Bayesian Profile Regression with Evaluation on Simulated Data

by

Dongmeng Liu

B.Sc., Nankai University, 2014

Project Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Spring 2016

Approval

Name:	**Dongmeng Liu**
Degree:	**Master of Science**
Title:	***Bayesian Profile Regression with Evaluation on Simulated Data***
Examining Committee:	**Chair:** Dr. Tim Swartz Professor

Dr. Jinko Graham
Senior Supervisor
Professor

Dr. Joan Hu
Supervisor
Professor

Dr. Brad McNeney
Internal Examiner
Associate Professor

Date Defended: 6 January 2016

Abstract

Using regression analysis to make inference using data sets that contain a large number of potentially correlated covariates can be difficult. This large number of covariates have become more common in clinical observational studies due to the dramatic improvement in information capturing technology for clinical databases. For instance, in disease diagnosis and treatment, obtaining a number of indicators regarding patients' organ function is much easier than before and these indicators can be highly correlated. We discuss Bayesian profile regression, an approach that deals with the large numbers of correlated covariates for the binary covariates commonly recorded in clinical databases. Clusters of patients with similar covariate profiles are formed through the application of a Dirichlet process prior and then associated with outcomes via a regression model. Methods for evaluating the clustering and making inference are described afterwards. We use simulated data to compare the performance of Bayesian profile regression to the LASSO, a popular alternative for data sets with a large number of predictors. To make these comparisons, we apply the recently developed R package PReMiuM, to fit the Bayesian profile regression.

Keywords: Bayesian mixture model; Clustering; Dirichlet Process; Profile regression

Acknowledgements

I would first like to thank my supervisor Dr. Jinko Graham for helping me and for being patient with me. I feel lucky to receive professional guidance from a knowledgeable supervisor, who has rich experience and sophisticated knowledge. Her enlightening and numerous support help me in all the time of research and writing of this project. I would never have been able to finish my project without her tremendous help, and I really appreciate her unlimited patience and kindness to me. At the same time, I would like to thank my committee chair, Dr. Tim Swartz, and committee members, Dr. Brad McNeney and Dr. Joan Hu, for their insightful comments and reviews on this project.

I thank my fellow office mates for the stimulating discussions along the way, for all the hard-working time that we worked together, and for all the fun we have had. I have been blessed with such a friendly and cheerful group of colleagues and friends, and feel lucky for getting the opportunity to continue to Ph.D. study at Simon Fraser University for another several years.

Finally, I wish to thank my boyfriend, Yabin, for being good-tempered all the time and making me laugh every time when I get stressful. Also, I would like to express my special thanks to my parents. I am eternally grateful for how much you support and love me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In biomedical studies, hundreds or even thousands of factors measuring the patients' lifestyle, health indicators and even genetics may be reported and then taken into consideration in data analysis. It has become routine to collect massive amounts of data for each study subject. Traditional regression analysis was developed for smaller scale data and, as a result, is less helpful when the data set contains a large number of covariates that are potentially correlated. In these data sets, the association between a particular covariate and the response can be statistically significant by itself but not when many other correlated covariates are also included in the model. Traditional regression analysis is no longer effective to find the true pattern when there is collinearity in the covariates.

To deal with the above problems, one possible approach makes nonparametrically based inference on clusters reflecting covariate profiles. The approach takes a more global point of view that settles the problems caused by collinearity in the data set. Previously introduced methods to profile data using clustering include the k-means algorithm (see, e.g. Hartigan and Wong, 1979) and latent class analysis (see Patterson et al., 2002). However, Bayesian profile regression, which we describe in this report, offers a number of advantages over these methods. First, unlike the k-means algorithm, the method is able to take into account the uncertainty associated with clustering by using Markov chain Monte Carlo (MCMC) methods to sample from the posterior distribution of the latent cluster memberships. Given these cluster memberships, a Bayesian mixture model is fitted at each iteration. In this way, a number of different clusterings are obtained across sampling iterations. Second, unlike latent class analysis, the number of clusters does not have to be fixed in advance of analyzing the data. Third, unlike the k-means algorithm, the method links the clustering to the response variable so that they can inform each other. Fourth, like latent class analysis, the method is able to compute the uncertainty associated with the cluster membership but, unlike latent class analysis, it does so via a model-averaging approach that allows us to evaluate the "best" partition obtained from the Bayesian mixture model.

In the following chapters, we will describe the methods of Bayesian profile regression and discuss some ways to evaluate the clustering output. Then we will use a simulation study to demonstrate the utility of the method and compare the results with a regression method, LASSO. Finally, we will briefly show the usage of the R package `PReMiuM`, which has been recently developed for Bayesian profile regression, and provide some explanations of the functions.

Chapter 2

Methodology

In this chapter, we first review the idea of a Dirichlet process, and how it is used as the prior in Bayesian mixture models. Then we review two submodels based on a Dirichlet process. The "assignment submodel" is a Bayesian mixture model clustering the covariate profiles into groups. The "disease submodel" links the groups to an outcome of interest using a logistic regression model. MCMC methods are used to fit both submodels, as is common with Bayesian approaches.

2.1 Dirichlet Process Introduction

The Dirichlet distribution, denoted by $Dirichlet(\alpha_1, \alpha_2, \ldots, \alpha_C)$, is a family of continuous multivariate probability distributions with a parameter vector $(\alpha_1, \alpha_2, \ldots, \alpha_C)$ of positive reals. We let $\Psi = \{\psi_1, \psi_2, \ldots, \psi_C\}$ represent the random variates, and write

$$\Psi \sim Dirichlet(\alpha_1, \alpha_2, \ldots, \alpha_C),$$

if they have a probability density function given by

$$f(\psi_1, \psi_2, \ldots, \psi_C) = \frac{\Gamma(\sum_c \alpha_c)}{\prod_c \Gamma(\alpha_c)} \prod_{c=1}^{C} \psi_c^{\alpha_c - 1},$$

where $\alpha_1, \ldots, \alpha_C > 0$ and $\Gamma(\alpha_c) = \int_0^\infty x^{\alpha_c - 1} e^{-x} dx$. Samples from the distribution lie in the $(C-1)$-dimensional simplex defined by:

$$\begin{aligned} \psi_1, \ldots, \psi_{C-1} &> 0 \\ \psi_1 + \ldots + \psi_{C-1} &< 1 \\ \psi_C &= 1 - \psi_1 - \ldots - \psi_{C-1}. \end{aligned}$$

It is easy to see that the Beta distribution is a special case of the Dirichlet distribution with $C = 2$.

A Dirichlet process is a distribution over distributions, often denoted by $DP(\alpha, G_0)$, where G_0 is a base distribution and α is a positive "concentration" parameter. Let G be a random probability measure drawn from a Dirichlet process, written $G \sim DP(\alpha, G_0)$. Then G has a support set contained within the support of G_0. For example, consider a uniform distribution on the interval (0,1), given by $G_0 = Beta(1, 1)$, and draw samples from the Dirichlet process using different α (see Figure 1). We can see that G is a discrete distribution, made up of a countable number of point masses. Blackwell and MacQueen (1973) showed that the distributions sampled from a Dirichlet process are discrete almost surely. The sampled G becomes more like the base distribution as the concentration parameter increases.

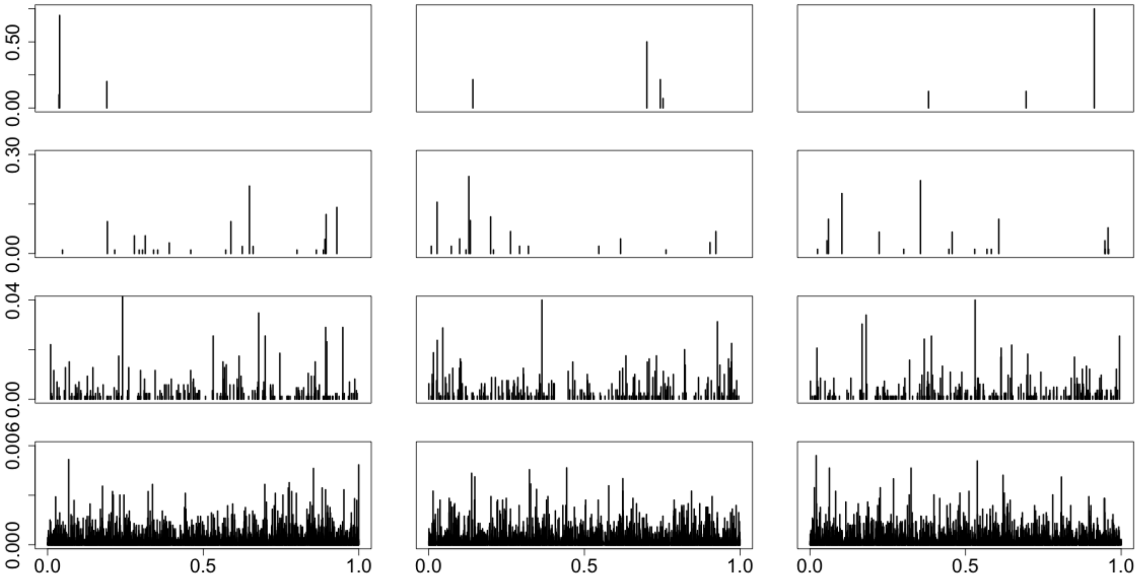


Figure 2.1: Draws of distribution G from the Dirichlet process $DP(\alpha, Beta(1, 1))$ using increasing values of the concentration parameter α (top to bottom: 1, 10, 100 and 1000). Each row consists of 3 repetitions of the same experiment. The vertical axis represents the probability masses and the horizontal axis represents the support points.

Assume we draw samples ϕ_i^*, for $i = 1, \dots, n$, from G where $G \sim DP(\alpha, G_0)$. Then the ϕ_i^*'s are i.i.d. given G. To obtain the predictive distribution of ϕ_i^*, the idea is to start with the joint distribution of the ϕ_i^*'s. Conceptually, this joint distribution is obtained by marginalizing out G as $P(\phi_1^*, \dots, \phi_n^*) = \int P(G) \prod_{i=1}^n P(\phi_i^* \mid G) dG$, where P demonstrates either a density or pmf, as appropriate. Assume we view ϕ_i^*'s in a specific order and are interested in the behaviour of ϕ_i^* given the previous $i - 1$ observations, we obtain the predictive distribution of ϕ_i^* from the joint distribution as,

$$\phi_i^* \mid \phi_1^*, \ldots, \phi_{i-1}^* = \begin{cases} \phi_k^* & \text{with probability } \frac{1}{i-1+\alpha} \\ \text{new draw from } G_0 & \text{with probability } \frac{\alpha}{i-1+\alpha} \end{cases}$$

where ϕ_k^* can be any one of $\phi_1^*, \ldots, \phi_{i-1}^*$ (Blei and Jordan, 2006). We can rewrite the above probabilities in the following form:

$$\phi_i^* \mid \phi_1^*, \ldots, \phi_{i-1}^* = \begin{cases} \phi_c & \text{with probability } \frac{num_{i-1}(\phi_c)}{i-1+\alpha} \\ \text{new draw from } G_0 & \text{with probability } \frac{\alpha}{i-1+\alpha} \end{cases} \tag{2.1}$$

where ϕ_c for $c \in 1, \ldots, C$ are the unique values of ϕ_i^* for $i \in 1, \ldots, n$, and $num_{i-1}(\phi_c)$ denotes the number of observations that equal to ϕ_c before the i^{th} observation. Thus a new observation tends to take on a value that has already been observed in the sample, in proportion to the number of times it has been observed. We also notice that the order of the samples, $\phi_1^*, \ldots, \phi_i^*$, does not have an impact on their distribution, but the unique values of these samples and how many of them take each unique value really matters. That is, the Dirichlet process realizations are exchangeable.

The above process is also called “Chinese restaurant process”, analogous to seating customers at tables in a Chinese restaurant. Think of it in this way: the first customer coming into a Chinese restaurant, where there are an infinite number of circular tables, each with infinite capacity, is seated at an unoccupied table with probability 1. At time i, a new customer comes in, and has the options to sit to the left of one of the $i-1$ customers that are already sitting at an occupied table, or at a new unoccupied table. Each occupied table corresponds to a component in the mixing model. From the above predictive distribution, we can see that a customer is more likely to sit at a table if there are already many people sitting there. However, with a probability proportional to α, the customer will sit at a new table.

2.2 Dirichlet Process Mixture Model

The idea of Dirichlet process mixture models goes back to Antoniak (1974). In this report, we will describe a standard discrete mixture model (see Shahbaba and Neal, 2009 and Neal, 2000). Suppose that we have covariate vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ on n individuals and assume that the covariate vectors are exchangeable random variables drawn independently from some unknown distribution. We can model their distribution as a mixture of simple distributions, with probability or density function

$$P(\boldsymbol{x}) = \sum_{c=1}^{C} \psi_c F(\boldsymbol{x} \mid \phi_c),$$

where ψ_c refer to the mixing proportions, and $F(\boldsymbol{x} \mid \phi)$ refers to the probability or density of $\boldsymbol{x}$ under a distribution, $F(\phi)$, in some simple class with parameters ϕ. For example, $F(\phi)$ could be the density function of a normal distribution, where ϕ refers to the parameters for the population mean, μ, and standard deviation, σ; that is, $\phi = (\mu, \sigma)$. For each cluster c, ϕ_c determines the distribution of observations within that cluster.

