated or located in these two images are totally independent.

In our work, co-segmentation refers to jointly segmenting multiple images of which the segmentation structure (i.e., spatial configuration of the segmentation result) is shared. Note that in our
framework, the image data from a shared segment on different images are not necessarily modeled by the same distribution, so our approach is less sensitive to noise and image artifacts. We present a Bayesian co-segmentation procedure, in which the shared structure information across images, as well as the information among neighboring vertices (pixels/voxels) in the same image, is coded in a Markov random field prior. A Swendsen-Wang type algorithm is developed for posterior sampling, which updates a block of vertices simultaneously and is more efficient than the alternative single-site Gibbs sampler.

The remaining sections are organized as follows: Section 2.2 discusses the model and prior specification; Section 2.3 presents the Bayesian inference via two Gibbs sampling algorithms; In Section 2.4, we illustrate the utility of our method on four MRI data sets.

2.2 Method

2.2.1 Gaussian Mixture Models

For convention, we first introduce some notations. Assume all $J$ MR images have the same size in each of the $l$ dimensions, $(d_1, \cdots, d_l)$, and have totally $n = d_1 \times \cdots \times d_l$ vertices. For example, $l = 2$ for 2D images and $l = 3$ for 3D images. The vertices are often called pixels for 2D images and voxels for 3D images. Denote the intensity measure at vertex $i$ in image $j$ by $X_{ji} \in \mathbb{R}^p$, for example, an image with RGB format has $p = 3$. In this paper, we focus on gray scale MR images, so $X_{ji} \in \mathbb{R}$ with $j = 1:J$ and $i = 1:n$.

The goal of MRI segmentation is to partition vertices in an image into different clusters, based on the homogeneity of their intensity measures. Suppose there are totally $K$ segments or clusters. For each vertex $i$ in image $j$, we introduce a latent variable $Z_{ji} \in \{1, \ldots, K\}$. It is a common practice in MRI segmentation to model the intensity measure $X_{ji}$ in the same cluster with a Gaussian distribution (Wells et al., 1996; Permuter et al., 2006; Lee and Lewicki, 2002),

$$p(X_{ji} | Z_{ji} = k) = \phi(X_{ji}; \mu_{jk}, \sigma_{jk}^2),$$

(2.1)

where $\phi(\cdot; \mu, \sigma^2)$ denotes a normal density function with mean $\mu$ and $\sigma^2$. Under this framework, segmentation becomes the problem of inferring the latent variables $Z_{ji}$’s. There are several advantages of this model-based approach. First, it is a soft segmentation approach in the sense that $Z_{ji}$ is not restricted to take one fixed value, but treated as a random variable and allowed to have uncertainty
over the $K$ clusters. So it can handle cases that have ambiguities in the structure definition due to the sampling artifacts or poor resolution. Further, such a generative model can be easily extended to a semi-supervised setting where some vertices can have known labels given by experts. At last, with a model-based approach, the selection of hyper-parameters, such as the number of clusters, can be formulated as a model selection problem, thus a range of criteria such as AIC and BIC can be applied.

We write the unknown parameters and latent variables in this model as $(Z_{..}, \mu_{..}, \sigma^2_{..})$, where the subscript dot is a shorthand notation for the set containing all possible values in that subscript location. In a Bayesian framework, we make our inference of the unknowns based on the posterior distribution

$$
\pi(Z_{..}, \mu_{..}, \sigma^2_{..} | X_{..}) \propto \prod_j \prod_i n_j \prod_{i=1}^n p(X_{ji} | Z_{ji}, \mu_j, \sigma^2_{ji}) \times \pi(Z_{..}) \times \prod_{j=1}^J \prod_{k=1}^K \pi(\mu_{jk}, \sigma^2_{jk}),
$$

where the first line is the likelihood of the data $X_{..}$ and the second line are the prior distributions over the latent variable $Z_{..}$ and unknown parameters.

In co-segmentation, prior distributions play a critical role, which will be discussed in the next subsection. The posterior distribution given above is not in closed form and we will employ a MCMC algorithm for posterior sampling, which will be discussed in Section 2.3.

### 2.2.2 Co-segmentation Priors

For computation efficiency, we use conjugate normal priors for the cluster mean $\mu_{jk}$ and inverse Gamma for variance $\sigma^2_{jk}$, namely,

$$
\pi(\mu_{jk}) = N(\mu_0, \tau_0), \quad \pi(\sigma^2_{jk}) = \text{InvGa}(\alpha_0, \beta_0).
$$

In our empirical experiments, we use default values for these hyper-parameters $(\mu_0, \tau_0, \alpha_0, \beta_0)$, which correspond to non-informative or vague prior choices. For example, in all of our experiments, we set

$$
\mu_0 = 0.5, \quad \tau_0 = 0.1, \quad \alpha_0 = 1, \quad \beta_0 = 0.01,
$$

15
In specific clinical scenario, however, the hyper-parameters should be tuned based on the prior knowledge and experience of radiologists, which will make co-segmentation more effective.

In traditional Bayesian mixture models, the latent variables $Z_{ji}$’s in (2.1) are modeled as i.i.d. discrete random variables. However, such an independent model does not work well with image data since the spatial dependence among intensity measures $X_{ji}$’s from nearby vertices should be incorporated into the model. For example, it is natural to assume that neighboring vertices are likely to belong to the same segment. For co-segmentation, in order to utilize information across images, we further extend the neighborhood on the same image to multiple images (e.g., image pairs, or adjacent images if images are obtained over time). We start with a graph on the $nJ$ vertices from the $J$ MR images: connect two vertices $(ji)$ and $(j'i')$, if they are neighboring vertices on the same image or vertices on different images but from roughly the same location. For example, we set the neighborhood as $|j - j'| \leq 1$ and $|i - i'| \leq 1$ in our experiments. Denote all the edges by set $E_0$, then construct a Potts model on $Z_{ji}$’s as the following

$$
\pi(Z_{..}) \propto \exp \left\{ \sum_{(ji) \sim (j'i')} \beta_{(ji)(j'i')} I[Z_{ji} = Z_{j'i'}] \right\}
$$

(2.3)

where $I[\cdot]$ is an indicator function, the pair $(ji) \sim (j'i')$ means they are connected by an edge from $E_0$, and $\beta_{(ji)(j'i')}$ is an edge-dependent tuning parameter representing the interaction strength. Our prior distribution above leads to information sharing over multiple images and meanwhile preserving the spatial structure within each individual image which has been destroyed by vectorization.

Note that if one permutes the labels $Z_{ji}$’s simultaneously for all images, for example, relabel the 1st cluster as the 2nd and the 2nd as the 1st for all images, the likelihood (2.1), the prior distribution (2.3), and therefore the posterior distribution stay the same. This is known as the label-switching issue in Bayesian mixture modeling (McLachlan and Peel, 2000). To make a coherent inference, we need to fix the order of the $K$ clusters on one particular image. In the later analysis, we will order the $K$ clusters on the first image by their cluster means, i.e., $\mu_{11} < \cdots < \mu_{1K}$.

Also note that if one permutes the labels on some (not all) images, the likelihood still stays the same, but the prior and the posterior change, since a priori we favor the configuration satisfying $Z_{ji} = Z_{j'i'}$ where $(ji)$ and $(j'i')$ are neighboring vertices on different images. So given a configuration $Z_{..}$, it is possible to shuffle the labels on image 2 to image $J$ (the labels on the 1st image have been fixed) to increase the prior (2.3), and therefore to increase the posterior probability. Finding the optimal order of the labels on the $(J - 1)$ images is time-consuming, so instead we give a simple
greedy algorithm to find a sub-optimal solution in the Appendix. This shuffling step is not a valid MCMC step, but is used to guide the chain to reach regions with high posterior probabilities, so in the MCMC algorithms in Section 2.3, this shuffling step is only called in the burning period.

2.3 MCMC Algorithms

With the Potts prior on $Z_\cdot$, the posterior distribution (2.2) is not in closed form. However, the conditional distributions for each parameter or latent variable given others and the data are from known parametric families, which leads to a simple Gibbs sampling scheme given in Section 2.3.1. We refer to it as a single-site Gibbs sampler (SSGS), since it updates the cluster membership $Z_{ji}$ sequentially over each vertex. A more efficient algorithm that updates a block of $Z_{ji}$’s based on Swendsen-Wang method is described in Section 2.3.2, which we refer to as a SW Gibbs sampler (SWGS).

2.3.1 A Single-site Gibbs Sampler

In the single-site co-segmentation, all parameters and latent variables are updated sequentially in each iteration as follows.

I. Initialization. Assign initial values for $(\mu_\cdot, \sigma^2_\cdot, Z_\cdot)$ such that $\mu_{11} < \cdots < \mu_{JK}$. Then, execute the following MCMC steps recursively:

II. At the $t$-th iteration,

1. Update $Z_{ji}$ sequentially for $i = 1 : n$ and $j = 1 : J$ with

$$P(Z_{ji} = k | X_\cdot, \cdots) = \frac{w_k}{\sum_{l=1}^{K} w_l},$$

where $(\cdots)$ denotes all other parameters and latent variables (except $Z_{ji}$) evaluated at their current values, and

$$w_k \propto \phi(X_{ji}; \mu_{jk}, \sigma^2_{jk}) \times \exp \left\{ \sum_{(j'i') \in \mathcal{N}(ji)} \beta_{(ji)(j'i')I}[Z_{j'i'} = k] \right\}, \tag{2.4}$$

where the set $\mathcal{N}(ji)$ denotes all the vertices $(j'i')$ that are neighbors of $(ji)$ as defined by the initial edge set $E_0$.
2. For each cluster $k = 1: K$ on the $j$-th image ($j = 1: J$), let $\overline{X}_{j|Z=k}$ denote the cluster mean for the $k$-th cluster and $n_{jk} = \sum_{i=1}^{n} I[Z_{ji} = k]$ the corresponding cluster size. Update

$$\mu_{jk}|\overline{X}_{.,.}: \cdots \sim N(\mu, \tau^2)$$

where $\mu = r \overline{X}_{j|Z=k} + (1 - r) \mu_0$, and

$$r = \frac{\tau_0^2}{\sigma_{jk}^2/ n_{jk} + \tau_0^2}, \quad \frac{1}{\tau^2} = \frac{1}{\sigma_{jk}^2/ n_{jk}} + \frac{1}{\tau_0^2}.$$

Then update

$$\sigma_{jk}^2|\overline{X}_{.,.}: \cdots \sim \text{InvGa}\left(\frac{n_{jk}}{2} + \alpha_0, \frac{1}{2} \sum_{i:Z_{ji} = k} (X_{ji} - \mu_{jk})^2 + \beta_0\right).$$

3. Label-switching: relabel the $K$ clusters such that $\mu_{11} < \mu_{12} < \cdots < \mu_{1K}$.

4. Shuffle the labels on image 2 to image $J$ when in the burning period (see Appendix A).

### 2.3.2 A Swendsen-Wang Gibbs Sampler

In SSGS, the labels $Z_{ji}$’s are updated sequentially. Such a single-site operation is inefficient for segmenting MR images in two aspects: (1) Traversing all vertices sequentially in each iteration is time-consuming due to the high dimension of MR images; (2) It might take many iterations to update a set of coupled vertices (Barbu and Zhu, 2005). In this subsection, we present a more efficient algorithm, which still operates image by image but on each image updates a block of vertices simultaneously.

Let $Z_j$ denote all the labels on the $j$-th image. Recall that the joint conditional distribution of $Z_j$, given the data and other unknowns is

$$\pi(Z_j|X_{.,.}: \cdots) \propto \exp\left\{\sum_{i=1}^{n} g_{ji}(Z_{ji})\right\} \times \exp\left\{\sum_{(ji) \sim (j'i')} \beta_{(ji)(j'i')} I[Z_{ji} = Z_{j'i'}]\right\}, \quad (2.5)$$

where

$$g_{ji}(k) = \log \phi(X_{ji}; \mu_{jk}, \sigma_{jk}^2) + \sum_{(ji) \sim (j'i')} \beta_{(ji)(j'i')} I[Z_{ji} = k].$$

The distribution (2.5) remains in the Potts family with the likelihood and prior contribution from other images as the external field term, So for each image we can sample a block of coupled $Z_{ji}$’s simultaneously using the Swendsen-Wang (SW) algorithm.
The SW algorithm, proposed by Swendsen and Wang (1987), is an efficient sampling method for Potts models. Instead of updating the \( Z \) values vertex by vertex, it updates the values block by block as follows. Given the current configuration of \( Z \)’s, connect neighboring vertices with the same \( Z \) values with certain probability that depends on \( \beta \) in (2.5). The connected vertices form disjointed blocks, and then SW algorithm updates the \( Z \) values in a block simultaneously. The SW algorithm can be justified by augmenting the space of \( Z \)’s by bonding variables (Edwards and Sokal, 1988; Higdon, 1997). Recently, Barbu and Zhu (2005) provided another justification for SW from the aspect of Metropolis-Hastings, which leads to the use of SW algorithm for non-Potts models.

Our SW Gibbs sampling algorithm, abbreviated as SWGS, is similar to SSGS given in the previous subsection, except at step II(1) we update \( Z_j \) for \( j = 1 : J \) as follows. Given the current labeling \( Z_j \) for the \( j \)-th image, connect two neighboring vertices (\( jij \) and \( jij' \)) that have the same \( Z \) value with probability \( 1 - \exp\{-\beta_{jij}(jij')\} \) where \( \beta_{jij}(jij') \) is the edge-dependent interaction parameter as in (2.3) and (2.5). Then the vertices on the \( j \)-th image are divided into \( m \) disconnected components \( (V_1, \ldots, V_m) \), where \( V_i \cap V_l = \emptyset \) and \( \bigcup_{l=1}^m V_l = \{1, \ldots, n\} \). For each component \( V \), we update their labels \( Z_jV \) simultaneously with

\[
P(Z_jV = k|X, \cdots) \propto \exp\left\{ \sum_{i \in V} g_{ji}(k) \right\}.
\]

Although our SWGS algorithm is designed for cases where the conditional distributions are from the Potts family, it can be easily extended to sample arbitrary posterior distributions. For example, for the bounding box prior in Lempitsky et al. (2009), and the prior used in Barbu and Zhu (2005) that encourages large and connected segments, the corresponding conditional distribution of \( Z_j \) is no longer from the Potts family. Nevertheless, we can still update the labels block by block through a Metropolis-Hastings step. We formulate the disjoint blocks as described before, then for block \( V \), propose to assign \( Z_jV \) a new label. The acceptance ratio for the new configuration \( Z'_j \) is given by

\[
\alpha(Z_j \rightarrow Z'_j) = \min\left\{ 1, \frac{q(Z_j, V|Z'_j) \pi(Z'_j, X, \cdots)}{q(Z'_j, V|Z_j) \pi(Z_j, X, \cdots)} \right\}.
\]

The proposal density \( q(Z_j, V|Z'_j) \) is difficult to evaluate since there are many different ways to obtain the same vertices set \( V \). An important result from Barbu and Zhu (2005) showed that the ratio of the two proposal densities \( q(Z_j, V|Z'_j)/q(Z'_j, V|Z_j) \) is of simple form. So the acceptance ratio can be computed easily. Further, with a particular choice of the proposal distribution for assigning labels for \( Z'_jV \), the acceptance ratio is 1.
2.3.3 Posterior Inference and Choice of Hyper-parameters

In the Bayesian framework, parameters estimation and inference on latent variable $Z_{ji}$’s become a matter of summarizing the posterior distribution. In general, we can report either posterior mean, median or mode over the MCMC samples. In our experimental studies, we adopted posterior means for parameter estimation and posterior modes for $Z_{ji}$’s.

