Bayesian Multilevel-multiclass Graphical Model

Jiali Lin

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Statistics

Inyoung Kim, Chair
Xinwei Deng
George Terrell
Feng Guo

May 15, 2019
Blacksburg, Virginia

KEYWORDS: Bayesian Method; Gaussian Graphical Model; Gaussian Process; Gene Co-expression Network; Ising Prior; Kernel Learning; Multilevel network; Pathway Analysis; Precision matrix estimation; Variational Bayesian; Variable Selection.

Bayesian Multilevel-multiclass Graphical Model

Jiali Lin

(ABSTRACT)

Gaussian graphical model has been a popular tool to investigate conditional dependency between random variables by estimating sparse precision matrices. Two problems have been discussed. One is to learn multiple Gaussian graphical models at multilevel from unknown classes. Another one is to select Gaussian process in semiparametric multi-kernel machine regression.

The first problem is approached by Gaussian graphical model. In this project, I consider learning multiple connected graphs among multilevel variables from unknown classes. I estimate the classes of the observations from the mixture distributions by evaluating the Bayes factor and learn the network structures by fitting a novel neighborhood selection algorithm. This approach is able to identify the class membership and to reveal network structures for multilevel variables simultaneously. Unlike most existing methods that solve this problem by frequentist approaches, I assess an alternative to a novel hierarchical Bayesian approach to incorporate prior knowledge.

The second problem focuses on the analysis of correlated high-dimensional data which has been useful in many applications. In this work, I consider a problem of detecting signals with a semiparametric regression model which can study the effects of fixed covariates (e.g. clinical variables) and sets of elements (e.g. pathways of genes). I model the unknown high-dimension functions of multi-sets via multi-Gaussian kernel machines to consider the possibility that elements within the same set interact with each other. Hence, my variable selection can be considered as Gaussian process selection. I develop my Gaussian process selection under the Bayesian variable selection framework.

Bayesian Multilevel-multiclass Graphical Model

Jiali Lin

(GENERAL AUDIENCE ABSTRACT)

A network can be represented by nodes and edges between nodes. Under the assumption of multivariate Gaussian distribution, a graphical model is called a Gaussian graphical model, where edges are undirected. Gaussian graphical model has been studied for years to understand conditional dependency structure between random variables. Two problems have been discussed.

In the first project, I consider learning multiple connected graphs among multilevel variables from unknown classes. I estimate the classes of the observations from the mixture distributions. This approach is able to identify the class membership and to reveal network structures for multilevel variables simultaneously. Unlike most existing methods that solve this problem by frequentist approaches, I assess an alternative to a novel hierarchical Bayesian approach to incorporate prior knowledge.

The second problem focuses on the analysis of correlated high-dimensional data which has been useful in many applications. In this work, I consider a problem of detecting signals with a semiparametric regression model which can study the effects of fixed covariates (e.g. clinical variables) and sets of elements (e.g. pathways of genes). I model the unknown high-dimension functions of multi-sets via multi-Gaussian kernel machines to consider the possibility that elements within the same set interact with each other. Hence, my variable selection can be considered as Gaussian process selection. I develop my Gaussian process selection under the Bayesian variable selection framework.

To my parents, for their unconditional love and everlasting support.

Acknowledgments

I would like to take this opportunity to express my deepest gratitude to my advisor, Dr. Inyoung Kim, for her invaluable patience, encouragement and guidance throughout my Ph.D. study. I am lucky to have Dr. Kim as my advisor and a mentor in my life.

I would also like to extend my gratitude to Dr. Xinwei Deng, Dr. George Terrell and Dr. Feng Guo. They graciously agreed to serve on my Ph.D. advisory committee. I greatly appreciate their instructive advice and everlasting support!

I would also like to thank all the faculties in the Department of Statistics for their inspiring courses and guidance, and all the staff members in the Department of Statistics for their everyday support to make my study and research possible.

To all of my friends in my research group, in the department and outside the department, thank you all for your friendship and company. Thank you!

Last but not least, big thanks to my parents. I am forever grateful for their unconditional love and support throughout my life. I love you forever!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

To begin with, a gene co-expression network (GCN), an undirected graph, has been used to describe the relationships between genes. In mathematics, a graph is composed of nodes which are connected by edges. In this context, we visualize the interaction between genes by a network. In the case of Gaussian graphical models (GGMs), learning the graph topology (i.e. the conditional dependencies between the genes) is equivalent to estimating nonzero entries in the inverse covariance matrix (precision matrix). In other words, if the (i, j)th entry in the precision matrix is 0, gene i and j are conditionally independent, otherwise, they are dependent given all other genes.

Gaussian graphical model has been widely applied in science as it is able to describe the conditional dependencies between the variables. Learning the graph topology via Gaussian graphical model (GGMs) is equivalent to estimating nonzero entries in the inverse covariance matrix (precision matrix) $\mathbf{\Sigma}^{-1}$. However, estimating a precision matrix is known to be

statistically challenging especially for high dimensional data. One can compute the maximum likelihood estimator (MLE) for a non-decomposable Gaussian graphical model, also known as covariance selection (Dempster, 1972). In the high-dimensional case where the number of observations is less than the number of variables, the sample covariance matrix is singular and thus non-invertible. Another problem will also arise because MLE can hardly yield exact zeros in the precision matrix. In other words, MLE does not result in sparse graphs thus making it hard to interpret for network analysis.

Numerous methods to estimate the covariance matrix in high-dimensional problem have developed methods to learn sparse graphs by some forms of penalty functions that force zeros in the corresponding precision matrix. Meinshausen and Bühlmann (2006) proposed doing so by performing LASSO regression of all nodes to a target node. A penalized log-likelihood approach can be optimized by a variety of methods (Yuan and Lin, 2007), (Banerjee et al., 2008), (Rothman et al., 2008). Among them, the most popular one is known as the graphical lasso (Glasso), developed by Friedman et al. (2008). An alternative is to form Bayesian graphical lasso which uses Laplace priors on off-diagonal entries of the precision matrix (Wang, 2012). In recent years, researchers realize that it is more efficient to learn multiple graphs together as they may share certain characteristics. Guo et al. (2011) jointly learned multiple graphs by decomposing precision matrices to be a multiplication of common factors across groups and unique factors for each group. Danaher et al. (2014) generalized this jointly learning multiple graph approach and use two different penalty functions: the fused lasso and the group lasso.

1.2 Motivation

Estimation of Gaussian graphical model has a lot of applications. For example, biological networks. Unlike classic analysis on a single variable, biological networks provide further insight into biological processes, as evidence shows groups of genes by functional pathways operate together. It is now more concerned to incorporate phenotype information because these complex networks exhibit similar but different patterns across biological conditions.

However, the standard formulation for learning Gaussian graphical models assumes that classes of observations are given and learn the network structures within one level (e.g. pathways or genes). However, this assumption might be too optimistic. In some cases, we may have heterogeneous data at different classes. Often, categorizing these different classes are often ambiguous. For example, a present-day gene expression data can involve hundreds of pathway and gene variables, while lacking the case-control status. Hence, we propose a technique for inferring multiple connected networks from observations belonging to unknown classes following mixture distributions. For genomic and proteomic data, it is reasonable that genes can be clustered or grouped into pathways for particular functions. On the other hand, pathways are not isolated, either. Instead, they work together to accomplish certain tasks. One may also find it necessary to study their connectivity or relationship patterns. To tackle the problem of the multilevel networks, we propose multiple Gaussian graphical models for multilevel variables from unknown classes that are able not only to estimate the class membership and but also to learn the networks at a different level.

Thus, the task of the first project is to learn multiple Gaussian graphical models at multilevel from unknown classes. In most cases, however, we may usually have heterogeneous data obtained at different levels. Therefore, we consider learning multiple connected graphs among multilevel variables from unknown classes. We are also interested in the problem where the

outcome is continuous and we aim to find detecting signals with a model which is able to study the effects of fixed covariates (e.g. clinical variables) and sets of elements (e.g. pathways of genes). For example, genetic effects can have a huge impact on disease. Particularly, it is important to identify significant genetic pathway effects associated with biomarkers.

1.3 Overview

The rest of the dissertation is organized as follows. In Chapter 2, we propose Bayesian multiple Gaussian graphical models. We estimate the classes of the observations from the mixture distributions by evaluating the Bayes factor and learn the network structures by fitting a novel neighborhood selection algorithm. Our approach is based on a generative model that allows incorporating prior knowledge. We demonstrate the unique advantages of our methods through several simulations. An application on breast cancer shows us the results gained from the model can provide insight into biological studies. In Chapter 3, we propose Gaussian process semiparametric regression model. We model the unknown high-dimension functions of multi-sets by Gaussian process and propose to select Gaussian process. Furthermore, we incorporate prior knowledge for structural sets by imposing an Ising prior on the model. Our approach can be easily applied in high-dimensional space where the sample size is smaller than the number of elements. An efficient variational Bayes algorithm is developed. The advantages of our approach are demonstrated through simulation studies and type II diabetes genetic pathway analysis. In Chapter 4, we give a general review of the contributions of this dissertation, as well as discuss directions for future research.

Chapter 2

Bayesian Multiple Gaussian Graphical Models for Multilevel Variables from Unknown Classes

2.1 Introduction

Gaussian graphical model has been widely applied in science as it is able to describe the conditional dependencies between the variables. Learning the graph topology via Gaussian graphical model (GGMs) is equivalent to estimating nonzero entries in the inverse covariance matrix (precision matrix) $\mathbf{\Sigma}^{-1}$. However, estimating a precision matrix is known to be statistically challenging especially for high dimensional data. One can compute the maximum likelihood estimator (MLE) for a non-decomposable Gaussian graphical model, also known as covariance selection (Dempster, 1972). In the high-dimensional case where the number of observations is less than the number of variables, the sample covariance matrix is singular and thus non-invertible. Another problem will also arise because MLE can hardly yield exact

zeros in the precision matrix. In other words, MLE does not result in sparse graphs thus making it hard to interpret for network analysis.

Numerous methods to estimate the covariance matrix in the high-dimensional problem have developed methods to learn sparse graphs by some forms of penalty functions that force zeros in the corresponding precision matrix. Meinshausen and Bühlmann (2006) proposed doing so by performing LASSO regression of all nodes to a target node. A penalized log-likelihood approach can be optimized by a variety of methods (Yuan and Lin, 2007), (Banerjee et al., 2008), (Rothman et al., 2008). Among them, the most popular one is known as the graphical lasso (Glasso), developed by Friedman et al. (2008). An alternative is to form Bayesian graphical lasso which uses Laplace priors on off-diagonal entries of the precision matrix (Wang, 2012). In recent years, researchers realize that it is more efficient to learn multiple graphs together as they may share certain characteristics. Guo et al. (2011) jointly learned multiple graphs by decomposing precision matrices to be a multiplication of common factors across groups and unique factors for each group. Danaher et al. (2014) generalized this jointly learning multiple graph approach and use two different penalty functions: the fused lasso and the group lasso.

However, the standard formulation for learning Gaussian graphical models assumes that classes of observations are given and learn the network structures within one level (e.g. pathways or genes). However, this assumption might be too optimistic. In some cases, we may have heterogeneous data at different classes. Often, categorizing these different classes are often ambiguous. For example, a present-day gene expression data can involve hundreds of pathway and gene variables, while lacking the case-control status. Hence, we propose a technique for inferring multiple connected networks from observations belonging to unknown classes following mixture distributions. For genomic and proteomic data, it is reasonable that genes can be clustered or grouped into pathways for particular functions. On

the other hand, pathways are not isolated, either. Instead, they work together to accomplish certain tasks. One may also find it necessary to study their connectivity or relationship patterns. To tackle the problem of the multilevel networks, we propose multiple Gaussian graphical models for multilevel variables from unknown classes that are able not only to estimate the class membership and but also to learn the networks at a different level.

In this chapter, we develop multiple Gaussian graphical models for multilevel variables from unknown classes under the Bayesian hierarchical framework for inferring multiple connected networks from observations belonging to unknown classes following mixture distributions. Our goal is to simultaneously explore conditional dependency structures among variables of two levels (i.e. set level and element level) when classes are drawn from the mixture distributions. We investigate a two-step solution to this problem using Bayesian methods. The basic idea of the proposed model is to assess class membership first and then learn the multilevel networks.

Our approach has several novel features: (1) it is able to reveal networks for multilevel variables simultaneously; (2) it evaluates class membership for the observations from Gaussian graphical model; (3) it is fully model-based, thus has nice probabilistic interpretations. It is known that clustering data for the Gaussian graphical model is challenging and only a few works focus on the relevant problem so far.

The rest of the chapter is organized as follows. In Chapter 2.2, we first describe Bayesian multiple Gaussian graphical models (denoted as "BMGGM") for multilevel variables from unknown classes. Chapter 2.3 proposes a probability model to infer high-level variable network from the lower-level variable network. Chapter 2.4 contains Bayesian generative model and posterior inference. In Chapter 2.6, we illustrate the performance of the proposed model through simulated data. Chapter 2.7 demonstrates the application of a case study. Chapter 2.8 contains concluding remarks.

2.2 Multiple Gaussian Graphical Model for Unknown Classes

2.2.1 Problem Setup and Notation

We start by describing the problem setup and some notation that will be used throughout this chapter.

Suppose we are given a data set that has C possible heterogeneous classes with a total of P variables, where these variables belong to K pre-specified groups. For a given group, it has p_k variables such that $\sum_{k=1}^{K} p_k = P$. We refer to the groups as "sets" (or "higher-level variables") and the variables within higher-level variables as elements (or "lower-level variables"). Because our data has a hierarchical structure, we shall refer to these variables as "multilevel variables".

We index classes by c where $c = 1, \ldots, C$, index matrix entries for lower-level variables by $\{j, j'\}$ where $1 < j \neq j' < P$ and index matrix entries for higher-level variables by $\{k, k'\}$ where $1 < k \neq k' < K$.

2.2.2 The Background and General Formulation

For a given class c, we have a $n_c \times P$ data matrix, where n_c is the sample size in the class c and P is the number of variables. In other words, the cth class contains n_c observations $\{\mathbf{x}_{c,1}, \ldots, \mathbf{x}_{c,n_c}\}$, where $\mathbf{x}_{c,i} = \{x_{c,i,1}, x_{c,i,2}, \ldots, x_{c,i,p}\} \in \mathbb{R}^p$, $i = 1, \ldots n_c$. We assume that within each class c, $\mathbf{x}_{c,1}, \ldots, \mathbf{x}_{c,n_c} \in \mathbb{R}^p$ are independent and identically drawn from a

$P-$variate multivariate Gaussian distribution within each class,

$$\mathbf{x}_{c,i} \sim N(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), \quad i = 1, \ldots, n_c, \tag{2.1}$$

where $\boldsymbol{\mu}_c = \{\mu_{c,1}, \ldots, \mu_{c,p}\} \in \mathbb{R}^p$ is the mean vector and $\boldsymbol{\Sigma}_c$ is a symmetric and positive definite $P \times P$ matrix.

Further, we assume all the data points are generated from a mixture of a $C-$component Gaussian distributions with unknown class membership for an individual. Following standard practice, we introduce a $C-$dimensional indicator z_i to represent the latent state. Then, the prior distribution of z can be specified in terms of the mixing coefficients π_c by

$$p(z_{c,i} = c) = \pi_c, \tag{2.2}$$

where $\sum_{c=1}^{C} \pi_c = 1$.

As each base distribution in the mixture is a multivariate Gaussian with mean $\boldsymbol{\mu}_c$ and covariance matrix $\boldsymbol{\Sigma}_c$, the joint distribution has the form

$$p(\mathbf{x}_{c,i}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \sum_{c=1}^{C} \pi_c N(\mathbf{x}_{c,i}\,|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \quad i = 1, \ldots, n_c.$$

2.2.3 Learning Sparse Gaussian Graphical Model

Our goal is to infer graphs that have sparse structure so that we can describe the relationships among elements. Equivalently, we can estimate the non-zeros in the inverse matrix $\boldsymbol{\Sigma}_c^{-1}$,

which corresponds to edges in the graphs,

$$\boldsymbol{\Sigma}_c^{-1} = \begin{pmatrix} \omega_{c,1,1} & \cdots & \omega_{c,1,P} \\ \vdots & \ddots & \vdots \\ \omega_{c,P,1} & \cdots & \omega_{c,P,P} \end{pmatrix}. \tag{2.3}$$

The basic idea of learning a sparse Gaussian graphical model is to represent the inverse covariance matrix by a modified Cholesky decomposition and then apply the neighborhood selection with priors that can lead to sparsity.

Specifically, in order to learn a sparse covariance matrix with statistically interpretable parameterization, we adopt the idea of a modified Cholesky decomposition (Rue and Held, 2005) of the precision matrix, $\boldsymbol{\Sigma}_c^{-1} = \mathbf{L}_c^\top \mathbf{D}_c \mathbf{L}_c$, where $\mathbf{L}_c$ is a upper triangular matrix and $\mathbf{D}_c$ is diagonal matrix. With this decomposition, the multivariate normal model can be transformed into a regression problem.

For given element j, $x_{c,i,j}$ can be written as

$$x_{c,i,j} = \mu_{c,j} + \sum_{j'<j} \beta_{c,j'j}(x_{c,i,j'} - \mu_{c,i,j'}) + \epsilon_{c,j}, \quad j = 2, \ldots, P,$$

where $\epsilon_{c,j}$ is random error for a given class c and element j.