We start by assuming that the number of mixing components, C, is finite. In this case, a common prior distribution for ψ_c is a symmetric Dirichlet distribution whose density function is defined as

$$f(\psi_1, \ldots, \psi_C) = \frac{\Gamma(\alpha)}{\Gamma(\alpha/C)^C} \prod_{c=1}^{C} \psi_c^{(\alpha/C)-1},$$

where $\psi_c \geq 0$ and $\sum \psi_c = 1$. An allocation variable, $Z_i = c$, indicates the cluster to which individual i is assigned. Let $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_C)$, then the model can be written as follows:

$$\begin{aligned} &\boldsymbol{X}_i \mid Z_i, \boldsymbol{\phi} \sim F(\phi_{Z_i}) \\ &Z_i \mid \psi_1, \ldots, \psi_C \sim Multinomial(\psi_1, \ldots, \psi_C) \\ &\psi_1, \ldots, \psi_C \sim Dirichlet(\alpha/C, \ldots, \alpha/C) \\ &\phi_Z \sim G_0 \end{aligned} \tag{2.2}$$

We can eliminate the mixing proportions, ψ_c, and obtain the predictive distribution for Z_i by integrating the joint distribution of $(Z_i, \psi_1, \ldots, \psi_C)$ over the Dirichlet prior for $\psi_1, \ldots, \psi_C$ to get the joint distribution of $Z_1, \ldots Z_n$ and then applying Bayes theorem:

$$Pr(Z_i = c \mid Z_1, \ldots, Z_{i-1}) = \frac{num_{i-1}(c) + \alpha/C}{i - 1 + \alpha},$$

where $num_{i-1}(c)$ denotes the number of observations before the i^{th} observation assigned to component c and is a function of $Z_1, \ldots, Z_{i-1}$.

When we let C go to infinity, we obtain:

$$\begin{aligned} &Pr(Z_i = c \mid Z_1 \ldots, Z_{i-1}) \rightarrow \frac{num_{i-1}(c)}{i - 1 + \alpha} \\ &Pr(Z_i \neq Z_j \text{ for all } j < i \mid Z_1 \ldots, Z_{i-1}) \rightarrow \frac{\alpha}{i - 1 + \alpha} \end{aligned}$$

As a result, the conditional distribution for ϕ_{Z_i}, becomes

$$\phi_{Z_i} \mid \phi_{Z_1}, \ldots, \phi_{Z_{i-1}} \sim \frac{1}{i - 1 + \alpha} \sum_{j<i} \delta_{\phi_{Z_j}} + \frac{\alpha}{i - 1 + \alpha} G_0,$$

where $\delta_{\phi_{Z_j}}$ denotes the point mass distribution at ϕ_{Z_j}. Note that the observations are assumed to be exchangeable, and so we can view any observation, i, as the last observation, and thus obtain the conditional distribution in the following form:

$$\phi_{Z_i} \mid \phi_{Z_{-i}} \sim \frac{1}{n-1+\alpha} \sum_{j \neq i} \delta_{\phi_{Z_j}} + \frac{\alpha}{n-1+\alpha} G_0,$$

where $\phi_{Z_{-i}}$ denotes all the ϕ_{Z_j} for $j \neq i$. This is the same conclusion reached in equation (2.1) above, taking ϕ_i^* in the equation to be ϕ_{Z_i} here.

An equivalent form of the above Dirichlet process is the "stick-breaking" process. Sethuraman (1994) introduced stick-breaking as a constructive way of forming G, which is expected to be distributed according to a Dirichlet process. If $G \sim DP(G_0, \alpha)$, a simplified constructive definition of the Dirichlet process is

$$\begin{aligned} &G = \sum_{c=1}^{\infty} \psi_c \delta_{\phi_c} \\ &\phi_c \sim G_0 \quad \text{i.i.d for } c \in Z^+ \\ &\psi_c = V_c \cdot \prod_{l<c} (1 - V_l) \quad \text{i.i.d for } c \in Z^+ \setminus \{1\} \\ &\psi_1 = V_1 \\ &V_c \sim Beta(1, \alpha) \quad \text{i.i.d for } c \in Z^+, \end{aligned} \tag{2.3}$$

where ϕ_c is independent of V_c for $c \in Z^+$ and Z^+ denotes the positive integers. This formulation for V and ψ is the so-called "stick-breaking" distribution. Imagine that we start with a unit-length stick and in the l^{th} step we break off a piece of the remaining stick according to V_l and then assign this broken-off portion to ψ_l. It is important to note that the distribution, G, is discrete, because draws of ϕ's from G_0 can only take the values in the set $\{\phi_c : c \in Z^+\}$.

2.3 Assignment Submodel

One of the most common applications of the Dirichlet process is in clustering data, where the Dirichlet distribution serves as the prior distribution for the mixing-proportion parameters, $\Psi_c = (\psi_1, \ldots, \psi_C)$, in the mixture model, given the number of clusters C. We now describe an allocation submodel of the probability that an individual is assigned to a particular cluster (see Molitor et al., 2010). This assignment submodel is based on a stick-breaking formulation of the Dirichlet process.

We denote a vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ as the observed covariate profile for individual i. ψ_c denotes the probability of assignment to the c^{th} cluster and $\phi_c^j(x)$ denotes the probability of the j^{th} covariate in cluster c being x for a discrete covariate. The parameters, $\phi_c^1, \phi_c^2, \ldots, \phi_c^p$ are the prototypical profile for cluster c. The covariates are assumed to be independent conditional on the cluster assignment. The basic mixture model for allocation of individual i to a group based on his or her observed covariate vector $\boldsymbol{x}_i$ has the following probability

$$Pr(\boldsymbol{x}_i) = \sum_{c=1}^{C} Pr(Z_i = c) \cdot f(\boldsymbol{x}_i \mid Z_i = c)$$

where the mixture weights, $Pr(Z_i = c) = \psi_c$ are modeled according to the "stick-breaking" prior in equation (2.3). The probability of observed covariate vector $\boldsymbol{x}_i$ can be rewritten with parameters $\phi_c^1, \ldots, \phi_c^p$, so that

$$\begin{aligned} Pr(\boldsymbol{x}_i) &= \sum_{c=1}^{C} Pr(Z_i = c) \prod_{j=1}^{p} Pr(x_{ij} \mid Z_i = c) \\ &= \sum_{c=1}^{C} \psi_c \prod_{j=1}^{p} \phi_c^j(x_{ij}). \end{aligned}$$

An individual's covariate values are assumed to be conditionally independent given their cluster membership. The mixture model for $Pr(\boldsymbol{x}_i)$ incorporates a Dirichlet process (DP) prior through the covariate profiles' distribution $F(\boldsymbol{x}_i \mid \phi_c) = \prod_{j=1}^{p} \phi_c^j(x_{ij})$ and through the mixing proportions ψ_c for the clusters.

Larger values of the parameter α imply less clustering. To avoid computational difficulties from α values that are too small, Ohlsson et al. (2007) suggest a lower bound of 0.3. Since there is little *a priori* information about α, Molitor et al. (2010) assign it a uniform prior distribution on the interval (0.3, 10). The infinite cluster model in the Dirichlet process is approximated by considering a maximum number of clusters, C. Molitor et al. (2010) note that the value of C needs to be large enough to give a good approximation but small enough to avoid having to estimate a large number of unnecessary cluster parameters and allocation probabilities for very small clusters. The R package `PReMiuM` does not require specification of C in its profile regression function. However, the function does have an argument that requires the initial number of clusters. For our simulated data, we find that neither specifying the value of the initial number of clusters being as small as 2, nor as high as 20 will impact the Bayesian profile regression output, as long as the MCMC iterations are enough (for example, 20000 sweeps). In this report, we only analyze data sets with binary covariates and binary outcomes (i.e., an individual is either diseased or non-diseased).

2.4 Disease Submodel

The assignment submodel clusters individuals into groups and the allocation variables, $Z_i = c, c = 1, \ldots, C$ indicating the cluster to which individual i is assigned, can be used as categorical variables for predicting the outcome via a regression model. We let θ_{Z_i} measure the influence of Z_i on the outcome (on the logistic scale) and $\boldsymbol{w}_i = (w_{i1}, \ldots, w_{ip})$ denote the confounding variables for individual $i; i = 1, \ldots, n$. The disease submodel, associating group membership and the confounding variables with the outcome, is

$$logit(p_i) = \theta_{Z_i} + \boldsymbol{\beta} \boldsymbol{w}_i, \tag{2.4}$$

where p_i is the conditional probability of the binary disease outcome Y_i being 1 given Z_i and $\boldsymbol{w}_i$, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ denotes the regression parameter coefficients corresponding to the confounding covariates $\boldsymbol{w}_i$. When all the confounders are set to the "reference" values of zero, the left side only includes the intercept term θ_{Z_i}, which can be interpreted as the baseline log odds, p_{Z_i}, of developing disease for an individual in group Z_i; that is, $p_{Z_i} = exp(\theta_{Z_i})/[1 + exp(\theta_{Z_i})]$.

In MCMC simulation, at each iteration, individual covariate profiles are first assigned to clusters, and then each individual is assigned the risk associated with its cluster. The R package `PReMiuM` implements these methods.

Chapter 3

Evaluating Clustering Output

The data-fitting process gives us a large amount of posterior output. In this chapter, we first describe an effective way to use the posterior output to identify the optimal clustering. Next, we describe a way to use the output to evaluate the uncertainty associated with the chosen partition, via model averaging.

3.1 Characterizing the Best Partition

Since the number of clusters and the cluster labels are changing from iteration to iteration, there is no easy way to find an assignment that maximizes the average posterior probability across the iterations. To find a "best", we start by constructing a score matrix at each iteration of the MCMC sampler in the following way: set the element in row i, column j equal to 1 if individuals i and j are assigned to the same cluster, and 0 otherwise. A posterior probability matrix, also known as similarity matrix, $\mathbf{S}$, is estimated by taking the average of the score matrix over the MCMC iterations. Each element of the similarity matrix, S_{ij}, measures the proportion of times individuals i and j are assigned to the same group. The best partition is then chosen from all the partitions generated by the MCMC sampler as the one that minimizes the least-squared distance to the similarity matrix $\mathbf{S}$. This method was introduced by Dahl (2006), but is susceptible to Monte Carlo error since it assumes that one of the observed partitions is the best.

An alternative approach is to apply deterministic clustering methods, such as partitioning around medoids (PAM: Kaufman and Rousseeuw, 2005), to the similarity matrix. The best PAM partition is determined for each fixed number of clusters up to a specified maximum number of clusters. Among the resulting set of clusterings, an optimal partition is chosen by maximizing an associated clustering score such as the silhouette width (see Rousseeuw, 1987).

In practice, these two methods often provide very similar results and both are available in the package `PReMiuM`. However, the PAM approach requires a specification of at least two clusters when calculating the optimal clustering score, so that Dahl's approach may be preferable when the data have a weak structure (i.e. just one cluster).

3.2 Evaluating Uncertainty of Best Partition

Since MCMC samplers provide varying cluster assignments over iterations, it is important to examine whether or not the model clusters the individuals consistently with the optimal partition that we choose. An optimal partition should be accompanied by consistent clustering at each iteration and thus have narrow credible intervals in terms of the baseline risks and profile parameters defined below. The consistency is to a large degree based on the structure of the data. If the data have strong clustering structure, we will get good certainty regarding the clustering parameter estimates. Otherwise, when the data are "noisy" and individuals do not tend to group consistently into the same clusters, the chosen partition will be accompanied by great uncertainty, meaning that the "best partition" is highly haphazard.

We denote by z^{best} the optimal partition. Given the optimal partition, we define the average baseline risks, $\overline{p}_c$, over all the individuals within a particular cluster, c, of z^{best} by

$$\overline{p}_c = \frac{1}{n_c} \sum_{i:z_i^{best}=c} p_{z_i},$$

where z_i is the observed cluster allocation for the i^{th} individual, p_{z_i} is the probability of the outcome in cluster z_i, and n_c denotes the number of individuals assigned to cluster c of the optimal partition, z^{best}. We estimate the posterior mean of the average baseline risks by taking their average over the MCMC iterations. If the posterior distribution of average baseline risks is skewed, we may instead estimate their posterior median by taking their median over the MCMC iterations. Credible intervals can be used to make inference. A consistent partition leads to narrower credible intervals for the average baseline risks. The average of cluster parameters from the assignment submodel, $\phi^j_{z_i}$, can also be computed as

$$\overline{\phi}^j_c = \frac{1}{n_c} \sum_{i:z_i^{best}=c} \phi^j_{z_i}. \tag{3.1}$$

We can estimate the posterior median or mean as well as the credible intervals for the profile parameters from the MCMC output. For example, to estimate the posterior median of $\overline{\phi}^j_c$ we may take the median of $\overline{\phi}^j_c$ over the MCMC iterations.

Given the estimated cluster-specific parameters, we may investigate the associations among covariates by drawing a heatmap of the dissimilarity matrix among covariate profiles. We take the vector of cluster specific parameters $(\overline{\phi}^j_1, \ldots, \overline{\phi}^j_C)$ as the coordinates for the j^{th} covariate and the covariates' dissimilarity is obtained by computing the Euclidean distance between pairs of covariate profiles. Similarly, the associations among individuals may be investigated by plotting the heatmap of the individuals' dissimilarity matrix. As described in section 3.1, a similarity matrix has been constructed at each MCMC iteration, and thus can be used to investigate whether individuals tend to cluster. The R package `PReMiuM` implements a function to produce the heatmap of the individual dissimilarity matrix.

Chapter 4

Data

4.1 Data Generation

We generate a data set of 48 subjects and 70 binary covariates. Among the 70 covariates, $X_1, \ldots, X_{28}$ are based on latent variable Z_1; $X_{29}, \ldots, X_{56}$ are based on latent variable Z_2; $X_{57}. \ldots, X_{66}$ are based on $Z_1 + Z_2$; and $X_{67}, \ldots, X_{70}$ are binary noise. The probability of covariates being a success conditional on the corresponding latent variable is given by

$$Pr(X = 1 \mid Z) = \frac{exp(t + \beta Z)}{1 + exp(t + \beta Z)},$$

where Z is one of Z_1, Z_2 and $Z_1 + Z_2$, as appropriate, and t is the intercept term that ensures $\int_z Pr(X = 1 \mid z) Pr(Z = z) dz = 0.5$. We set $\beta = 1.2$ for both latent variables and $Z_1 + Z_2$.

In this simulation study, we assign the 48 individuals to 4 groups, with each group corresponding to a combination of (Z_1, Z_2):

- individuals 1-12 are in group 1 with $(Z_1, Z_2) = (1, 1)$;
- individuals 13-24 are in group 2 with $(Z_1, Z_2) = (1, 3)$;
- individuals 25-36 are in group 3 with $(Z_1, Z_2) = (3, 1)$; and
- individuals 37-48 are in group 4 with $(Z_1, Z_2) = (3, 3)$.