Our algorithm requires the number of clusters $K$ to be pre-given. In some applications, we can set $K$ to be the number of different tissues in MR images. For example, it is common to set $K = 3$ in segmenting MR brain images, which correspond to gray-matter, white-matter, and cerebrospinal-fluid tissue. Alternatively, we can use model selection criteria such as AIC and BIC to select $K$.

The choice of $\beta_{(ji)(j'i')}$, the edge-dependent interaction parameter as in (2.3), plays an important role in co-segmentation: large value leads to high influence of neighboring vertices. To the best of our knowledge, there is no optimal way to select $\beta$. We could try different $\beta$ values and then select them by some model selection criteria, which, however, is time-consuming since there are many $\beta$’s (same as the number of edges). In our empirical study, we set

$$
\beta_{(ji)(j'i')} = -\log\left(1 - e^{-h|X_{ji} - X_{j'i'}|}\right),
$$

where $h$ is a positive tuning parameter, implying high influence of neighboring vertices when their intensity measures are close. Then choose $h$ as follows. For each image, we plotted the histogram for the intensity difference among neighboring vertices, $|X_{ji} - X_{j'i'}|$, where $(ji)$ and $(j'i')$ are neighbors. (The intensity measures have been normalized with $X_{ji} \in [0, 1]$.) We found that there was no valleys between 0 and 0.05 in all histograms, that is, two vertices would be classified in the same cluster when $|X_{ji} - X_{j'i'}| < 0.05$, if using a simple thresholding rule. Then we chose $h$ such that neighboring vertices with intensity difference equal to 0.05 had a 50% chance to be connected in the SW algorithm, that is, we set $1 - \exp\{-\beta\} = e^{-0.05 \times h} \approx 0.5$. Such an equation leads to $h \approx 14$. Eventually, we adopted a less informative choice $h = 15$.

2.4 Experiments

We demonstrate the efficiency and accuracy of our co-segmentation algorithm on four different types of MRI data sets. We compare the performance of co-segmentation with individual segmentation
on the first three data sets: (i) a pair of synthetic brain images; (ii) five real corpus callosum MRI slices from a volume data, and (iii) three 2D abdominal MR images scanned over time. We then demonstrate the efficiency of SWGS versus SSGS on a 3D dynamic cardiac data set. The individual segmentation procedure is almost the same as the co-segmentation, except that $\beta_{(j)ij'}$ in (2.3) is set to be zero when $j \neq j'$, that is, information is shared among nearby vertices on the same image, but not across images.

Evaluating the accuracy of medical images segmentation is difficult due to the lack of “true” segmentation results. Since segmentation for medical images is mainly used to identify ROIs and true ROIs can be provided by experts, after obtaining the segmentation result, we label some clusters as ROI and the other clusters as non-ROI. In all the figures, we display only this two-class segmentation result: ROI versus non-ROI. We employ quantitative evaluations based on the proportion of correctly identified ROI, as introduced in Fenster and Chiu (2005). Let $V_T$ and $V_S$ denote the regions enclosed by the true boundary of ROI (provided by human experts) and the estimated boundary (from the segmentation algorithm) respectively. Define the true positive (TP) volume as the volume enclosed by both the true and estimated boundaries, i.e., $V_{TP} = V_S \cap V_T$, the false positive (FP) volume is $V_{FP} = V_S - V_T$, the false negative (FN) volume is $V_{FN} = V_T - V_S$, and $V$ denotes the total region. Then define

$$\text{True Position Fraction (TPF)} = \frac{V_{TP}}{V_T},$$

$$\text{False Position Fraction (FPF)} = \frac{V_{FP}}{V - V_T},$$

$$\text{False Fraction (FF)} = 1 - \frac{V_{FP} + V_{FN}}{V_T}.$$

Following Fenster and Chiu (2005), we name the last measure “False Fraction”, although it is the measure that we prefer to be large.

For each data set, we first normalize the intensity measures such that $Y_{ji} \in [0, 1]$. Then apply some standard pre-processing in image segmentation to reduce the number of vertices. We first use Laplacian of Gaussian (a compound operator that combines a smoothing Gaussian-shape operation with a differentiation Laplacian operation) to obtain the closed contours of boundary for each image. The contours give rise to a fine partition of an image and vertices falling into the same boundary can be treated as a super-vertex. To keep the consistency of super-vertices over $J$ images, we use the overlap of the $J$ partitions to form super-vertices. Figure 2.1 depicts the final super-vertices of three sequential images.
2.4.1 Synthetic Contaminated Brain Image Pair

In this experiment, we construct a synthetic contaminated image pair based on a brain image (of size $181 \times 217$) from the Simulated Brain Database\(^1\), which is derived from an average of 27 T1-weighted images of a normal brain. We add white noises at two different regions of the brain, one contaminated region on one image. This pair of synthetic data is shown on the top row of Figure 2.2. We want to demonstrate that the contaminated region in each image could be retrieved much better under the supervision of the other one.

We set $K = 3$ that correspond to GM (gray matter), WM (white matter) and CSF (cerebrospinal-fluid tissue). The results for individual segmentation and co-segmentation for WM are shown in Figure 2.2 and the quantitative evaluation is in Table 2.1. Although co-segmentation does slightly worse than individual segmentation on FF (approximately 2% higher than the individual segmentation), it extracts 17% more ROI than the individual segmentation as shown in TPF. As we expect, co-segmentation provides a much better overall result than individual segmentation since it utilizes the information from both images that are known to be similar.

\(^1\)http://www.bic.mni.mcgill.ca/brainweb/
Figure 2.2: Data and results for the contaminated brain image pair: original images (top row), results from individual segmentation (middle row), and results from co-segmentation (bottom row).

<table>
<thead>
<tr>
<th></th>
<th>TPF</th>
<th>FPF</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-seg</td>
<td>0.98</td>
<td>0.3</td>
<td>0.03</td>
</tr>
<tr>
<td>Individual</td>
<td>0.81</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>Co-seg</td>
<td>0.97</td>
<td>0.96</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.81</td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>TPF</th>
<th>FPF</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-seg</td>
<td>0.99</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Individual</td>
<td>0.82</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>Co-seg</td>
<td>0.96</td>
<td>0.94</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.82</td>
<td>0.93</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>TPF</th>
<th>FPF</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-seg</td>
<td>0.98</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.99</td>
<td>0.93</td>
<td>0.01</td>
</tr>
<tr>
<td>Co-seg</td>
<td>0.96</td>
<td>0.94</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.89</td>
<td>0.93</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>TPF</th>
<th>FPF</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-seg</td>
<td>0.98</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>1.00</td>
<td>0.91</td>
<td>0.01</td>
</tr>
<tr>
<td>Co-seg</td>
<td>0.96</td>
<td>0.94</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.79</td>
<td>0.93</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>TPF</th>
<th>FPF</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-seg</td>
<td>0.94</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.99</td>
<td>0.91</td>
<td>0.01</td>
</tr>
<tr>
<td>Co-seg</td>
<td>0.96</td>
<td>0.94</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.79</td>
<td>0.93</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>TPF</th>
<th>FPF</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-seg</td>
<td>0.94</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.95</td>
<td>0.88</td>
<td>0.01</td>
</tr>
<tr>
<td>Co-seg</td>
<td>0.93</td>
<td>0.90</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.87</td>
<td>0.90</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>TPF</th>
<th>FPF</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-seg</td>
<td>0.93</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.95</td>
<td>0.88</td>
<td>0.01</td>
</tr>
<tr>
<td>Co-seg</td>
<td>0.93</td>
<td>0.90</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.48</td>
<td>0.90</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>TPF</th>
<th>FPF</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-seg</td>
<td>0.92</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.95</td>
<td>0.90</td>
<td>0.01</td>
</tr>
<tr>
<td>Co-seg</td>
<td>0.93</td>
<td>0.90</td>
<td>0.01</td>
</tr>
<tr>
<td>Individual</td>
<td>0.77</td>
<td>0.90</td>
<td>0.01</td>
</tr>
</tbody>
</table>
2.4.2 Multiple Corpus Callosum Slices

Several studies have indicated that the size and shape of the corpus callosum (CC) in human brain are correlated to sex, age, brain growth, and various types of brain dysfunction. In order to find such correlations, computer-assisted segmentation is needed (Lundervold et al., 1999). Five CC slices, each of size $70 \times 100$, from the Simulated Brain Database are shown in Figure 2.3 (top row).

We applied co-segmentation and individual segmentation algorithms on this data set with various choice of $K$. As discussed in Lundervold et al. (1999), the number of tissue types in CC images is less than 10. So $K$ ranges from 2 to 10 as in the plot of AIC and BIC versus $K$ shown in Figure 2.4. The most significant reduction of AIC and BIC occurs when $K$ changes from 4 to 5. Both criteria continue decreasing after $K = 5$, but with a much slower rate. So we select $K = 5$. The segmentation results are shown in the middle and the last rows in Figure 2.3. We can see that co-segmentation is roughly the same as the individual segmentation on the first four images, but is much better on the last one. As shown in Table 2.1, individual segmentation of the last image CC5 has FPF $= 0.17$ and FF $= -0.98$ indicating serve over segmentation, i.e., a huge false positive error. However, this is avoided by our co-segmentation procedure, which has FPF $= 0.00$ and FF $= 0.83$ for CC5, since it utilizes the information across images.
2.4.3 Sequential Liver Images

Liver perfusion is a quantitative measurement of blood flow in the liver, which provides useful information on the assessment and treatment of liver diseases. Precise segmentation of livers under perfusion is an important preliminary step for further analysis. The difficulty of liver segmentation is the ambiguity of the boundaries that are connected with other organs like the heart. In this experiment, we use sequential liver images from Chen and Gu (2006), which are 2D abdominal MR images (of size $145 \times 118$) scanned over time and are shown in Figure 2.5.

The bottom panel of Figure 2.4 presents the selection of $K$ using AIC and BIC, both of which select $K = 6$. According to Figure 2.5 and Table 2.1, our co-segmentation algorithm outperforms the individual one.

2.4.4 4D cardiac data

Cardiovascular disease is one of the leading death causes in the world. Segmenting cardiac MR images is a common medical practice since it provides clinically useful indicators of heart function. In this experiment, we use a series of three 3D MR images (of size $101 \times 101 \times 12$) from Andreopoulos and Tsotsos (2008), which are scanned along a cardiac circle. The purpose of segmentation here is to delimitate the endocardium. As shown in Figure 2.6, the nearby images (along the time domain)
Figure 2.5: Data and results for sequential liver images: original images (top row), results from individual segmentation (middle row), and results from co-segmentation (bottom row).

have strong similarities, which motivates the use of co-segmentation.

For this set of data, the two co-segmentation algorithms, SWGS and SSGS, output roughly the same result in terms of accuracy. SWGS, however, has an apparent advantage in terms of speed. Figure 2.7 (left) compares their convergence rates in terms of the number of iterations, and indicates that SSGS needs almost 9 times more iterations than SWGS to reach convergence. This is because in each iteration, SWGS updates the cluster labels block by block, while SSGS does it pixel by pixel. Further, the way SWGS updates the cluster labels requires less computation, therefore less time, to finish an iteration than SSGS, as supported by the right panel of Figure 2.7 that compares the convergence speed in terms of time.
Figure 2.6: Data and results for the cardiac data: one slice of the 3D image is presented at 3 time points (top panel, from left to right), and results from co-segmentation (lower panel).

Figure 2.7: Convergence comparison between SSGS and SWGS. The left plot is the convergence rates of SSGS and SWGS which indicates SWGS needs only 1/10 iterations steps to be convergent. The right plot shows the speed in time for convergency which evaluates the SWGS is around 25 time faster than SSGS.
Benefit segmentation, that is, grouping consumers into different segments based on their product preference, is an essential problem of marketing theory and practice. Modern marketing environments impose some new challenges to traditional segmentation methods. For example, companies are adding more and more features (large $p$) into a single product, while the data we could collect from each consumer is of relatively small size (small $n$). Although most methods in benefit segmentation assume consumers use every product feature in their decision making, recent research has shown that consumers only consider a subset of features. Further, the heterogeneity among consumers in selecting important product features should be used as an additional index for market segmentation and for new product development.

In responding to these challenges, we propose a Bayesian approach for collaborative inference among consumers. In addition to introducing a hierarchical layer to bridge the population-level model and the individual-level model, we build a similarity graph/network among consumers from auxiliary variables and then incorporate such structures into our prior specification to improve the segmentation result. In terms of statistical methodology, the proposed method is a Bayesian approach for multi-task learning problems with structured sparsity, where the structures we consider are stochastic groups and graphs. Connections with existing work on structured sparsity are discussed. And we demonstrate the utility of our method on several simulated data sets and a real case study example on online shopping websites.

3.1 Marketing Segmentation

Market segmentation is an essential element in marketing theory and practice (Wedel and Kamakura, 1999). The concept of market segmentation was originally introduced by Smith (1956), “... viewing a heterogeneous market as a number of smaller homogeneous markets, in response to different preferences, attributable to the desires of consumers for more precise satisfaction of their varying
wants.” The variables or criteria used to segment a market, or in other words, to group customers, are called the bases for market segmentation. Choices of bases include demographical, cultural, geographical variables, and personality or life-style, which are variables not depending on products. On the other hand, we can also segment a market based on product-specific variables such as usage frequency, store loyalty, and product benefit. Specifically “benefit segmentation” refers to segmenting markets based on the way consumers respond to product features in their decision making, which are widely used in marketing research and practice for better advertising and distribution of new products (Haley, 1968; Calantone and Sawyer, 1978; Wind, 1978).

Conjoint analysis is a common approach for benefit segmentation, and regarding statistical methodology, mixture models are often used. Consider a simple binary conjoint experiment, where each subject/consumer has to choose between two products of different combinations of the $p$ product features. Repeat this experiment $n$ times for each subject. Let $C_i$ denote the $n$ choices made by the $i$th subject, which depend on the product features via the following generalized linear model

$$C_i = g(X_i \beta_i + \epsilon_i), \quad i = 1, \ldots, m,$$

where $\epsilon_i$ denotes the error term and $X_i$ denotes the corresponding $n \times p$ design matrix. The unknown individual regression coefficients $\beta_i$'s are further modeled by a mixture with each component corresponding to a market segment or a cluster of consumers. For a review on methodology for segmentation, see Wedel and Kamakura (1999).

Although mixture models and conjoint analysis have been successfully applied on many real cases, the modern marketing environment imposes some new challenges.

The first challenge is the increasing number of product features (large $p$). Companies are adding more and more features into a single product such as a cellphone or a laptop. This new trend restricts the use of some traditional methods, for example, conjoint analysis is recommended not to handle more than six features (Green and Srinivasan, 1978). In responding to this new challenge, researchers in marketing science have proposed some recent alternatives, including hybrid conjoint techniques using self-explication (Johnson, 1987) that relies on subjects in the experiment to tell the research what the important product features are, and dimension reduction techniques such as multidimensional scaling (DeSarbo et al., 2008). These new approaches, however, either do not reflect the real-life choice scenario or are difficult to interpret. From the managerial standpoint, of major interest is the relationship between a single feature and consumer’s utility, so managers
can focus their limited resources on a few important features, instead of spread over all features. From the consumers’ perspective, given such a long list of product features, they usually make their choice only based on a subset of important features, due to convenience, cost of thinking or lack of expertise about some features (Gilbride et al., 2006), in contrast to the assumption held in traditional segmentation methods that consumers consider every feature for their product choice. Further, consumers’ heterogeneity in feature selection should be incorporated into marketing models as a new index for market segmentation.

Another challenge comes from the restriction that the sample size \( n \), the number of observations researchers collect from each subject, cannot be large, before the subject gets bored or fatigued. So when the dimension \( p \) gets large, each individual model (3.1) falls into the typical “high dimension low sample-size” paradigm, which makes inference on feature selection for each individual model challenging.