This expression can be further expanded to

$$\begin{aligned} x_{c,i,2} &= \mu_{c,2} + \beta_{c,12}(x_{c,i,1} - \mu_{c,i,1}) + \epsilon_{c,2}, \\ x_{c,i,3} &= \mu_{c,3} + \sum_{j'=1}^{2} \beta_{c,j'3}(x_{c,i,j'} - \mu_{c,i,j'}) + \epsilon_{c,3}, \\ &\vdots \end{aligned}$$

$$x_{c,i,p} \quad = \quad \mu_{c,p} + \sum_{j'=1}^{p-1} \beta_{c,j'p}(x_{c,i,j'} - \mu_{c,i,j'}) + \epsilon_{c,p}.$$

By these formulations, the upper triangular $\mathbf{L}$ has entries of regression coefficients

$$\mathbf{L}_c^\top = \begin{pmatrix} 1 & & & & \\ -\beta_{c,12} & 1 & & & \\ -\beta_{c,13} & -\beta_{c,23} & 1 & & \\ \vdots & \vdots & & \ddots & \\ -\beta_{c,1P} & -\beta_{c,2P} & \dots & -\beta_{c,PP-1} & 1 \end{pmatrix}. \tag{2.4}$$

For simplicity, each data are centered such that $\boldsymbol{\mu}_c = 0$. From equation (2.4), we have $\mathbf{L_c X_c} = \boldsymbol{\epsilon}_c$. Define $\mathbf{D}_c = \text{var}(\boldsymbol{\epsilon}_c)$, then $\mathbf{D}_c$ is a diagonal matrix of variance. The advantage of this parameterization is that it enforces the symmetry and positive-definiteness of the precision matrix.

We shall realize the precision matrix obtained from Meinshausen and Bühlmann (2006) is not necessarily positive definite. Other approaches (Yuan and Lin, 2007), (Banerjee et al., 2008) require constraint that precision matrix be positive definite. Therefore, to get around these obstacles, we look for the estimated precision matrix to be unconstrained and statistically meaningful. It is obvious that the variables of the modified Cholesky decomposition depend on the order in which the variables appear in the data matrix. However, this problem becomes less important as we aim to find meaningful patterns across the classes while $\boldsymbol{\Sigma}_c^{-1}$ in each class is guaranteed to be valid.

We show the ordering problem with a toy example. We generate data based on the following element network and corresponding precision matrix by approach defined in Chapter 2.6.1.

The diagram is

$$\text{element1} \to \text{element2} \quad \text{and} \quad \text{element2} \to \text{element3}.$$

(See Figure 2.1).

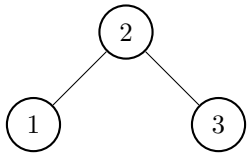


Figure 2.1: Diagram for the toy example.

$$\mathbf{\Sigma}^{-1} = \begin{pmatrix} 1.00 & 0.50 & 0.00 \\ 0.50 & 1.00 & 0.50 \\ 0.00 & 0.50 & 1.00 \end{pmatrix}.$$

Our Bayesian multiple Gaussian graphical models give an estimation

$$\hat{\mathbf{\Sigma}}^{-1} = \begin{pmatrix} 1.00 & 0.50 & -0.01 \\ 0.50 & 1.25 & 0.54 \\ -0.01 & 0.54 & 1.30 \end{pmatrix}.$$

Now we switch column 2 to 3. Then the true diagram becomes

$$\text{element1} \to \text{element3} \quad \text{and} \quad \text{element2} \to \text{element3}.$$

Our Bayesian multiple Gaussian graphical models give an estimation

$$\hat{\mathbf{\Sigma}}^{-1} = \begin{pmatrix} 1.00 & 0.00 & 0.53 \\ 0.00 & 1.00 & 0.52 \\ 0.53 & 0.52 & 1.56 \end{pmatrix}.$$

We can see that the modified Cholesky decomposition gives a different estimation of the precision matrix but would able to detect the correct adjacency matrix in Figure 2.2.

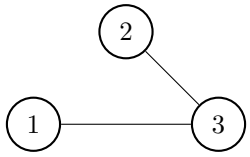


Figure 2.2: Diagram for the toy example after 2nd column and 3rd column switched.

2.3 Multilevel Network Probability Model

In this chapter, we propose a multilevel network probability model. We illustrate how to infer the network structure at the set level. In our example, sets are high-level variables and elements are lower-level variables within sets. We estimate set interaction probabilities using element interaction probabilities by implementing the idea of Kim et al. (2007) which originally proposed the set-set interaction estimation through element-element probabilities.

We build a multilevel network model using the following assumption: (A1) two sets interact if at least one element pair from the two sets interact.

First we define some notations. Define $g_{c,j,j'}^{(k,k')} = 1$ if element j interacts with the element j' in set pair k and k' and $g_{c,j,j'}^{(k,k')} = 0$ otherwise. Then, $\Pr(g_{c,jj'} = 1)$ is the probability that the element j interacts with element j'. Let $P_{c,kk'}$ represent the interaction event between sets pair k and k', with $P_{c,kk'} = 1$ if they interact and $P_{c,kk'} = 0$ otherwise.

By using (A1), we can calculate the interaction probabilities between two sets using element interaction probabilities:

$$\Pr(P_{c,kk'}=1) = 1 - \prod_{\{g^{(k,k')}_{c,j,j'} \in P_{c,kk'}\}} \{1 - Pr(g_{c,j,j'})^{\rho}\}, \tag{2.5}$$

where $\{g^{(k,k')}_{c,j,j'} \in P_{c,kk'}\}$ is all pairs of element from the sets k and k' and ρ is a adjusting constant value using the number of element pairs which will be explained in shortly.

We consider the following three conditions:

- If $\Pr(g^{(k,k')}_{c,j,j'}=1)=1$ for at least one lower-level variable pair, $\Pr(P_{c,kk'}=1)=1$;
- If $\Pr(g^{(k,k')}_{c,j,j'}=1)=0$ for all lower-level pairs, $\Pr(P_{c,kk'}=1)=0$;
- If $\Pr(g^{(k,k')}_{j,j'}=1)=1/2$ for all lower-level pairs, $\Pr(P_{c,kk'}=1)=1/2$;
- If $0<\Pr(g^{(k,k')}_{c,j,j'}=1)<1$, we have $0 \leq \Pr(P_{c,kk'}=1) \leq 1$.

Since the number of elements of set vary, we adjust the set probability. The ρ can be derived from the third condition as

$$\rho = \frac{\log\{1-(1/2)^{\frac{1}{M_{c,k,k'}}}\}}{\log(1/2)},$$

where $M_{c,k,k'}$ represents the total number of elements pairs between sets k and k'. This adjustment is motivated from observing that the value of equation (2.5) increases as the number of elements increases.

The third condition can be ignored and we set $\rho = 1$ to compute $\Pr(P_{c,kk'}=1)$. We will compare these two scenarios in chapter 2.6.

2.4 Bayesian Hierarchical Framework

We seek to learn the network (both set and element) structure for each class so as to obtain the inverse covariance matrix $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_C^{-1}$. Since we do not know each class belongs to what cluster, we estimate the class membership for each observation by fitting finite mixture model.

In this chapter, we describe how to estimate the parameters under the Bayesian hierarchical framework and the MCMC algorithm to fit this model.

2.4.1 Estimate Class Membership in Gaussian Mixture Models

We treat class membership estimation as a clustering problem. With the distribution of $p(\mathbf{x}_{c,i}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, we can fit the mixture model to compute $p(z_{c,i}|\mathbf{x}_{c,i})$, which indicates posterior probability that the entry belongs to the cth component with $\mathbf{x}_{c,i}$ having been observed on it. This value can be easily evaluated by applying Bayes theorem

$$p(z_{c,i}|\mathbf{x}_{c,i}) = \frac{\pi_c N(\mathbf{x}_{c,i}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c'=1}^{C} N(\mathbf{x}_{c',i}|\boldsymbol{\mu}_{c'}, \boldsymbol{\Sigma}_{c'})}. \tag{2.6}$$

The cth mixing proportion π_c can be viewed as the prior probability that entity belongs to the cth component of the mixture $c = 1, \ldots, C$. The posterior probability $p(\mathbf{x}_{c,i}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ is the corresponding conditional probability known as the responsibility once we have observed.

Note that a label switching problem arises if we want samples from MCMC. That is if we calculate $p(z_{c,i}|\mathbf{x}_{c,i})$ for each iteration, we are free to relabel all the classes, without affecting the segments of the data. This is normal since it is basically a clustering problem where the estimation of parameters does not rely on how the components are labeled. Ignoring this problem will result in severe consequences because the algorithm will never converge. One

way to avoid this is to impose identifiability constraint on the parameter space such as:

$$\pi_1 < \pi_2 < \ldots < \pi_C, \tag{2.7}$$

which is applied to our study described in the Chapter 2.6.

2.4.2 Prior Specification

In this subchapter, we provide the specification of priors in our models.

Let $\theta_{c,k,k'}$ represent the prior probability of being an edge between two sets k and k'. Let $g_{c,j,j'}$ be a latent indicator variable to represent whether elements j and j' are connected. These elements can be within the same set or across different higher levels. To be specific, if the pair of elements j and j' come from same set k, then the $\theta_{c,k,k'}$ becomes $g^k_{c,j,j'}$; if the pair of elements j and j' come from different set k and k', then the $\theta_{c,k,k'}$ becomes $g^{k,k'}_{c,j,j'}$. The matrix with entries $g_{j,j'}$ is called "adjacency matrix".

For each pair of element $1 \leq j \leq j' \leq p$, the prior distribution of regression coefficient $(\beta_{c,j,j'})$ for each edge depends on $g_{c,j,j'}$ and consider as the following distribution

$$p(\beta_{c,j,j'}|g_{c,j,j'}) = \begin{cases} N(0, v_1^2), & \text{if } g_{j,j'} = 1 \\ N(0, v_0^2), & \text{if } g_{j,j'} = 0, \end{cases} \tag{2.8}$$

where v_1 is far from zero but v_0 is close to zero, $v_1 \geq v_0 > 0$.

If $g_{c,i,j} = 0$, $\beta_{c,j,j'}$ has a prior with small variance v_0. We set $v_1 = \lambda v_0$. Parameter λ_c influences the prior probability of edge inclusion. In the context of Bayesian variable selection (George and McCulloch, 1993), λ_c should be chosen carefully so that it is larger enough to support edges in graphs but not so large for non-edges. We adopt the suggestion in the chapter: set

λ on the grid $10, 100, 1000, 1000, 1000000$ and we pick $\lambda = 1000$ for illustration through out this work. The prior called "Bernoulli-Gaussian model", has been widely used in the work of Bayesian variable selection (Kuo and Mallick, 1998). This is very similar to standard "spike and slab" prior. But rather than the "spike and slab" prior, this prior does not require summing over the "irrelevant" parameters. Thus, with "Bernoulli-Gaussian model" on the entries of the adjacency matrix, we are able to estimate the network structure and thus to obtain sparse inverse covariance matrix.

To facilitate designing the MCMC scheme, we would like to have conjugate priors for the rest of the parameters. Thus, the prior distributions of v_0^2, v_1^2, $g_{c,j,j'}$, and $\theta_{c,k,k'}$ are specified as inverse-gamma, Bernoulli, and Beta distributions, respectively, where are defined as

$$\begin{aligned} p(v_0^2) &= \frac{f^e}{\Gamma(e)}(v_0^2)^{-e-1}\exp(-\frac{f}{v_0^2}), \\ p(g_{c,j,j'}|\theta_{c,k,k'}) &= (\theta_{c,k,k'})^{g_{c,j,j'}}(1-\theta_{c,k,k'})^{(1-g_{c,j,j'})}, \\ p(\theta_{c,k,k'}) &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}(\theta_{c,k,k'})^{a-1}(1-\theta_{c,k,k'})^{b-1}. \end{aligned}$$

With these priors, one can easily derive a simple Gibbs sampling scheme, which will be fully discussed in Chapter 2.6.

The choice of hyperparameters plays an important role in the performance of our Bayesian multiple Gaussian graphical models. We now discuss them. First, we can tell the variance v_0^2 depends on hyperparameters e and f. It increases as e decreases and f increases. Similarly, it decreases as e increases and f decreases. Then, a large value of f and small value of e yield sparse low-variable network. Preliminary cross validation experiments showed that setting $e = 10,000$ and $f = 100$ led to good performance with regard to adjacency matrix estimation for a given class. These settings will be applied in all the case studies simulations. Then, we set Beta hyperparameters on $\theta_{c,k,k'}$ to $a = 1, b = 2$ no matter what $k = k'$ or $k \neq k'$. A

similar setting was used by Marlin and Murphy (2009), except in our work we want both lower-variable network and higher-variable to be sparse.

2.5 Posterior Sampling

In this chapter, we describe a sampling scheme for Bayesian multiple Gaussian graphical models. For estimating Bayesian multiple Gaussian graphical models from unknown classes, we conduct two steps: in the first step, we compute class membership probabilities; in the second step, we then estimate network structure for elements and then infer network structure for sets using elements.

This procedure is model-based with data observed only and other parameters unobserved. The unobserved parameters inferred from Chapter 2.2 have statistical meanings. The procedure of the first step can be illustrated as a directed graph as shown in Figure (2.3). Note that there the data $\mathbf{x}$ can be affected by both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. The focus of this chapter is an approach based on the covariance matrix $\boldsymbol{\Sigma}$.

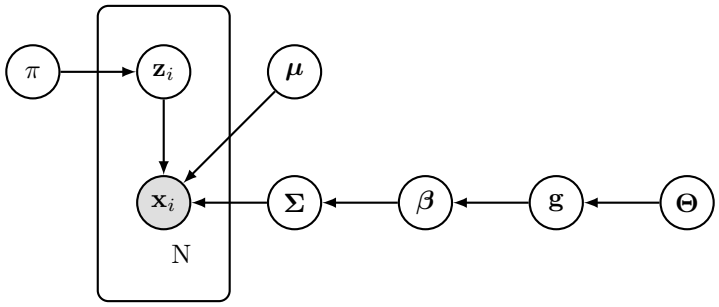


Figure 2.3: Directed acyclic graphical model representing the Bayesian mixture of Gaussians model, Only the $\mathbf{X}$ matrix is observed, all other quantities are inferred.

Fore the second step, the procedure of estimating element networks is described in Chapter 2.2.3. The procedure of learning set networks is demonstrated in the block array below. By using notation in Chapter 2.3, $P_{c,kk'}$ means the interaction event between sets pair k and k'

for given class c.

$$\begin{array}{c} \begin{matrix} 1 & \dots & k & \dots & K \end{matrix} \\ \begin{pmatrix} 1 & \dots & P_{c,1k} & \dots & P_{c,1K} \\ \dots & \ddots & \dots & \dots & \dots \\ P_{c,k1} & \dots & 1 & \dots & P_{c,kK} \\ \dots & \dots & \dots & \ddots & \dots \\ P_{c,K1} & \dots & P_{c,Kk} & \dots & 1 \end{pmatrix} \begin{matrix} 1 \\ \vdots \\ k \\ \vdots \\ K \end{matrix} \end{array}$$

2.5.1 Full Conditional Distribution

Let denote the parameters of interest in a vectorized fashion: $\boldsymbol{\beta}_{c,j} = [\beta_{c,j,1}, \beta_{c,j,2,}, \dots, \beta_{c,j,j-1}]^\top$, $\mathbf{g}_{c,j} = [g_{c,j,1}, g_{c,j,2,}, \dots, g_{c,j,j-1}]^\top$, $\mathbf{x}_{c,j} = [x_{c,1,j}, x_{c,2,j}, \dots, x_{c,N,j}]^\top$. Let $\boldsymbol{\Psi}_c$ be the set of parameters of interest.

We let $p(\mathbf{x}_{c,j}|\boldsymbol{\beta}_{c,j})$ represent the likelihood obtained from multivariate normal distribution. $p(\boldsymbol{\beta}_{c,j}|g_{c,j})$ is the prior distribution of $\boldsymbol{\beta}_{c,j}$. $p(\mathbf{g}_{c,j}|\theta_{c,k,k'})$ is the prior distribution of $\mathbf{g}_{c,j}$, and $p(\theta_{c,k,k'})$ is the prior distribution. Then, the joint posterior has the following form:

$$p(\boldsymbol{\Psi}_c) \propto \prod_{j=1}^{P-1} p\{(\mathbf{x}_{c,j}|\boldsymbol{\beta}_{c,j}) \cdot p(\boldsymbol{\beta}_{c,j}|\mathbf{g}_{c,j})\} \cdot \prod_{k=1}^{K-1} \{p(\mathbf{g}_{c,j}|\theta_{c,k,k'}) \times p(\theta_{c,k,k'})\}.$$

Since all the prior distributions are conjugate, the full conditional distributions have the closed forms which are summarized as follows:

- The full conditional distribution of $\boldsymbol{\beta}_j$:

$$\begin{aligned} p(\boldsymbol{\beta}_{c,j}|-) &\propto p(\mathbf{x}_{c,j}|\boldsymbol{\beta}_{c,j}) \cdot p(\boldsymbol{\beta}_{c,j}|\mathbf{g}_{c,j}); \\ &\propto \exp\{-\frac{1}{2\sigma_c^2}[\mathbf{x}_{c,j} - \mathbf{x}_{c,-j}\boldsymbol{\beta}_{c,j}]^\top[\mathbf{x}_{c,j} - \mathbf{x}_{c,-j}\boldsymbol{\beta}_{c,j}]\} \times \exp\{-\frac{1}{2}\boldsymbol{\beta}_{c,j}^\top \boldsymbol{\Sigma}_{c,g}^{-1}\boldsymbol{\beta}_{c,j}\}; \\ &\propto \exp\{-\frac{1}{2}[\boldsymbol{\beta}_{c,j}^\top(\sigma^{-2}\mathbf{x}_{c,-j}\mathbf{x}_{c,-j} + \boldsymbol{\Sigma}_{c,g}^{-1})\boldsymbol{\beta}_{c,j} - 2\boldsymbol{\beta}_{c,j}^\top(\sigma_c^2\mathbf{x}_{c,-j}\mathbf{x}_{c,j})]\}. \end{aligned}$$

  One can show that $[\boldsymbol{\beta}_{c,j}|-] \sim\ N(\boldsymbol{\mu}_c^N, \mathbf{V}_c^N)$, where

$$\begin{aligned} \boldsymbol{\mu}_c^N &= (\sigma_c^{-2}\mathbf{x}_{c,-j}\mathbf{x}_{c,-j} + \boldsymbol{\Sigma}_g^{-1})^{-1}(\sigma_c^2\mathbf{x}_{c,-j}\mathbf{x}_{c,j}), \\ \mathbf{V}_c^N &= (\sigma_c^{-2}\mathbf{x}_{c,-j}\mathbf{x}_{c,-j} + \boldsymbol{\Sigma}_{c,g}^{-1})^{-1}, \\ \boldsymbol{\Sigma}_{c,g} &= \text{diag}\{v_1^2\mathbf{g}_{c,j} + v_0^2(\mathbf{1} - \mathbf{g}_{c,j})\}. \end{aligned}$$

- The full conditional distribution of $\mathbf{g}_j$:

$$\begin{aligned} p(\mathbf{g}_{c,j}|-) &\propto p(\boldsymbol{\beta}_{c,j}|\mathbf{g}_{c,j}) \cdot p(\mathbf{g}_{c,j}|\theta_{c,k,k'}) \\ &\propto |\boldsymbol{\Sigma}_{c,g}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}\boldsymbol{\beta}_{c,j}^\top \boldsymbol{\Sigma}_{c,g}^{-1}\boldsymbol{\beta}_{c,j}\} \times (\theta_{c,k,k'})^{\mathbf{g}_{c,j}}(\mathbf{1} - \theta_{c,k,k'})^{1-\mathbf{g}_{c,j}}, \end{aligned}$$

  and sample $\mathbf{g}_{c,j}$ with probability $\Pr(\mathbf{g}_{c,j} = 1)$.