The four combinations of latent variables corresponding to the assigned groups are shown in Figure 4.1. We assign the binary outcome y_i such that all individuals in

- group 1 have probability 0.5 of being 1;
- group 2 have probability 0.1 of being 1;
- group 3 have probability 0.3 of being 1; and

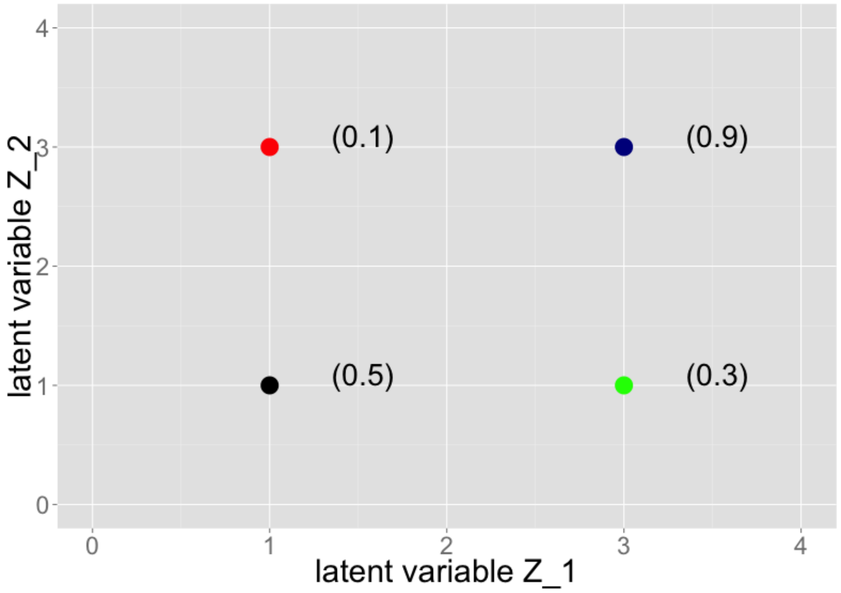


Figure 4.1: Schematic of the groups underlying the simulated data. The points at (1,1), (1,3), (3,1) and (3,3) are the centers of group 1, 2, 3, and 4, respectively. The numbers in parentheses beside the points are the outcome probabilities for the group.

- group 4 have probability 0.9 of being 1.

We generate missing values in the following way:

1. Among the 70 covariates, randomly select 35 covariates denoted by $X_{i_1}, \dots, X_{i_{35}}$, and denote the other 35 covariates by $X_{i_{36}}, \dots, X_{i_{70}}$.

2. For each X_{i_k}; $k = 1, \dots, 35$; generate a random variable u_{i_k} from a uniform distribution on $(0, 1)$, and reassign a value to that covariate:

$$X_{i_k} = \begin{cases} NA & u_{i_k} < 0.3 \\ X_{i_k} & otherwise \end{cases}$$

3. For each X_{i_k}; $k = 36, \dots, 70$, similarly, randomly generate a random variable u_{i_k} from a uniform distribution on $(0, 1)$ and then assign

$$X_{i_k} = \begin{cases} NA & u_{i_k} < 0.1 \\ X_{i_k} & otherwise \end{cases}$$

In the resulting simulated data set, the percentage of missing values for each covariate is below 0.45 and the percentage of missing values for each individual is below 0.35, which are reasonable proportions in the real data sets motivating this report. Missing values are such that a conventional "complete case" analysis would lead to no patients to analyze. For further information regarding the simulated data set, we take a look at the sample mean for each covariate. Before missing values are generated, the average over all individuals for each covariate is around 0.5 (mean 0.51, range 0.35-0.67), as expected. After the missing values are generated, the average remains around 0.5 (mean 0.52, range 0.36-0.68).

4.2 Exploratory Summaries

To get a global view of the covariate patterns, Figure 4.2 shows a grid of colored rectangles with different colors corresponding to the values of covariates, where red corresponds to covariate absent, blue to covariate present and white to missing values. The dendograms added to the left side and to the top demonstrate the cluster membership of patients and covariates, respectively, as determined by agglomerative hierarchical clustering based on Euclidean distance. The dendogram of covariates clusters $X_1, \ldots, X_{28}$, which reflect Z_1. The dendogram also clusters the noise and covariates reflecting $Z_1 + Z_2$ with covariates reflecting Z_2. Individuals cluster together as expected given how the data were simulated. Referring to Figure 4.2, the first 12 individuals tend to have all covariates (except the noise) absent; the second 12 individuals tend to have the Z_1-determined covariates absent, Z_1+Z_2-determined covariates present about half the time and the Z_2-determined covariates present; the third 12 individuals tend to have the Z_2-determined covariates absent, Z_1+Z_2-determined covariates present about half the time and Z_1-determined covariates present; and the last 12 individuals tend to have all covariates present.

Multiple correspondence analysis (MCA) is a data analysis technique for categorical variables, used to detect latent structures in a data set. MCA can be viewed as the counterpart of principle component analysis for categorical data (see, e.g. Le Roux and Rouanet, 2004). The R package `FactoMineR` (Lê et al., 2008) provides the `mca()` function for analysis as well as plots that visually depict the data structure. Figure 4.3 shows the MCA factor map of the simulated data for the first two principal coordinates. The first two principle coordinates can be approximately obtained by appropriately rotating and mirroring the Z_1 and Z_2 coordinates from Figure 4.1. We see that MCA manages to capture the latent data structure and clusters the individuals as expected based on the first two principal coordinates of variation in the covariate data.

Next, we explore how the outcome relates to the covariates using the estimated log odds ratios and approximate 95% confidence intervals. The results are shown in Figure 4.4, where we can see that most of the covariates have confidence intervals centered at zero, though several covariates seem to be shifted away from the dashed line. We also explore the relationship between outcome and groups using the corresponding log odds ratios and approximate 95% confidence intervals, taking the first group as the baseline against which to compare the others. If the true groups were known, their estimated association with outcome (as summarized by the logarithm of the ratio of the odds of outcome in a group relative to the odds in group 1) would be shown in Figure 4.5.

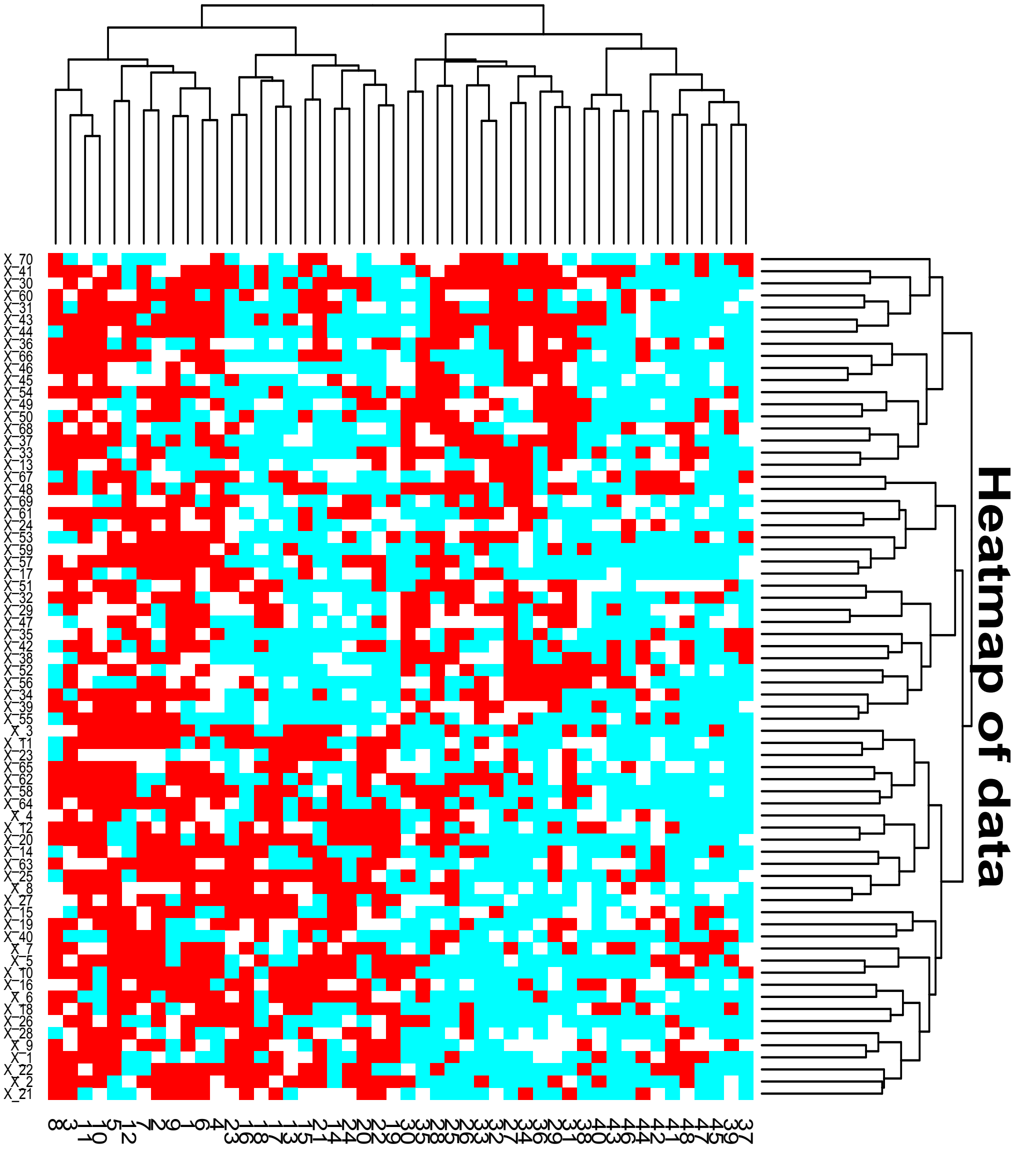


Figure 4.2: Heatmap of 70 covariates (columns) for 48 patients (rows) with dendograms added to the left and to the top; covariate absent=red, present=blue, missing/unknown=white.

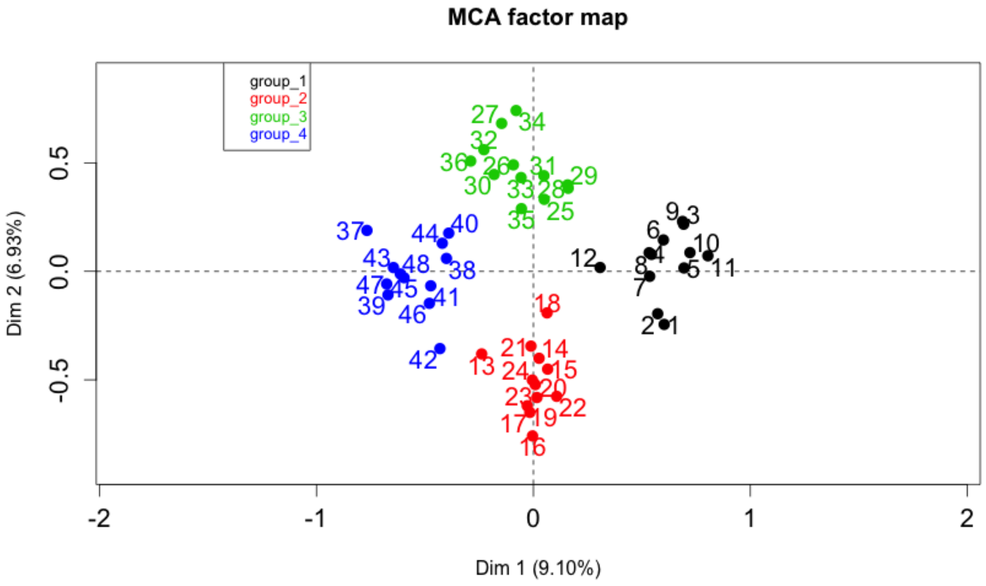


Figure 4.3: Multiple correspondence analysis factor map of the first two principal coordinates showing the latent structure in the data

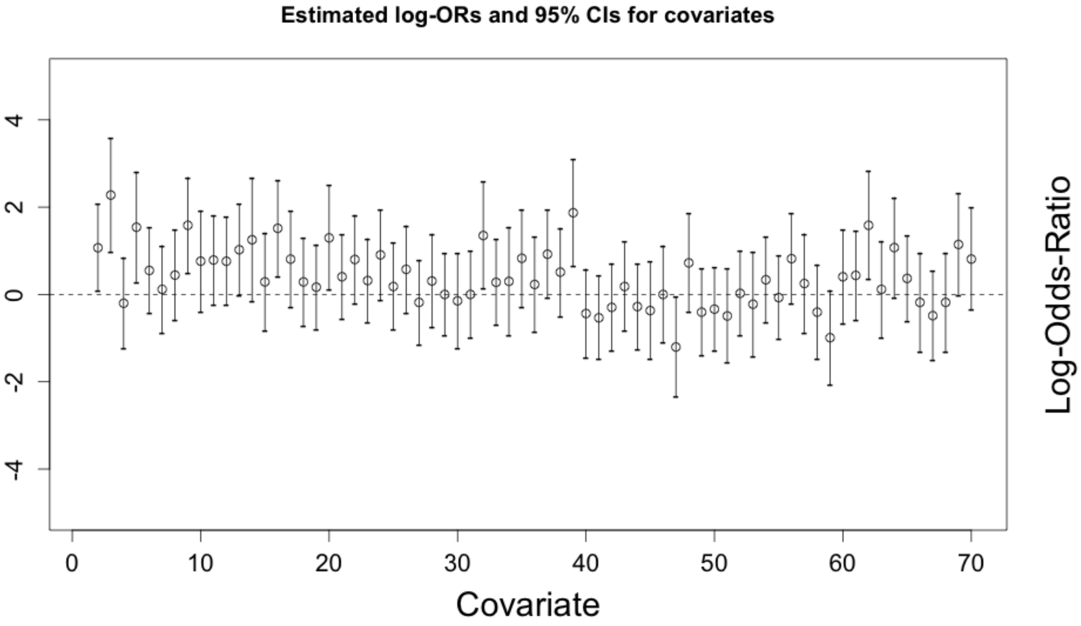


Figure 4.4: Estimated associations between outcome and covariates and their approximate 95% confidence intervals. Associations are estimated from 2*2 contingency tables cross-classifying the covariate and the outcome.