To address these issues we propose to encourage similar consumers to share their experience. The idea is to construct a similarity graph \( G = (V, E) \) on the subjects based on social networks or pairwise correlation calculated using auxiliary psychological, cultural, or demographic data. The vertex set \( V \) denotes all the subjects, and \( E \) contains all the edges \( i \sim j \) (i.e., neighbors in the graph) indicating that subjects \( i \) and \( j \) are similar. In some graphs there may be an additional similarity weight \( e_{ij} > 0 \) attached to each edge. We will incorporate this graph structure into prior specification with the rationale that similar consumers may respond to product features in a similar way, therefore similar consumers share their information for statistical inference. This is why we call our approach collaborative.

In Section 3.2, we state market segmentation as a multi-task learning problem and draw its connection with structured sparsity. In Section 3.3, we develop a Bayesian approach for this problem and address issues with prior specification and computation for posterior inference. Section 4.4 presents simulation studies and Section 4.5 presents our analysis on a real example.

### 3.2 Collaborative Inference with Structured Sparsity

In modeling the \( m \) consumers’ decision making, we simultaneously fit \( m \) generalized linear models (GLM). We can also write models in (3.1) compactly as a single GLM by stacking \( C_i \)’s and \( \beta_i \)’s together and \( X_i \)’s along the diagonal

\[
C_{mn \times 1} = g(X \beta_{mp \times 1} + \text{error}), \quad X_{mn \times mp} = \text{diag}(X_i)_{i=1}^m.
\]
The model dimension $mp$ is large and increases with the sample size $mn$, falling into the typical “high dimension low sample-size” paradigm. A common premise held in such high-dimensional data analysis is that only a small fraction of those variables is relevant to the prediction of the response variable. In addition to sparsity, it is natural to impose some structure dependence among $\beta_i$’s due to the dependence among the $m$ models. Structures play two roles here: 1) they provide additional information on $\beta_i$’s sparsity pattern and therefore lead to estimation efficiency and improved accuracy, and 2) in many cases the assumed structure naturally leads to information sharing across different models. In this framework, it is key to utilize the dependence among the $m$ models in estimation and variable selection, which we term as “collaborative inference”.

A natural way to impose dependence among the $m$ models is to assume that relevant variables are shared by the $m$ models. This corresponds to the so-called group sparsity, in which elements of the regression vector $\beta$ in (3.2) are divided into disjoint groups or blocks. The sparsity assumption is applied at the group level, that is, coefficients in the same group are zero or non-zero simultaneously. Yuan and Lin (2006) introduced gLasso for sparse recovery with group structures. The penalty of gLasso takes the following form in the context of linear regression with squared error loss

$$\arg\min_{\beta} \left[ \|Y - X\beta\|^2 + \lambda \sum_j \|\beta_{g_j}\| \right]$$

(3.3)

where $\| \cdot \|$ denotes the $L_2$ norm and $\beta_{g_j}$ denotes the elements of $\beta$ in the $j$th group. Note that the penalty term in (3.3) employs $L_2$ norm within groups and $L_1$ norm between groups, and has the advantage of excluding or including variables in the same group simultaneously. The use of gLasso to incorporate knowledge of group structures has been extended for multi-task learning with linear regression models (Lounici et al., 2009), for general loss functions (Kim et al., 2006), for logistic models (Meier et al., 2006), and for nonlinear modeling with multiple reproducing kernel Hilbert spaces (Bach, 2008). Statistical properties of gLasso and its extensions have been studied by Huang and Zhang (2009), Koltchinskii and Yuan (2008), Nardi and Rinaldo (2008), and others. An extension of this simple disjoint group structure is a hierarchical group structure where groups may be overlapped and ordered. To handle hierarchical groups, Zhao et al. (2009) proposed the composite absolute penalty.

However, gLasso type penalties cannot be applied for our market segmentation model. The group assumption imposed on the $m$ models implies that the set of relevant product features is the same for all consumer, which is too strong and ignores the possible heterogeneity among the $m$ models. It
is reasonable to assume that the models can be divided into $K$ groups and models in the same group share the same set of relevant variables. However, different from the group sparsity considered by the aforementioned work, here in market segmentation, the group structure is not pre-determined, but part of the statistical inference.

Recall that we need to fit a GLM for each consumer, where the individual sample size $n$ is relatively small comparing to the number of product features $p$. To obtain a reliable statistical inference with a limited sample size, we propose to pool information across similar consumers. In most marketing experiments, besides their choice making, one also collect psychological, cultural, or demographic data for each individual. Those information can be used to define a similarity measure among consumers, which induces a graph structure $G = (V, E)$ among the $m$ models, where the vertex set $V = \{1, \ldots, m\}$ contains the $m$ models, and the edge set $E$ contains all the neighborhood pairs $i \sim j$. In some graphs there may be an additional similarity weight $e_{ij} > 0$ attached to each edge. Graph structures have been considered in the literature before. For example, in the context of gene network analysis, Li and Li (2008) propose to use a graph Laplacian penalty on $\beta$, i.e.,

$$
\sum_{i \sim j} e_{ij} \| \beta_i - \beta_j \|^2,
$$
in additional the sparse $L_1$ penalty. In their recent work, Li and Li (2010) further established the asymptotic properties of their method including the error bound and model selection consistency.

A limitation of the Laplacian penalty proposed by Li and Li (2008) is that their method only leads to information sharing among models in an adjacent neighborhood, i.e., a local collaboration, in contrast to the global collaboration led by the group assumption. In our proposed Bayesian approach, we bridge these two approach by utilizing both the graph and group structures, which leads to an information sharing at both the local and global levels. As an alternative to regularization approaches, Bayesian methods offer an effective and conceptually appealing approach to incorporate structure information: the assumed structure can be utilized in prior specification, the use of hyperparameters can remove the dependence caused by the structure assumption and make computation more efficient, and the Bayesian hierarchical modeling naturally leads to information sharing across connected structure components. For example, Bayesian approaches to utilize group and graph type structures have been proposed for compressive sensing (Ji et al., 2009), for analyzing fMRI data (Smith and Fahrmeir, 2007), for applications in genomics (Li and Zhang, 2010), and for identifying differentially expressed genes via multiple hypotheses testing (Dahl and Newton, 2007; Wei and Li, 2008; Wei and Pan, 2008).
3.3 Proposed Bayesian Approach

3.3.1 Model and Priors

For simplicity, we focus on binary choice data, but extensions to multivariate outcomes are straightforward (Chib et al., 1998; Imai and van Dyk, 2005). We model the binary choice data by a probit model at the individual level for \( i = 1, \ldots, m \),

\[
C_i = I[Y_i > 0], \quad Y_i = X_i \beta_i + \epsilon_i,
\]

(3.4)

where \( I[\cdots] \) is an indicator function operated on the continuous latent vector \( Y_i \) and \( \epsilon_i \sim N(0, \sigma^2 I) \). Although the scale parameter \( \sigma \) is often set to be 1 to make the model identifiable (Albert and Chib, 1993), we employ the parameter expansion approach (Liu and Wu, 1999; Meng and van Dyk, 1999; Hobert and Marchev, 2008) to improve the mixing of the Markov chain Monte Carlo (MCMC) algorithm. In the parameter expansion approach, the model is overparameterized and each value of \( \sigma \) corresponds to an equivalent class or orbit of the model, but the use of the Haar measure

\[
\pi(\sigma^2) = \text{InvGa}(0, 0)
\]

(3.5)

still gives rise to a valid MCMC chain on the original model.

Assume there are totally \( K \) segments among the consumers. Then we model the individual coefficient \( \beta_i \) by a mixture of \( K \) components. Introduce a segment indicator variable \( H_i \in \{1, \ldots, K\} \), then

\[
\pi(\beta_i \mid H_i = k) = g_k(\beta_i \mid \mu_k),
\]

where \( g_k \) could be a point mass at \( \mu_k \) or any symmetric continuous distribution, such as normal or heavy tailed distributions, centered at \( \mu_k \). Of interest are the coefficients \( \mu_k \)'s at the segment level. To adapt to the assumption that consumers use only a small fraction of product features in their decision making, we use a spike and slab type prior on \( \mu_k \) for the purpose of variable selection (Mitchell and Beauchamp, 1988; George and McCulloch, 1993; Ishwaran and Rao, 2005).

For \( k = 1 : K \) and \( j = 1 : p \)

\[
\pi(\mu_{kj} \mid Z_{kj}) = 0 \cdot I[Z_{kj} = 0] + N(0, \tau^2) \cdot I[Z_{jk} = 1],
\]

(3.6)

\[
\pi(Z_{kj}|w) = \text{Bernoulli}(w),
\]

(3.7)
where $Z_{kj}$ is the binary latent variable indicating whether $\mu_{kj} = 0$ or not.

We consider two choices for $g_k$ in the prior specification for $\beta_i$. One is what we refer to as “sparsity at the individual level”, in which

$$
\pi(\beta_{ij} | \mu_{kj}, H_i = k) = 0 \cdot I[\mu_{kj} = 0] + N(\mu_{kj}, v) \cdot I[\mu_{kj} \neq 0].
$$

(3.8)

That is, the individual coefficient vector $\beta_i$ share the same sparsity pattern as the corresponding segment coefficient $\mu_k$. The other is what we refer to as “sparsity at the segment level”, in which

$$
\pi(\beta_{ij} | \mu_{kj}, H_i = k) = N(\mu_{kj}, v).
$$

(3.9)

That is, the individual coefficient vector $\beta_i$ is no longer sparse, in terms of $L_0$ norm, but such a dense coefficient vector is needed to model random effects for individual consumers for some cases.

In traditional Bayesian mixture models, the segment indicator variables $H_i$’s are modeled by independent discrete distributions over $\{1, \ldots, K\}$. Such an independent model, however, does not take into account the prior dependence among the $m$ consumers. Given a similarity graph $G = (V, E)$, we propose to model $H = (H_1, \ldots, H_m)$ jointly by a Potts model

$$
\pi(H) \propto \exp\left\{ \sum_{i=1}^{m} \alpha_i(H_i) + \lambda \sum_{i \sim i'} e_{ij} I[H_i = H_{i'}] \right\},
$$

(3.10)

where $i \sim i'$ indicates consumer $i$ and consumer $i'$ are neighbors in the graph $G$ and $e_{ij}$ denotes the corresponding similarity weight. The prior distribution above is a special case of the discrete Markov random field on graph $G$. The $\alpha_i$’s can be used to reflect prior knowledge on the cluster size. For our market segmentation application, we will make no prior assumption on the size of each segment, so set $\alpha_i = 0$. The prior distribution over $H$ favors configurations where neighbors with high similarity weights are assigned to the same segment. Consequently, those similar consumers will share their information in estimating the sparse segment coefficient $\mu_k$. The use of Markov random fields in Bayesian variable selection has been explored by others (Smith and Fahrmeir, 2007; Li et al., 2010), however, their methods only lead to local collaboration—only adjacent neighbors in the graph share their experience. The novel contribution of our approach is that in addition to the graph structure, we also incorporate a (stochastic) group structure which leads to another layer of global collaboration—individuals in the same group/segment who may not be neighbors also share their experience in estimating $\mu_k$’s.
For our market segmentation model, the latent variables are the continuous outcomes $Y$ for the probit model, the variable selection indicators $Z$, and the segment indicators $H$; the parameters of interest are the segment coefficients $\mu_k$'s and the individual coefficients $\beta_i$'s; other parameters include proportion of relevant variables $w$ and several variance parameters $(\sigma^2, \tau^2, v)$. We have elicited the prior distributions for all the parameters except $(w, \tau^2, v)$, for which we use conjugate priors for the sake of computation efficiency

$$
\pi(w) = \text{Beta}(a_0, b_0), \quad \pi(\tau^2) = \text{InvGa}(s_1, s_2), \quad \pi(v) = \text{InvGa}(t_1, t_2), \quad (3.11)
$$

where $a_0, b_0, s_1, s_2, t_1, t_2$ are hyper-parameters.

### 3.3.2 Computation

We employ Gibbs samplers for posterior inference over $(Y, Z, H, \mu, \beta)$, as well as $(w, \tau^2, v)$, by iteratively sampling from each parameter or a block of them conditioning on the data and other parameters.

In a probit model with binary response $C$, without observing the latent variable $Y$, the mixing rate might be slow (Liu, 2002). To speed up the MCMC chain, we employ the parameter expansion data augmentation (PX-DA) approach as described in Liu (2002). In PX-DA, we do not need to sample $\sigma^2$ but a working parameter $c$ from a chi-square distribution, then rescale the model outcome $Y$ and other parameters.

Depending on which prior distribution of $\beta$, (3.8) and (3.9), is used, the sampling scheme is a little different. When using (3.9), we sample $(\mu, Z)$ together since they are coupled. Note that prior (3.8) on $\beta$ can be expressed as

$$
\pi(\beta_{ij}|Z_{kj}, H_i = k, \mu_{kj}) = 0 \cdot I[Z_{kj} = 0] + N(\mu_{kj}, v) \cdot I[Z_{kj} \neq 0].
$$

So $(\mu, Z, \beta)$ are coupled and we have to sample all three of them together. The two Gibbs samplers are given in the Appendix B and C as Algorithm I and Algorithm II.

### 3.3.3 Choice of Tuning Parameters

An important tuning parameter in our model is $\lambda$ in the Potts prior (3.10). To the best of our knowledge, there is no computational efficient way to choose $\lambda$. As a simple approach, we do a grid search to select the optimal value of $\lambda$ based on model selection criteria such as AIC or BIC.
The number of segments $K$ is another tuning parameter. Note that our model includes two “extreme” approaches for multi-task learning: one is to ignore the dependence among the $m$ models and fit them individually, which corresponds to $K = m$; the other is to impose a strong group assumption that the $m$ models share the same set of parameters, which corresponds to $K = 1$. To adaptively select $K$, we again use AIC or BIC to choose the optimal one among a range of values.

### 3.4 Simulation Studies

We conduct simulation studies to evaluate the performance of the proposed Bayesian collaborative model ($BCM$). We will consider both regression and binary outcomes. Note that our algorithm, which is given for probit models, can be used for regression models too by ignoring the steps related to the latent variable $Y$. In all the simulation studies, we assume the design matrix $X$ is the same for all $m$ models. Each entry of $X$ is generated by $N(0, 1)$ and then each column is standardized to have zero mean and unit variance.

We compare $BCM$ with three alternatives: Oracle, Independent and $gLasso$ (Group Lasso). Oracle knows the true relevant variables and estimates the coefficients using least squares for regression outcomes and logistic for binary outcomes. Independent fits each model independently and uses forward stepwise variable selection. $gLasso$ assumes all the models share the same set of relevant variables and estimates the coefficients via the R package grplasso (Meier et al., 2006).

We evaluate the performance of each method based on the prediction mean-squared error (PMSE) or prediction misclassification error (PMSE) on an independent test set and three variable selection error rates: False Positive Rate (FPR), False Negative Rate (FNR) and False Discovery Rate (FDR), where FPR is the percentage of falsely selected irrelevant variables and FNR is the percentage of falsely excluded relevant variables. All the reported quantities are averages over 50 simulations.

In our $BCM$, we need a similarity graph on the $m$ models, which can be constructed using auxiliary information in practice. In simulation studies, we generate the graph structure $G = (V, E)$ this way: if model $i$ and model $i'$ are in the same segment, we set $e_{ii'} = 1$ with probability .95, and 0 otherwise; if they are from different segments, we set $e_{ii'} = 1$ with probability .05, and 0 otherwise. In other words, the prior neighborhood structure is not, but close to be, accurate.
Table 3.1: Comparison of variable selection for BCM, gLasso, and Independent in Scenario I and II for simulation study with regression models.