- The full conditional distribution of $\theta_{c,k,k'}$:

$$\begin{aligned} p(\theta_{c,k,k'}|-) &\propto p(\mathbf{g}_{c,j}|\theta_{c,k,k'}) \cdot p(\theta_{c,k,k'}|a,b) \\ &\propto (\theta_{c,k,k'})^{\sum_{j'>j}\mathbf{g}_{c,j}}(\mathbf{1} - \theta_{c,k,k'})^{\sum_{j'>j}(1-\mathbf{g}_{c,j})} \times (\theta_{c,k,k'})^{a-1}(\mathbf{1} - \theta_{c,k,k'})^{b-1} \\ &\propto (\theta_{c,k,k'})^{\sum_{j'>j}\mathbf{g}_{c,j}+a-1}(\mathbf{1} - \theta_{c,k,k'})^{\sum_{j'>j}(1-\mathbf{g}_{c,j})+b-1} \\ &\propto (\theta_{c,k,k'})^{a^N-1}(\mathbf{1} - \theta_{c,k,k'})^{b^N-1}, \end{aligned}$$

  where $a^N = \sum_{j'>j}\mathbf{g}_{c,j} + a$ and $b^N = \sum_{j'>j}(1 - \mathbf{g}_{c,j}) + b$.

We now present the MCMC algorihm 1 for solving Bayesian multiple GGMs with unknown classes in greater detail.

Algorithm 1 MCMC algorihm for Bayesian multiple GGMs with unknown classes.

initialize $\Psi_c = \{\boldsymbol{\pi}_c, \boldsymbol{\mu}_c, \boldsymbol{\beta}_c, \mathbf{g}_c, \boldsymbol{\Theta}_c\}$ for each class c.
Select prior
for each iteration **do**:
    Evaluate the data class membership by $p(z_{c,i}|\mathbf{x}_{c,i}) = \frac{\pi_c N(\mathbf{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c'=1}^{C} N(\mathbf{x}_i|\boldsymbol{\mu}_{c'}, \boldsymbol{\Sigma}_{c'})}$.
    for each class $c = 1, \ldots, C$ **do**
        for each elements $j = 2, \ldots, P$ **do**
            Update $\boldsymbol{\beta}_{c,jj'}$ and $\mathbf{g}_{c,jj'}$
        end for
        for each set $k = 1, \ldots, K$ **do**
            Update $\theta_{c,k,k'}$
        end for
        Compute estimated precision matrix for each class by $\boldsymbol{\Sigma}_c^{-1} = \mathbf{L}_c^{\top} \mathbf{D}_c \mathbf{L}_c$
    end for
end for
Return: posterior samples of $\mathbf{g}_c, \mathbf{z}_c$ for each class c.

2.5.2 Posterior Inference

In this subchapter, we represent two ways to estimate the posterior of edge inclusion (i.e. $g_{c,jj'} = 1$) for each edge. Both approaches select the edges marginally.

For the very first practical approach : we obtained the posteriors of edge inclusion after the burn-in. Then, we choose the edges whose posteriors of inclusion are larger than 0.5, which was used by Barbieri and Berger (2004) and Peterson et al. (2015). We refer to this approach as "BMGGM1".

The second approach is based on testing. We treat selection of edges as hypothesis testing problem for Bernoulli distribution where $\hat{g}_{c,jj'} \sim N\big(g_{c,jj'}, SE(\hat{g}_{c,jj'})\big)$, where $SE(\hat{g}_{c,jj'}) = \sqrt{\frac{\hat{g}_{c,jj'}(1-\hat{g}_{c,jj'})}{N}}$ and N is the number of MCMC simulations. The central limit theorem tells

us that for large N

$$Z = \frac{g_{c,jj'} - \hat{g}_{c,jj'}}{SE(\hat{g}_{c,jj'})} \sim N(0,1).$$

Then the approximate $1-\alpha$ confidence intervals for $g_{c,jj'}$ be

$$\big[\hat{g}_{c,jj'} - Z^*_{\alpha/2} \times SE(\hat{g}_{c,jj'}), \quad \hat{g}_{c,jj'} + Z^*_{\alpha/2} \times SE(\hat{g}_{c,jj'})\big],$$

where $Z_{\alpha/2}$ is a critical value and α is the significance level. Finally, we select the edges if the corresponding confidence interval does not contain 0. We refer to this approach as "BMGGM2".

Furthermore, we may compare differential networks among different classes. Here, we define differential element networks if the value of $|g_{c,jj'} - g_{c',jj'}| = 1$. It means the element pair (j, j') exist in the network of either class c or that of class c' but not both. Similarly, we would like to test the following null hypothesis against the given alternative

$$H_0 : g_{c,jj'} = g_{c',jj'} \quad \text{versus} \quad H_a : g_{c,jj'} \neq g_{c',jj'}.$$

So the test statistic is

$$Z = \frac{\hat{g}_{c,jj'} - \hat{g}_{c',jj'}}{\sqrt{\hat{g}_{jj'}(1-\hat{g}_{jj'})(\frac{1}{N}+\frac{1}{N})}},$$

where $\hat{g}_{jj'} = \frac{N\hat{g}_{c,jj'} + N\hat{g}_{c',jj'}}{N+N} = \frac{1}{2}(\hat{g}_{c,jj'} + \hat{g}_{c',jj'})$

Then the approximate $1-\alpha$ confidence intervals for $g_{c,jj'}$ be

$$\big[\hat{g}_{c,jj'} - \hat{g}_{c',jj'} - Z^*_{\alpha/2} \times \sqrt{\hat{g}_{jj'}(1-\hat{g}_{jj'})(\frac{1}{N}+\frac{1}{N})}, \quad \hat{g}_{c,jj'} - \hat{g}_{c',jj'} + Z^*_{\alpha/2} \times \sqrt{\hat{g}_{jj'}(1-\hat{g}_{jj'})(\frac{1}{N}+\frac{1}{N})}\big].$$

Likewise, we claim the edge to be differential if the corresponding confidence interval does

not contain 0.

A similar idea can be extended to more than two differential networks by using multiple comparisons.

In the later experiments, we found approach 1 (BMGGM1) with a fixed threshold of 0.5 resulted in better performance than approach 2 (BMGGM2). We will focus on approach 1 through out this chapter unless specified otherwise.

2.6 Simulation

In this chapter, we present two simulation studies that evaluate the performance of our Bayesian multiple Gaussian graphical models. We consider two simulation settings. In the first experiment, we estimate all the parameters of interest and make the inference of Gaussian graphical model, assuming that the class labels of observations are given. We also compare our BMGGM approach with the existing methods. The second experiment is similar to the first one except that we assume class labels are unknown. We will evaluate the ability of the model to make predictions by learning differential graphs across the classes.

2.6.1 Simulation Study for Known Classes

Simulation Setting

In this subchapter, we run a simulation to evaluate the performance of our algorithm when the class is given. We start by drawing $n = 100$ samples independently and identically from multivariate Gaussian distribution $N(\mathbf{0}, \mathbf{\Sigma}_c^{-1})$. Since the class is given, we simplify out notation by letting $\mathbf{\Sigma} = \mathbf{\Sigma}_c$.

The inverse covariance matrix $\mathbf{\Sigma}$ depends on the corresponding network structure. To form the adjacency matrix, we create a matrix with ones on its diagonal and with zeroes on entries not corresponding to network edges. To simulate the adjacency matrix for both elements and sets, we proceed the following steps,

1. We create $K = 6$ set networks first based on one of the network types (AR(2) network, chain network, and scale-free network). Each set network has $p_k = 10$ element. Thus, we can have corresponding adjacency matrices on the diagonal block of $\mathbf{\Sigma}$.

2. For the off-diagonal block of $\mathbf{\Sigma}$, we set entries to zeros except for $g_{4,17} = 1$ where 4th element and 17th element belong to set 1 and 2 respectively. In this way, we make set 4 and 17 connected based the assumption A1 from Chapter 2.3. That is, $g_{4,17}^{(1,2)} = 1$ because element 4 interacts with the element 17 in set pair 1 and 2 for a given class.

For elements, we consider three types of simulated network: AR(2) network, chain network, and scale-free network which are shown in Figure 2.4.

We explain how to generate these networks in detail:

- AR(2) network: This is also called *second-order autoregression* model. For a given set, the corresponding precision matrix $\mathbf{\Omega}$ is set with entries $\omega_{j,j} = 1$, for $j = 1, \cdots, P$. $\omega_{j,j+1} = \omega_{j+1,j} = 0.5$, for $j = 1, \cdots, P-1$. $\omega_{j,j+2} = \omega_{j+2,j} = 0.4$ for $j = 1, \cdots, P-2$.

- Chain network: this network structure corresponds to a band matrix which can be created by adding nonzero entries of diagonal matrix. It resembles a tridiagonal precision matrix (Fan and Li, 2001).

- Scale-free network: we generate set networks using the Barabasi-Albert algorithm (Barabási and Albert, 1999), each with a power law degree distribution. That is,

for a given set, the degree distribution

$$p(k) \propto k^{-\alpha}, \tag{2.9}$$

where α is some prefixed parameter. Power-law degree distributions can mimic the network structure of biological data (Chen and Sharp, 2004) and are usually more difficult to learn the other type of network structure (Peng et al., 2009).

To create a symmetric and positive-definite covariance matrix $\mathbf{\Sigma}$, we first replace the first in the adjency matrix with non-zeros values from a uniform distribution on $[-1, -0.5] \cup [0.5, 1]$. Then, we proceed the following Steps 2.1-2.4 used by Danaher et al. (2014), where we divide each off-diagonal elements in its row and average the matrix with its transpose:

1. Convert the network structure to corresponding adjacency matrix (i.e. $(0, 1)$ -matrix with zeros on its diagonal);
2. Replace the first entries with other nonzeros entries $\sim$ uniform$(0.5, 1)$;
3. Make the matrix positive definite by dividing each off-diagonal entry by the sum of the absolute values of the off-diagonal entries in its row, and averaging the matrix with its transpose;
4. Each observation is drawn independently and identically from multivariate Gaussian distribution $N(\mathbf{0}, \mathbf{\Sigma}^{-1})$.

2.6.2 Evaluation Metrics

Following the standard practice for the Gaussian graphical model, we assess the performance of the Gaussian graphical model by focusing on the estimation of adjacency matrix or network

structure for each set ks, (where $k = 1, \ldots, 6$). The key measure of the model should reveal predictive power to obtain an accurate sparse matrix, not only the ability to detect the edges but also non-edges.

A number of metrics are worth assessing: true positive rate (TPR), true negative rate (TNR), and accuracy (ACC). To calculate these metrics, we first define false positive (FP), true positive (TP), false negative (FN), and true negative (TN) for the edge status of pair of nodes, $g_{j,j'}$

$$\begin{aligned}
\mathrm{TP}^k &= \sum_{1<j<j'<P} \mathbf{I}\{g^k_{j,j'} = 1, \hat{g}^k_{j,j'} = 1\};\\
\mathrm{FP}^k &= \sum_{1<j<j'<P} \mathbf{I}\{g^k_{j,j'} = 0, \hat{g}^k_{j,j'} = 1\};\\
\mathrm{TN}^k &= \sum_{1<j<j'<P} \mathbf{I}\{g^k_{j,j'} = 0, \hat{g}^k_{j,j'} = 0\};\\
\mathrm{FN}^k &= \sum_{1<j<j'<P} \mathbf{I}\{g^k_{j,j'} = 1, \hat{g}^k_{j,j'} = 0\}.
\end{aligned} \tag{2.10}$$

Accordingly, we can calculate the true positive rate (TPR), true negative rate (TNR) and accuracy (ACC) for each higher-level variable and average them:

$$\begin{aligned}
\mathrm{TPR} &= \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\mathrm{TP}^k}{\mathrm{TP}^k + \mathrm{FN}^k}\right);\\
\mathrm{TNR} &= \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\mathrm{TN}^k}{\mathrm{FP}^k + \mathrm{TN}^k}\right);\\
\mathrm{ACC} &= \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\mathrm{TP}^k + \mathrm{TN}^k}{\mathrm{TP}^k + \mathrm{FP}^k + \mathrm{FN}^k + \mathrm{TN}^k}\right).
\end{aligned} \tag{2.11}$$

Since the inverse covariance matrix is rather sparse, so we have far more non-edges than edges for a given graph. We also introduce precision and recall which are widely used in

information retrieval and binary classification. We can define them as the follows,

$$\begin{aligned} \text{Precision} &= \frac{1}{K}\sum_{k=1}^{K}(\frac{\text{TP}^k}{\text{TP}^k + \text{FP}^k}); \\ \text{Recall} &= \frac{1}{K}\sum_{k=1}^{K}(\frac{\text{TP}^k}{\text{TP}^k + \text{FN}^k}). \end{aligned} \tag{2.12}$$

The precision measures "how many selected items are relevant" and the recall measures "how many relevant items are selected".

2.6.3 Simulation Result

We implement a Bayesian MCMC procedure described in Chapter 2.5 for obtaining samples from the posterior probabilities. To make inference, we run our Gibbs sampler for 10,000 iterations with another 10,000 as burn-in. In this subchapter, we will showcase our experiment results. We only show results of scale-free network as it is the hardest network to learn and the rest would be on the appendix. We compare our method with the other three methods on the same dataset. We first apply the graphical lasso by Friedman et al. (2008) (referred as "Glasso"). Then, we apply the group graphical lasso (referred as "GGL") and fused graphical lasso (referred as "FGL") proposed by Danaher et al. (2014). All the tuning parameters are selected to give the lowest cross-validation average error.

We begin with assessing whether the MCMC procedure can converge to the stationary distribution. Figure 2.5 demonstrates the trace plots for the number of edges for each set network, which indicates good mixing rates of Markov chains.

To estimate adjacency matrix, we obtain the posterior probability of edge inclusion by calculating the average for the MCMC samples of $g_{j,j'}$. The heatmaps for posterior probability of edge inclusion are displayed in Figure 2.6, 2.7 and 2.8, the patterns of which show good

recovery of true network structure.

Figure 2.9 shows receiver operator curves (ROC) for each set network with the number of true positive rate and false positive rate described in Chapter 2.6.

We compare our method with the other three methods Glasso, GGL, and FGL on the same dataset. The result of the estimated network structure is summarized in Table 2.1. We assess the accuracy of the estimation of network structure via precision, recall and ACC together instead of ACC alone since the graphs we have are rather sparse. We average these results over 4 set networks to make comparisons with other methods.

	TNR	Precision	Recall	ACC
GLasso	0.45(0.07)	0.31(0.03)	1.00(0.00)	0.56(0.06)
FGL	0.98(0.02)	0.82(0.12)	0.52(0.00)	0.88(0.05)
GGL	0.92(0.03)	0.77(0.08)	1.00(0.00)	0.93(0.02)
BMGGM1	0.98 (0.01)	0.92(0.07)	0.78(0.10)	0.94(0.02)
BMGGM2	0.79 (0.01)	0.57(0.05)	0.98(0.01)	0.82(0.05)

Table 2.1: Comparison of five methods in terms of four measures with standard error (SE) over 100 simulated runs under scale-free network case; GLasso=graphical lasso (Friedman et al, 2008); GGL=group graphical lasso (Danaher et al, 2014); FGL=fused graphical lasso (Danaher et al, 2014); BMGGM1=our proposed method with thresholding; BMGGM2=our proposed method with testing.

Results suggest that our Bayesian multiple Gaussian graphical models have the best accuracy overall. Group graphical lasso perform slightly better than fused graphical lasso, which is implied in Danaher et al. (2014). Graphical lasso has the lowest accuracy, due to the fact that it learns each graph individual without considering the connection between the set network. With these simulation settings, compared to joint graphical lasso, our Bayesian multiple Gaussian graphical models will overestimate the edges, which lead to relatively higher true positive rate and recall, and relatively lower true negative rate and precision.