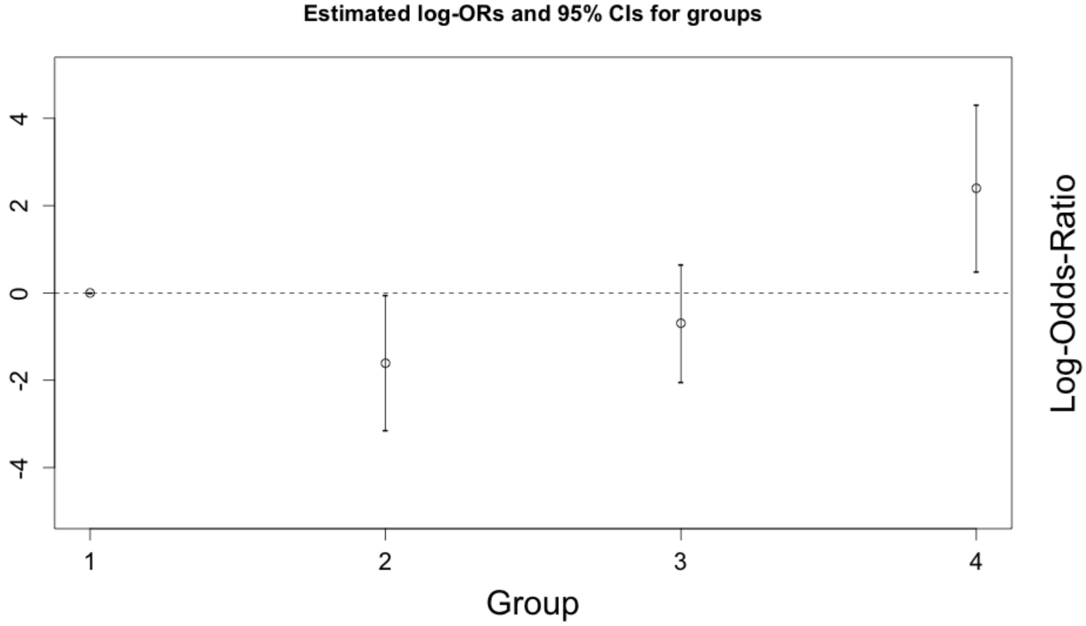


Figure 4.5: Estimated associations between outcome and groups and their approximate 95% confidence intervals. The odds-ratios for all groups are calculated with group 1 as the baseline; hence, the confidence interval for group 1 is not defined.

Chapter 5

Application

5.1 Bayesian Profile Regression Analysis

We use the default values for all hyperparameters suggested by Hastie et al. (2015): in the assignment submodel, we specify a uniform distribution on the interval $(0, 1)$ as the base distribution, G_0, for ϕ_c^j, the probability that binary covariate X_j is one in cluster c. The uniform distribution is equivalent to a symmetric $Dirichlet(1, 1)$ distribution, as described in equation (2.2), and to a $Beta(1, 1)$ base distribution, as shown in Figure 2.1. We specify a $Gamma(2, 1)$ prior for α in the stick-breaking process described in equation (2.3), where a $Gamma(a, b)$ random variable has density function $\frac{b^a}{\Gamma(a)}x^{a-1}e^{-bx}$ for $x > 0$. In the disease submodel referred to equation (2.4), we specify a $t_7(0, 2.5)$ prior for the cluster-specific effect θ_{z_i}, where a $t_\gamma(\mu, \sigma)$ random variable has density function $\frac{\Gamma(\frac{\gamma+1}{2})}{\Gamma(\frac{\gamma}{2})\sqrt{\pi\gamma}\sigma}\left[1 + \frac{1}{\gamma}(\frac{x-\mu}{\sigma})^2\right]^{-\frac{\gamma+1}{2}}$. In fitting the model, we initialized all chains allocating subjects randomly to 10 groups, and ran the chain for 20,000 iterations after a burn-in sample of 300 iterations. Setting the argument `seed` gives the same set of random numbers at each run. We can include the outcome variable when modeling by setting the argument `excludeY=FALSE` or only model the covariates by setting `excludeY=TRUE` in the call to the `profRegr()` function. Including outcome helps provide more information when fitting the assignment submodel, which will be discussed later, and enables the function to provide estimates of cluster-specific disease risks. Bayesian profile mixture models can be fitted as follows.

```
R>library("PReMiuM")
R> runInfoObj <- profRegr(yModel = "Bernoulli", xModel = "Discrete",
+  nSweeps =20000, nBurn = 300, data = DATA[,1:71],
+  output = "output_DATA", covNames = colnames(DATA)[1:70],
+  nClusInit = 10, run = TRUE, seed = 3459, excludeY = FALSE)
R> dissimObj<-calcDissimilarityMatrix(runInfoObj)
R> clusObj<-calcOptimalClustering(dissimObj)
R> riskProfileObj<-calcAvgRiskAndProfile(clusObj)
```

Table 5.1 and Table 5.2 show the confusion matrices with outcome excluded and included, respectively. We can see that, when the outcomes are excluded, 2 individuals from group 3 are extracted to form a small group, whereas including the outcome gives us perfect clustering. These results demonstrate that including outcome can significantly improve the clustering.

		true cluster membership			
		1	2	3	4
Bayesian profile regression clustering	1	12	0	0	0
	2	0	12	0	0
	3	0	0	10	0
	4	0	0	2	0
	5	0	0	0	12

Table 5.1: Bayesian profile clustering based on the covariates only

		true cluster membership			
		1	2	3	4
Bayesian profile regression clustering	1	12	0	0	0
	2	0	12	0	0
	3	0	0	12	0
	4	0	0	0	12

Table 5.2: Bayesian profile clustering based on the covariates and outcome

The object `riskProfileObj` is a list including the empirical mean of the outcome for each cluster, the cluster sizes, the cluster-specific covariate profile and so forth. Table 5.3 shows an excerpt of the estimates of cluster parameters; a full report of the estimated cluster parameters can be found in the Appendix, in Table A.1. For the j^{th} covariate in the c^{th} cluster, the MCMC sampler returns dependent realizations of the profile parameter $\overline{\phi}_c^j$ described in equation (3.1), which are sampled from the posterior distribution. The posterior mean of $\overline{\phi}_c^j$ can be estimated by taking the mean over all "sweeps" or MCMC iterations. Likewise, credible intervals can be estimated by taking the appropriate quantiles of the $\overline{\phi}_c^j$'s over all the sweeps. Since the posterior distribution of the cluster specific parameters over the MCMC samplers is symmetric (results not shown), the point estimates using the mean and the median are almost identical. Most of the true values of cluster specific parameters are contained in the corresponding credible intervals, though several credible intervals fail to cover the true value, which is assumed to be the result of high percentages of missing values or randomness. We also assume that some credible intervals are wide due to much missing information. In our data simulation, the strength of association between covariates and latent variables, which is denoted by β, determines to what degree the simulated values are consistent with the latent variable values. That is, given the same proportion of missing covariate data, we expect the credible intervals to be less variable with a higher value of β. Generally, the estimates capture the data structure, and are effective for describing the clusters. The empirical mean of the cluster-specific risks (`risk`) the cluster size (`clustersize`) and cluster-specific covariate profile parameters (`phi`) can be obtained as follows:

```
R> risk <- riskProfileObj$empiricals
R> clustersize <- riskProfileObj$riskProfClusObj$clusterSizes
R> phi <- riskProfileObj$profile
```

	Group 1	**Group 2**	**Group 3**	**Group 4**
p_c (true)	0.50	0.10	0.30	0.90
point estimate	0.50	0.17	0.33	0.92
95% C.I.	(0.25,0.75)	(0.04,0.43)	(0.12,0.61)	(0.68,0.99)
$\overline{\phi}_c^1$ (true)	0.23	0.23	0.77	0.77
point estimate	0.50	0.33	0.83	0.54
95% C.I.	(0.23,0.77)	(0.11,0.61)	(0.58,0.98)	(0.27,0.79)
$\vdots$				
$\overline{\phi}_c^{28}$ (true)	0.23	0.23	0.77	0.77
point estimate	0.46	0.20	0.77	0.82
95% C.I.	(0.19,0.74)	(0.03,0.49)	(0.47,0.97)	(0.55,0.97)
$\overline{\phi}_c^{29}$ (true)	0.23	0.77	0.23	0.77
point estimate	0.34	0.62	0.17	0.78
95% C.I.	(0.09,0.65)	(0.29,0.90)	(0.02,0.42)	(0.47,0.97)
$\vdots$				
$\overline{\phi}_c^{56}$ (true)	0.23	0.77	0.23	0.77
point estimate	0.08	0.61	0.66	0.92
95% C.I.	(0.21,0.79)	(0.55,0.97)	(0.07,0.55)	(0.39,0.89)
$\overline{\phi}_c^{57}$ (true)	0.08	0.5	0.5	0.92
point estimate	0.15	0.46	0.50	0.83
95% C.I.	(0.00,0.26)	(0.35,0.85)	(0.39,0.89)	(0.73,1.00)
$\vdots$				
$\overline{\phi}_c^{70}$ (true)	0.5	0.5	0.5	0.5
point estimate	0.42	0.56	0.40	0.50
95% C.I.	(0.21,0.72)	(0.24,0.85)	(0.14,0.70)	(0.23,0.77)

Table 5.3: An excerpt of estimates of cluster-specific parameters. The cluster-specific profile parameters, $\overline{\phi_c}$, are defined in equation (3.1).

The function `plotRiskProfile()` offers a helpful and intuitive way to investigate the posterior distribution of the covariates and probabilities of the response by showing their box-plots for each cluster. It would be impractical to draw box-plots for as many as 70 covariates, and so we choose a subset of the covariates and use the function `plotRiskProfile()` to show the box-plots in Figure 5.1. The covariates X_1, X_{28} (related to latent variable Z_1), X_{29}, X_{56} (related to Z_2), X_{57} (related to Z_1+Z_2) and X_{67} (binary noise) are chosen to represent different latent variables, and the estimates of their cluster-specific profile parameters are reported in Table 5.3. From Figure 5.1, we can see that the risks for each cluster agree closely with the probabilities of outcome set in the simulation. The function plots estimated cluster-specific profile parameters, $\overline{\phi}_c^j$, for $j \in \{1, 28, 29, 56, 57, 67\}$, with the average value over clusters drawn as a horizontal line. The red-coloured boxes indicate that the 90% credible intervals for the cluster-specific profile parameter are above the average over clusters, the green-coloured boxes indicate that the 90% credible intervals include the average, and the blue-coloured boxes indicate that the 90% credible intervals are below the average. We expect X_1 and X_{28} to have the same trends in their estimated profile parameters across clusters, and also, X_{29} and X_{56}. As expected, X_1 and X_{28} tend to have small values of the cluster-specific profile parameter in cluster 1 and 2 but relatively large values in cluster 3 and 4, since the latent variable Z_1 equals 1 in the first two clusters and equals 3 in the last two clusters. Though the estimated profile parameter for X_1 falls below the mean in cluster 4, we assume it to be the result of randomness involved in such a small data set. Similarly, X_{29} and X_{56} have small profile parameter estimates in cluster 1 and 3 where Z_2 equals 1, and large values in cluster 2 and 4, where Z_2 equals 3. X_{57} has the lowest estimated profile parameter in cluster 1 where $Z_1 + Z_2$ equals 2, an estimated value around the average in clusters 2 and 3 where $Z_1 + Z_2$ equals 4, and a higher value in cluster 4 where $Z_1 + Z_2$ equals 6. The noise covariate, X_{67}, has an estimated profile parameter around 0.5 for all the clusters, as expected.

To gain insight into the latent structure underlying the covariates, we draw a heatmap of the dissimilarity matrix among covariate profiles. We take the vector of cluster specific parameters $(\overline{\phi}_1^j, \overline{\phi}_2^j, \overline{\phi}_3^j, \overline{\phi}_4^j)$ as the coordinates for the j^{th} covariate and compute the Euclidean distance between pairs of covariate profiles to obtain the elements of the covariate dissimilarity matrix. Figure 5.2 shows the resulting heatmap. The covariates are generally grouped into three clusters: the first 28 covariates reflecting latent variable Z_1 are grouped into one cluster on the bottom right, the second 28 covariates reflecting latent variable Z_2 are grouped into one cluster at the top left, and the smaller cluster in the center contains most $Z_1 + Z_2$-determined covariates, which tend to be more correlated with Z_1-determined covariates in these data. We can also take a look at the heatmap of the patients' dissimilarity matrix to get a sense of individual clustering certainty. The package offers a function `heatDissMat()` to draw a heat map of the patients using the object `dissimObj` returned by function `calcDissimilarityMatrix()`, which is used for calculating the optimal partition. The shade corresponds to the degree of individuals' similarity to each other based on

Bayesian profile regression. Figure 5.3 demonstrates that the simulated individuals have a strong signal of clustering.