<table>
<thead>
<tr>
<th>Scenario I</th>
<th>Scenario II</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCM</td>
<td>gLasso</td>
</tr>
<tr>
<td>FPR</td>
<td>0.005</td>
</tr>
<tr>
<td>FNR</td>
<td>0.028</td>
</tr>
<tr>
<td>FDR</td>
<td>0.005</td>
</tr>
</tbody>
</table>

3.4.1 Linear Regression Models

We fit $m = 200$ regression models simultaneously, each of which has $p = 20$ variables and $n = 16$ observations. The 200 models are randomly divided into $K = 3$ segments. The three sparse segment coefficients are given by

\[
\mu_1 = \{-1.5, -1.5, 0.05, 0.05, 0.85, 0.85, 1.0, 1.0, 0, \ldots, 0\},
\]

\[
\mu_2 = \{0.6, 1.0, 0.1, 0.8, 1.2, 1.2, 0, \ldots, 0\},
\]

\[
\mu_3 = \{1.5, 1.5, 0.4, 0, \ldots, 0\}.
\]

Given that the $i$th model is from the $k$th segment, we generate the individual coefficient vector $\beta_i \sim N(\mu_k, \text{diag}(\nu_l)_{l=1}^p)$. We consider two scenarios here.

In Scenario I, “sparsity at the individual level,” we set

\[
\nu_l = 0 \cdot \mathbb{I}[\mu_{kl} = 0] + 0.1 \cdot \mathbb{I}[\mu_{kl} \neq 0].
\]

That is, $\beta_i$ has the same sparsity pattern as the segment coefficients $\mu_k$. We repeat this simulation 50 times, for each simulated data set, evaluate the PMSE on an independent test data (16 test samples from each of the 200 models) for the four methods. We use the PMSE from Independent as the baseline, and express the PMSEs from the other three methods as percentages of the baseline performance. In Figure 3.1, we draw a boxplot for the scaled PMSEs over 50 iterations for each of the four methods. It is not surprising that Oracle returns the smallest prediction error most of the time, since it knows the true set of relevant variables. Our approach BCM is slightly worse than Oracle, but still better than gLasso. Independent is the worst since the coefficient can hardly be accurately estimated due to the $p > n$ setting. Table 3.1 lists the comparison of these approaches for variable selection: BCM is the best in terms of all three criteria.

In the simulation study, we observed that the set of relevant variables returned by gLasso is often
the union of relevant variables for each segment, which is consistent with the assumption imposed by \textit{g}Lasso. For this particular simulation setting, there is a big overlap among the three relevant variable sets, so the performance of \textit{g}Lasso is not too bad. However, if the overlap among relevant variables from different segments is small, we expect \textit{g}Lasso’s performance will deteriorate.

In Scenario II, “sparsity at the segment level,” we just use $\nu_l = 0.1$, that is, generate $\beta_{ij} \sim N(0, \mu_{kj}, 0.1)$. So in terms of $L_0$ norm the coefficient of each individual model is no longer sparse. Since $p > n$, \textit{Oracle} will not be able to apply the least squares method using all $p$ variables. So in this case, \textit{Oracle} uses the relevant variables at the segment level. This is why \textit{Oracle}’s PMSE is worse than our \textit{BCM} as shown in Figure 3.2. \textit{g}Lasso is worse than \textit{Oracle} and \textit{Independent} is the worst. We also summarize the variable selection accuracy for Scenario II in Table 3.1, in which our method \textit{BCM} is the best.

Throughout this simulation study, we use 1000 iterations for burning and 1000 for sampling for our MCMC, with hyper-parameters in (3.11) fixed to be

$$a_0 = 1, \ b_0 = 1, \ s_1 = 2, \ s_2 = 0.5, \ t_1 = 2, \ t_2 = 0.01.$$
Figure 3.2: Compare the PMSEs among BCM, Oracle, gLasso, and Independent in Scenario II for simulation study with regression models. The y-coordinate indicates the percentage of PMSE relative to the PMSE from Independent on the same data set.

We use AIC to select tuning parameters $K$ and $\lambda$ on a grid. For example, for one data set, our BCM suggested to select $K = 3$ and $\lambda = 0.05$.

### 3.4.2 Probit Models

We still consider $m = 200$ model with binary outcomes, which form $K = 3$ segments with the segment coefficients are given by

\[
\begin{align*}
\mu_1 &= \{-2.25, -2.25, 0.075, 0.075, 1.275, 1.275, 1.5, 1.5, 0, \ldots, 0\}; \\
\mu_2 &= \{0.9, 0.9, 1.5, 1.5, 1.5, 1.5, 0, \ldots, 0\}; \\
\mu_3 &= \{2.25, 2.25, 0.6, 0, \ldots, 0\}.
\end{align*}
\]

As in the linear regression case, we generate the individual coefficient $\beta_i$'s under both Scenario I and II.

For each method, we evaluate the prediction misclassification error (PMSE) on an independent
Figure 3.3: Compare the PMSE’s among BCM, Oracle, gLasso, and Independent in Scenario I for simulation study with probit models. The y-coordinate indicates the percentage of PMSE relative to the PMSE from Independent on the same data set.

Table 3.2: Comparison of variable selection for BCM, gLasso, and Independent in Scenario I and II for simulation study with probit models.

<table>
<thead>
<tr>
<th></th>
<th>Scenario I</th>
<th></th>
<th>Scenario II</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BCM</td>
<td>gLasso</td>
<td>Independent</td>
<td>BCM</td>
</tr>
<tr>
<td>FPR</td>
<td>0.069</td>
<td>0.139</td>
<td>0.057</td>
<td>0.017</td>
</tr>
<tr>
<td>FNR</td>
<td>0.029</td>
<td>0.046</td>
<td>0.128</td>
<td>0.031</td>
</tr>
<tr>
<td>FDR</td>
<td>0.072</td>
<td>0.076</td>
<td>0.260</td>
<td>0.045</td>
</tr>
</tbody>
</table>

test set. Figure 3.3 and 3.4 present the boxplot of scaled PMCE’s (relative to the PMCE from Independent) for Scenario I and II. We also evaluate various variable selection error rates for each method except Oracle, which is summarized in Table 3.4.2. It is easy to see that the proposed method BCM is an overall winner. Note that in Scenario I, different from regression models, BCM performs better than Oracle, because with such a small sample size $n = 16$, some data sets are well separated, which makes the logistic fitting not converge.

For BCM, we use 5000 iterations for burning and 5000 for sampling for our MCMC algorithm. All the hyper-parameters in (3.11) are fixed to be

$$a_0 = 10, \ b_0 = 40, \ s_1 = 2, \ s_2 = 0.5, \ t_1 = 2, \ t_2 = 0.01.$$
3.5 A Case Study Example

The proposed model is used to analyze the data from a marketing experiment we have conducted at University of Illinois at Urbana-Champaign (UIUC), where the goal of the experiment is to explore important web features for online shopping.

3.5.1 Study Design

A choice-based conjoint study was conducted on 234 undergraduate students at UIUC. In the study, all subjects sequentially viewed 16 slides; each slide displayed two websites both offering the 22-inch Samsung LCD monitor; then the subject indicated which website he/she was willing to make the purchase. The pair of websites has different features, see Figure 3.5. We consider 15 web features, such as whether it provides product review from other consumers, security statement, free shipping option, rebate etc. Among the features 6 of them are three-level features and 9 are two-level ones. We coded them into 21 dummy variables. Then for simplicity, we renamed the variables from three-level features in the format of “feature 1” (level 3 vs others) and “feature 2” (level 2 vs others). The details of the features are given in Table 3.4. We use orthoplan in SPSS to provide 32 combinations
Figure 3.5: Demonstration of one pair of websites offering the 22-inch Samsung LCD monitor. Totally 15 web features are considered which include 6 three-level features and 9 two-level features.

of the 15 features, and randomly divided them into 16 pairs, which are then used to generate websites in the experiment.

For all the participants, we collected additional survey responses of behavioral characteristics associated with Internet behaviors, such as number of hours using Internet per day, the level of prior knowledge about online shopping, perceived risks of purchasing products through websites etc. Let \( w_i \) denote the survey responses from the \( i \)th participant. We construct a similarity graph on the 234 participants based on \( w_i \)'s as follows: first calculate the \( L_2 \) distance between a pair of participants \( d_{ii'} = \|w_i - w_{i'}\|^2 \), then add an edge between \( i \) and \( i' \) in the graph if \( d_{ii'} > \delta \) where \( \delta \) is a chosen threshold. By looking at the histogram of the pairwise distances, we set \( \delta = 5 \) so the graph is neither too dense nor too sparse. The similarity weight \( e_{ii'} \) on each edge is set to be a constant 1.

3.5.2 Results

We applied Algorithm I on this data set, which corresponds to using prior (3.8) on \( \beta_i \)'s. That is, the individual coefficient \( \beta_i \) has the same sparsity pattern as the segment coefficient. The output on AIC and BIC for the selection of tuning parameters \( K \) and \( \lambda \) are displayed in Table 3.3. When \( \lambda = .05 \), the prior influence on the segment structure is too strong and all the results returned
Table 3.3: AIC (BIC) scores ($\times 10^4$) for the online shopping data with prior (3.8), where “NA” indicates the outcome has size 0 segment, so it is not consistent with the pre-specified $K$ value.

$$\lambda = 0 \quad \lambda = .005 \quad \lambda = .01 \quad \lambda = .03 \quad \lambda = .05$$

<table>
<thead>
<tr>
<th>$K$ = 1</th>
<th>$K$ = 2</th>
<th>$K$ = 3</th>
<th>$K$ = 4</th>
<th>$K$ = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$7.814$ ($10.719$)</td>
<td>$7.813$ ($10.718$)</td>
<td>$7.814$ ($10.719$)</td>
<td>$7.814$ ($10.719$)</td>
<td>$7.813$ ($10.718$)</td>
</tr>
<tr>
<td>$7.472$ ($10.249$)</td>
<td>$7.800$ ($10.705$)</td>
<td>$7.055$ ($9.670$)</td>
<td>$6.969$ ($9.546$)</td>
<td>NA (NA)</td>
</tr>
<tr>
<td>$7.203$ ($9.876$)</td>
<td>$5.368$ ($7.339$)</td>
<td>$6.008$ ($8.226$)</td>
<td>$7.164$ ($9.829$)</td>
<td>NA (NA)</td>
</tr>
<tr>
<td>$5.588$ ($7.641$)</td>
<td>$7.117$ ($9.764$)</td>
<td>$6.143$ ($8.412$)</td>
<td>$5.567$ ($7.614$)</td>
<td>NA (NA)</td>
</tr>
</tbody>
</table>

Table 3.4: Estimated Segment Coefficients using Algorithm I

<table>
<thead>
<tr>
<th>Features</th>
<th>Segment 1</th>
<th>Segment 2</th>
<th>Segment 3</th>
<th>Segment 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  Free shipping</td>
<td>0</td>
<td>0.12</td>
<td>0.38</td>
<td>0.72</td>
</tr>
<tr>
<td>2  Coupon 1</td>
<td>0.26</td>
<td>0.39</td>
<td>0.27</td>
<td>0.51</td>
</tr>
<tr>
<td>3  Coupon 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4  Price Discount 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5  Price Discount 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.23</td>
</tr>
<tr>
<td>6  Price Comparison</td>
<td>0.26</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7  Tracking Order</td>
<td>0.52</td>
<td>0.36</td>
<td>0.18</td>
<td>0.14</td>
</tr>
<tr>
<td>8  Product Picture</td>
<td>0</td>
<td>0</td>
<td>0.36</td>
<td>0.22</td>
</tr>
<tr>
<td>9  Description</td>
<td>0</td>
<td>0</td>
<td>0.19</td>
<td>0</td>
</tr>
<tr>
<td>10 Rebate 1</td>
<td>-0.16</td>
<td>-0.16</td>
<td>-0.20</td>
<td>0</td>
</tr>
<tr>
<td>11 Rebate 2</td>
<td>0</td>
<td>0</td>
<td>0.18</td>
<td>0.17</td>
</tr>
<tr>
<td>12 Combo</td>
<td>-0.20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13 Web Exclusivity</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14 Web Security</td>
<td>0</td>
<td>0.23</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15 Return/Cancellation 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16 Return/Cancellation 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17 Warranty 1</td>
<td>0.40</td>
<td>0.36</td>
<td>0.40</td>
<td>0</td>
</tr>
<tr>
<td>18 Warranty 2</td>
<td>0</td>
<td>0</td>
<td>0.22</td>
<td>0</td>
</tr>
<tr>
<td>19 Delivery Time</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20 Rating From Others 1</td>
<td>0.66</td>
<td>0.43</td>
<td>0.32</td>
<td>0</td>
</tr>
<tr>
<td>21 Rating From Others 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-0.43</td>
</tr>
</tbody>
</table>

just one segment (i.e., other segments have size zero). Since the results are not consistent with the pre-specified $K$ values, we label them as “NA”. Both AIC and BIC suggested to select $K = 4$ and $\lambda = 0.005$. The estimation of the segment coefficient $\mu_k$’s are given in Table 3.4. All four segments select features like “Coupon” and “Tracking Order”. Majority select features like “Free shipping”, “Rebate”, “Warranty”, and “Rating from others”. The segment sizes are 20, 14, 163, and 38. The 3rd segment dominated the others, probably due to the homogeneity of the participants who are college students. We can also see that Segment 1, 2 and 3 are similar, but Segment 4 is different from them. This agrees with our later analysis using Algorithm II, which selects $K = 2$ segments.

We applied Algorithm II on this data set, which corresponds to using prior (3.9) on $\beta_i$’s. That is, the individual coefficient $\beta_i$ is no longer sparse but it has the flexibility of modeling small departures from the segment coefficient. As in Table 3.5, the minimum of AIC is achieved by $K = 2$ and $\lambda = 0.005$, which is quite close to the minimum BIC score. So we set $K = 2$ and $\lambda = 0.005$. 

43
Table 3.5: AIC (BIC) scores (×10^4) for the online shopping data with prior (3.9), where “NA” indicates the outcome has size 0 segment, so it is not consistent with the pre-specified K value.