Lastly, our Bayesian multiple Gaussian graphical models are able to identify the connection

between the sets, where the $\Pr(P_{12} = 1) = 1$, as we can clearly see Figure 2.10 where there is an edge across the set 1 and set 2.

We also computed the adjacency matrix without condition 3 and discoved several false positives: $\Pr(P_{16} = 1) = 0.55$, $\Pr(P_{34} = 1) = 0.69$, and $\Pr(P_{56} = 1) = 0.62$. Thus, to calculate the set connection, using condition 3 has a better performance.

2.6.4 Simulation Study for Unknown Classes

Simulation Setting

In this subchapter, we run a simulation to evaluate the performance of our algorithm when the class is not given. For each class, we start by drawing samples independently and identically from multivariate Gaussian distribution $N(\mathbf{0}, \mathbf{\Sigma}_c^{-1})$, where $c = 1, 2$. The sample size here is 500, with 100 cases and 400 controls.

For the covariance matrices, we consider $K = 10$ sets and $p_k = 10$ elements each. $\mathbf{\Sigma}_1$ is created the same way we described in Subchapter 2.6.1. To form $\mathbf{\Sigma}_2$, we remove the edges randomly in $\mathbf{\Sigma}_1$ (i.e. setting the non-zero entries in corresponding adjacency matrix to zeros) and add new edges randomly by replacing zero entries with non-zero values. We let the proportion of differential edge for these two graphs be 20%.

Evaluation Metrics

The evaluation criterion would be the same in Subchapter 2.6.2. Likewise, we redefine false positive (FP), true positive (TP), false negative (FN), true negative (TN) and accuracy

(ACC) for the class of observations:

$$\begin{aligned} \text{TP} &= \sum_{i=1}^{n} \mathbf{I}\{c=1, \hat{c}=1\}, \quad \text{FP} = \sum_{i=1}^{n} \mathbf{I}\{c=2, \hat{c}=1\}; \\ \text{TN} &= \sum_{i=1}^{n} \mathbf{I}\{c=2, \hat{c}=2\}, \quad \text{FN} = \sum_{i=1}^{n} \mathbf{I}\{c=1, \hat{c}=2\}; \\ \text{ACC} &= \frac{1}{N} \sum_{i=1}^{n} \left(\frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}\right). \end{aligned} \tag{2.13}$$

Results

As this is a clustering problem, a label switching problem may rise at each iteration. We solve this by adding a constraint $n_1 < n_2.$, that is, $\pi_1 < \pi_2$. Figure 2.11 displays the algorithm converges as posterior predictions of the number of cases and controls are more spread out at two ends. With this, the algorithm is able to achieve 81.2% accuracy over 100 simulated runs.

2.7 Application

In this chapter, we apply our Bayesian multiple Gaussian graphical models to reveal the dependence structure for the breast cancer gene expression data where genes can be viewed as elements and sets can be viewed as sets.

According to literature, American white women were slightly more likely to develop breast cancer than African American women (Chlebowski et al., 2005). Therefore, our goal is to compare differential networks among racial groups to better understand the genetic difference. These differential networks include not only the pathway network but also the gene network with each pathway. Specifically, we consider a gene-gene pair to be differential if

the true value of $|g_{1,j,j'} - g_{2,j,j'}| = 1$ in our example which reflects that edge (j, j') is included in either white women or non-white women but not both. This means the edge between a gene-gene pair is different in white women and non-white women.

The human breast cancer data set was collected from the University of Texas M.D. Anderson Cancer Center (Shi et al., 2010) containing $22,283$ genes expression measurements across 176 white patients and 102 non-white patients. Furthermore, the genes were mapped into $1,320$ pathways using the Canonical pathways (CP) from the Molecular Signatures Database (MsigDB), with the number of genes within each pathway ranges from 4 to 778. We apply our method to the top 3 significant pathways (P711, P717, and P956) expressed between white and non-white women breast cancer patients. So in this dataset, we have $n = 278$ samples, $c = 2$ classes and $K = 3$ sets. The number of genes in the these 3 pathways (P711: REACTOME ORC1 REMOVAL FROM CHROMATIN, P717: REACTOME SIGNALING BY ERBB2, P956: REACTOME DOWNSTREAM SIGNAL TRANSDUCTION) are $p_{P956} = 59$, $p_{P711} = 21$ and $p_{P717} = 17$. The dimension of the data matrix is 97.

We normalize all the genes with zero means and one standard deviation for each class. Since our goal is to generate meaningful patterns and create a hypothesis, we set the hyperparameters to yield sparse pathway network and gene network. We choose Beta hyperparameters on $\theta_{c,k,k'}$ to $a = 1, b = 2$ no matter what $k \neq k'$ or $k \neq k'$. We set Gamma hyperparameters $e = 10,000$ and $f = 1000$ to further enforce the sparsity.

In Figure 2.12, 2.13, we present gene networks for white women and non-white women, where the color represents the pathways (genes can be grouped into pathways). The results show that the element structures for these two groups are very different. Furthermore, the graph of white women has more edges (144 edges) than that of non-white women (114 edges), and only 51 edges were overlapped. Comparing the vertex degrees (i.e. the number of nearest neighbors of a vertex) of the two networks, we found that the white women network had

more genes with large degrees (2.14).

On the pathway level, we identified pathway P956 are conditionally dependent on pathway P717 in white women while P956 and P711 pathways are conditionally independent. The adjacency matrix for these two groups are summarized below:

$$\mathbf{\Theta}_{\text{white women}} = \begin{pmatrix} 1.000 & 1.000 & 1.000 \\ 1.000 & 1.000 & 1.000 \\ 1.000 & 1.000 & 1.000 \end{pmatrix}.$$

$$\mathbf{\Theta}_{\text{non white women}} = \begin{pmatrix} 1.000 & 1.000 & 1.000 \\ 1.000 & 1.000 & 0.006 \\ 1.000 & 0.006 & 1.000 \end{pmatrix}.$$

Analysis of gene network and pathway network both reveal results consistent with the breast cancer genomics (Keenan et al., 2015) that American white women were slightly more likely to develop breast cancer than African American women.

Overall, the novel application of our Bayesian multiple Gaussian graphical models to understand breast cancer networks has yielded results consistent with the known literature and identified potential biomarkers and pathways for future research.

On the other hand, we are also interested in applying Bayesian multiple Gaussian graphical models when the class is unknown. We treat race (i.e. white and non-white women) as a class and evaluate the performance in terms of prediction rate.

We apply the same settings in the previous chapter. That is, we choose P956, P711, and P717 as pathways. We normalize all the genes with zero means and one standard deviation. The

hyperparameters are also same in order to yield sparse pathway network and gene network: set Beta hyperparameters on $\theta_{c,k,k'}$ to $a = 1, b = 2$ no matter what $k \neq k'$ or $k \neq k'$ and Gamma hyperparameters $e = 10,000$ and $f = 1000$ to further enforce the sparsity.

In addition, we force a constraint ($n_1 < n_2$, where n_1 is the number of cases and n_2 is the number of controls) to prevent the label switching problem, the proposed method is able to achieve prediction rate of 64.7%. The reason that we have a prediction rate of 64.7% is that all the samples have diseases. we discovered a very unbalanced clustering result, that is one class covered almost all the observations while the other class had only a few samples.

2.8 Discussion

We have proposed the Bayesian multiple Gaussian graphical models, a framework for learning networks structure of multiple groups that could be possibly connected to each other. Given a data set that has a hierarchical structure, this model is able to uncover the network structure for set and element simultaneously. Furthermore, our approach can evaluate the Bayes factor to decide the class membership for each class. We design and employ a simple Gibbs sampling scheme to solve the Bayesian multiple Gaussian graphical models.

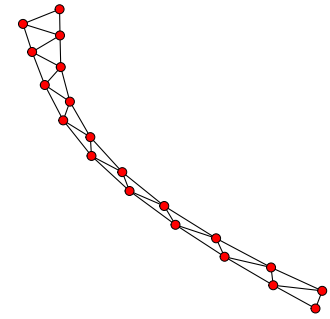
Our method is comparable to the existing method in terms of accuracy of adjacency matrix estimation. Our method also provides a good statistical interpretation through posterior probabilities for each parameter, because our generative model is more explainable and interpretable over deterministic models. We define the model by explaining the generative mechanism in a top-down fashion 2.3.

In this chapter, we make several novel contributions: (1) development of a Bayesian multiple Gaussian graphical models that is able to reveal set networks and element networks simul-

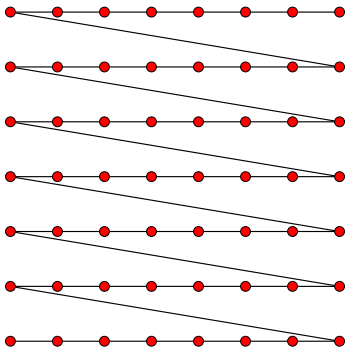
taneously; (2) establishment of a Bayesian factor approach to evaluate class membership for the observations from Gaussian graphical model; (3) estimation of the model by using Bayesian rather than frequentist approach. This Bayesian formulation is fully model-based, thus has nice probabilistic interpretations. Clustering data for the Gaussian graphical model is very difficult and only a few works focus on the relevant problem so far.

There are several potentials for future work. For instance, when it comes to clustering the observations, one could discuss the properties of the Bayesian method in terms of convergence. There are rooms for improving clustering accuracy. One may also work on developing a fast and scalable Bayesian algorithm to apply to larger datasets. Another possibility is to extend our model to allow discrete data.

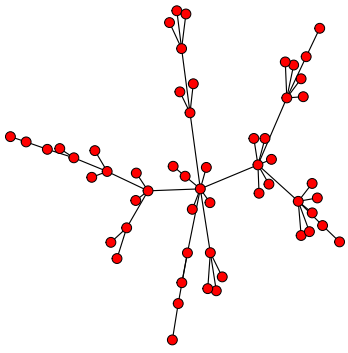
An `R` package implementing Bayesian multiple Gaussian graphical models with unknown classes is available on the author's Github repository.



(a)



(b)



(c)

Figure 2.4: (a) AR(2) network. (b) Chain network. (c) Scale-free network.

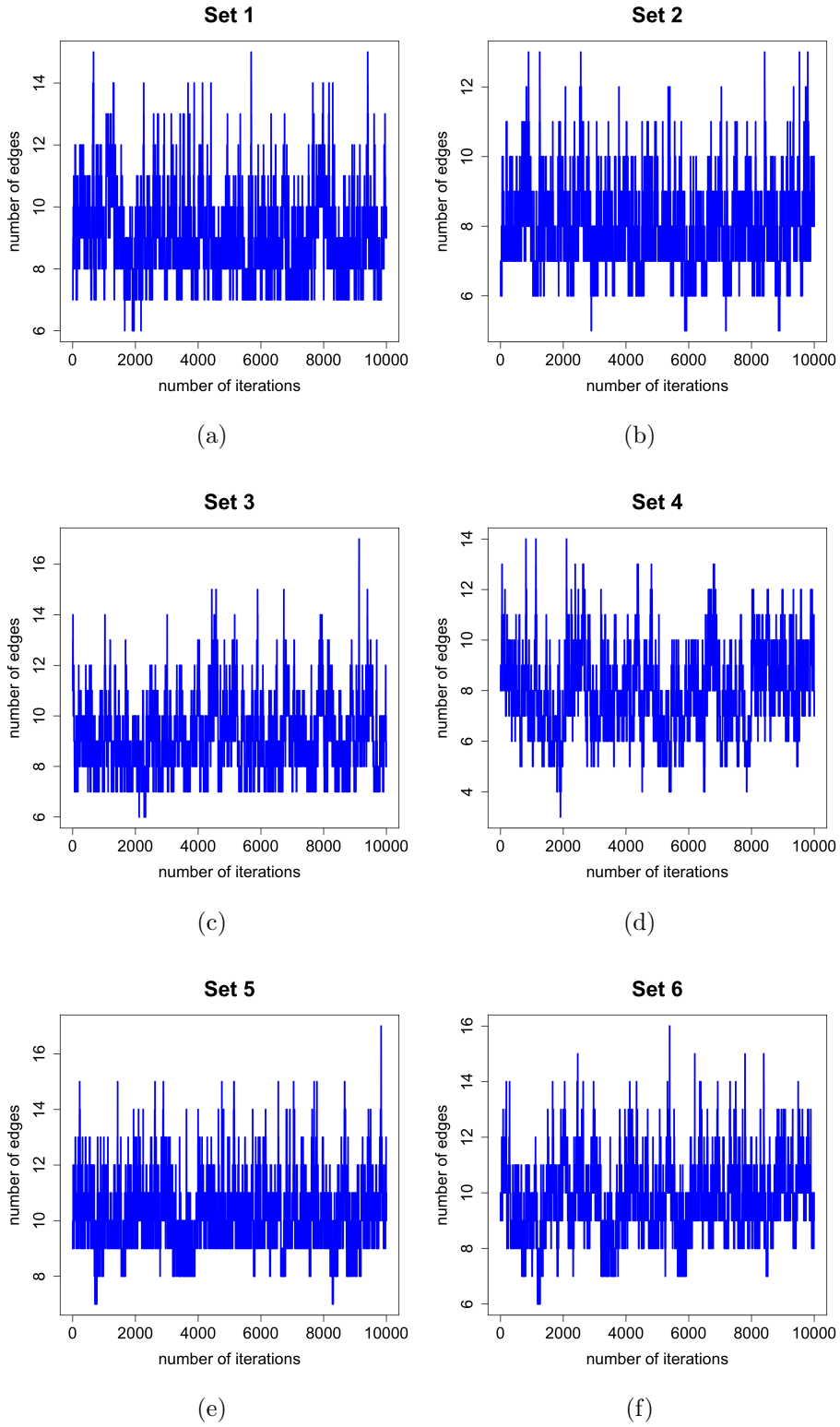


Figure 2.5: Trace plot of the number of edges for 6 sets based on 10, 000 MCMC iterations, where x-axis is the number of iterations and y-axis is the number of edges under scale-free network case.

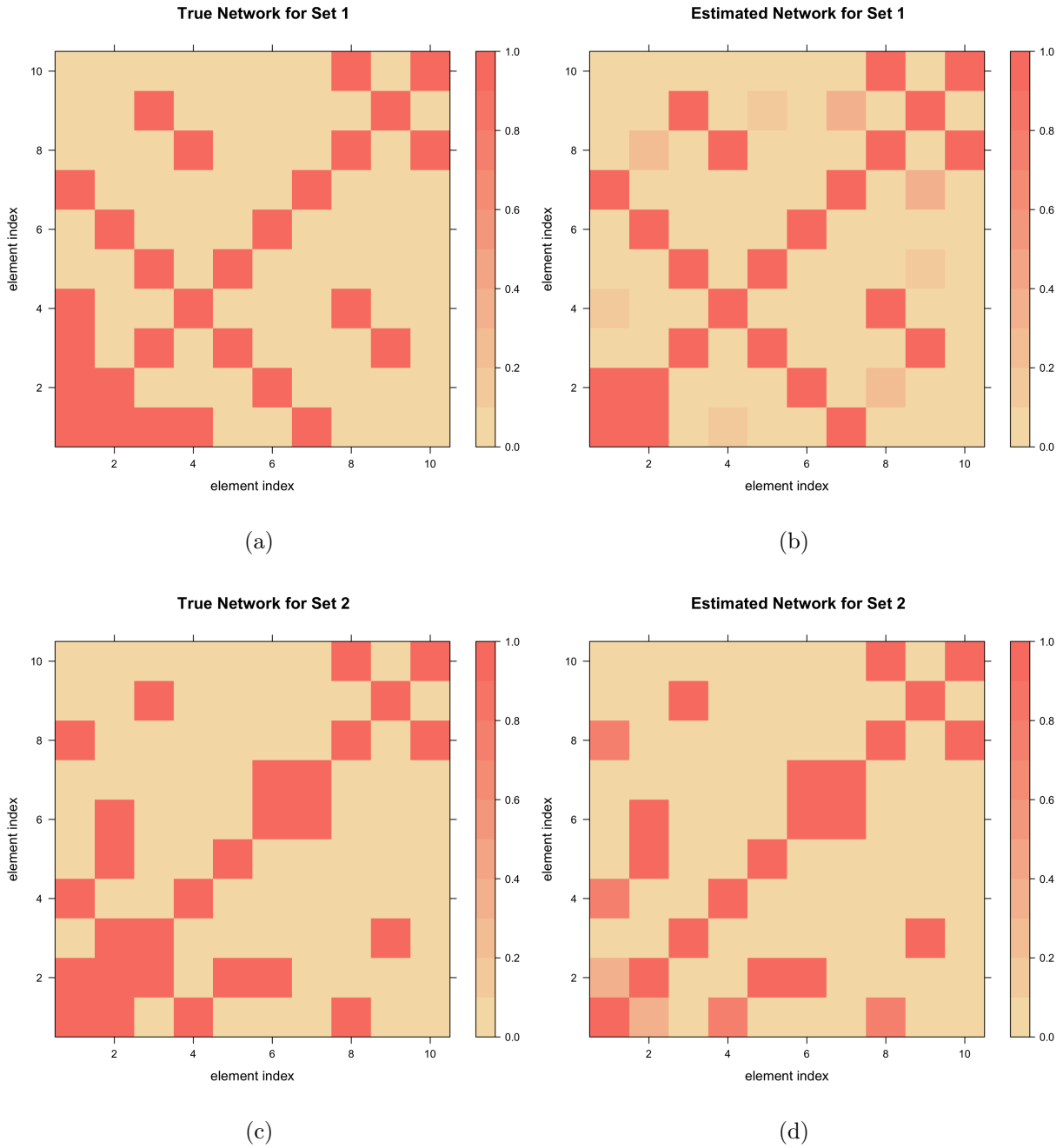


Figure 2.6: True and estimated heat maps for the set 1-2 simulated sets networks under scale-free network case: Figures (a) and (c) are ground truth and Figures (b) and (d) are estimated.