The function `globalParsTrace()` is implemented in the package to provide a basic diagnostic plot of the trace of some global parameters such as the number of clusters and the concentration parameter α in the stick-breaking. For more convergence diagnostics, the R package `coda` (Plummer et al., 2006) is helpful. Although there are no parameters that can be used to definitively demonstrate convergence of the algorithm, there are methods to investigate whether there is evidence against convergence. The following code can be used to reproduce the trace plot and autocorrelation plot for both the number of clusters and the parameter α in Figure 5.4 and Figure 5.5. Referring to Figure 5.4, we see that our chains do not seem to get stuck in certain areas for α and the number of clusters, when the initial number of clusters is set to 10. Referring to Figure 5.5, the autocorrelations for both global parameters are relatively low for our chain, which indicate no evidence against convergence. Another way to investigate convergence is to monitor the distribution of α across multiple runs initialised with different numbers of clusters (see Hastie et al., 2014). We obtain the posterior distribution of α for different numbers of initial clusters with three repetitions per initialisation and 20,000 sweeps after a burn-in of 300 samples. Figure 5.6 shows the boxplot of the posterior distribution of α as a function of the initial numbers of clusters. We see that the posterior distribution of α stabilises for all these initial number of clusters, which suggests convergence. The cluster specific parameters cannot be plotted as easily due to label switching across the iterations of the MCMC sampler and so assessing their convergence is difficult. Hastie et al. (2014) introduce the marginal model posterior, defined as $pr(Z \mid \boldsymbol{X})$, where Z represents the cluster membership and $\boldsymbol{X}$ the covariate vector, as a tool to assess convergence for Dirichlet process mixtures. However, in the `PReMiuM` R package, no missing value handling technique has been implemented for the marginal model posterior which prevents its use if missing values are present. The following R code was used to generate the trace plot of the number of clusters and parameters shown in Figure 5.4 and Figure 5.5.

```
R> par(mfrow=c(1,2))
R> runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
+  nSweeps=20000, nBurn=300, data=DATA[,1:71],
+  covNames=colnames(DATA)[1:70], output="convergence_DATA",
+  nClusInit=10, run=TRUE, seed=3459, excludeY=FALSE, reportBurnIn=TRUE)
R> globalParsTrace(runInfoObj, parameters="nClusters", plotBurnIn=TRUE)
R> globalParsTrace(runInfoObj, parameters="alpha", plotBurnIn=TRUE)

R> library("coda")
R> nclusterchain<-mcmc(read.table("convergence_DATA_nClusters.txt"))
R> autocorr.plot(nclusterchain)
```

```
R> alphachain<-mcmc(read.table("convergence_DATA_alpha.txt"))
R> autocorr.plot(alphachain)
```

5.2 Comparison with LASSO

To compare the performance of Bayesian profile regression to LASSO on the simulated data, we replace the missing values in the LASSO analysis with their true values before they are taken away. The LASSO thus has the benefit of perfectly imputed data, in contrast to Bayesian profile regression which must deal with the missing values. Using perfectly imputed data for LASSO but not for Bayesian profile regression gives maximum advantage to LASSO in the comparison. If Bayesian profile regression beats LASSO under these circumstances, we are confident to say that Bayesian profile regression can perform better than LASSO in real life situations, where there is no way to obtain the true values. We choose two commonly used values of the LASSO regularization parameter λ: one gives the most regularized model such that the cross-validated error is within one standard error of the minimum (denoted by LASSO/1se for short) and the other gives minimum mean cross-validated error (LASSO/min for short). LASSO/1SE selects none of these covariates as important factors, and thus makes uniform predictions to all individuals. LASSO/min selects only X_{20} as important, which is related to Z_1. However, according to our data generation scheme, the two latent variables, Z_1 and Z_2, have an interaction effect and thus should both have an impact on the outcome. Clearly, LASSO is not adequately capturing the structure in the data.

We are able to obtain the information about risks as well as the cluster membership from Bayesian profile regression. Bayesian profile regression makes risk predictions that agree closely to the probabilities of outcome assigned to each group: individuals 1-12 (group 1) all have predicted probabilities around 0.50; individuals 13-24 (group 2) are mostly predicted to be around 0.20; individuals 25-36 (group 3) are predicted to be in the range of 0.30 to 0.40; and individuals 37-48 (group 4) are all predicted to be 0.90. We see that individuals in group 2 and group 3 have slightly higher predicted probabilities of outcome than the expected values of 0.1 and 0.3, respectively, and we assume it to be the result of higher observed probability of outcome than expected when randomness comes into the simulated data. The observed and predicted outcome are shown in Table 5.4. Bayesian profile regression appears to do a better job of capturing the structure of the outcome and makes more precise predictions. To compute the mean squared prediction error (MSPE), we generate a test set, which has 8 individuals within each of the four groups and 70 binary covariates using the same scheme as the training set. Bayesian profile regression and both LASSO solutions are applied to the test set to make predictions. The training set's mean squared error (MSE) as well as the MSPE for each method is shown in Table 5.5. Both the MSE and MSPE are computed on the probabilities of outcome instead of the observed outcome. Bayesian profile regression gives the smallest MSE and MSPE on both training set and test set. In addition, Bayesian profile regression even gives slightly smaller error on the test set than on the training set, which demonstrates that it prevents the problem of overfitting. In summary, Bayesian profile regression seems to be better suited than LASSO for our simulated data set.

ID	**observed**	**true**	**predicted outcome**		
	outcome	**probability**	Bayesian profile	LASSO/min	LASSO/1SE
1	1	0.50	0.49	0.52	0.48
2	1	0.50	0.49	0.44	0.48
3	0	0.50	0.49	0.44	0.48
4	1	0.50	0.49	0.44	0.48
5	1	0.50	0.49	0.52	0.48
6	1	0.50	0.49	0.44	0.48
7	0	0.50	0.49	0.44	0.48
8	0	0.50	0.49	0.44	0.48
9	0	0.50	0.49	0.44	0.48
10	0	0.50	0.49	0.44	0.48
11	0	0.50	0.49	0.44	0.48
12	1	0.50	0.52	0.52	0.48
13	0	0.10	0.20	0.44	0.48
14	0	0.10	0.14	0.44	0.48
15	0	0.10	0.15	0.44	0.48
16	0	0.10	0.20	0.44	0.48
17	0	0.10	0.20	0.44	0.48
18	0	0.10	0.20	0.44	0.48
19	0	0.10	0.20	0.44	0.48
20	1	0.10	0.20	0.44	0.48
21	0	0.10	0.14	0.52	0.48
22	0	0.10	0.20	0.44	0.48
23	0	0.10	0.20	0.52	0.48
24	1	0.10	0.21	0.44	0.48
25	0	0.30	0.32	0.44	0.48
26	1	0.30	0.37	0.52	0.48
27	0	0.30	0.37	0.52	0.48
28	0	0.30	0.36	0.44	0.48
29	0	0.30	0.37	0.52	0.48
30	0	0.30	0.37	0.52	0.48

Continued on next page

Table 5.4 – continued from Table 5.4

ID	**observed**	**true**	**predicted outcome**		
	outcome	**probability**	Bayesian profile	LASSO/min	LASSO/1SE
31	1	0.30	0.37	0.52	0.48
32	0	0.30	0.37	0.52	0.48
33	1	0.30	0.37	0.52	0.48
34	1	0.30	0.37	0.52	0.48
35	0	0.30	0.31	0.44	0.48
36	0	0.30	0.37	0.52	0.48
37	1	0.90	0.90	0.52	0.48
38	0	0.90	0.90	0.52	0.48
39	1	0.90	0.90	0.52	0.48
40	1	0.90	0.90	0.52	0.48
41	1	0.90	0.90	0.52	0.48
42	1	0.90	0.90	0.52	0.48
43	1	0.90	0.90	0.52	0.48
44	1	0.90	0.90	0.52	0.48
45	1	0.90	0.90	0.52	0.48
46	1	0.90	0.90	0.52	0.48
47	0	0.90	0.90	0.52	0.48
48	1	0.90	0.90	0.52	0.48

Table 5.4: Observed and predicted outcome by different methods.

	Bayesian profile regression	LASSO/min	LASSO/1SE
MSE	0.003	0.078	0.088
MSPE	0.002	0.073	0.088

Table 5.5: MSE and MSPE by different methods.

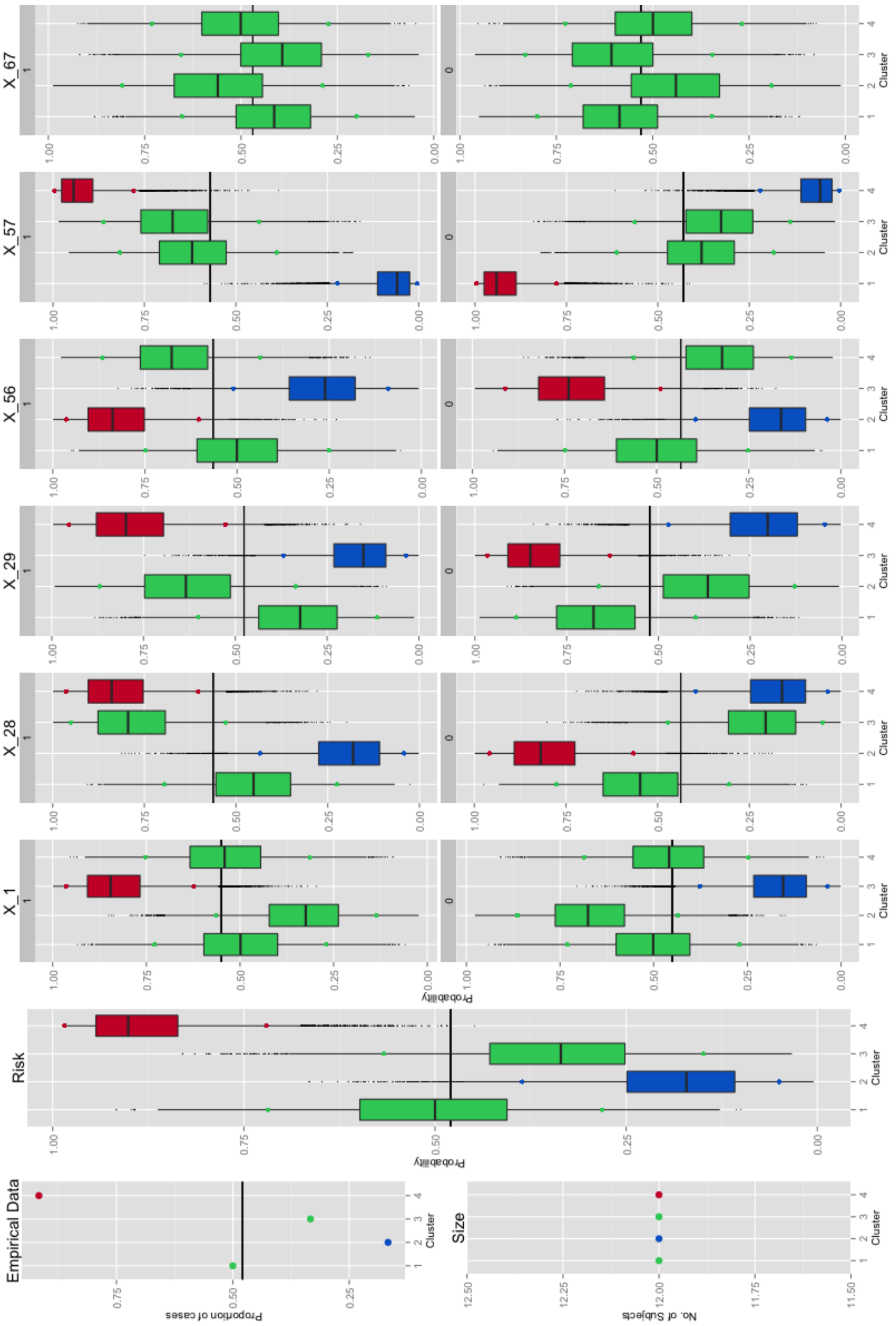


Figure 5.1: Posterior distributions of the parameters for the representative clustering. The red-coloured boxes indicate that the 90% credible intervals for the cluster-specific profile parameter are above the average over clusters, the green-coloured boxes indicate that the intervals include the average, and the blue-coloured boxes indicate that the intervals are below the average.