<table>
<thead>
<tr>
<th>K</th>
<th>λ = 0</th>
<th>λ = .005</th>
<th>λ = .01</th>
<th>λ = .03</th>
<th>λ = 0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.616</td>
<td>1.604</td>
<td>1.560</td>
<td>1.551</td>
<td>1.554</td>
</tr>
<tr>
<td></td>
<td>(1.816)</td>
<td>(1.804)</td>
<td>(1.741)</td>
<td>(1.733)</td>
<td>(1.735)</td>
</tr>
<tr>
<td>2</td>
<td>1.583</td>
<td>1.524</td>
<td>1.678</td>
<td>1.533</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>(1.728)</td>
<td>(1.653)</td>
<td>(1.852)</td>
<td>(1.684)</td>
<td>(NA)</td>
</tr>
<tr>
<td>3</td>
<td>1.657</td>
<td>1.708</td>
<td>1.540</td>
<td>1.649</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>(1.784)</td>
<td>(1.863)</td>
<td>(1.668)</td>
<td>(1.814)</td>
<td>(NA)</td>
</tr>
<tr>
<td>4</td>
<td>1.635</td>
<td>1.706</td>
<td>1.578</td>
<td>1.702</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>(1.760)</td>
<td>(1.836)</td>
<td>(1.682)</td>
<td>(1.823)</td>
<td>(NA)</td>
</tr>
<tr>
<td>5</td>
<td>1.551</td>
<td>1.534</td>
<td>1.731</td>
<td>1.563</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>(1.660)</td>
<td>(1.651)</td>
<td>(1.876)</td>
<td>(1.676)</td>
<td>(NA)</td>
</tr>
</tbody>
</table>

Table 3.6: Estimated Segment Coefficients using Algorithm II

<table>
<thead>
<tr>
<th>Features</th>
<th>Segment 1</th>
<th>Segment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Free shipping</td>
<td>0</td>
<td>0.60</td>
</tr>
<tr>
<td>2. Coupon 1</td>
<td>0.19</td>
<td>0.35</td>
</tr>
<tr>
<td>3. Coupon 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4. Price Discount 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5. Price Discount 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6. Price Comparison</td>
<td>0.20</td>
<td>0</td>
</tr>
<tr>
<td>7. Tracking Order</td>
<td>0.32</td>
<td>0</td>
</tr>
<tr>
<td>8. Product Picture</td>
<td>0</td>
<td>0.51</td>
</tr>
<tr>
<td>9. Description</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10. Rebate 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11. Rebate 2</td>
<td>0.29</td>
<td>0</td>
</tr>
<tr>
<td>12. Combo</td>
<td>-0.15</td>
<td>0</td>
</tr>
<tr>
<td>13. Web Exclusivity</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14. Web Security</td>
<td>0.32</td>
<td>0</td>
</tr>
<tr>
<td>15. Return/Cancellation 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16. Return/Cancellation 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17. Warranty 1</td>
<td>0.64</td>
<td>0.38</td>
</tr>
<tr>
<td>18. Warranty 2</td>
<td>0.27</td>
<td>0.35</td>
</tr>
<tr>
<td>19. Delivery Time</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20. Rating From Others 1</td>
<td>0.33</td>
<td>0</td>
</tr>
<tr>
<td>21. Rating From Others 2</td>
<td>0</td>
<td>-0.30</td>
</tr>
</tbody>
</table>

in our later analysis. The estimation of the segment coefficient $\mu_k$’s are given in Table 3.6. The two segments are composed of 132 and 102 participants respectively. Participants in Segment 1 value web features such as “Price Comparison”, “Tracking Order”, “Combo”, and “Rebate”. On the other hand, Segment 2 likes “Freed Shipping” and “Product Picture”. Features selected by both segments are “Coupon”, “Warranty” and “Rating From Others”. Web marketing managers for targeting different segments can maximize effectiveness of Web space usage by stressing the important features which are found by our model.
Chapter 4

Multiple Partition Process

Identifying the subtypes of certain cancer, like breast cancer, is critical to the treatment. Due to the complexity of the cancer, the patients can be clustered into different ways. In other words, the objects could have multiple clustering structures that is different from the traditional clustering methods. The traditional clustering methods assume that objects in all $p$ features share the same clustering structure. The mosaic clustering relaxes the all-feature-in assumption of the traditional clustering and allows that different features could belong to different clustering structures. Unfortunately, this all-object-in assumption in mosaic clustering is still strong in some application, because it can results in fragments in some features which is not desired in the cancer patients clustering. In this chapter, we propose a novel model, Multiple Partition Process abbreviated as MPP, which relaxes both the all-feature-in and all-object-in assumptions. The MPP assumes that different cells in the data matrix could belong to different clustering structures. In the multiple-structure clustering, how to choose the cluster numbers of the structures is challenging. In MPP, we assign different DP priors for different structures. Eventually the MPP is evaluated on simulations and a real data set of breast cancer.

4.1 Statistical Clustering

Clustering methods are widely used in many fields, such as biology, medicine, engineering and marketing. Consider a data matrix $X_{n \times p}$, where we have $n$ objects with each being measured by $p$ features. In the traditional approaches, various clustering algorithms are applied on the $p$-dimensional vectors, such as K-means and hierarchical clustering. The clustering structure, or partition, over objects is consistent in each of the $p$ features. For example, the entry $(i,j)$ in $X$ denotes the expression level for the $i$-th sample and $j$-th gene in microarray analysis. The traditional clustering analysis partition the $n$ objects into some clusters, say three clusters, across the $p$ features, see the left panel of Figure 4.1. Nowadays many applications violate the assumption of partition
sharing among all $p$ features and the data may exhibit different partitions when being associated with different sets of features. In the microarray analysis example, the data structure might exhibit like this: some features divide the objects into two clusters, some divide the objects into three clusters, and there is no clustering structure for the remaining features, or in other words, the $n$ objects form just one cluster with those features, see the middle panel of Figure 4.1.

A natural extension of the traditional all-features-in approaches is the mosaic type clustering. The mosaic type clustering in this paper is defined as multiple clustering structures, or partitions, in different non-overlap sets of features. Biclustering (Hartigan, 1972) is such a mosaic type clustering technique which simultaneously clusters the rows and columns of a data matrix. The biclustering always generates biclusters each of which is a subset of rows which exhibit similar behavior across a subset of columns, or vice versa. A variety of biclustering algorithms can be found in the literatures.

In Hartigan (1972), the author proposes a method so called direct clustering which begins with the entire data as a single block and then iteratively finds the row and column split of every block into two pieces. Cheng and Church (2000) constructs one bicluster at a time using some criteria. Once a bicluster is created, its entries are replaced by random numbers, and the procedure is repeated iteratively. A nonparametric Bayesian biclustering is discussed in Meeds et al. (2007). Two independent Dirichlet Process (DP) priors are introduced over row and column clusters separately.

Although the mosaic type clustering relaxes the all-feature-in assumption and can retrieve multiple partitions simultaneously, it might cause undesired fragments in some of the clustering structures. For instance, the data displayed on the right in Figure 4.1 presents some new challenges to the mosaic clustering. There are three partitions in the data, same as the ones presented in the middle of Figure 4.1. However, each column of the data matrix involves at least one of the three partitions, while in mosaic clustering, each column is assumed to follow one and only one partition. So, the unexpected clustering result is caused by the improper assumption that each feature can only join in one and only one clustering structure. In order to avoid this fragmentation issue, we model the multiple partitions in a more flexible manner: each of the $p$ features can belong to multiple rather than single partition. More precisely, in each feature, different objects might belong to different partitions. This flexibility makes each clustering structure consider the overall features and leads to a novel clustering method – MPP.

The remainder of the paper is organized as follows. In Section 4.2 we propose the model and priors of MPP and discuss the identifiability issue. Section 4.3 develops the MCMC-based sampling scheme. Simulation studies are presented in Section 4.4. The breast cancer application is discussed
Figure 4.1: Different clustering scenarios for the matrix data. From left to right panel, the plots are for: the traditional consistent clustering over all $p$ features; the mosaic clustering with 3 partitions and the clustering scenario with mixed partitions over each feature. Top panel: the original data with three scenarios from left to right; Middle panel: rows of the data matrices have been re-arranged for a better visual effect for the first partition. Bottom panel: The re-arranged matrices for the second partition if applicable. For better visualization of clustering effect of the third scenario, only related entries are plotted as seen the most right plots in the middle and bottom panels.
in Section 4.5.

Recently, Dunson (2009) proposed Local Partition Process (LPP) that is constructed through a locally-weighted mixture of global and local clustering structures. The local clustering in LPP is a partition over single feature while the global clustering is overall partition. In LPP, each object is associated with one global cluster and $p$ local clusters. The information borrowing in LPP can be induced across subjects through both global and local clustering. The LPP can be considered a special case of MPP with $p + 1$ partitions each of which is associated with one single feature. The subtle differences between MPP and LPP will be discussed in section 4.2.

4.2 Multiple Partitions Process

4.2.1 Model and Priors

Assume the matrix-type data $X_{n \times p}$ contains totally $s$ partitions. Each entry of $X$ is associated with one of all $s$ partitions, so a partition indicator matrix $S_{n \times p}$ is introduced where $S_{ij} = l$, $l \in 1 : s$, implies that the data point of the $i$-th object in $j$-th feature is associated with the $l$-th partition. Therefore, each object possesses multiple cluster memberships, while each feature is a mixture of partitions. If the entry $(i, j)$ is belong to the $l$-th partition, i.e., $S_{ij} = l$, then we have $X_{ij} \sim N(\Theta_{ij}^{(l)}, \sigma^2)$, where $\Theta_{ij}^{(l)}$ is the normal mean for entry $(i, j)$ in $l$-th partition. In MPP, in each partition, we favor that the mean parameters over all objects form small number of clusters. This can be accomplished through specifying DP priors (Ferguson, 1973, 1974) for different $s$ partitions. For $l = 1 : s$,

$$\Theta_{ij}^{(l)} \sim G_l,$$

$$G_l \sim DP(\alpha_l, G_0),$$

where $\alpha_l$ is a scaling parameter and $G_0$ is a base prior which is $N(\mu_0, \sigma^2_0)$ in our model. The DP priors make the mean parameters in one partition form clusters each of which possesses a unique value. Assume that the cluster size of the $l$-th partition is $c_l$. For the clustering purpose, we re-denote the notation $\Theta_{kj}^{(l)}$ as the $k$-th cluster mean parameter in $j$-th feature in the $l$-th the partition, where $k \in 1 : c_l$. Meanwhile, a cluster indicator matrix $C_{s \times n}$ is introduced for $s$ partitions, where the $l$-th row of $C$, $C_l = (C_{l1}, \ldots, C_{ln})$, is the membership set for the $l$-th partition and $C_{li} \in 1 : c_l$. We note that the first partition is constrained to have just one single cluster, or in other words, the
first partition has no clustering structure. So \( C_{1i} = 1 \), for \( i = 1 : n \). Given \( S_{ij} = l \) and \( C_{li} = k \), we have \( X_{ij} \sim N(\Theta_{kj}^{(l)}, \sigma^2) \).

The summary of the notations is:

- \( X_{n \times p} \): data matrix with \( n \) objects (rows) and \( p \) features (columns);
- \( S_{n \times p} \): partition indicator matrix, where \( S_{ij} \in 1 : s \);
- \( C_{s \times n} \): cluster indicator matrix over \( s \) partitions, where \( C_{li} \in 1 : c_l \);
- \( \Theta^{(l)}_{c_l \times p} \): normal means for the \( l \)-th partition over all \( p \) features.

The model for MPP is:

\[
p(X|S, \theta, \sigma^2, C) = \prod_{i=1}^{n} \prod_{j=1}^{p} N(X_{ij}; \Theta_{C_{li}}^{(q)}, \sigma^2),
\]

where \( q = S_{ij} \) and \( \theta \) is short term for all parameters of normal means, i.e., \( \theta = \Theta^{(\cdot)} \).

In additional to the DP priors above for mean parameters, we also have the priors:

- \( \alpha_l \sim \text{Gamma}(a_{\alpha}, b_{\alpha}), \quad \text{for} \ l = 2 : s; \)
- \( \sigma^2 \sim \text{IG}(\gamma_1, \gamma_2); \)
- \( S_{ij} \sim \text{Multi}(B_{1j}, \ldots, B_{sj}). \)

The non-identifiability problems, however, could occur in the following a couple of scenarios:

1. Suppose the \( j \)-th feature does not contain the first partition structure and it is a mixture of the remaining \( s-1 \) partitions, i.e., \( S_{ij} \neq 1 \), for all \( i \in 1 : n \). The model remains the same if we re-label any cluster in any partition over \( j \)-th feature as the first partition structure.

2. Suppose the \( j \)-th feature contains a few first partitions but there is no \( l \)-th \( (l > 1) \) partition. For \( i \in f_1 \), \( S_{ij} = 1 \). Suppose \( C_{lt_1} = \cdots = C_{lt_m} \), where \( t_1, \cdots, t_m \in f_1 \). If we relabel entries \((t_1, j), \cdots, (t_m, j)\) from the first structure to the \( l \)-th partition, the model remains the same.

In order to overcome these non-identifiability issues above, the prior of \( S \) is updated to be: for the \( j \)-th column,

\[
S_{j}|\lambda \sim \text{Polya’s Urn}(d_1, \cdots, d_s, \lambda),
\]

where \( \lambda \) is the initial number of balls and \( d_l \) is the proportion of balls with color \( l \) and \( d_1 > d_l \), for
4.3 Posterior Computation

4.3.1 Gibbs Sampler

We propose to use an adaptation of the slice sampling approach (Walker, 2007; Dunson, 2009) to implement posterior computation. In slice sampling, the DP priors are represented via the stick-breaking construction: for \( l = 2 : s, \ i = 1 : n, \)

\[
\begin{align*}
\Theta_{kj}^{(l)} &\sim N(\mu_0, \sigma^2); \\
C_{li} &\propto \prod_{h=1}^{\infty} \pi_h^{(l)} \delta_h, \quad \pi_h^{(l)} = \pi_\star^{(l)} \prod_{t<h} (1 - \pi_t^{(l)}), \quad \pi_t^{(l)} \sim \text{Beta}(1, \alpha_t).
\end{align*}
\]

Until now, the unknown parameters include \( S, C, \theta, \pi^\star, \alpha, \) and \( \sigma^2, \) where \( \pi^\star = \{\pi_{1:\infty}^{(1)}, \cdots, \pi_{1:\infty}^{(s)}\} \) and \( \alpha = \{\alpha_1, \cdots, \alpha_s\}. \) In addition, a latent variable \( u = \{u_1, \cdots, u_s\} \) is adopted, where \( u_l = \{u_{l1}, \cdots, u_{ln}\}. \) Then the complete data joint likelihood of \( X, u \) and \( S \) is

\[
\left( \frac{1}{\sqrt{2\pi\sigma^2}} \right)^n \prod_{i=1}^n \left\{ \prod_{j=1}^p \exp \left( -\frac{(X_{ij} - \Theta_{C_{si}j}^{(q)})^2}{2\sigma^2} \right) \prod_{l=2}^s I[u_{li} < \pi_{C_{li}}^{(l)}] \right\},
\]

where \( q = S_{ij} \) and the \( u_{li} \) are constrained to fall in \((0,1).\)

A collapsed Gibbs sampler is employed. In the collapsed Gibbs sampler, parameters \( \theta \) is integrated when update \( S \) which make the sampling more efficient. There are two main steps in the collapsed Gibbs sampler:

1. Update \( S|[-\theta] \cdots, \) where \([-\theta] \) means \( \theta \) is not included and \( (\cdots) \) denote the remaining parameters.

2. Update parameters \( u, C, \theta, \pi^\star, \alpha, \sigma^2. \)

The algorithm of Step 1 is discussed in section 4.3.2. The step 2 is the similar to the corresponding steps in Dunson (2009) and the details are given in Appendix D.
4.3.2 Updating S

In order to make the MCMC more efficient, a collapsed Gibbs sampler is adopted which integrates parameters \( \theta \) when update \( S \). For simplicity, we suppress the subscript \( j \) in all the notations in section 4.3.2 and define \( S_i \) as the structure indicator for \( i \)-th subject and \( S_{[-i]} \) as all the subjects except \( i \)-th subject.

\[
P(S_i = l | X, S_{[-i]}, C, \sigma^2) \propto P(X | S_{[-i]}, S_i = l, C, \sigma^2) P(S_i = l | S_{[-i]})
\]

Note that the sampling of \( S_i \) does not depend on \( \theta \), i.e., we integrate over \( \theta \). The partition matrix \( C \) and structure indicator \( S \) divide the objects (except the \( i \)-th object) into several groups, say \( m \) groups. All the objects in each group share the same normal mean parameter which is integrated out. When we consider sampling \( P(S_i = l | X, S_{[-i]}, C, \sigma^2) \), the \( i \)-th object joins the following group:

\[
T_{il} = \{ t : t \in 1:n, t \neq i, S_t = l, C_{lt} = C_{li} \},
\]

which is the set of objects, excluding \( i \), in cluster \( C_{li} \) of structure \( l \). The term \( P(X | S_{[-i]}, S_i = l, C, \sigma^2) \) can be written as multiplication of \( m \) factors most of which remain the same when \( l \) varies. So, we just need to calculate these factors varies over different \( l \). Thus, we have

\[
P(X | S_{[-i]}, S_i = l, C, \sigma^2) \propto \frac{P(X_{T_{il} \cup \{i\}} | S_{[-i]}, S_i = l, C, \sigma^2)}{P(X | S_{[-i]}, C, \sigma^2)}.
\]

Incorporating with the fact that the integrated density for objects in \( X_1, \cdots, X_d \sim N(\Theta, \sigma^2) \) with prior \( \Theta \sim N(\mu_0, \sigma_0^2) \) is given by

\[
\left( \frac{1}{\sqrt{2\pi\sigma^2}} \right)^d \frac{1}{\sqrt{\sigma_0^2}} \sqrt{d/\sigma^2 + 1/\sigma_0^2} \exp \left\{ \frac{(\sum_{t \in 1:d} X_t^2/\sigma^2 + \mu_0^2/\sigma_0^2)}{2(d/\sigma^2 + 1/\sigma_0^2)} - \sum_{t \in 1:d} X_t^2/\sigma^2 + \mu_0^2/\sigma_0^2 \right\},
\]

we can obtain the conditional distribution. Besides, the case with \( T_{il} \) empty should be considered. The detail of the derivation and exact conditional distribution is given in Appendix E.