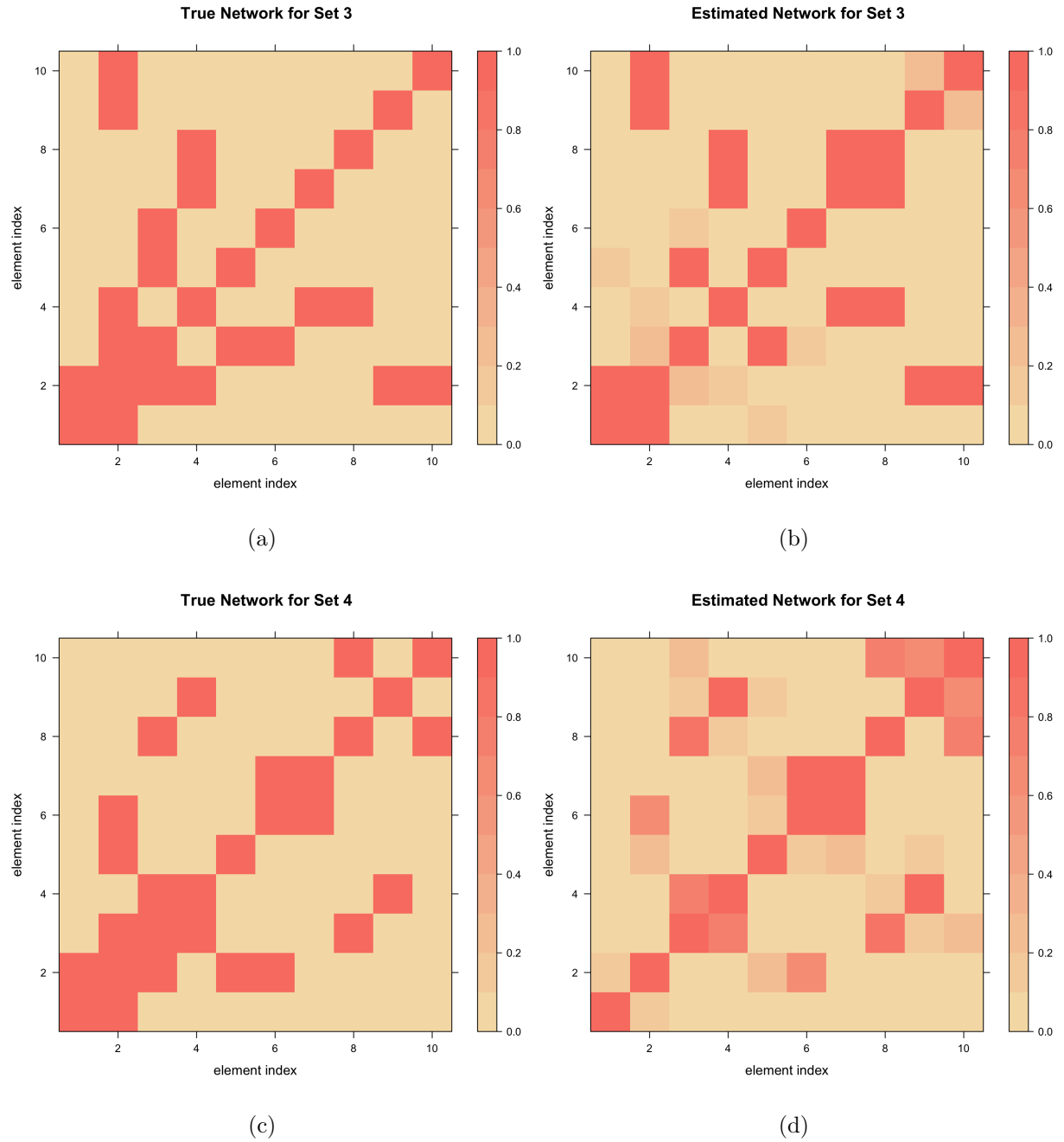


Figure 2.7: True and estimated heat maps for the set 3-4 simulated sets networks under scale-free network case: Figures (a) and (c) are ground truth and Figures (b) and (d) are estimated.

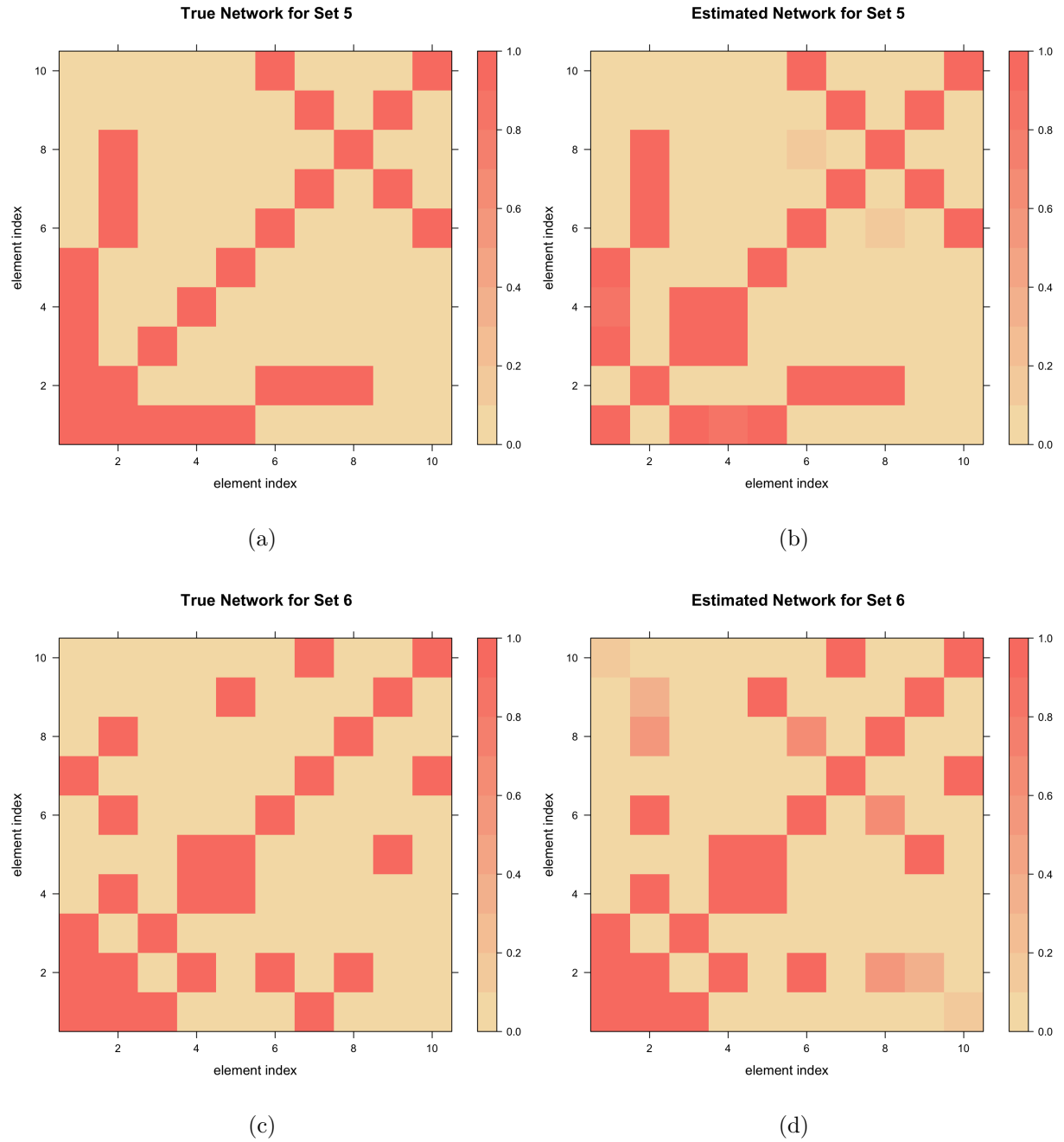


Figure 2.8: True and estimated heat maps for the set 5-6 simulated sets networks under scale-free network case: Figures (a) and (c) are ground truth and Figures (b) and (d) are estimated.

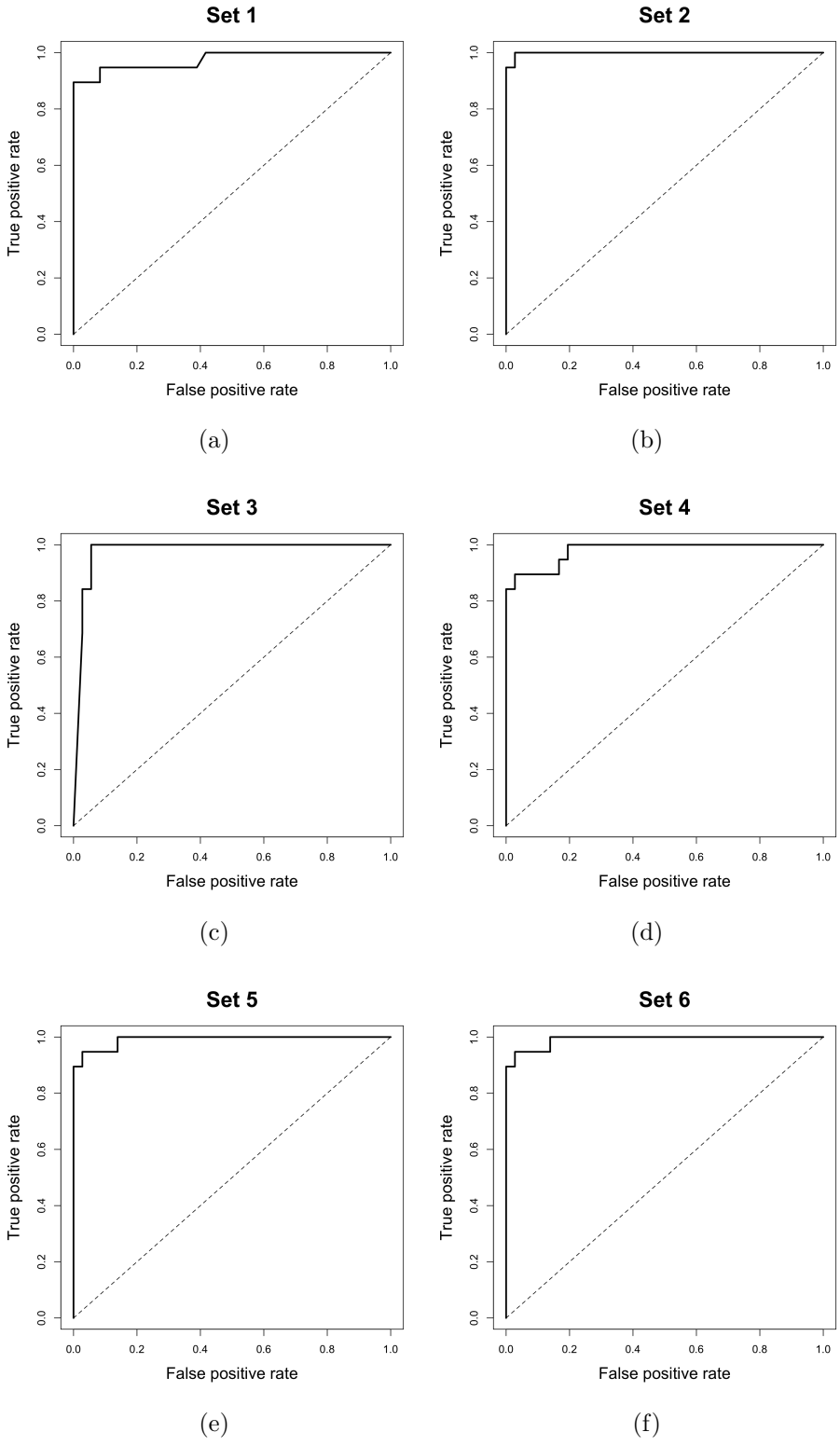


Figure 2.9: ROC curves for the posterior probability of edge inclusion with varying thresholds for 6 set networks under scale-free network case.

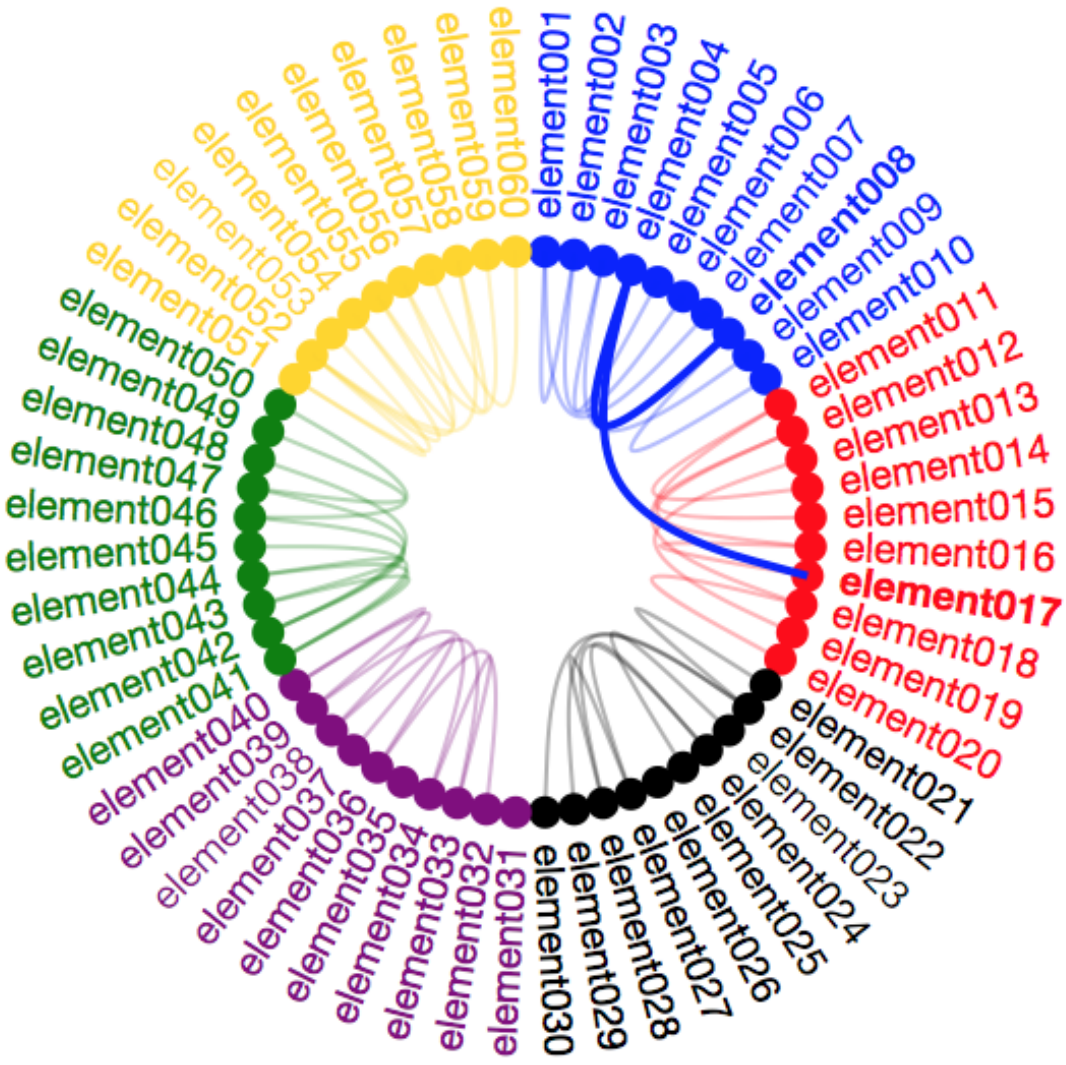


Figure 2.10: Inferred networks for set and element variable under scale-free network case.

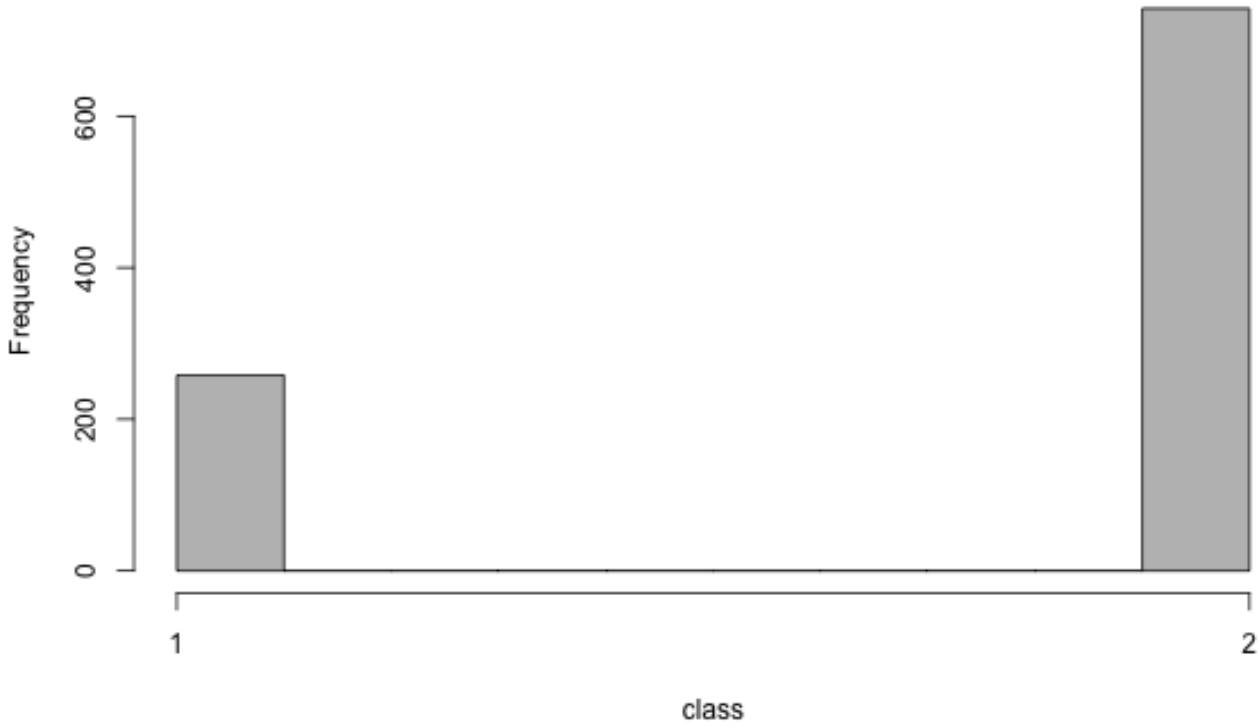


Figure 2.11: Posterior predictions of the number of cases and controls by Bayesian multiple Gaussian graphical model.