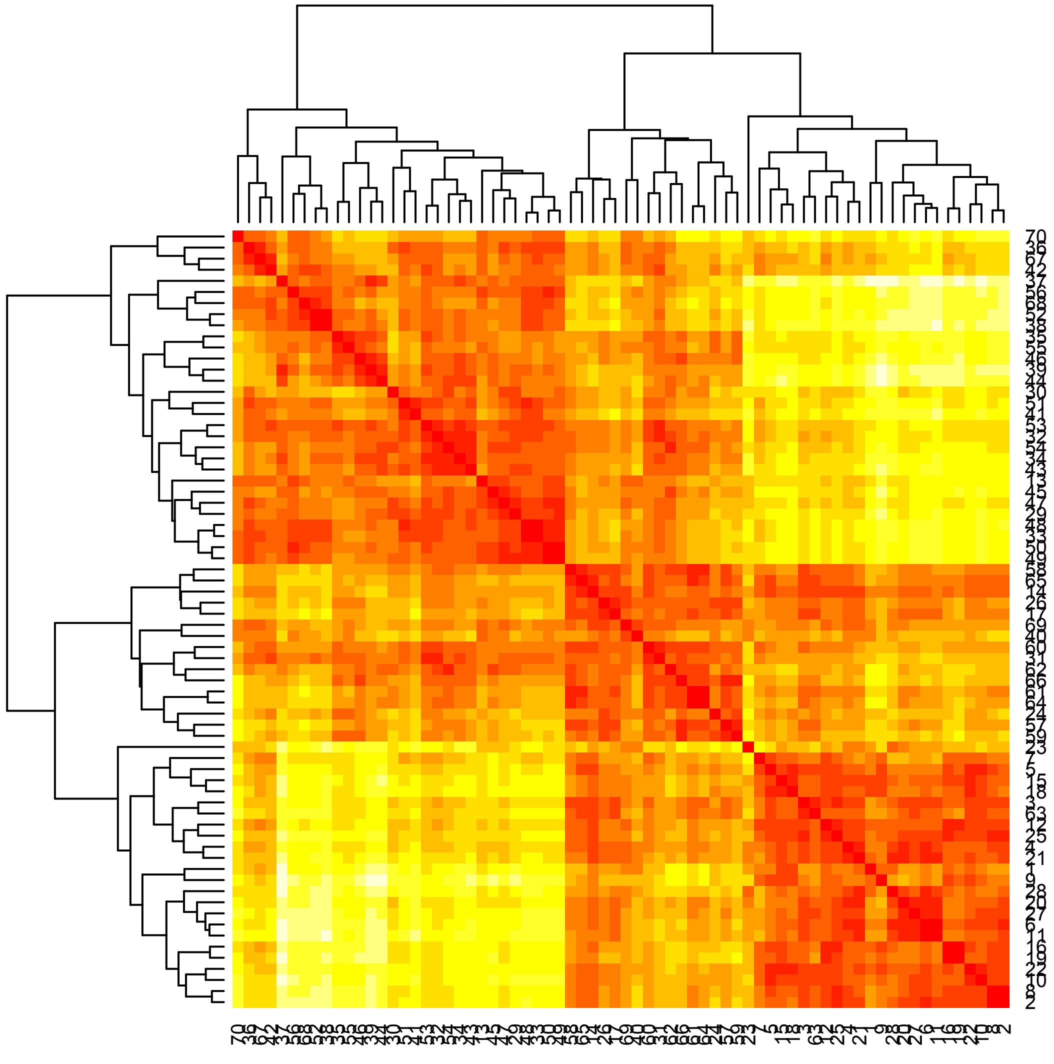


Figure 5.2: Heatmap of the covariates' dissimilarity matrix

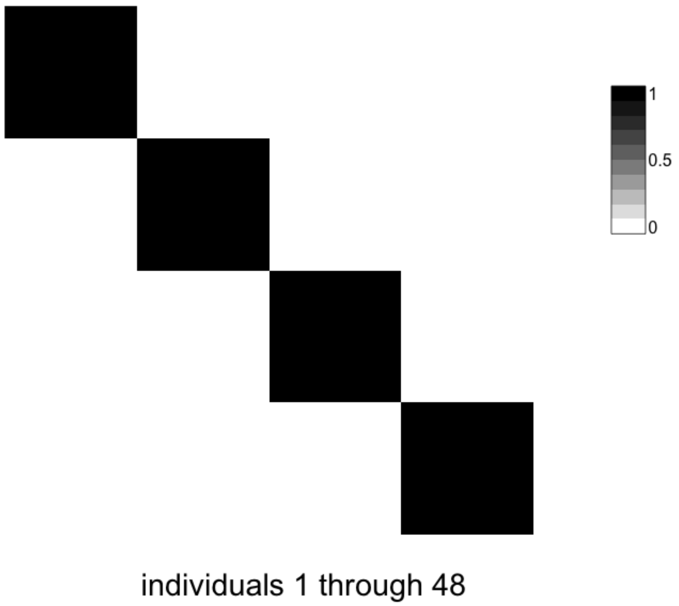


Figure 5.3: Heatmap of the patients' dissimilarity matrix

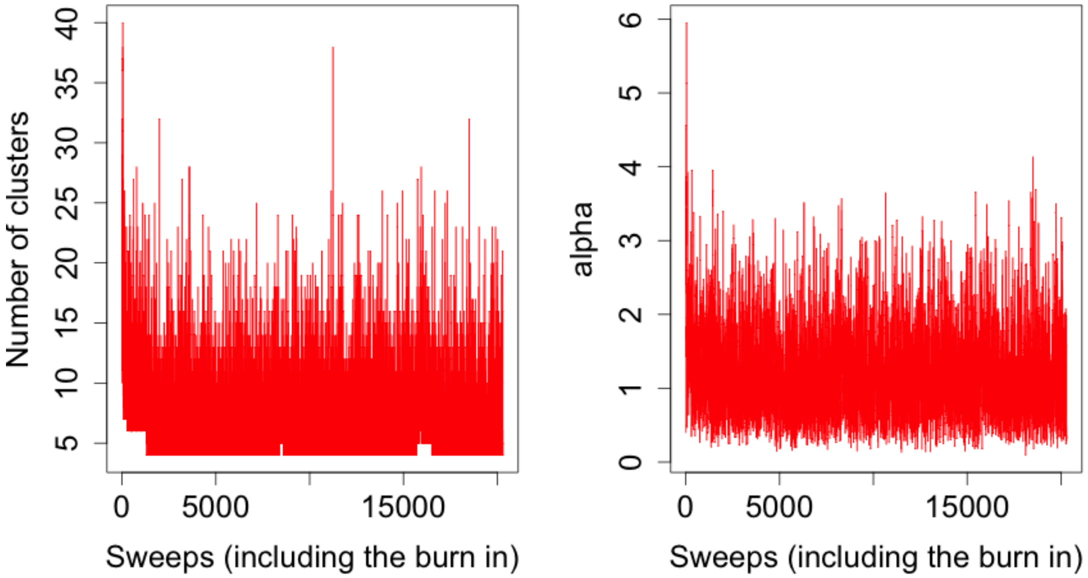
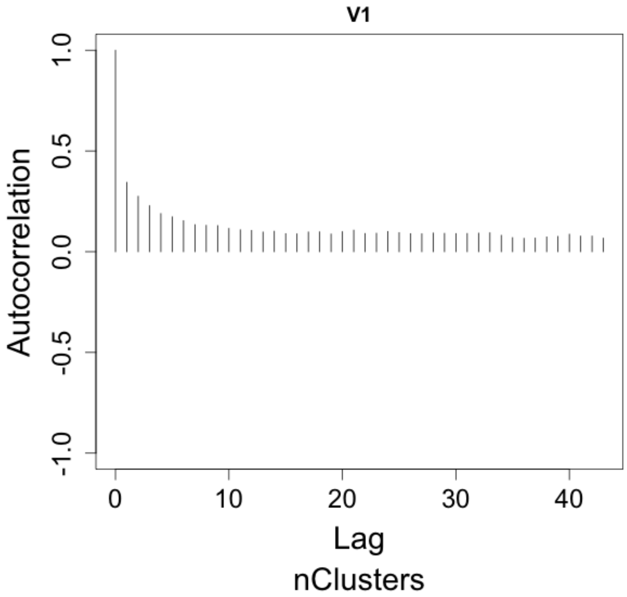


Figure 5.4: The trace of the number of clusters and parameters α through the iterations of the MCMC sampler.



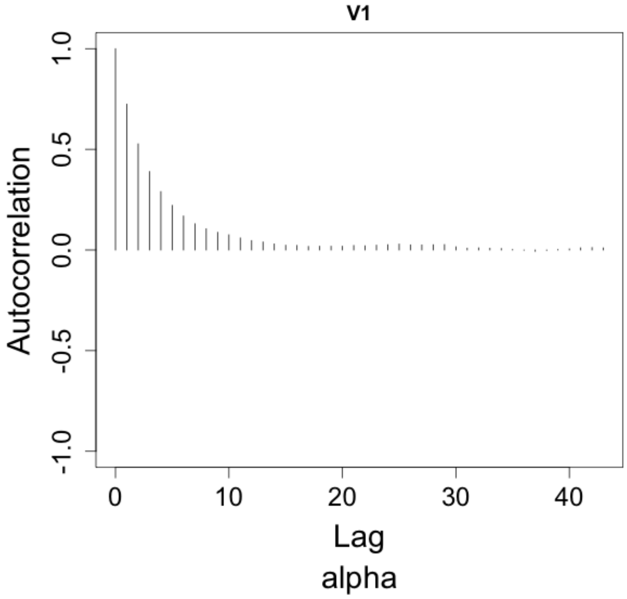


Figure 5.5: Autocorrelation plot of α obtained from `coda`.

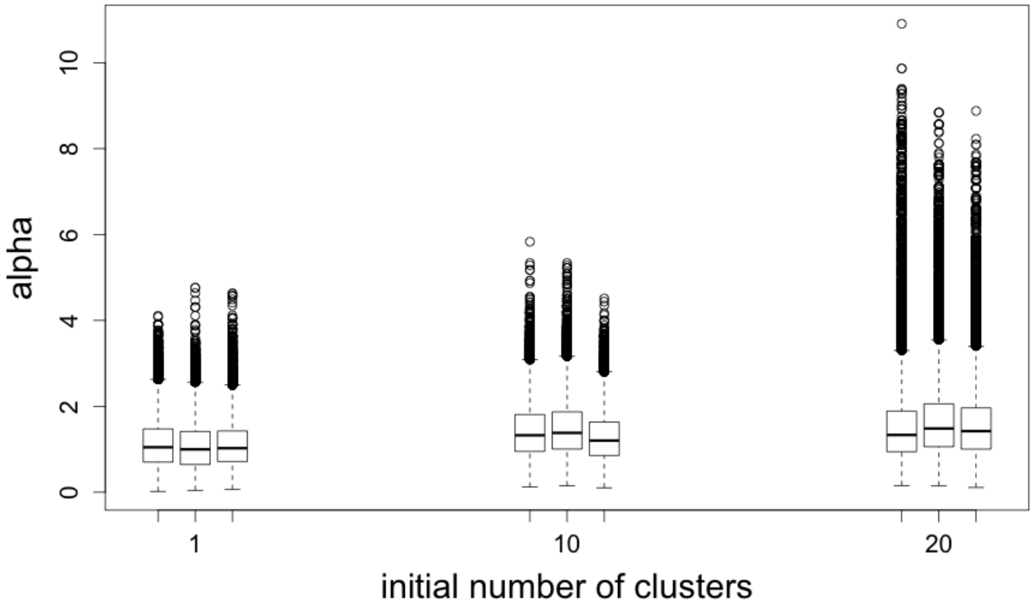


Figure 5.6: The posterior distribution of α with different initial numbers of clusters.

Chapter 6

Discussion

The database that motivates this report collects clinical questionnaire data on pediatric vasculitis patients across the world with the aim of developing better tools for disease classification, and assessment of disease activity and damage. There are two vasculitis subtypes that are of particular interest for classification. Debate among vasculitis researchers includes whether these two subtypes are really separate or one type along a continuum. A major challenge in any pediatric vasculitis research is the disease rarity. Disease incidence is estimated to be 23 in 100,000 in children and vasculitis is estimated to contribute only 2-10% of all conditions evaluated in pediatric rheumatology clinics (Weiss, 2012). In addition, vasculitis is not a single disease but a group of complex conditions which can affect any organ. Thus, the incidence of any one of these subtypes is even lower and for some it is extremely low. The investigators are trying to increase the number of patients by global recruitment of incident patients and inclusion of retrospective patients with some information available in archives. However, this data collection strategy leads to challenges with missing data because different centers (countries) have different diagnostic procedures and the types of tests/evaluations done also change over time. As a result, every covariate and every patient has missing information. In addition, extracting information on retrospective patients and merging it with the current questionnaire data, as well as cross-checking and fixing any inconsistencies in the responses to items in the current questionnaire is labour-intensive and time-consuming. Another challenge is that since the database is based on comprehensive questionnaire data about clinical and lab investigation covariates in patients, many of the variables are correlated. For example, there may be a number of variables that all refer to different tests done to determine kidney function.

In this report, we have focused on understanding Bayesian profile regression. In addition to Bayesian profile regression, another method to identify the unobserved cluster membership is latent class analysis or LCA (see, e.g. Lazarsfeld and Henry, 1968 and McCutcheon, 1987). LCA relates a set of observed covariates (discrete and/or continuous) to a set of latent classes which are discrete. A class is characterized by a “profile”; that is, a pattern of conditional probabilities that indicate the chance that variables take on certain values. LCA attempts to detect the presence of latent classes in the covariates. Thus, LCA can be used to group subjects in clinical observational studies much like profile regression. However, one unresolved issue in the application of LCA is that there is not one commonly accepted statistical indicator for deciding the number of classes in advance. A traditionally used criterion for determining the number of classes is the Bayesian Information Criteria (BIC). Nylund et al. (2007) evaluated the ability of likelihood-based tests and various information criteria (ICs) to correctly identify the number of classes and found that bootstrap likelihood ratio tests perform the best. A version of LCA suitable for continuous normally distributed variables is called 'latent profile analysis' (Lazarsfeld and Henry, 1968), and is based on normal mixture modeling. In R, normal mixture models may be fitted using the `mclust` package (Fraley, 1999). The implementation of LCA with binary covariates is available in the R package `poLCA` (Linzer and Lewis, 2011; R Core Team, 2012), which can also incorporate polytomous categorical covariates. The software 'Mplus' is a proprietary statistical package for the analysis of latent variables and can handle a combination of categorical and continuous variables. Latent class analysis may be extended to incorporate a disease submodel (see, e.g., Wang, 2015 for a special case). However, to our knowledge, no general software implementing these extensions is available.

Both Bayesian profile regression and extended versions of latent class analysis find clusters in the covariates and link cluster membership to the outcome. When group structures underlie the data, these approaches are expected to perform better than standard regression approaches which do not model the latent groups. In contrast to latent class analysis, Bayesian profile regression does not require the number of classes to be determined in advance. The number of latent classes is a parameter of intrinsic interest that we wish to learn from the data. Therefore, not having to specify this parameter in advance is an advantage for us. Furthermore, profile regression embraces the presence of far more covariates than patients, which is a problem for standard regression-based approaches. Additional covariates enable better resolution of the latent groups in the profile regression model. Missing covariate values are also easily dealt with by the profile regression approach, which automatically imputes them as part of the fitting process. For our simulated data, this method did a better job of making predictions and capturing the data structure than LASSO which is known as an effective tool for prediction and variable selection. Though Bayesian profile regression is recently developed, the method as well as the R package is worthy of consideration.

In addition to clustering and risk estimation, we are also interested in which covariates affect the clustering and risks. As mentioned above, the database has a large number of covariates that are potentially correlated and may be unrelated to the outcome. We may waste time by measuring redundant predictors that provide no additional information about the clustering or outcome. Unnecessary predictors also decrease the efficiency of estimation. In future work, we plan to investigate how variable selection works in Bayesian profile regression and use it to exclude unimportant covariates. The aim is to get a reduced and therefore more cost-effective set of predictors that will improve the prediction performance of the Bayesian profile regression model. Ultimately, we plan to apply Bayesian profile regression to data sets extracted from the vasculitis database to understand whether patients' clinical characteristics at diagnosis are correlated with treatment outcomes, such as remission rates one year after diagnosis.

Bibliography

Antoniak, C. E. (1974). Mixture of Dirichlet process with applications to Bayesian nonparametric problems. *Annals of Statistics*, 273(5281):1152–1174.

Blei, D. M. and Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143.

Dahl, D. (2006). *Model-based clustering for expression data via a Dirichlet process mixture model, in Bayesian inference for gene expression and proteomics.* Cambridge: Cambridge University Press.

Fraley, C. (1999). Mclust: software for model-based cluster analysis. *Journal of Classification*, 12:297–306.

Hartigan, J. and Wong, M. (1979). A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108.

Hastie, D. I., Liverani, S., and Richardsom, R. (2014). Sampling from Dirichlet process mixture models with unknown concentration parameter: mixing issues in large data implementations. *Statistics and Computing*, 25(5):1023–1037.

Hastie, D. I., Liverani, S., and Richardsom, R. (2015). PReMiuM: Dirichlet process Bayesian clustering, profile regression.

Kaufman, L. and Rousseeuw, P. J. (2005). *Finding groups in data: an introduction to cluster analysis.* Wiley Series in Probability and Mathematical Statistics. Wiley-Interscience, 1 edition.