The algorithm 2 in Neal (2000) suggests an additional step in the MCMC of DP: re-sample the unique values for all the existing clusters. This re-sampling step is able to make the whole MCMC more efficient. In our MPP, accordingly, a re-sampling step of \( S \) is added after the updating of \( S \) as described above.

Define \( M_l \) \( (l \in 1:s) \) the objects that are labeled in the \( l \)-th partition and \( \bar{S}_{M_l} \) is the unique
structure indictor value of the set \( M_l \). The re-sampling step in MPP is to update \( \tilde{S}_{M_l} \) which might take value from 1 to \( s \). Similar to the updating of \( S \) above, the re-sampling of \( S \) will be conditional on all other parameters but \( \theta \). In other words, the mean parameter \( \theta \) will be integrated over again:

\[
P(\tilde{S}_{M_l} = h|X, C, \sigma^2, \tilde{S}_{[-M_l]}) \propto P(X|S_{M_l} = h, S_{[-M_l]}, C, \sigma^2)P(\tilde{S}_{M_l} = h).
\]

The integration of \( \theta \) should be paid attention. When calculate \( P(X_{M_l}|S_{M_l} = h, S_{[-M_l]}, C, \sigma^2) \), the objects in set \( M_l \) take value \( h \in 1 : s \), but the integration of mean parameter \( \Theta_h \) should also involve all other objects belonging to the \( l \)-th structure. The details of the re-sampling of \( S \) are described in Appendix F.

4.3.3 Posterior Inference

Among all the parameters, the partition indicator \( S \) and the cluster indicator \( C \) are of interest. For \( S \), we use \( \hat{S} \) to denote the estimate of \( S \) where \( \hat{S}_{ij} \), the cell \( (i, j) \) in \( \hat{S} \), takes the posterior mode. The inference of \( C \) is more complicated and two ways in our paper are provided to make inference on it:

1. Conditional on all the samples with cluster size \( \hat{c}_l \), \( \hat{C}_{ij} \) takes the value of the sample mode;

2. \( \hat{C} \) is the sample of which the association matrix is closest to the averaged association matrix (Dahl and Newton, 2007).

4.3.4 Choice of \( s \)

We use average Pseudomarginal Likelihood, or ALPML (Geisser and Eddy, 1979; Gelfand and Mallick, 1995), to choose the parameter \( s \). the ALMPL is defined based on Conditional Predictive Ordinate (CPO) which is

\[
CPO_i = f(y_i|D^{-i}) = \int f(y_i|x_i, \theta) \pi(\theta|D^{-i})d\theta,
\]

where \( (x_i, y_i) \) is the data for \( i \)-th subject, \( D^{-i} \) denotes the data with \( i \)-th subject deleted. Observe that

\[
CPO_i = f(y_i|D^{-i}) = \int \frac{1}{f(y_i|x_i)} \pi(\theta|D) d\theta.
\]
Figure 4.2: The data has 120 subjects and 20 features. The left panel is the data in the generated order (by row); the middle one is the data shuffled according to the memberships of structure 2; the right panel is shuffled according to the memberships of structure 3.
4.4 Simulation Studies

To illustrate the approach, a simulation example is considered. The data set $X$ is generated this way: $X$ contains $n = 120$ subjects (rows) and $p = 20$ features (columns), see the left panel of Figure 4.2. There are $s = 3$ partition structures. Structure indicator matrix $S$ is constructed as follows. The columns 1 to 5 are all “2” (structure 2); The columns 9 to 12 are “3” (structure 3); The columns 6-8 are mixtures of “2” and “3” (50%-50%) and the remaining columns are “1” (structure 1). The structure 1 has just one cluster, i.e. $c_1 = 1$, with $\Theta_{1j}^{(1)} = \text{Unif}(-0.8, 0.8)$, $j = 1 : p$. The structure 2
has $c_2 = 3$ clusters with $\Theta^{(2)}$

$$
\Theta_1^{(2)} = \{0.2, 0.8, 0.1, 0.8, 0.4, -0.4, -0.5, 0.9, \ldots\}
$$

$$
\Theta_2^{(2)} = \{-0.1, 0.4, 0.9, -0.4, 0.5, 0.1, -0.4, \ldots\}
$$

$$
\Theta_3^{(2)} = \{-0.9, -0.1, -0.4, -0.6, -0.7, 0, 0.6, -0.1, \ldots\}
$$

The middle panel of Figure 4.2 shuffles the data to make the first 5 columns ordered by the corresponding memberships. The structure 3 possesses $c_3 = 2$ clusters with $\Theta^{(3)}$

$$
\Theta_1^{(3)} = \{\cdots, 0.2, -0.2, 0.4, 0.9, -0.4, 0.4, 0.6, \cdots\}
$$

$$
\Theta_2^{(3)} = \{\cdots, -0.9, 0.8, -0.7, 0, 0.5, -0.2, -0.3, \cdots\}
$$

In addition, $\sigma^2 = 0.2$.

The hyper-parameters are: $\mu_0 = 0$, $\sigma_0^2 = 0.1$, $a_\alpha = 0.3$, $b_\alpha = 0.1$, $\gamma_1 = 1$, $\gamma_2 = 0.1$, $d = (\frac{1}{s+3}, \frac{1}{s}, \ldots, \frac{1}{s})$ and $\lambda = 2$. The initial values of parameters are: $\Theta_{l}^{(1)} \sim N(\mu_0, \sigma_0^2)$, $\alpha_l = 1$, $l = 1 : 3$, $C. = 1$, $S. = 1$ and $\sigma^2 = 0.001$.

We use 2,000 iterations for burning-up and 1,000 as samples for inference. We run $s = 1, 2, 3, 4$
and 5 sequentially. The values of ALMPL are plotted in Figure 4.3. Eventually we set $s = 3$, because 1) after $s \geq 3$, the ALPML values are very close; and 2) the $S$’s for cases $s = 4$ and 5 just contain 3 different structures.

Given $s = 3$, the estimation of $S$, $C$ and others are listed as follows.

- Accuracy of $S$: Table 4.1 shows the structure labeling results.

<table>
<thead>
<tr>
<th>$(S)$ Structure 1</th>
<th>Structure 2</th>
<th>Structure 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(true $S$) Structure 1</td>
<td>960</td>
<td>2</td>
</tr>
<tr>
<td>(true $S$) Structure 2</td>
<td>0</td>
<td>700</td>
</tr>
<tr>
<td>Structure 3</td>
<td>0</td>
<td>78</td>
</tr>
</tbody>
</table>

Table 4.2: $s = 3$: Clustering for Structure 2 (comparing with true $C_2$.)

<table>
<thead>
<tr>
<th>(true $C_2$) cluster 1</th>
<th>cluster 2</th>
<th>cluster 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster 1</td>
<td>41</td>
<td>0</td>
</tr>
<tr>
<td>cluster 2</td>
<td>0</td>
<td>33</td>
</tr>
<tr>
<td>cluster 3</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4.3: $s = 3$: Clustering for Structure 3 (comparing with true $C_3$.)

<table>
<thead>
<tr>
<th>(true $C_3$) cluster 1</th>
<th>cluster 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(true $C_3$) cluster 1</td>
<td>69</td>
</tr>
<tr>
<td>(true $C_3$) cluster 2</td>
<td>3</td>
</tr>
</tbody>
</table>

- Accuracy of $C$: The histogram of cluster sizes of structure 2 and 3, i.e, $c_2$ and $c_3$ are shown in Figure 4.4. So, structure 2 has 3 clusters, i.e. $\hat{c}_2 = 3$, and structure 3 has 2 clusters, i.e. $\hat{c}_3 = 2$, The estimation of partition $\hat{C}$ is given in table 4.2 and 4.3.

**4.5 A Case Study Example**

The breast cancer data set contains 517 patients with 9 features (517 rows and 9 columns). The data set is shown in Figure 4.5 (left panel). We run 3000 MCMC iterations (2000 burning up) for both cases for $s = 1, 2, 3, 4$ and 5. Figure 4.6 represents the selection of $s$ by ALMPL. Because $s = 3$ reaches the maximum, we have $s = 3$.

When $s = 3$, there are three clustering structures. For different features, the proportions of different structures in $S$ are listed as Table 4.4.
Figure 4.5: The breast cancer data set contains 517 patients and 9 features (left). The left panel is the data with objects ordered by original order; the middle panel is ordered by the cluster membership in structure 2; the right panel is ordered by the cluster membership in structure 3.

Figure 4.6: Select $s$ by ALPML for the breast cancer data. We have $\hat{s} = 3$, because ALPML reaches the maximum when $s = 3$. 

Breast Cancer Data: Selection of $s$
The first structure always has one cluster, i.e. $\hat{c}_1 = 1$. For the cluster sizes of structures 2 and 3, we have $\hat{c}_2 = 2$ and $\hat{c}_3 = 4$. In a summary, the 2 clusters in structure 2 have 365 and 152 objects, the 4 clusters in structure 3 have 154, 308, 5 and 50 objects respectively.
Chapter 5

Multiple Partitions for Latent Features

The MPP is inefficient when the feature number \( p \) is large. For instance, the aCGH data matrix \( X_{n \times p} \) involves \( n \) cancer patients (objects) and each patient has millions of measurements (\( p \) is huge) along all chromosomes. A natural way to overcome the large-\( p \) problem is to perform dimension reduction before applying MPP. In this chapter, a two-stage clustering approach is proposed: 1) perform dimension reduction; 2) apply MPP to the reduced “new” data matrix \( Z \) with relatively small dimension. Due to the properties of the aCGH data, we employ Indian Buffet Process (IBP) to reduce the dimension. This two-stage approach, however, is not optimal, and our goal is to propose an integrated model that could simultaneously exact the latent features and perform multiple-structure clustering over objects.

5.1 Multiple Clustering for ACGH Data

ACGH is a technique to detect genomic copy number variations at a higher resolution level than chromosome-based comparative genomic hybridization (CGH) (Shinawi and Cheung, 2008). The resulting data consists of log fluorescence ratios as a function of the genomic DNA location and provides a cytogenetic representation of the relative DNA copy number variation. In Pinkel and Albertson (2005), the authors show that genomic abnormalities in the number of DNA copies in a cell are associated with cancer development and progression. So, we may expect that shared genomic regions with common DNA copy alterations in a particular subtype of cancer patients may contain genes that are crucial in characterizing this population (Aladandayuthapani et al., 2010). Thus, the detection of these shared regions for some specific cancer and the clustering of patients play important roles in the treatment of cancer. Figure 5.1 shows the aCGH data consisting of six lung cancer patients from three different lung cancer subtypes: “TN”, “HER2+”, and “ER2+”.

In the literatures, several aCGH data clustering algorithms have been developed (Wieringen et al., 2008; Sohrab and et al., 2009; Roquain and van de Wiel, 2010). These algorithms, however,
Figure 5.1: The aCGH data set from six different lung cancer patients from three different lung cancer subtypes: “TN,” “HER2+” and “ER2+.”
still have some limitations. Firstly, the potential multiple clustering structures are not considered. In clinical practice, multiple clustering structures could help patients get better treatment. Secondly, these algorithms usually require the fixed cluster number. This fixed cluster number, however, is usually difficult to obtain. Besides, making the cluster size unfixed is desirable because it can help us to discover more subtle cancer subtypes. The MPP is a model which is able to identify multiple clustering structures and treat the cluster sizes unfixed. Unfortunately, MPP is inefficient when \( p \) is large. In order to make MPP applicable for the aCGH data, a two-stage procedure is proposed and discussed in section 5.2. Because the two-stage is not optimal, as our future work, we will work on a model which is able to identify the shared regions and make clustering simultaneously. Section 5.4 will represent our future work.

5.2 Two-stage Clustering Approach

In order to overcome the large-\( p \) problem before applying MPP on aCGH data set, a two-stage clustering approach is proposed: 1) perform dimension reduction; 2) apply MPP to the dimension reduced “new” data. One important property of the aCGH data is that all the objects can be better captured by representing each one as possessing multiple latent features. Thus, for aCGH data, we could reduce the dimension in this way: each “new” dimension refers to a latent features and the “new” data matrix is a feature possession matrix with entry \((i, j)\) taking 0/1 values (absence/presence) to indicate whether the \( i \)-th object possesses the \( j \)-th latent feature or not. Usually the “new” features has relatively small dimensions. There are two advantages of this dimension reduction: the important patterns critical to clustering can be identified and the the original dimensions are preserved which is important to the cancer treatment. IBP can be used as a such dimension reduction tool. We will describe the IBP in the following section.

5.2.1 Indian Buffet Process

The typical clustering algorithms represent data in terms of which cluster each object belongs to. Clustering models are restrictive, because they do not have distributed representations. For example, we can describe a person as “student”, “female”, “Asian”, “married” and so on. These features are latent and the number of these potential features are always unlimited. Meanwhile, each object could be represented as possessing multiple latent features. Several methods exist for representing objects in terms of latent features, such as Blei et al. (2003) and Ueda and Saito (2003). These methods,
however, still could not solve one critical question: how many latent features are needed to express
the latent structure responsible for the observed data. In Griffiths and Ghahramani (2005), the
authors “take the idea of defining priors over infinite combinatorial structures from nonparametric
Bayesian statistics, and use it to develop methods for unsupervised learning in which each object
is represented by a sparse subset of an unbounded number of features”. Assume that data matrix
$X_{n \times p}$ involve $n$ objects and the measurement of the $i$-th object, $X_i$, is a $p$-dimensional vector. We have

$$X_{n \times p} = Z_{n \times K} F_{K \times p} + \epsilon,$$

(5.1)

where $Z_{n \times K}$ is a feature indicator matrix with entry $(i,j), Z_{ij}$, takes 0/1 values (absence/presence)
to indicate whether the $i$-th object possesses the $j$-th latent feature or not and $F_{K \times p}$ is a feature
matrix with $j$-th rows representing the $j$-th feature $f_j$. Besides, $K$ denotes the number of features
and $\epsilon$ is an error term. Because the feature number $K$ is assumed to be unbounded, we have $K = \infty$.

A distribution over $Z$ is designed to be used as a prior in probabilistic models that represent
objects using a potentially infinite array of features. However, when $K \to \infty$, the probability of
matrix $Z$, $P(Z)$, goes to 0. So, before giving the probability of matrices, the concept of equivalence
class, denoted as $[\cdot]$, is introduced. The equivalence classes are defined with respect to a function on
binary matrices, $lof(\cdot)$. This function maps binary matrices to left-ordered binary matrices. Any
two binary matrices $Y$ and $Z$ are $lof$-equivalent if $lof(Y) = lof(Z)$. The $lof$-equivalence class of a
binary matrix $Z$, denoted $[Z]$, is the set of binary matrices that are $lof$-equivalent to $Z$. Given the
equivalence class, we can have:

$$P([Z])_{K \to \infty} = \frac{\alpha_{K+} K!}{\prod_{h=1}^{K} K_h!} \exp\{-\alpha H_n\} \prod_{k=1}^{K} \frac{(nm_k)!(m_k - 1)!}{n!},$$

(5.2)

where $H_n = \sum_{i=1}^{n} \frac{1}{i}, K_h$ is the count of the number of columns with history $h$, $m_k = \sum_{i=1}^{n} z_{ik}$ and
$K_+$ is the number of active features possessed by at least one object.