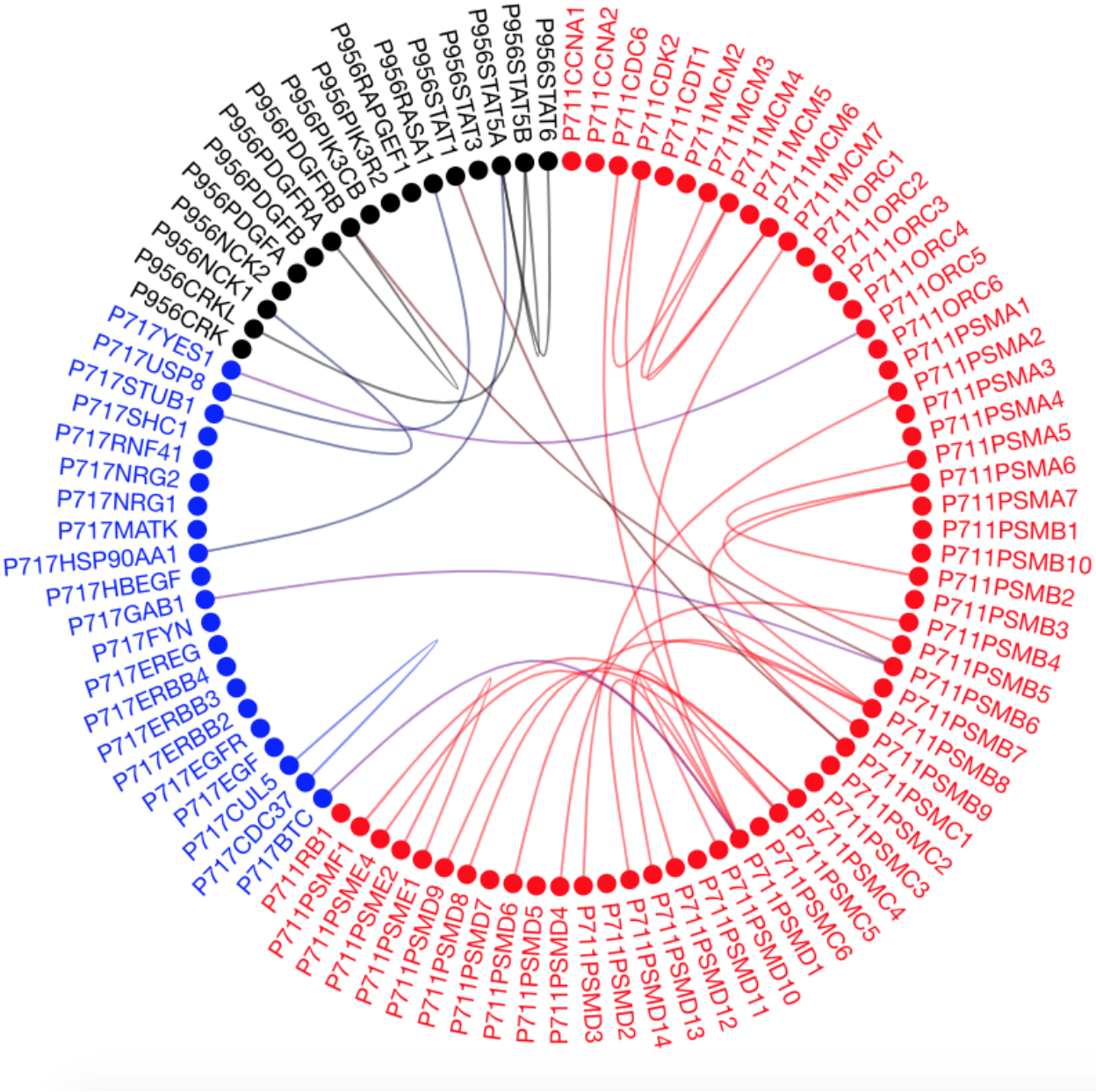


Figure 2.12: Estimated biological gene pathway for American white women in breast cancer gene expression application data.

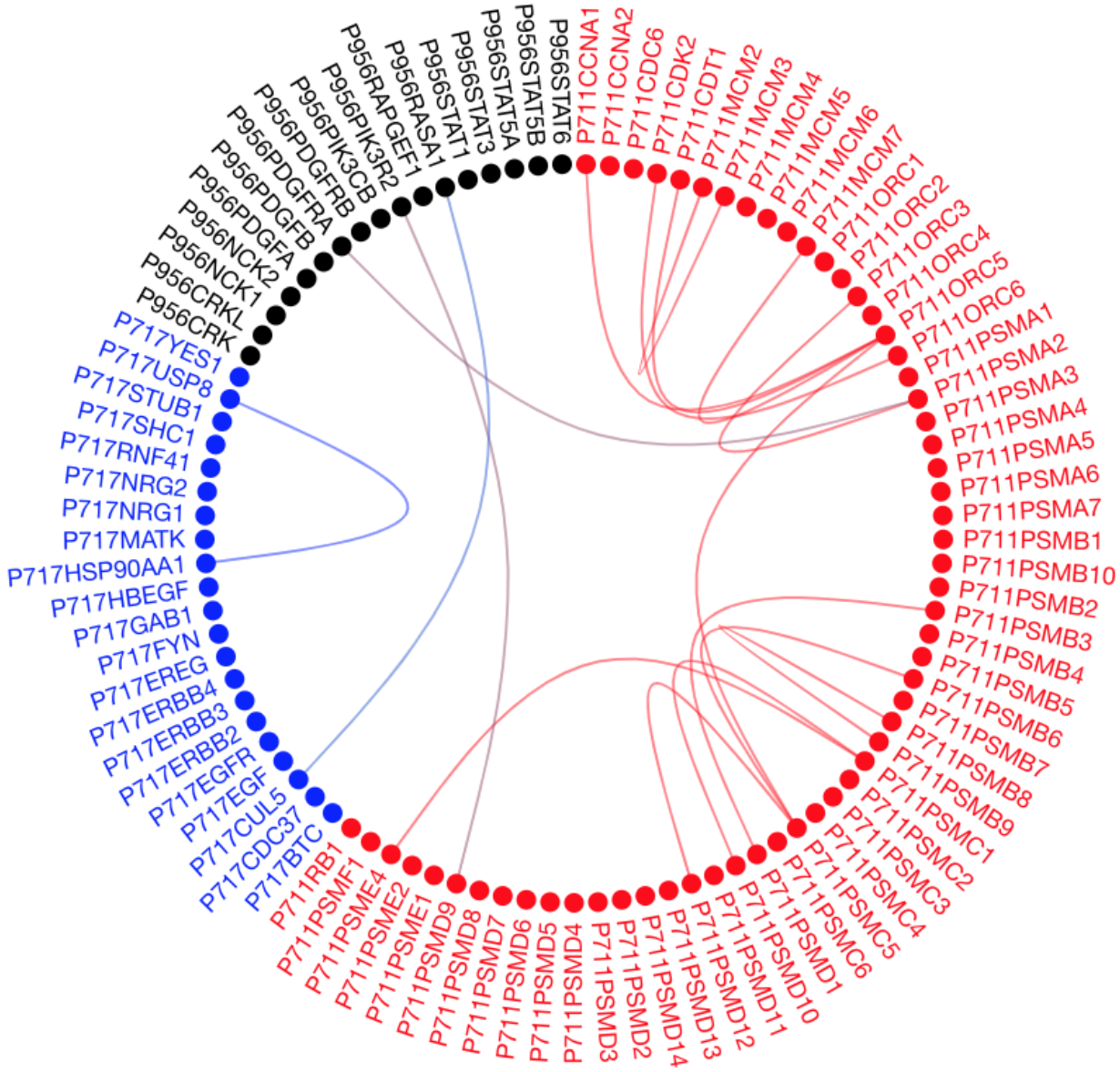


Figure 2.13: Estimated biological gene pathway for non-white women in breast cancer gene expression application data.

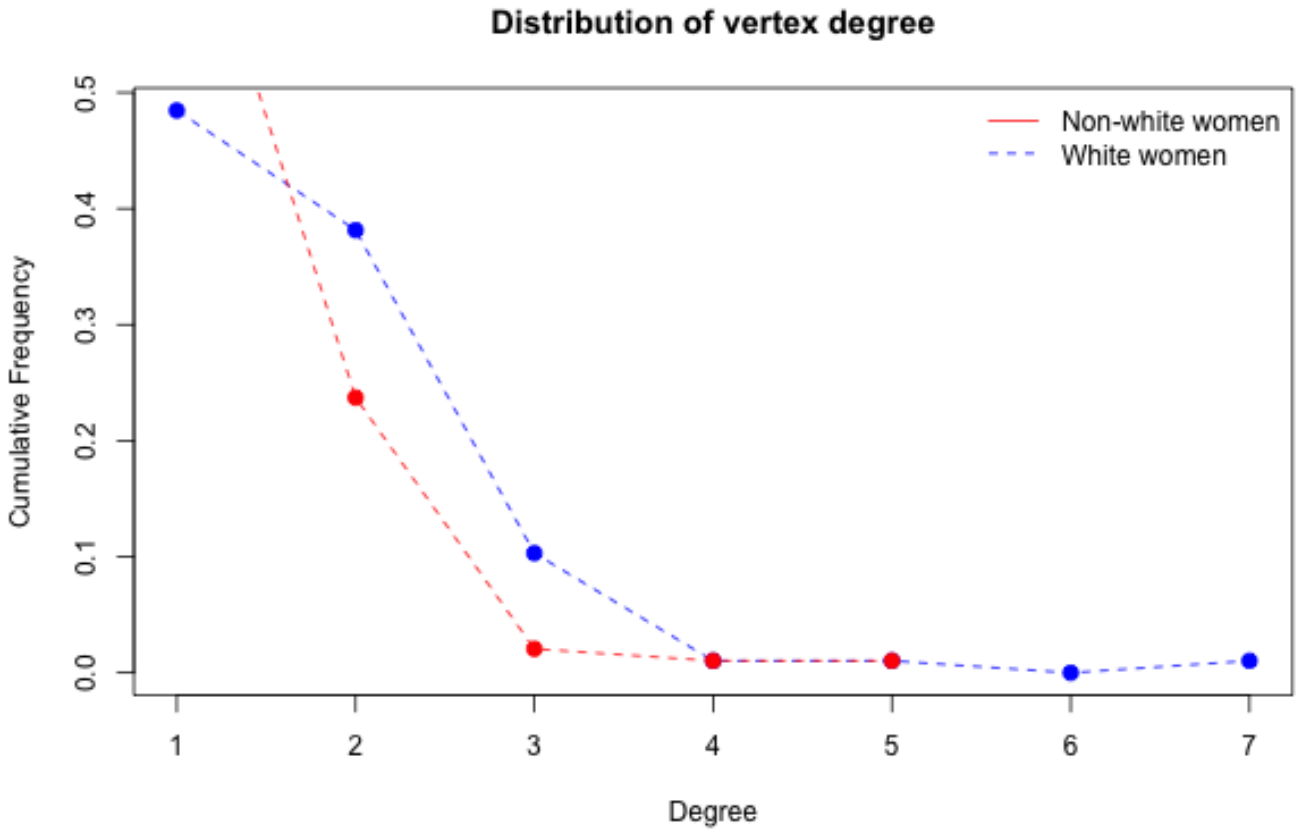


Figure 2.14: Estimated vertex degrees in breast cancer gene expression application data.

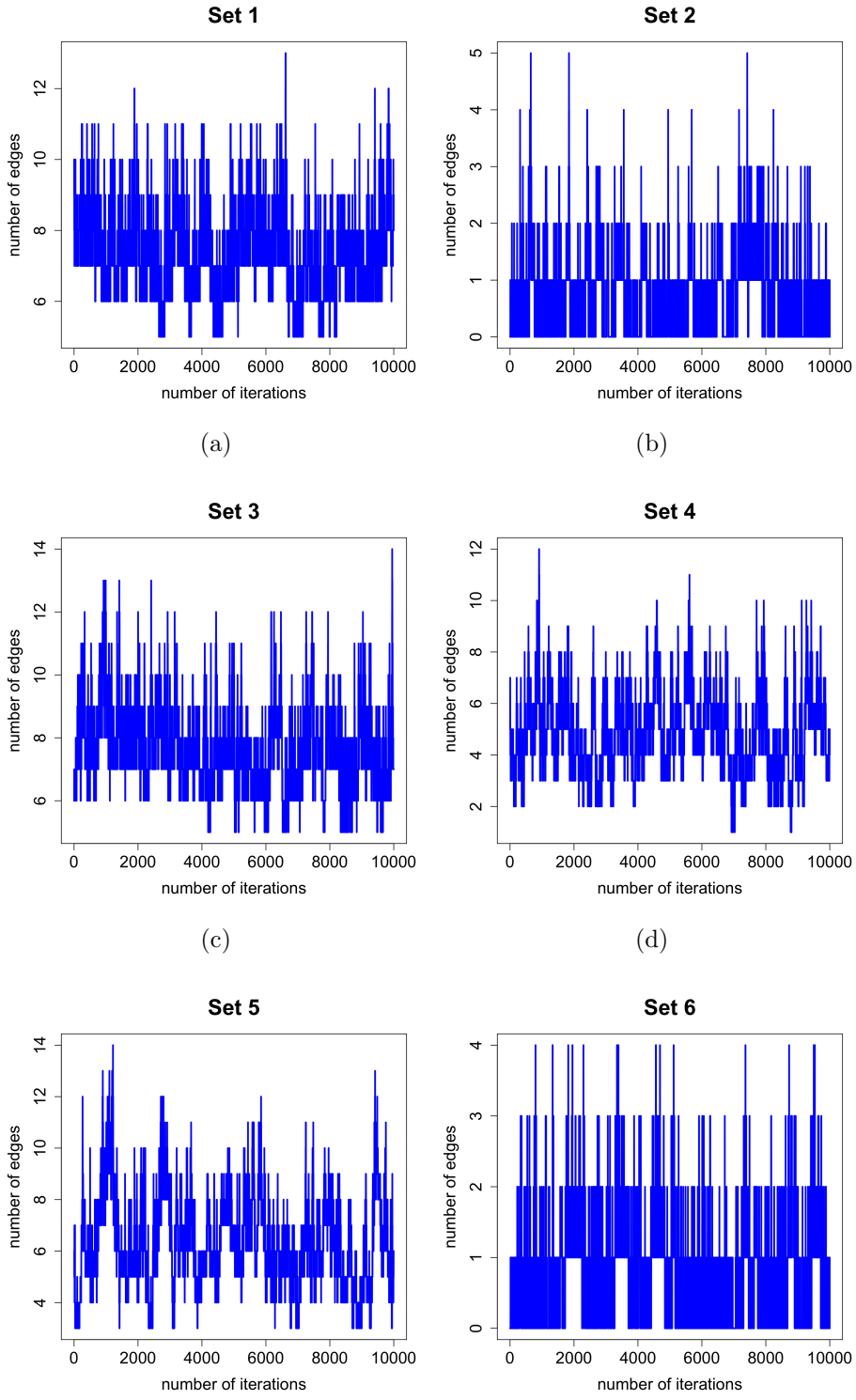


Figure 2.15: Trace plot of number of edges for 6 sets based on 10,000 MCMC iteration, where x-axis is the number of iterations and y-axis is the number of edges under chain network case.