Lazarsfeld, P. F. and Henry, N. (1968). *Latent structure analysis.* Boston, MA: Houghton Mifflin.

Lê, S., Josse, J., and Husson, F. (2008). FactoMineR: an R package for multivariate analysis. *Journal of Statistical Software*, 25(1).

Le Roux, B. and Rouanet, H. (2004). *Geometric data analysis, from correspondence analysis to structured data analysis.* Kluwer Academic Publishers.

Linzer, D. A. and Lewis, J. B. (2011). poLCA: an R package for polytomous variable latent class analysis. *Journal of Statistical Software*, 42(10).

McCutcheon, A. L. (1987). *Latent class analysis*, volume 64 of *Latent Class Analysis.* Sage.

Molitor, J., Papathomas, M., Jerrett, M., and Richardson, S. (2010). Bayesian profile regression with an application to the National survey of children's health. *Biostatistics*, 11(3):484–498.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265.

Nylund, K. L., Asparouhov, T., and Muthén, O. B. (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling: a Monte Carlo simulation study. *Structural Equation Modeling*, 14(4):535–569.

Ohlsson, D., Sharples, L., and Spiegelhalter, D. (2007). Flexible random-effects models using Bayesian semi-parametric models: applications to institutional comparisons. *Statistics in Medicine*, 26:721–728.

Patterson, B. H., Dayton, C. M., and Graubard, B. I. (2002). Latent class analysis of complex sample survey data: application to dietary data. *Journal of the American Statistical Association*, 97(459):721–728.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). **coda**: convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11.

Rousseeuw, P. J. (1987). Sihouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, 20:53–65.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(1994):639–650.

Shahbaba, B. and Neal, R. M. (2009). Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research*, 10:1829–1850.

Wang, H. (2015). Statistical inference ender latent class models, with application to risk assessment in cancer survivorship studies. Unpublished PhD thesis, Simon Fraser University.

Weiss, P. F. (2012). Pediatric vasculitis. *Pediatric Clinics North America*, 59(2):407–423.

Appendix A

Table of Cluster-specific Parameter Estimates

	GRP1	**GRP2**	**GRP3**	**GRP4**
size	12	12	12	12
p_k (true)	0.50	0.10	0.30	0.90
Mean	0.50	0.17	0.33	0.92
95% C.I.	(0.25,0.75)	(0.04,0.43)	(0.12,0.61)	(0.68,0.99)
$\overline{\phi}_c^1$ (true)	0.21	0.21	0.79	0.79
Mean	0.50	0.33	0.83	0.54
95% C.I.	(0.23,0.77)	(0.11,0.61)	(0.58,0.98)	(0.27,0.79)
$\overline{\phi}_c^2$ (true)	0.21	0.21	0.79	0.79
Mean	0.17	0.15	0.69	0.67
95% C.I.	(0.02,0.42)	(0.02,0.38)	(0.43,0.90)	(0.51,0.95)
$\overline{\phi}_c^3$ (true)	0.21	0.21	0.79	0.79
Mean	0.17	0.33	0.71	0.79
95% C.I.	(0.02,0.41)	(0.11,0.61)	(0.46,0.91)	(0.55,0.95)
$\overline{\phi}_c^4$ (true)	0.21	0.21	0.79	0.79
Mean	0.33	0.29	0.67	0.80
95% C.I.	(0.09,0.65)	(0.09,0.54)	(0.39,0.89)	(0.52,0.97)
$\overline{\phi}_c^5$ (true)	0.21	0.21	0.79	0.79
Mean	0.16	0.38	0.77	0.58
95% C.I.	(0.02,0.39)	(0.15,0.65)	(0.51,0.94)	(0.31,0.83)
$\overline{\phi}_c^6$ (true)	0.21	0.21	0.79	0.79
Mean	0.29	0.18	0.70	0.71
95% C.I.	(0.09,0.54)	(0.03,0.45)	(0.54,0.95)	(0.59,0.98)
$\overline{\phi}_c^7$ (true)	0.21	0.21	0.79	0.79
Mean	0.17	0.40	0.64	0.45
95% C.I.	(0.02,0.41)	(0.14,0.70)	(0.35,0.88)	(0.18,0.74)
$\overline{\phi}_c^8$ (true)	0.21	0.21	0.79	0.79
Mean	0.13	0.18	0.70	0.71

Continued on next page

Table A.1 – continued from Table A.1

	GRP1	**GRP2**	**GRP3**	**GRP4**
95% C.I.	(0.00,0.41)	(0.03,0.44)	(0.40,0.92)	(0.35,0.96)
$\overline{\phi}_c^{9}$ (true)	0.21	0.21	0.79	0.79
Mean	0.27	0.27	0.87	0.43
95% C.I.	(0.07,0.55)	(0.07,0.56)	(0.59,1.00)	(0.12,0.77)
$\overline{\phi}_c^{10}$ (true)	0.21	0.21	0.79	0.79
Mean	0.14	0.29	0.85	0.69
95% C.I.	(0.02,0.36)	(0.09,0.53)	(0.62,0.98)	(0.43,0.90)
$\overline{\phi}_c^{11}$ (true)	0.21	0.21	0.79	0.79
Mean	0.29	0.22	0.82	0.90
95% C.I.	(0.09,0.54)	(0.05,0.45)	(0.56,0.97)	(0.67,1.00)
$\overline{\phi}_c^{12}$ (true)	0.21	0.21	0.79	0.79
Mean	0.27	0.27	0.63	0.64
95% C.I.	(0.07,0.55)	(0.07,0.56)	(0.34,0.88)	(0.35,0.88)
$\overline{\phi}_c^{13}$ (true)	0.21	0.21	0.79	0.79
Mean	0.54	0.72	0.40	0.78
95% C.I.	(0.26,0.81)	(0.36,0.96)	(0.14,0.70)	(0.48,0.97)
$\overline{\phi}_c^{14}$ (true)	0.21	0.21	0.79	0.79
Mean	0.25	0.43	0.61	0.72
95% C.I.	(0.06,0.52)	(0.19,0.68)	(0.35,0.84)	(0.46,0.91)
$\overline{\phi}_c^{15}$ (true)	0.21	0.21	0.79	0.79
Mean	0.34	0.31	0.80	0.60
95% C.I.	(0.11,0.61)	(0.10,0.57)	(0.52,0.97)	(0.30,0.86)
$\overline{\phi}_c^{16}$ (true)	0.21	0.21	0.79	0.79
Mean	0.27	0.18	0.70	0.55
95% C.I.	(0.07,0.55)	(0.03,0.44)	(0.40,0.92)	(0.24,0.84)
$\overline{\phi}_c^{17}$ (true)	0.21	0.21	0.79	0.79
Mean	0.25	0.50	0.69	0.92
95% C.I.	(0.06,0.52)	(0.21,0.79)	(0.43,0.90)	(0.72,1.00)
$\overline{\phi}_c^{18}$ (true)	0.21	0.21	0.79	0.79
Mean	0.28	0.39	0.85	0.58
95% C.I.	(0.07,0.56)	(0.15,0.65)	(0.64,0.98)	(0.30,0.83)
$\overline{\phi}_c^{19}$ (true)	0.21	0.21	0.79	0.79
Mean	0.27	0.13	0.66	0.56
95% C.I.	(0.07,0.55)	(0.00,0.42)	(0.39,0.89)	(0.25,0.84)
$\overline{\phi}_c^{20}$ (true)	0.21	0.21	0.79	0.79
Mean	0.28	0.22	0.69	0.92
95% C.I.	(0.09,0.54)	(0.05,0.45)	(0.43,0.90)	(0.72,1.00)
$\overline{\phi}_c^{21}$ (true)	0.21	0.21	0.79	0.79
Mean	0.36	0.29	0.75	0.82
95% C.I.	(0.12,0.65)	(0.04,0.64)	(0.48,0.94)	(0.55,0.97)
$\overline{\phi}_c^{22}$ (true)	0.21	0.21	0.79	0.79
Mean	0.23	0.31	0.84	0.64

Continued on next page

Table A.1 – continued from Table A.1

	GRP1	**GRP2**	**GRP3**	**GRP4**
95% C.I.	(0.05,0.48)	(0.10,0.57)	(0.61,0.98)	(0.39,0.86)
$\overline{\phi}_c^{23}$ (true)	0.21	0.21	0.79	0.79
Mean	0.67	0.27	0.60	0.92
95% C.I.	(0.28,0.95)	(0.07,0.56)	(0.30,0.86)	(0.71,1.00)
$\overline{\phi}_c^{24}$ (true)	0.21	0.21	0.79	0.79
Mean	0.22	0.75	0.75	0.70
95% C.I.	(0.03,0.52)	(0.43,0.96)	(0.42,0.96)	(0.40,0.92)
$\overline{\phi}_c^{25}$ (true)	0.21	0.21	0.79	0.79
Mean	0.21	0.23	0.69	0.59
95% C.I.	(0.05,0.46)	(0.06,0.49)	(0.43,0.90)	(0.43,0.90)
$\overline{\phi}_c^{26}$ (true)	0.21	0.21	0.79	0.79
Mean	0.27	0.58	0.69	0.86
95% C.I.	(0.07,0.55)	(0.31,0.83)	(0.43,0.90)	(0.64,0.98)
$\overline{\phi}_c^{27}$ (true)	0.21	0.21	0.79	0.79
Mean	0.20	0.23	0.80	0.89
95% C.I.	(0.03,0.48)	(0.05,0.49)	(0.51,0.97)	(0.63,1.00)
$\overline{\phi}_c^{28}$ (true)	0.21	0.21	0.79	0.79
Mean	0.46	0.20	0.77	0.82
95% C.I.	(0.19,0.74)	(0.03,0.49)	(0.47,0.97)	(0.55,0.97)
$\overline{\phi}_c^{29}$ (true)	0.21	0.79	0.21	0.79
Mean	0.34	0.62	0.17	0.78
95% C.I.	(0.09,0.65)	(0.29,0.90)	(0.02,0.42)	(0.47,0.97)
$\overline{\phi}_c^{30}$ (true)	0.21	0.79	0.21	0.79
Mean	0.25	0.43	0.16	0.83
95% C.I.	(0.06,0.52)	(0.19,0.68)	(0.02,0.39)	(0.59,0.98)
$\overline{\phi}_c^{31}$ (true)	0.21	0.79	0.21	0.79
Mean	0.22	0.67	0.46	0.67
95% C.I.	(0.05,0.45)	(0.39,0.89)	(0.21,0.73)	(0.39,0.89)
$\overline{\phi}_c^{32}$ (true)	0.21	0.79	0.21	0.79
Mean	0.18	0.67	0.34	0.69
95% C.I.	(0.02,0.45)	(0.35,0.91)	(0.11,0.62)	(0.43,0.90)
$\overline{\phi}_c^{33}$ (true)	0.21	0.79	0.21	0.79
Mean	0.39	0.69	0.22	0.61
95% C.I.	(0.15,0.65)	(0.43,0.90)	(0.05,0.46)	(0.34,0.85)
$\overline{\phi}_c^{34}$ (true)	0.21	0.79	0.21	0.79
Mean	0.16	0.77	0.31	0.77
95% C.I.	(0.02,0.39)	(0.52,0.94)	(0.10,0.58)	(0.51,0.94)
$\overline{\phi}_c^{35}$ (true)	0.21	0.79	0.21	0.79
Mean	0.22	0.92	0.50	0.71
95% C.I.	(0.03,0.53)	(0.74,1.00)	(0.23,0.77)	(0.46,0.91)
$\overline{\phi}_c^{36}$ (true)	0.21	0.79	0.21	0.79
Mean	0.42	0.50	0.27	0.58

Continued on next page

Table A.1 – continued from Table A.1

	GRP1	**GRP2**	**GRP3**	**GRP4**
95% C.I.	(0.17,0.69)	(0.19,0.82)	(0.07,0.55)	(0.31,0.83)
$\overline{\phi}_c^{37}$ (true)	0.21	0.79	0.21	0.79
Mean	0.28	0.92	0.17	0.82
95% C.I.	(0.09,0.53)	(0.71,1.00)	(0.02,0.42)	(0.55,0.97)
$\overline{\phi}_c^{38}$ (true)	0.21	0.79	0.21	0.79
Mean	0.33	0.92	0.28	0.54
95% C.I.	(0.08,0.65)	(0.71,1.00)	(0.07,0.55)	(0.28,0.79)
$\overline{\phi}_c^{39}$ (true)	0.21	0.79	0.21	0.79
Mean	0.23	0.92	0.29	0.83
95% C.I.	(0.03,0.54)	(0.72,1.00)	(0.04,0.64)	(0.59,0.98)
$\overline{\phi}_c^{40}$ (true)	0.21	0.79	0.21	0.79
Mean	0.57	0.60	0.63	0.71
95% C.I.	(0.32,0.81)	(0.30,0.86)	(0.34,0.88)	(0.46,0.91)
$\overline{\phi}_c^{41}$ (true)	0.21	0.79	0.21	0.79
Mean	0.18	0.62	0.17	0.50
95% C.I.	(0.03,0.44)	(0.34,0.85)	(0.02,0.41)	(0.25,0.75)
$\overline{\phi}_c^{42}$ (true)	0.21	0.79	0.21	0.79
Mean	0.38	0.69	0.46	0.50
95% C.I.	(0.15,0.65)	(0.42,0.90)	(0.21,0.72)	(0.23,0.76)
$\overline{\phi}_c^{43}$ (true)	0.21	0.79	0.21	0.79
Mean	0.14	0.69	0.21	0.77
95% C.I.	(0.02,0.36)	(0.43,0.90)	(0.05,0.45)	(0.52,0.95)
$\overline{\phi}_c^{44}$ (true)	0.21	0.79	0.21	0.79
Mean	0.15	0.86	0.25	0.92
95% C.I.	(0.02,0.39)	(0.64,0.98)	(0.06,0.52)	(0.72,1.00)
$\overline{\phi}_c^{45}$ (true)	0.21	0.79	0.21	0.79
Mean	0.42	0.70	0.30	0.91
95% C.I.	(0.12,0.77)	(0.40,0.93)	(0.08,0.60)	(0.69,1.00)
$\overline{\phi}_c^{46}$ (true)	0.21	0.79	0.21	0.79
Mean	0.20	0.88	0.42	0.91
95% C.I.	(0.03,0.48)	(0.59,1.00)	(0.17,0.69)	(0.69,1.00)
$\overline{\phi}_c^{47}$ (true)	0.21	0.79	0.21	0.79
Mean	0.40	0.60	0.25	0.80
95% C.I.	(0.14,0.71)	(0.30,0.86)	(0.04,0.58)	(0.51,0.97)
$\overline{\phi}_c^{48}$ (true)	0.21	0.79	0.21	0.79
Mean	0.34	0.71	0.22	0.61
95% C.I.	(0.11,0.61)	(0.46,0.91)	(0.05,0.46)	(0.35,0.85)
$\overline{\phi}_c^{49}$ (true)	0.21	0.79	0.21	0.79
Mean	0.44	0.67	0.20	0.73
95% C.I.	(0.15,0.76)	(0.35,0.91)	(0.03,0.49)	(0.44,0.93)
$\overline{\phi}_c^{50}$ (true)	0.21	0.79	0.21	0.79
Mean	0.45	0.72	0.25	0.71