The probability distribution in (5.2) can be derived from a simple stochastic process, called
exchangeable IBP (Griffiths and Ghahramani, 2005): $n$ customers enter a restaurant. Each customer
encounters a buffet consisting of infinitely many dishes. The first customer starts at the left of the
buffet and stops after a Poisson($\alpha$) number of dishes. The $i$-th customer moves along the buffet and
makes a single decision for each set of dishes with the same history. If there are $K_h$ dishes with
history $h$, under which $m_h$ previous customers have sampled each of those dishes, then the customer
samples a Binomial($\frac{m_h}{\alpha}, K_h$) number of those dishes, starting at the left. And then tries a Poisson($\frac{\alpha}{h}$)

62
new dishes.

In Gibbs Sampler, we can derive the conditional distribution from the exchangeable IBP:

$$P(Z_{ik} = 1 | Z_{-i,k}) = \frac{m_{-i,k}}{n},$$

where $Z_{-i,k}$ is the set of assignments of other objects, excluding the $i$-th object for the $k$-th feature, and $m_{-i,k}$ is the number of objects possessing feature $k$ excluding the $i$-th object. Similar to Chinese Restaurant Process, IBP contains a rich get richer phenomenon.

5.2.2 A Gibbs Sampler

Given the model as in 5.1 and the priors $F \sim N(0, \sigma^2_F I_p)$, $\sigma^2 \sim \text{InvGam}(s_1, s_2)$ and $\sigma^2_F \sim \text{InvGam}(t_1, t_2)$, a Gibbs sampler is employed to sample the posterior:

- Initialize $K_+, Z, F, \sigma^2, \sigma^2_F$ and Set the largest feature number increase $K_I$;

- In step $t$,

  1. For $i = 1 : n$ and $k = 1 : K_+$, consider three cases:

     (a) $m_{-i,k} > 0$: calculate $M_{-i} = M - \frac{MZ^T z_i M}{Z_i M Z_k^T - 1}$, if $Z_{ik} = 0$, $M_0 = M_{-i} - \frac{M_{-i} Z_i^T M_{-i}}{Z_i M_{-i} Z_k^T + 1}$; otherwise $M_1 = M_{-i} - \frac{M_{-i} Z_i^T M_{-i}}{Z_i M_{-i} Z_k^T + 1}$. Then update $Z_{ik} | X, Z_{-i,k}, \sigma^2, \sigma^2_F \sim \text{Bernoulli}(r_1 + r \sigma^2)$, where

     $$r = \frac{Z_{ik} = 1 | \cdots | M_{-i}^T M_{-i}}{\sum_j Z_{ik} = 0 | \cdots | M_{-i}^T M_{-i}} \propto \frac{\left(\frac{m_{-i,k}}{n}\right)^{p/2}}{|M_{-i}^T M_{-i}|} \exp\left(-\frac{1}{2\sigma^2} \text{tr}(X^T (Z_{0} M_{0} Z_0^T - Z_1 M_1 Z_1^T) X)\right) \times \frac{m_{-i,k}}{n},$$

     and $M_0 = M$;

     (b) $m_{-i,k} = 0$: delete $Z[,]$ and $A[,]$. Meanwhile $K_+ = K_+ - 1$;

     (c) $k = K_+$: new feature, $n_f$, for object $i$. $n_f \sim \text{Multinomial}(0, 1, 2, \cdots, K_I)$, $(p_0, p_1, p_2, \cdots, p_{K_I})$,

     where

     $$p_i \propto \left(\frac{\sigma^2}{\sigma^2_F}\right)^{\alpha_2/2} |\Sigma|^{\alpha/2} \exp\left(-\frac{1}{2\sigma^2} \text{tr}(X^T (1 - Z M Z^T) X)\right) \frac{\left(\frac{\alpha}{\sigma^2}\right)^\alpha \exp\left(-\frac{\alpha}{\sigma^2}\right)}{\alpha!};$$

  2. Add $n_f$ columns to $Z$, add $n_f$ rows to $F$ and update $K_+ = K_+ + n_f$;

  3. Update $F \sim \text{MN}(MZ^TX, I, \sigma^2 M)$, where $\text{MN}(\mu, \Omega, \Sigma)$ is the matrix normal distribution with mean $\mu$ and covariance $\Omega$ and $\Sigma$;

  4. Update $\sigma^2 \sim \text{InvGam}(\frac{np}{2} + s_1, \frac{\text{tr}(X^T X - ZF)^T (X - ZF)}{2} + s_2)$;
Figure 5.2: The first row are four objects generated by the four features shown in the second row; the third row are the first 4 most frequent features in MCMC.

5. Update $\sigma^2_A \sim \text{InvGam}(K_p + t_1, \frac{tr(F^TF)}{2} + t_2)$;

6. Update $\alpha \sim \text{Gamma}(K_+ + 1, H_n)$.

Note that we integrates $F$ out when updating $Z$. This collapsed Gibbs sampler makes the MCMC much more efficient.

5.2.3 Inference

After the burning period, we record all the features over the remaining MCMC steps for inference. In addition, we should record those “removed features” in each MCMC steps. In Gibbs Sampler, when updating $Z$, we could find some features possessed by just one object. This kind of features will be removed and will not appear in the next MCMC step. But, how to identify them to be same feature over two sequential MCMC steps? We adopt a shuffling operation as follows: in step $t$ and $t + 1$, we obtain $Z^{(t)}$ and $Z^{(t+1)}$. We know, some features might be removed from $Z^{(t)}$ according to the record. Let’s denote the remaining dishes as $Z_r^{(t)}$. Suppose $Z_r^{(t)}$ has $K_1$ features and $Z^{(t+1)}$ has $K_2$ features. Construct a $K_1 \times K_2$ table $T$ with cell $(i,j)$ equal to the common subjects number shared by feature $i$ in step $t$ and feature $j$ in step $t + 1$. Next, find the maximum cell in table $T$,
say \((a_0, b_0) = \arg \max_{a=1:K_1, b=1:K_2} T(a, b)\). Switch \(a_0\) and \(b_0\) columns in step \(t + 1\) and delete \(a_0\)-th row and \(b_0\)-th in \(T\). Keep doing the above procedure until only one column or one row left. For all the MCMC steps for inference, we can get a list of features and their corresponding frequencies. Eventually, we choose the most frequent \(\hat{K}\) features where \(\hat{K}\) is the posterior mode of \(K_+\).

### 5.2.4 Simulations

A simulation is considered to evaluate IBP. The data matrix we generate contains \(n = 100\) objects each of which has dimension \(p = 36\). Four latent features \((K_+ = 4)\) we use are shown in the second row of Figure 5.2. The \(Z_{100 \times 4}\) is generated by sampling each cell as Bernoulli(0.5). In addition, we set \(\sigma^2 = 0.25\). Eventually we generate the data matrix \(X\) based on \(Z, F\) and \(\sigma^2\). The First row of Figure 5.2 are the first four subjects.

The initial values in Gibbs Sampler are \(\sigma^2 = 1, \sigma^2_F = 1, \alpha = 1\). We run 1000 MCMC iterations, the sampling of parameters \(K_+, \sigma^2, \sigma^2_F, \alpha\), are represented in Figure 5.3. Set the first 500 as burning-up, and employ the remaining 500 steps as the samples of the posterior distribution. The histogram of \(K_+\) for the 500 samples are shown in Figure 5.4 and the mode is \(K_+ = 4\). Figure 5.2 (third row) represents the 4 most frequent features over MCMC steps. The frequencies of the features appearing
in MCMC are given in Figure 5.5.

5.3 MPP on Absence/Presence Matrix

In section 4.2, MPP is applied to a matrix with entries taking continuous values and these entries are assumed to follow Normal distributions. In the second step of the two-stage approach, the data matrix $Z$ takes binary values 0 or 1 to indicate the feature possession, so the entries are modeled as $Z_{ij} \sim \text{Bernoulli}(\Theta_{ij}^{(l)})$, where $\Theta_{ij}^{(l)}$ is the coefficient for entry $(i, j)$ in $l$-th partition. Because in the matrix $Z$, only $K_+$ columns are active and meaningful for clustering, $Z$ in this chapter only contains the $K_+$ active columns. Denote $S_{n \times K_+}$ as the structure indicator matrix and $C_{s \times n}$ as membership indicator matrix as in section 4.2. The model for MPP on absence/presence matrix is:

$$p(Z|S, \theta, \sigma^2, C) = \prod_{i=1}^{n} \prod_{j=1}^{K_+} \text{Bernoulli}(Z_{ij}; \Theta_{qij}^{(q)})$$

where $q = S_{ij}$. An conjugate prior of $\Theta_{qij}^{(q)}$ $\text{Beta}(1, b_q)$ is assigned. In terms of other priors and computation, they are similar to corresponding sections in chapter 4.
5.4 Future Work

The two-stage clustering approach extracts the latent features by IBP and then perform clustering by MPP based on the results of IBP. Apparently, this two-stage approach is not optimal. It will be more efficient if we integrate these two steps in one model. In other words, this integrated model is able to simultaneously extract the latent features over features and clustering the objects.
Appendix A

Shuffling

For \( j = 1 : (J - 1) \), perform the following steps sequentially,

1. Save the current configuration \( Z_· \) as \( \tilde{Z}_· \).

2. Based on \( Z_j \) and \( Z_{(j+1)} \), construct a \( K \times K \) table \( T \) whose \((a,b)\)-th entry is defined to be

\[
T_{a,b} = \sum_{i=1}^{n} I[Z_{ji} = a] \times I[Z_{(j+1)i} = b].
\]

Here the row indices correspond to the clusters from image \( j \) and the column indices correspond to the clusters from image \((j + 1)\).

3. Let \((a_0, b_0) = \arg \max_{a \in 1:K, b \in 1:K} T_{a,b}\).

4. Switch the labels \( a_0 \) and \( b_0 \) for \( Z_{(j+1)·} \).

5. Delete the \( a_0 \)-th row and \( b_0 \)-th column from table \( T \), and repeat the above procedure on the remaining \((K - 1) \times (K - 1) \) table, until reaching a table with only one entry.

6. If \( \pi(Z_·) > \pi(\tilde{Z}_·) \), that is,

\[
\sum_{(ji) \sim (j'j') \colon j' = j+1} \beta_{(ji)(j'j')}(I[Z_{ji} = Z_{j'j'}] - I[\tilde{Z}_{ji} = \tilde{Z}_{j'j'}]) > 0,
\]

accept the new configuration, otherwise, reset \( Z_· = \tilde{Z}_· \).
Appendix B

Algorithm I of BCM

For simplicity we assume $X_i = X$, for $i = 1, \cdots, m$, which is the case for our simulation examples and the real case study example. The two Gibbs sampling algorithms are given below, where Algorithm I is with prior (3.8) where we sample $(\mu, Z, \beta)$ together and Algorithm II is with prior (3.9) where we sample $(\mu, Z)$ together. In the expression for each conditional distribution, we use $\cdots$ to denote the current value of any other parameters in the iteration.

- Initialization.

- At the $t$-th iteration,

1. Update $(\beta, \mu, Z)|Y, \cdots$ jointly.

   a. Update $Z_{kj}$ sequentially for $k = 1: K$ and $j = 1: p$ with

   $$Z_{kj}|Y, [-\beta], [-\mu_k], \cdots \sim \text{Benoulli}(R/(R+1)),$$

   where $[-A]$ means that $A$ is excluded and

   $$\log R = \log \frac{w}{1 - w} - \frac{1}{2} \log \left( \frac{1/m_k}{(\tau^2 + v/m_k)||X_j||^2 + 1/m_k} \right)$$

   $$+ \frac{1}{2} \left( \frac{\sum_{i:H_i=k} Y_i^T X_j}{m_k} \right)^2 / \left( \frac{||X_j||^2}{m_k} + \frac{1}{vm_k + \tau^2 m_k^2} \right).$$

   Here $m_k$ is the size of the $k$-th segment and $X_j$ denotes the $j$-th column of the design matrix $X$.

   b. For $k = 1: K$ and $j = 1: p$, if $Z_{kj} = 0$, set $\mu_{kj} = 0$; else update

   $$\mu_{kj}|Y, [-\beta], \cdots \sim N(\mu_0, \sigma_0^2),$$

   $69$
where

\[
\sigma_0^2 = \left( m_k ||X_j||^2 + \frac{||X_j||^4 m_k}{||X_j||^2 + \sigma^2/v} + \frac{1}{\tau} \right)^{-1},
\]

\[
\mu_0 = \sigma_0^2 \left( \sum_{i: H_i = k} Y_i^T X_j - \frac{\sum_{i: H_i = k} Y_i^T X_j ||X_j||^2}{||X_j||^2 + 1/v} \right).
\]

(c) For \( i = 1 : m \) and \( j = 1 : p \), given \( H_i = k \), if \( Z_{kj} = 0 \), set \( \beta_{ij} = 0 \); else update

\[
\beta_{ij} | Y, \cdots \sim N(\mu_b, \sigma_b^2),
\]

where

\[
\sigma_b^2 = 1/ (||X_j||^2 + \frac{1}{b}); \\
\mu_b = \sigma_b^2 \left[ (Y_i - X_{[-j]} \beta_{i[-j]})^T X_j + \frac{\mu_k j}{v} \right]
\]

with \( \beta_{i[-j]} \) is \( \beta_i \) without the \( j \)-th element and \( X_{[-j]} \) is \( X \) without the \( j \)-th column.

2. For \( i = 1 : m \), update \( H_i | H_{[-i]} \) sequentially via

\[
P(H_i = k | Y, \cdots) = p_k
\]

where

\[
p_k \propto \exp \left\{ - \frac{\|\beta_i - \mu_k\|^2 }{2v} \right\} \times \exp \left\{ \lambda \sum_{i \neq i'} e_{ii'} I[H_{ii'} = k] \right\}.
\]

3. Update hyper-parameters \((v, \tau^2, w)\)

\[
v | Y, \cdots \sim \text{InvGa} \left( \frac{mp}{2} + t_1, \sum_{i,j} ||\beta_{ij} - \mu_{H_{ij}}||^2 / 2 + t_2 \right)
\]

\[
\tau^2 | Y, \cdots \sim \text{InvGa} \left( \sum_{k,j} Z_{kj}/2 + s_1, \sum_{k,j} \mu_{H_{ij}}^2 I[Z_{kj} = 1]/2 + s_2 \right)
\]

\[
w | Y, \cdots \sim \text{Beta} \left( \sum_{k,j} Z_{kj} + a_0, \sum_{k,j} (1 - Z_{kj}) + b_0 \right)
\]

4. Update the latent variable \( Y_{il} \) sequentially for \( i = 1 : m \) and \( l = 1 : n \) with

\[
Y_{il} = \Phi^{-1}(U) + X_l \beta_l,
\]
where \( \Phi(\cdot) \) is the c.d.f. of standard normal distribution and

\[
U \sim \text{Unif}[0, 1 - \Phi(X_i \beta_i)], \text{ if } C_{it} = 0;
\]

\[
U \sim \text{Unif}[1 - \Phi(X_i \beta_i), 1], \text{ if } C_{it} = 1.
\]

5. Calculate the working coefficient

\[
c = \left[ \sum_{i=1}^{m} ||Y_i - X_i \beta||^2 / \chi^2_{mn} \right]^{1/2}
\]

where \( \chi^2_{mn} \) denotes a random sample from chi-square distribution with \( mn \) degrees of freedom. Then, rescale other parameters:

\[
Y = Y/c, \ \beta = \beta/c, \ \mu = \mu/c, \ \tau^2 = \tau^2/c^2, \text{ and } v = v/c^2.
\]

6. Recorder the labels for the \( K \) segments, such that \( \mu_{11} < \mu_{21} < \cdots < \mu_{K1} \).
Appendix C

Algorithm II of BCM

Algorithm II is almost the same as Algorithm I, except in steps 1 and 3.