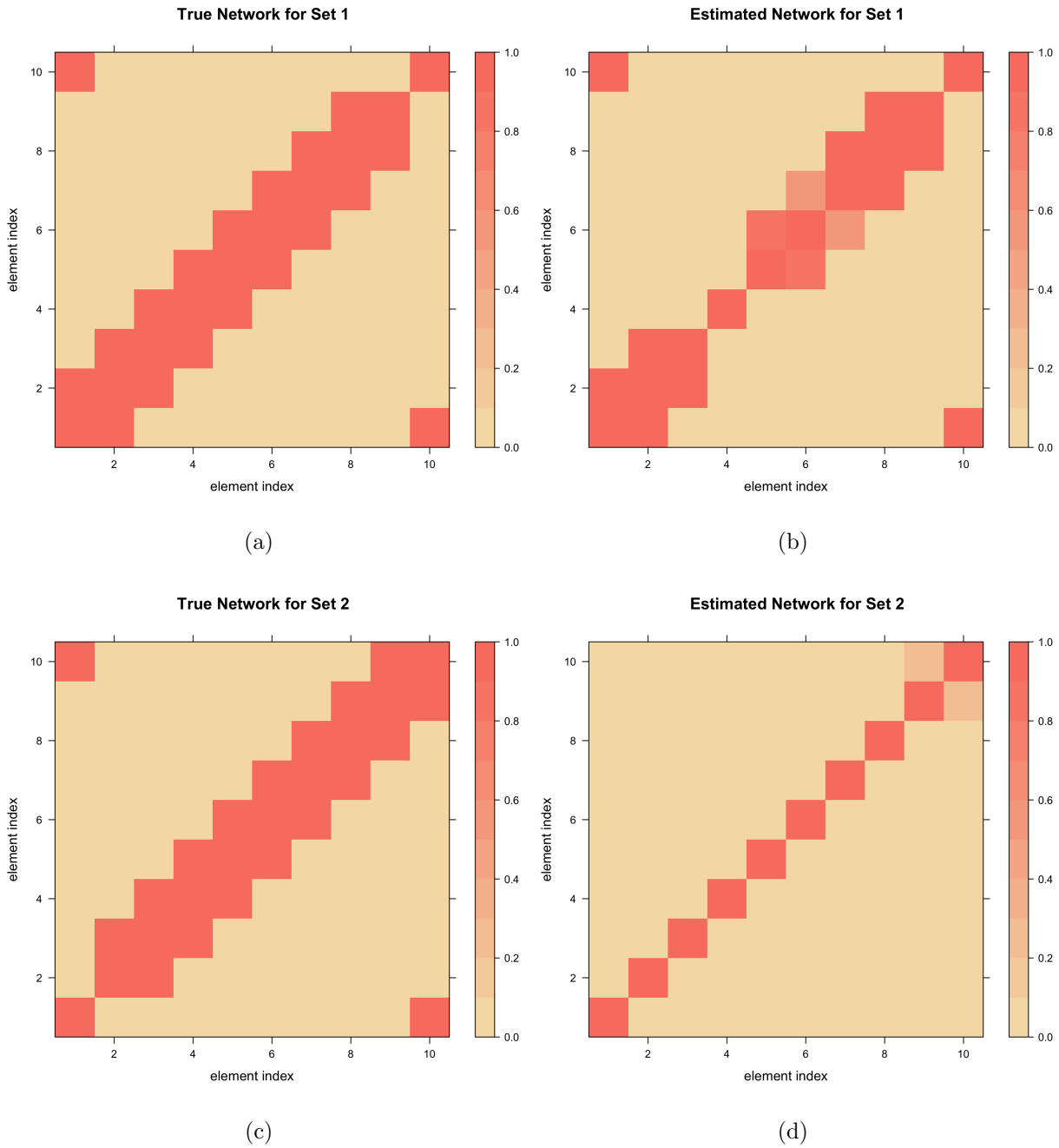


Figure 2.16: True and estimated heat maps and estimated ones; True heat maps for the set 1-2 simulated sets networks under chain network case; Figures (a) and (c) are ground truth and figures (b) and (d) are estimated.

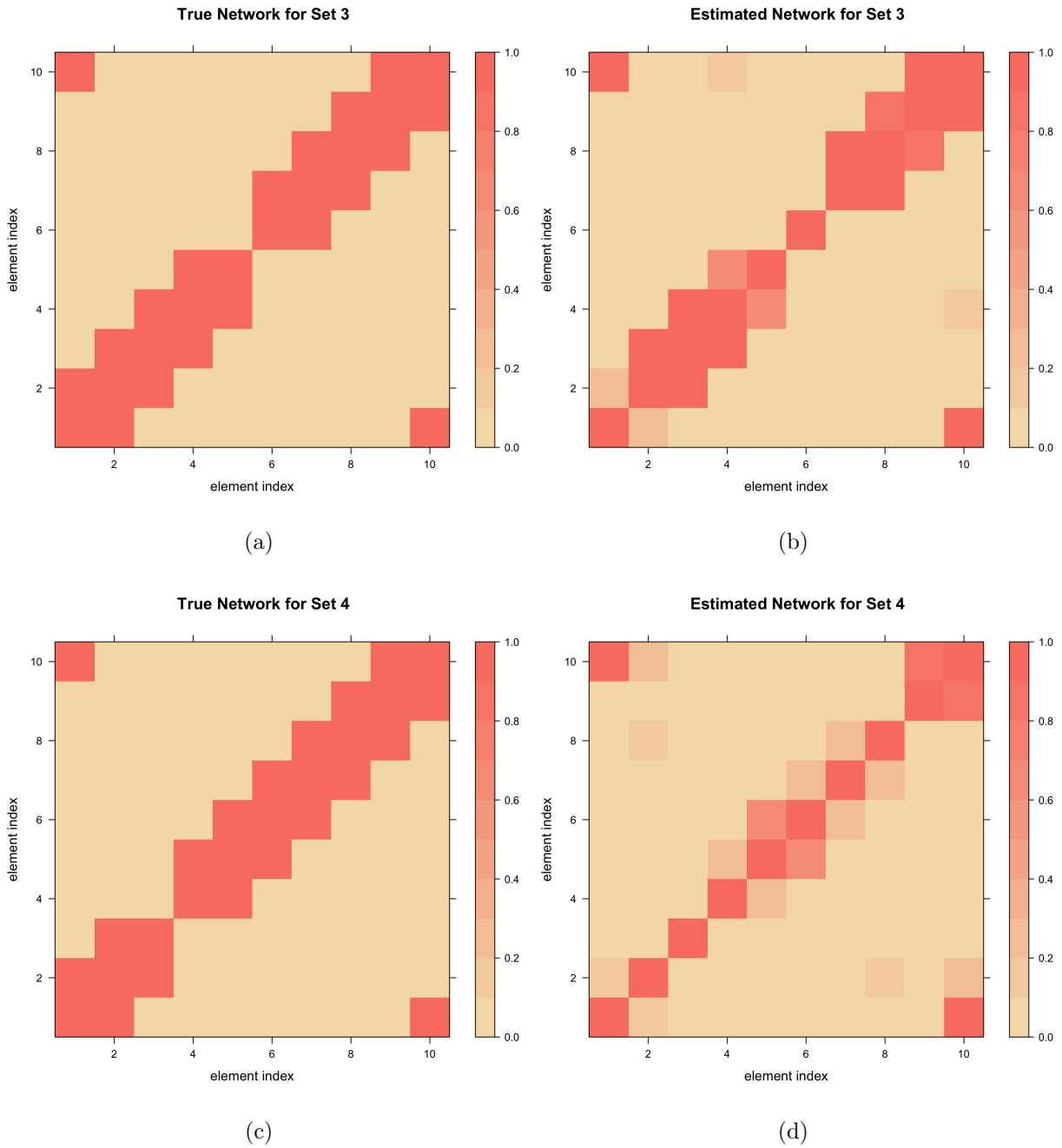


Figure 2.17: True and estimated heat maps; True heat maps for the set 3-4 simulated sets networks under chain network case; Figures (a) and (c) are ground truth and Figures (b) and (d) are estimated.

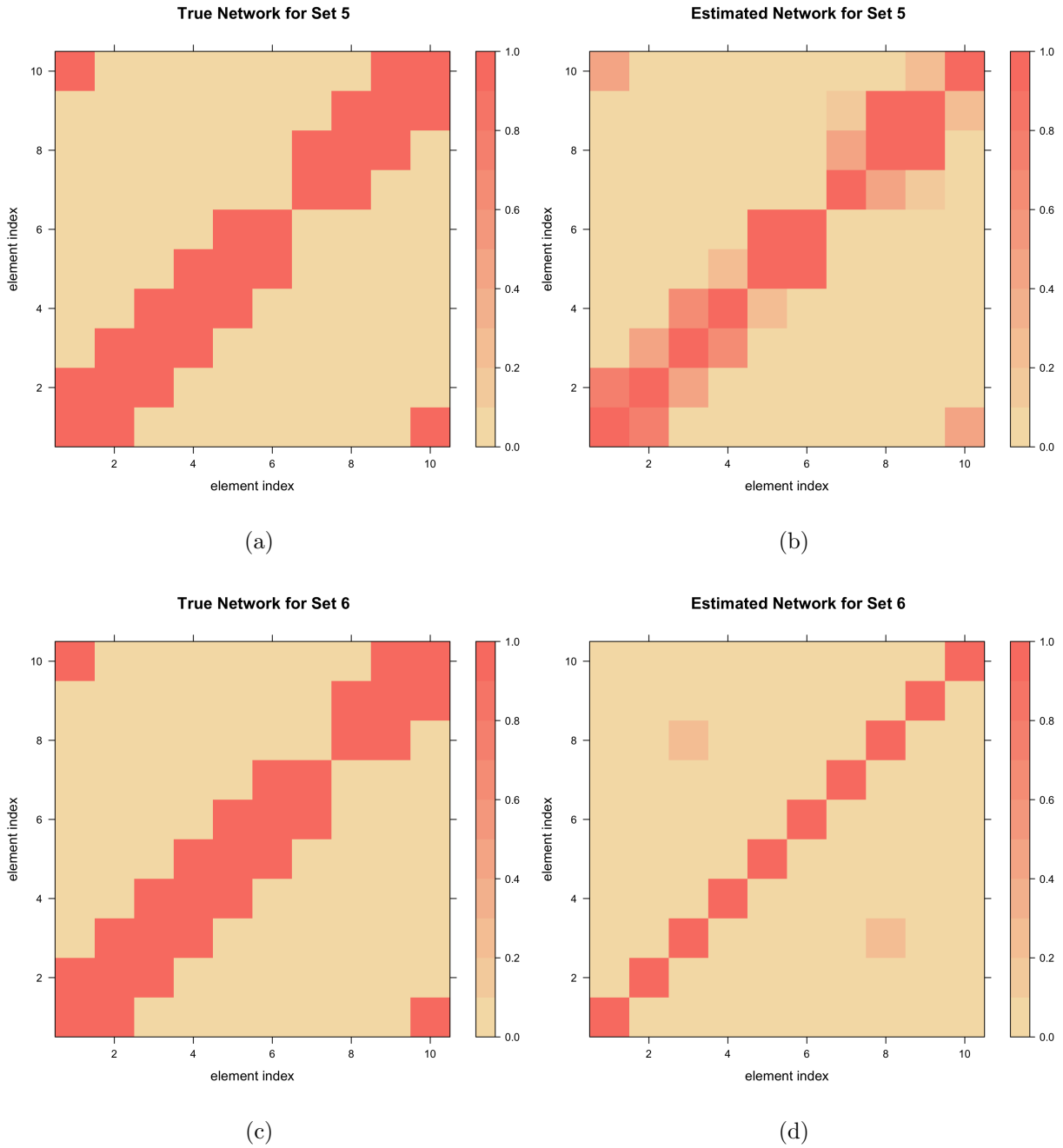


Figure 2.18: True and estimated heat maps for the set 5-6 simulated sets networks under chain network case; Figures (a) and (c) are ground truth and Figures (b) and (d) are estimated.

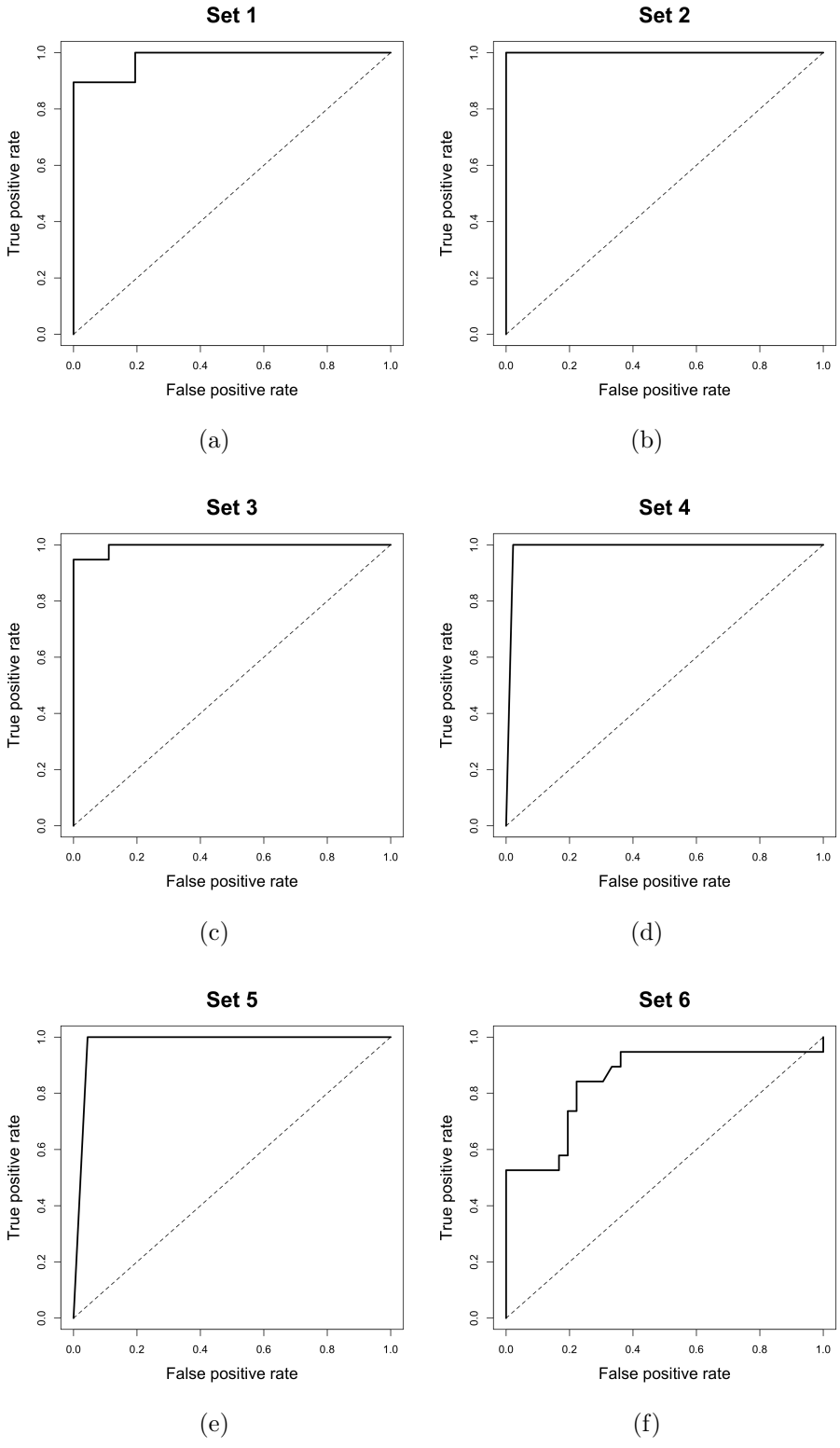


Figure 2.19: ROC curves for the posterior probability of edge inclusion with varying thresholds for 6 set networks under chain network case.

	TNR	Precision	Recall	ACC
FGL	1.00(0.10)	NaN	0.00(0.03)	0.89(0.06)
GGL	1.00(0.04)	1.00(0.04)	0.15(0.04)	0.90(0.09)
GLasso	0.62(0.03)	0.20(0.10)	1.00(0.01)	0.67(0.08)
BMGGM1	0.90(0.03)	0.50(0.03)	0.50(0.10)	0.85(0.09)
BMGGM2	0.89(0.01)	0.48(0.10)	0.52(0.07)	0.84(0.08)

Table 2.2: Comparison of five methods in terms of four measures with standard error (SE) over 100 simulated runs under AR(1) network case; GLasso=graphical lasso (Friedman et al, 2008); GGL=group graphical lasso (Danaher et al, 2014); FGL=fused graphical lasso (Danaher et al, 2014); BMGGM1=our proposed method with thresholding; BMGGM2=our proposed method with testing.