Continued on next page

Table A.1 – continued from Table A.1

	GRP1	GRP2	GRP3	GRP4
95% C.I.	(0.19,0.74)	(0.46,0.91)	(0.06,0.52)	(0.46,0.91)
$\overline{\phi}_c^{51}$ (true)	0.21	0.79	0.21	0.79
Mean	0.22	0.61	0.20	0.60
95% C.I.	(0.03,0.52)	(0.35,0.85)	(0.03,0.49)	(0.19,0.94)
$\overline{\phi}_c^{52}$ (true)	0.21	0.79	0.21	0.79
Mean	0.38	0.89	0.31	0.55
95% C.I.	(0.10,0.71)	(0.63,1.00)	(0.10,0.57)	(0.26,0.82)
$\overline{\phi}_c^{53}$ (true)	0.21	0.79	0.21	0.79
Mean	0.25	0.73	0.36	0.69
95% C.I.	(0.06,0.51)	(0.44,0.93)	(0.12,0.66)	(0.43,0.90)
$\overline{\phi}_c^{54}$ (true)	0.21	0.79	0.21	0.79
Mean	0.21	0.64	0.27	0.85
95% C.I.	(0.05,0.46)	(0.39,0.86)	(0.07,0.55)	(0.61,0.98)
$\overline{\phi}_c^{55}$ (true)	0.21	0.79	0.21	0.79
Mean	0.36	0.92	0.55	0.77
95% C.I.	(0.14,0.62)	(0.71,1.00)	(0.26,0.81)	(0.52,0.95)
$\overline{\phi}_c^{56}$ (true)	0.21	0.79	0.21	0.79
Mean	0.50	0.82	0.28	0.67
95% C.I.	(0.21,0.79)	(0.55,0.97)	(0.07,0.55)	(0.39,0.89)
$\overline{\phi}_c^{57}$ (true)	0.07	0.5	0.5	0.93
Mean	0.08	0.61	0.66	0.92
95% C.I.	(0.00,0.26)	(0.35,0.85)	(0.39,0.89)	(0.73,1.00)
$\overline{\phi}_c^{58}$ (true)	0.07	0.5	0.5	0.93
Mean	0.15	0.46	0.50	0.83
95% C.I.	(0.02,0.39)	(0.21,0.72)	(0.25,0.75)	(0.58,0.98)
$\overline{\phi}_c^{59}$ (true)	0.07	0.5	0.5	0.93
Mean	0.11	0.75	0.69	0.86
95% C.I.	(0.00,0.38)	(0.48,0.94)	(0.43,0.90)	(0.64,0.98)
$\overline{\phi}_c^{60}$ (true)	0.07	0.5	0.5	0.93
Mean	0.18	0.58	0.50	0.70
95% C.I.	(0.03,0.45)	(0.31,0.83)	(0.21,0.78)	(0.40,0.92)
$\overline{\phi}_c^{61}$ (true)	0.07	0.5	0.5	0.93
Mean	0.08	0.50	0.54	0.90
95% C.I.	(0.00,0.27)	(0.21,0.79)	(0.28,0.79)	(0.66,1.00)
$\overline{\phi}_c^{62}$ (true)	0.07	0.5	0.5	0.93
Mean	0.22	0.58	0.39	0.92
95% C.I.	(0.05,0.45)	(0.31,0.84)	(0.15,0.65)	(0.73,1.00)
$\overline{\phi}_c^{63}$ (true)	0.07	0.5	0.5	0.93
Mean	0.11	0.42	0.80	0.80
95% C.I.	(0.00,0.36)	(0.17,0.69)	(0.51,0.97)	(0.52,0.97)
$\overline{\phi}_c^{64}$ (true)	0.07	0.5	0.5	0.93
Mean	0.08	0.54	0.50	0.92

Continued on next page

Table A.1 – continued from Table A.1

	GRP1	**GRP2**	**GRP3**	**GRP4**
95% C.I.	(0.00,0.29)	(0.28,0.79)	(0.24,0.76)	(0.73,1.00)
$\overline{\phi}_c^{65}$ (true)	0.07	0.5	0.5	0.93
Mean	0.09	0.45	0.60	0.75
95% C.I.	(0.00,0.31)	(0.19,0.74)	(0.30,0.86)	(0.42,0.96)
$\overline{\phi}_c^{66}$ (true)	0.5	0.5	0.5	0.5
Mean	0.09	0.69	0.54	0.85
95% C.I.	(0.00,0.29)	(0.43,0.90)	(0.27,0.79)	(0.62,0.98)
$\overline{\phi}_c^{67}$ (true)	0.5	0.5	0.5	0.5
Mean	0.42	0.56	0.40	0.50
95% C.I.	(0.17,0.69)	(0.24,0.85)	(0.14,0.70)	(0.23,0.77)
$\overline{\phi}_c^{68}$ (true)	0.5	0.5	0.5	0.5
Mean	0.50	0.80	0.30	0.50
95% C.I.	(0.21,0.78)	(0.52,0.97)	(0.08,0.60)	(0.21,0.79)
$\overline{\phi}_c^{69}$ (true)	0.5	0.5	0.5	0.5
Mean	0.45	0.50	0.55	0.78
95% C.I.	(0.16,0.76)	(0.21,0.79)	(0.27,0.81)	(0.47,0.97)
$\overline{\phi}_c^{70}$ (true)	0.5	0.5	0.5	0.5
Mean	0.66	0.58	0.30	0.58
95% C.I.	(0.35,0.91)	(0.22,0.88)	(0.08,0.61)	(0.31,0.83)

Table A.1: Estimates of cluster-specific parameters. The cluster-specific profile parameters, $\overline{\phi_c}$, are defined in equation (3.1).

Appendix B

Code for Bayesian Profile Regression

```
library(PReMiuM)
# Bayesian profile regression with the covariates only
runInfoObj.yes<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[,1:71],
output="output_DATA", covNames=colnames(DATA)[1:70],
nClusInit=20, run=TRUE, seed=3459, excludeY=TRUE)
# Calculate the dissimilarity matrix.
dissimObj.yes<-calcDissimilarityMatrix(runInfoObj.yes)
# Calculate the optimal clustering.
clusObj.yes<-calcOptimalClustering(dissimObj.yes)
# Calculate the estimates ot risks and cluster-specific parameters.
riskProfileObj.yes<-calcAvgRiskAndProfile(clusObj.yes)
cluster.yes<-clusObj.yes$clustering
# Confustion matrix of the clustering results using the Bayesian
# profile regression and the true classes.
table(cluster.yes, DATA$group)


# Bayesian profile regression with the covariates and outcome
runInfoObj.no<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[,1:71],
output="output_DATA", covNames=colnames(DATA)[1:70],
nClusInit=10, run=TRUE, seed=3459, excludeY=FALSE)
dissimObj.no<-calcDissimilarityMatrix(runInfoObj.no)
clusObj.no<-calcOptimalClustering(dissimObj.no)
riskProfileObj.no<-calcAvgRiskAndProfile(clusObj.no)
cluster.no<-clusObj.no$clustering
table<(cluster.no, DATA$group)

# Plot the patients’ heatmap based on the dissimilarity matrix
# obtained by the function calcDissimilarityMatrix.
heatDissMat(dissimObj.no)

# Plot the posterior distributions of risks and some of the cluster-specific
# parameters: X_1, X_28, X_29, X_56, X_57 and X_67.
clusterOrderObj<-plotRiskProfile(riskProfileObj.no, "summary_DATA.png",
whichCovariates=c(1,28,29,56,57,67), orderBy="ClusterSize")

# Obtain the empirical mean of the cluster-specific risks.
risk<-riskProfileObj.no$empiricals

# Obtain the cluster sizes
clustersize<-riskProfileObj.no$riskProfClusObj$clusterSizes

# Obtain the cluster-specific parameters
phi<-riskProfileObj.no$profile

# Bayesian profile regression predictions
runInfoObj.p<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[,1:71],
output="pred_DATA", covNames=colnames(DATA)[1:70],
predict=DATA_num[,1:71], seed=81290)
dissimObj.p<-calcDissimilarityMatrix(runInfoObj.p)
clusObj.p<-calcOptimalClustering(dissimObj.p)
riskProfileObj.p<-calcAvgRiskAndProfile(clusObj.p)
pred.Bayesian<-calcPredictions(riskProfileObj.p,
fullSweepPredictions=FALSE, fullSweepLogOR=FALSE)
# Calculate the mean squared error using the
# true outcome probabilites.
prob <- c(rep(.5, 12), rep(.1, 12), rep(.3, 12), rep(.9, 12))
mean((pred.Bayesian$predictedY-prob)^2)

# Calculate the predictions for the test set.
runInfoObj.t<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[,1:71],
covNames=colnames(DATA)[1:70], predict=test[,1:71],
seed=81290, excludeY=FALSE)
dissimObj.t<-calcDissimilarityMatrix(runInfoObj.t)
clusObj.t<-calcOptimalClustering(dissimObj.t)
riskProfileObj.t<-calcAvgRiskAndProfile(clusObj.t)
pred.Bayesian.t<-calcPredictions(riskProfileObj.t,
fullSweepPredictions=FALSE, fullSweepLogOR=FALSE)
# Calculate the mean squared prediction error using
# the true outcome probabilities of the test set.
prob.t <- c(rep(.5, 8), rep(.1, 8), rep(.3, 8), rep(.9, 8))
mean((prob.t-pred.Bayesian.t$predictedY)^2)
```

```
# Plot the global trace of the parameters:
# alpha and the number of clusters.
par(mfrow=c(1,2))
runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[1:71],
output="converge_DATA", covNames=colnames(DATA)[1:70],
nClusInit=10, run=TRUE, seed=3459, excludeY=FALSE,
reportBurnIn=TRUE)
globalParsTrace(runInfoObj, parameters="nClusters", plotBurnIn=TRUE)
globalParsTrace(runInfoObj, parameters="alpha", plotBurnIn=TRUE)

# Investigate the parameter convergence using the R package coda.
library("coda")
alphachain<-mcmc(read.table("converge_DATA_alpha.txt"))
autocorr.plot(alphachain, sub="alpha")
nclusterchain<-mcmc(read.table("converge_DATA_nClusters.txt"))
autocorr.plot(nclusterchain, sub="nClusters")

# Plot the posterior distribution of alpha with different initial numbers
# of clusters, and run three repitions for each per initialisation.
runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[1:71],
output="converge_DATA", covNames=colnames(DATA)[1:70],
nClusInit=1, run=TRUE, seed=100, excludeY=FALSE,
reportBurnIn=FALSE)
chain1.1 <- read.table("converge_DATA_alpha.txt")
runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[1:71],
output="converge_DATA", covNames=colnames(DATA)[1:70],
nClusInit=1, run=TRUE, seed=200, excludeY=FALSE,
 reportBurnIn=FALSE)
chain1.2 <- read.table("converge_DATA_alpha.txt")
runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[1:71],
output="converge_DATA", covNames=colnames(DATA)[1:70],
nClusInit=1, run=TRUE, seed=300, excludeY=FALSE,
reportBurnIn=FALSE)
chain1.3 <- read.table("converge_DATA_alpha.txt")
runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[1:71],
output="converge_DATA", covNames=colnames(DATA)[1:70],
nClusInit=10, run=TRUE, seed=100, excludeY=FALSE,
reportBurnIn=FALSE)
chain2.1 <- read.table("converge_DATA_alpha.txt")
runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[1:71],
output="converge_DATA", covNames=colnames(DATA)[1:70],
nClusInit=10, run=TRUE, seed=200, excludeY=FALSE,
reportBurnIn=FALSE)
chain2.2 <- read.table("converge_DATA_alpha.txt")
runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[1:71],
output="converge_DATA", covNames=colnames(DATA)[1:70],
nClusInit=10, run=TRUE, seed=300, excludeY=FALSE,
reportBurnIn=FALSE)
chain2.3 <- read.table("converge_DATA_alpha.txt")
runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[1:71],
output="converge_DATA", covNames=colnames(DATA)[1:70],
nClusInit=30, run=TRUE, seed=100, excludeY=FALSE,
reportBurnIn=FALSE)
chain3.1 <- read.table("converge_DATA_alpha.txt")
runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[1:71],
output="converge_DATA", covNames=colnames(DATA)[1:70],
nClusInit=30, run=TRUE, seed=200, excludeY=FALSE,
reportBurnIn=FALSE)
chain3.2 <- read.table("converge_DATA_alpha.txt")
runInfoObj<-profRegr(yModel="Bernoulli", xModel="Discrete",
nSweeps=20000, nBurn=300, data=DATA[1:71],
output="converge_DATA", covNames=colnames(DATA)[1:70],
nClusInit=30, run=TRUE, seed=300, excludeY=FALSE,
reportBurnIn=FALSE)
chain3.3 <- read.table("converge_DATA_alpha.txt")
boxplot(data.frame(chain1.1, chain1.2, chain1.3, chain2.1, chain2.2,
chain2.3, chain3.1, chain3.2, chain3.3), at=c(1,2,3, 11,12,13, 21,22,23))
```