1. Update $(\mu, Z)|Y, \cdots$ jointly.

   (a) Update $Z_{kj}$ for $k = 1 : K$ and $j = 1 : p$

   $$Z_{kj}|Y, [-\mu_k], \cdots \sim \text{Bernoulli}(R/(R+1)),$$

   where

   $$\log R = \log \frac{w}{1-w} + \frac{1}{2} \log \left[ \frac{v}{v + m_k \tau^2} + \frac{m_k^2 \tau^2 || \sum_{i:H_i=k} \beta_{ij}/m_k ||^2}{2v(v + m_k \tau^2)} \right].$$

   (b) For $k = 1 : K$ and $j = 1 : p$, if $Z_{kj} = 0$, set $\mu_{kj} = 0$; else update

   $$\mu_{kj}|Y, \cdots \sim N(\mu_0, \sigma_0^2),$$

   where

   $$\sigma_0^2 = \left(\frac{m_k}{v} + \frac{1}{\tau^2}\right)^{-1}, \quad \mu_0 = \frac{\sigma_0^2 m_k}{v} \left( \sum_{i:H_i=k} \beta_{ij} \right)$$

   Then update $\beta_{ij}$ for $i = 1 : m$ and $j = 1 : p$

   $$\beta_{ij}|Y, H_i = k, \cdots \sim N(\mu_b, \sigma_b^2),$$

   where

   $$\sigma_b^2 = \left( ||X_j||^2 + \frac{1}{\nu} \right)^{-1}$$

   $$\mu_b = \sigma_b^2 \left[ (Y_i - X_{i,-j} \beta_{i,-j})^T X_j + \frac{\mu_{kj}}{v} \right].$$

   ......
3. Update $v$

$$v | \mathbf{Y}, \cdots \sim \text{InvGa}\left( \sum_{k,j} I[\beta_{kj} = 0] / 2 + t_1, \sum_{i,j} ||\beta_{ij} - \mu_{H,j}||^2 / 2 + t_2 \right).$$
Appendix D

Gibbs Sampler of MPP

1. update $S$. For $i = 1 : n$, $j = 1 : p$, sequentially update $S_{ij} | Z, S_{[-i][j]}, C, \sigma^2, \cdots$ as Appendix II.

2. re-sample $S$. For $l = 1 : s$, $j = 1 : p$, sequentially re-sample $\bar{S}_{Mlj} = h | Z, C, \sigma^2, \bar{S}_{[-Mlj]}$ as Appendix III.

3. update $\theta$. For $l = 1 : s$, $k = 1 : c_l$, $j = 1 : p$,

\[
\Theta_{kj}^{(l)} \sim N(\mu_1, \sigma_1^2) \\
\sigma_1^2 = 1/(N_k/\sigma^2 + 1/\sigma_0^2) \times \mu_1 = (\sum_{i:C_{li}=k,S_{ij}=l} Z_{ij}/\sigma^2 + \mu_0/\sigma_0^2) \sigma_1^2,
\]

where $N_k = \sum_{i=1}^n I[C_{li} = k, S_{ij} = l]$.

4. update $u, C, \pi^*$. 

(a) update $u$. For $l = 2 : s$, $i = 1 : n$,

\[
u_{li} \cdots \sim \text{Unif}(0, \pi_{C_{li}}^{(l)}).
\]

(b) update $\pi^*$. For $l = 2 : s$,

if $h > c_l$, $\pi_{h}^{*(l)} \sim \text{Beta}(1, \alpha_l)$;
else $\pi_{h}^{*(l)} \sim \text{Beta}(1, \alpha_l) I[b_l \leq \pi_{h}^{*(l)} \leq b_h]$, where

\[
b_l = \max_{i:C_{li}=h} \left\{ \frac{u_{li}}{\prod_{t<h} (1 - \pi_{t}^{*(l)})} \right\}, \\
b_h = 1 - \max_{i:C_{li}>h} \left\{ \frac{u_{li}}{\pi_{C_{li}}^{(l)} \prod_{t<C_{li}, t\neq h} (1 - \pi_{t}^{*(l)})} \right\}.
\]
(c) update $C$. For $l = 2 : s$, $i = 1 : n$

$$P(C_{li} = h | \cdots) \propto I[h \in m_i] \prod_{j : S_{ij} = l}{\exp \left( \frac{-(Z_{ij} - \Theta^{(l)}_{C_{ij}})^2}{2\sigma^2} \right)},$$

where $m_i$ is the set $[1, 2, \cdots, b_i]$, and $b_i$ is the smallest value satisfying

$$\sum_{h=1}^{b_i} \pi^{*}_{h} \prod_{t<h}(1 - \pi^{*}_{t}) \geq u^*_l.$$

Here $u^*_l = \min_{i=1:n} u_{li}$.

5. update $\alpha$. For $l = 2 : s$,

$$\alpha_l = \text{Gamma}(a_\alpha + c_l, b_{\alpha} - \sum_{h=1}^{c_l} \log(1 - \pi^{*}_{h}))$$

6. update $\sigma^2$.

$$\sigma^2 = IG \left( \frac{np}{2} + \gamma_1, \frac{\sum_{l=1}^{s} \sum_{j=1}^{p} (Z_{ij} - \Theta^{(q)}_{C_{ij}})^2}{2} + \gamma_2 \right),$$

where $q = S_{ij}$. 
Appendix E

Updating of $S$ in Gibbs Sampler of MPP

For simplicity, we suppress the subscript $j$ in all the notations below and define $S_i$ as the structure indicator for $i$-th object and $S_{[-i]}$ as all the objects except $i$-th object.

$$P(S_i = l | Z, S_{[-i]}, C, \sigma^2) \propto P(Z | S_{[-i]}, S_i = l, C, \sigma^2) P(S_i = l | S_{[-i]})$$

$$= P(Z | S_{[-i]}, S_i = l, C, \sigma^2) \lambda d_l + \sum_{t \neq i} I[S_t = l].$$  \hspace{1cm} (E.1)

Note that the sampling of $S_i$ does not depend on $\theta$, i.e., we integrate over $\theta$. The partition matrix $C$ and structure indicator $S$ divide the objects (except the $i$-th object) into several groups. When we consider sampling $P(S_i = l | Z, S_{[-i]}, C, \sigma^2)$, the $i$-th object joins the following group:

$$T_{il} = \{ t : t \in 1 : n, t \neq i, S_i = l, C_t = C_l \},$$

which is the set of objects, excluding $i$, in cluster $C_l$ of structure $l$. Besides, define $N_{il} = |T_{il}|$, the number of objects in $T_{il}$. It is easy to check that only the density function over objects that are not in $\cup_{l=1}^{s} T_{il}$ does not depend on $S_i$. Therefore, the first term in Eq.(E.1) can be rewritten as

$$P(Z | S_{[-i]}, S_i = l, C, \sigma^2)$$

$$\propto P(Z_{T_{il} \cup \{i\}} | S_{[-i]}, S_i = l, C, \sigma^2) \prod_{l' \neq l} P(Z_{T_{il'}} | S_{[-i]}, C, \sigma^2)$$

$$\propto \frac{P(Z_{T_{il} \cup \{i\}} | S_{[-i]}, S_i = l, C, \sigma^2)}{P(Z_{T_{il}} | S_{[-i]}, C, \sigma^2)},$$

\hspace{1cm} (E.2)

where the last step is obtained by dividing the expression by $\prod_{l'=1}^{s} P(Z_{T_{il'}}, S_{[-i]}, C, \sigma^2)$ and $p(Z_{T}, S_{[-i]}, C, \sigma^2)$ denotes the density (integrated over $\theta$) for $Z_t : t \in T$.

To calculate the ratio in the last of step of Eq.(E.2), two cases are considered: (I) $T_{il}$ is nonempty; (II) $T_{il}$ is empty.
Case I: $T_{il}$ is nonempty. Recall the integrated density for objects in $X_1, \cdots, X_d \sim N(\Theta, \sigma^2)$ with prior $\Theta \sim N(\mu_0, \sigma_0^2)$ is given by

$$\int \frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{\sqrt{2\pi\sigma_0^2}} \int \exp\left\{ -\frac{\sum_{t \in 1:D}(X_t - \Theta)^2}{2\sigma^2} \right\} \exp\left\{ -\frac{(\Theta - \mu_0)^2}{2\sigma_0^2} \right\} d\Theta$$

Then, the ratio in the last step of Eq.(E.2) is simplified to be $P(Z_i|S_i = l, C, \sigma^2)$ and

$$P(Z_i|S_i = l, C, \sigma^2) = \frac{1}{\sqrt{2\pi} \sqrt{\sigma^2 + \sigma_0^2}} \exp\left\{ \frac{(Z_i/\sigma^2 + \mu_0/\sigma_0^2)^2}{2(1/\sigma^2 + 1/\sigma_0^2)} - \frac{\mu_0^2/\sigma_0^2 + Z_i^2/\sigma^2}{2} \right\}.$$

Case II: $T_{il}$ is empty. The ratio in the last step of Eq.(E.2) is simplified to be $P(Z_i|S_i = l, C, \sigma^2)$ and

$$P(Z_i|S_i = l, C, \sigma^2) = \frac{1}{\sqrt{2\pi} \sqrt{\sigma^2 + \sigma_0^2}} \exp\left\{ \frac{(Z_i/\sigma^2 + \mu_0/\sigma_0^2)^2}{2(1/\sigma^2 + 1/\sigma_0^2)} - \frac{\mu_0^2/\sigma_0^2 + Z_i^2/\sigma^2}{2} \right\}$$

which agrees with Eq. (E.3) by setting $N_{il} = 0$, $\sum_{t \in T_{il}} Z_i = 0$ and $\sum_{t \in T_{il}} Z_i^2 = 0$.

So, we have

$$P(Z_i|S_i = l, C, \sigma^2) = \frac{1}{\sqrt{2\pi} \sqrt{\sigma^2 + \sigma_0^2}} \exp\left\{ \frac{(Z_i/\sigma^2 + \mu_0/\sigma_0^2)^2}{2(1/\sigma^2 + 1/\sigma_0^2)} - \frac{\mu_0^2/\sigma_0^2 + Z_i^2/\sigma^2}{2} \right\}.$$

Eventually,

$$P(S_i = l|Z, S_{-i}, C, \sigma^2)$$

$$\propto P(Z|S_{-i}, S_i = l, C, \sigma^2) \frac{\lambda d_i + \sum_{t \neq i} I[S_t = l]}{n - 1 + \lambda}$$

(E.4)
Appendix F

Resampling of S in Gibbs Sampler of MPP

We again suppress the subscript $j$ in all the notations below. Define $S_{M_l}$ is set of structure indicators of objects in $M_l$ and $\bar{S}_{M_l}$ is the unique structure indicator value of the set $M_l$.

For $l = 1 : s$

$$\Pr(\bar{S}_{M_l} = h | Z, C, \sigma^2, \bar{S}_{[-M_l]}) \propto \Pr(Z | S_{M_l} = h, \bar{S}_{[-M_l]}, C, \sigma^2) \Pr(\bar{S}_{M_l} = h)$$

$$= \int \Pr(Z, \theta | S_{M_l} = h, \bar{S}_{[-M_l]}, C, \sigma^2) d\theta \Pr(\bar{S}_{M_l} = h)$$

If $h = l$, then

$$= \int \Pr(Z, \theta | S_{M_l} = h, \bar{S}_{[-M_l]}, C, \sigma^2, \theta) \Pr(\theta | \mu_0, \sigma_0^2) d\theta \Pr(\bar{S}_{M_l} = h)$$

$$\propto 1 \prod_{t=1}^{s} \prod_{k=1}^{c_t} \sqrt{\frac{1}{N_{tk}/\sigma^2 + 1/\sigma_0^2}} \exp \left\{ \frac{(\sum_{i \in H_{tk}} Z_i/\sigma^2 + \mu_0/\sigma_0^2)^2}{2(N_{tk}/\sigma^2 + 1/\sigma_0^2)} - \frac{\sum_{i \in H_{tk}} Z_i^2/\sigma^2 + \mu_0^2/\sigma_0^2}{2} \right\} dh. \tag{F.1}$$

If $h \neq l$, then

$$\Pr(\bar{S}_{M_l} = h | Z, C, \sigma^2, \bar{S}_{[-M_l]}) \propto \prod_{t \neq l, l \neq h} \prod_{k=1}^{c_t} \sqrt{\frac{1}{N_{tk}/\sigma^2 + 1/\sigma_0^2}} \exp \left\{ \frac{(\sum_{i \in H_{tk}} Z_i/\sigma^2 + \mu_0/\sigma_0^2)^2}{2(N_{tk}/\sigma^2 + 1/\sigma_0^2)} - \frac{\sum_{i \in H_{tk}} Z_i^2/\sigma^2 + \mu_0^2/\sigma_0^2}{2} \right\}$$

$$\times \prod_{k=1}^{c_h} \sqrt{\frac{1}{(N_{hk} + N_{tk})/\sigma^2 + 1/\sigma_0^2}} \exp \left\{ \frac{(\sum_{i \in \{H_{tk}, H_{hk}\}} Z_i/\sigma^2 + \mu_0/\sigma_0^2)^2}{2((N_{hk} + N_{lk})/\sigma^2 + 1/\sigma_0^2)} - \frac{\sum_{i \in \{H_{tk}, H_{hk}\}} Z_i^2/\sigma^2 + \mu_0^2/\sigma_0^2}{2} \right\} dh. \tag{F.2}$$

where $H_{tk}^h = \{t : S_t = l, C_{ht} = k\}$ and $N_{tk}^h = |H_{tk}^h|$. 78
Eq (F.1) and (F.2) can be further simplified by dividing the term

$$\prod_{t=1}^{s} \prod_{k=1}^{c_t} \sqrt{\frac{1}{N_{tk}/\sigma^2 + 1/\sigma_0^2}} \exp \left\{ \left( \frac{\sum_{i \in H_{tk}} Z_i/\sigma^2 + \mu_0/\sigma_0^2}{2(N_{tk}/\sigma^2 + 1/\sigma_0^2)} - \frac{\sum_{i \in H_{tk}} Z_i^2/\sigma^2 + \mu_0^2/\sigma_0^2}{2} \right) \right\}.$$ 

Thus,

if $h = l$, then

$$P(\bar{S}_{M_i} = h|Z, C, \sigma^2, \bar{S}_{[-M_i]}) \propto d_h;$$

else

$$P(\bar{S}_{M_h} = h|Z, C, \sigma^2, \bar{S}_{[-M_h]}) \propto \prod_{k=1}^{c_h} \sqrt{\frac{1}{(N_{hk} + N_{hl})/\sigma^2 + 1/\sigma_0^2}} \exp \left\{ \left( \frac{\sum_{i \in \{H_{tk}, H_{hk}\}} Z_i/\sigma^2 + \mu_0/\sigma_0^2}{2((N_{hk} + N_{hl})/\sigma^2 + 1/\sigma_0^2)} - \frac{\sum_{i \in \{H_{tk}, H_{hk}\}} Z_i^2/\sigma^2 + \mu_0^2/\sigma_0^2}{2} \right) \right\} \frac{1}{\prod_{t=l,h=1}^{c_t} \sqrt{\frac{1}{N_{tk}/\sigma^2 + 1/\sigma_0^2}} \exp \left\{ \left( \frac{\sum_{i \in H_{tk}} Z_i/\sigma^2 + \mu_0/\sigma_0^2}{2(N_{tk}/\sigma^2 + 1/\sigma_0^2)} - \frac{\sum_{i \in H_{tk}} Z_i^2/\sigma^2 + \mu_0^2/\sigma_0^2}{2} \right) \right\} d_h. $$
References


82