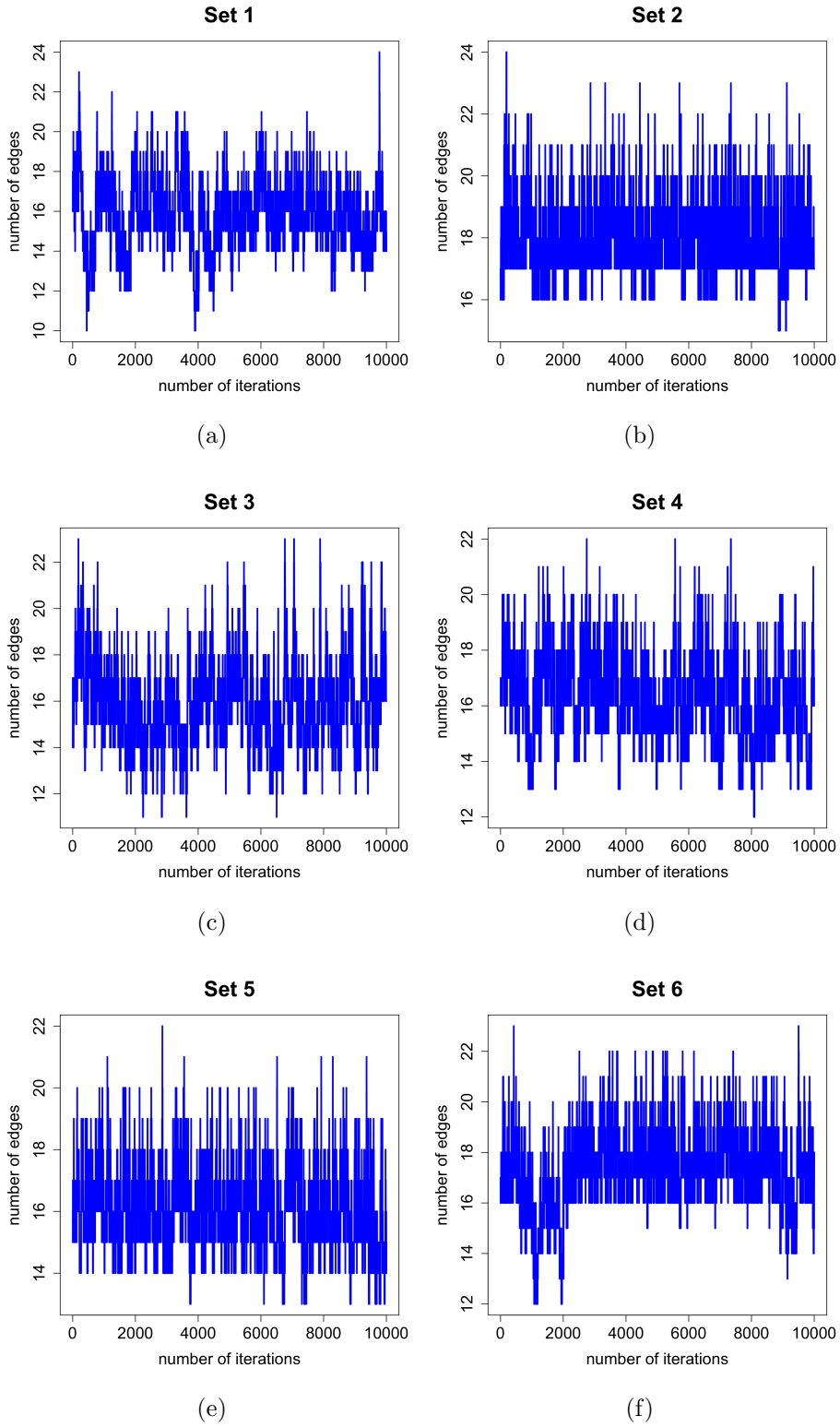
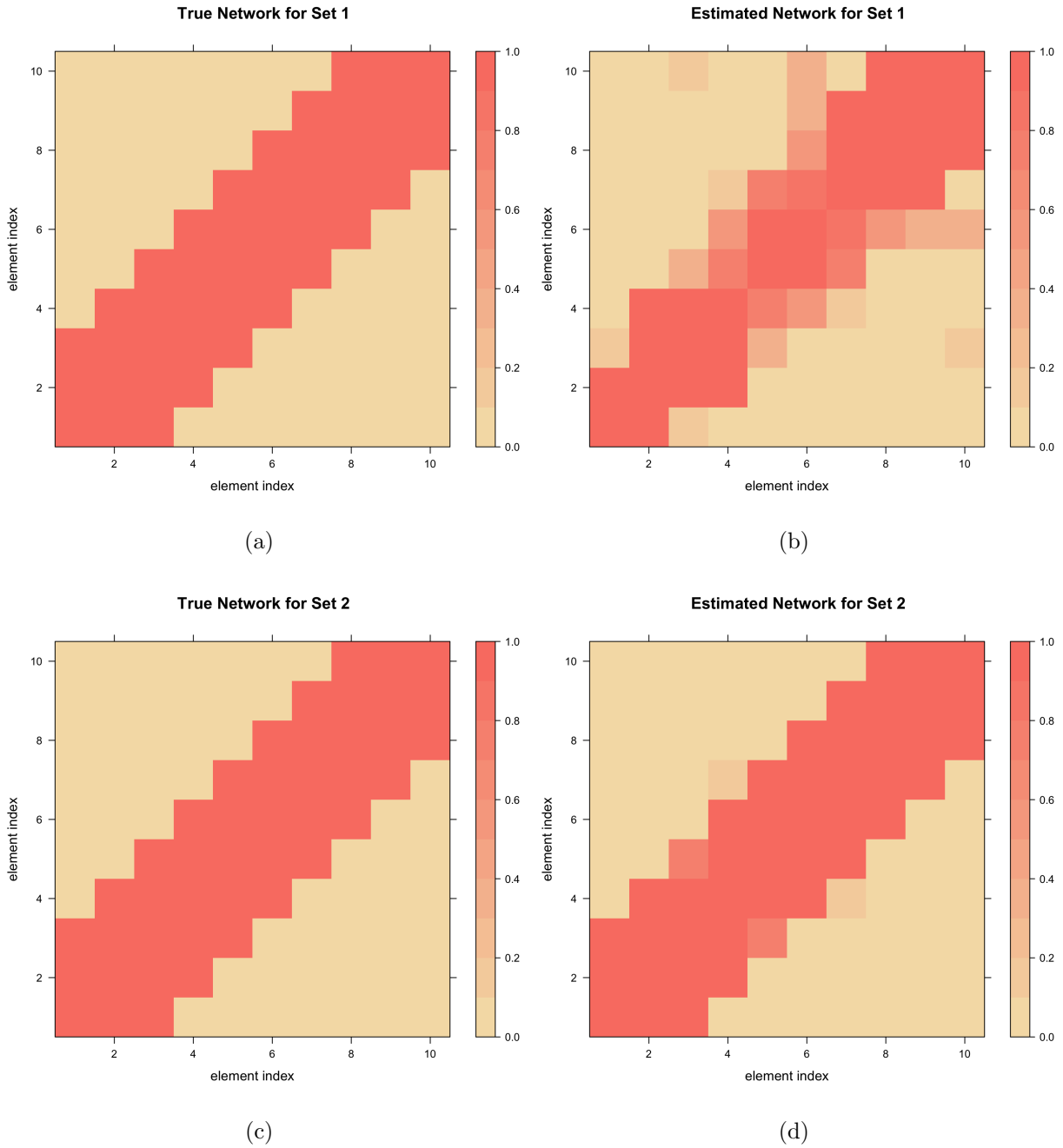


Figure 2.20: Trace plot of number of edges for 6 sets based on 10,000 MCMC iteration, where x-axis is the number of iterations and y-axis is the number of edges under AR(2) network case.



(a) (b)

(c) (d)

Figure 2.21: True and estimated heat maps for the set 1-2 simulated sets networks under AR(2) network case; Figures (a) and (c) are ground truth and Figures (b) and (d) are estimated.

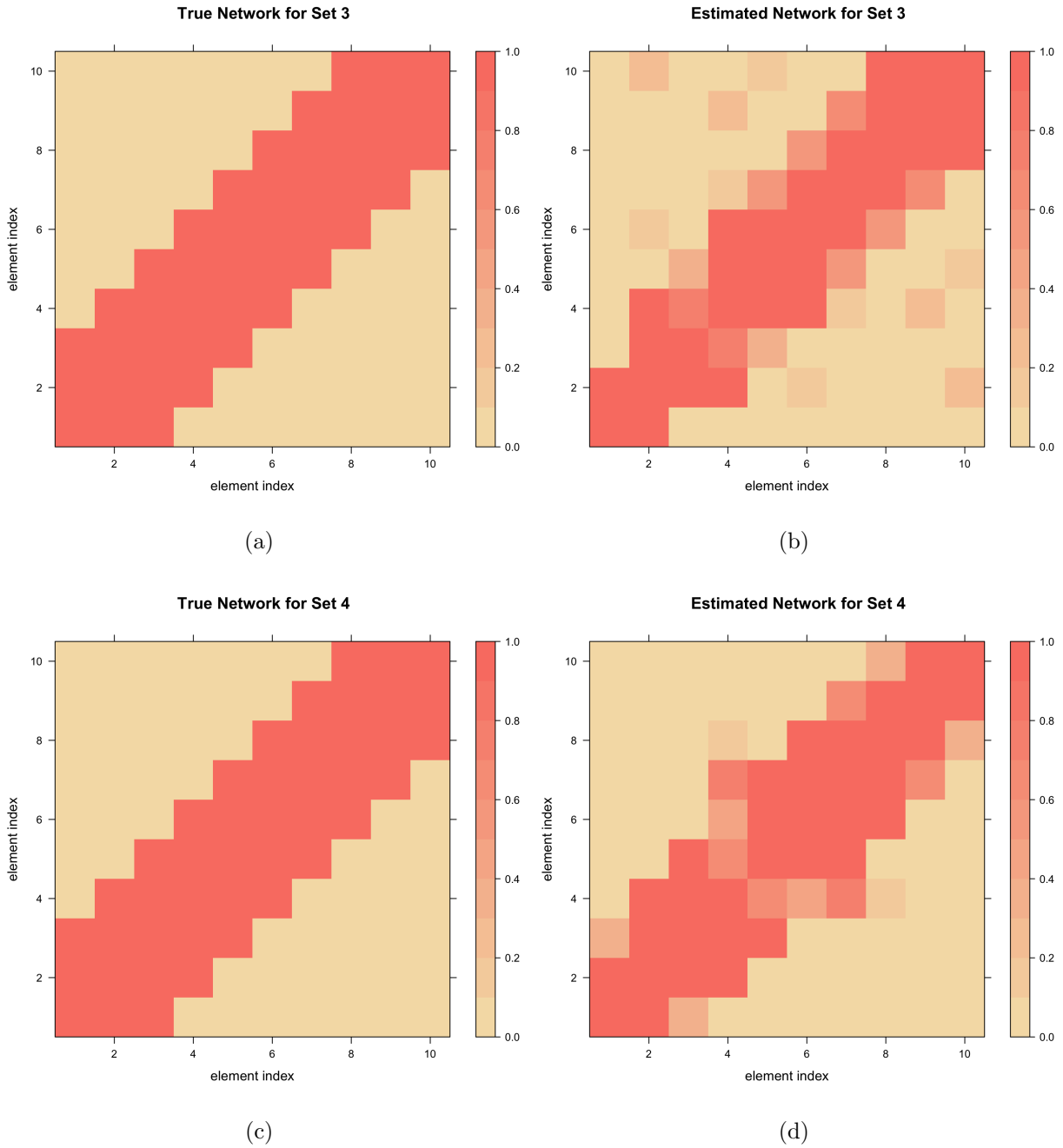
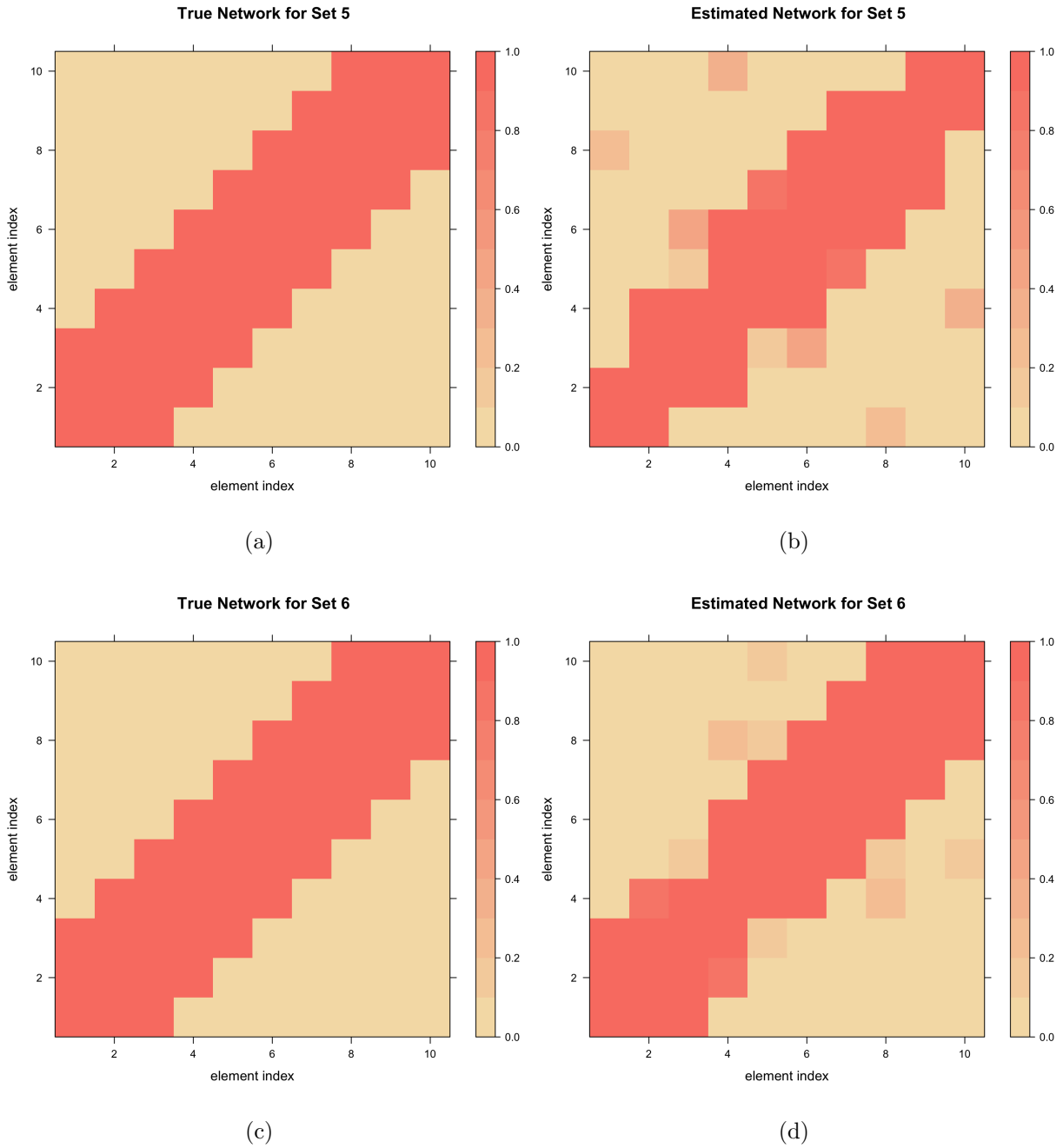


Figure 2.22: True and estimated heat maps for the set 3-4 simulated sets networks under AR(2) network case; Figures (a) and (c) are ground truth and Figures (b) and (d) are estimated.



(a) (b)

(c) (d)

Figure 2.23: True and estimated heat maps for the set 5-6 simulated sets networks under AR(2) network case; Figures (a) and (c) are ground truth and Figures (b) and (d) are estimated.

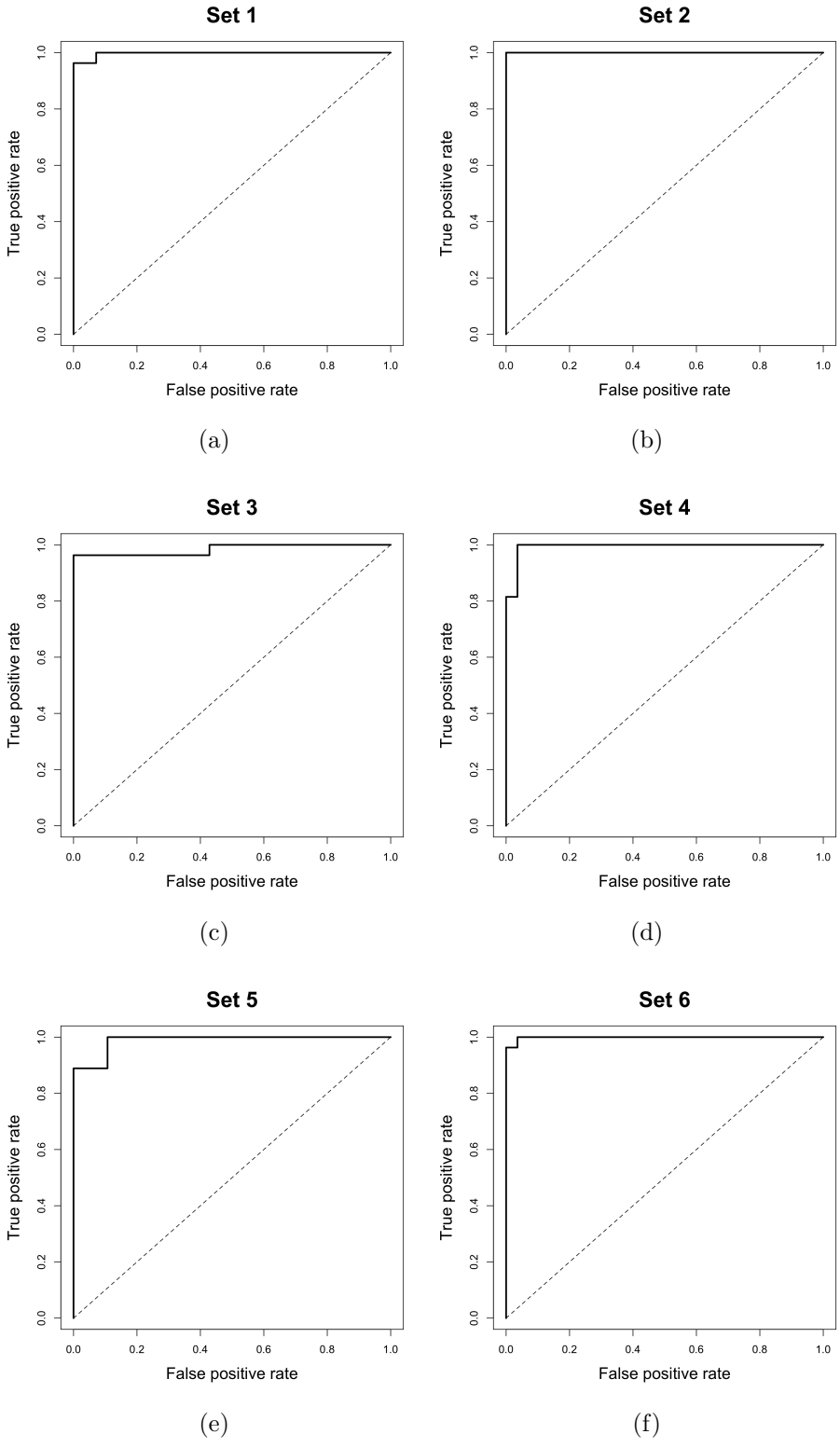


Figure 2.24: ROC curves for the posterior probability of edge inclusion with varying thresholds for 6 set networks under AR(2) network case.

	TNR	Precision	Recall	ACC
FGL	1.00(0.03)	1.00(0.03)	0.88(0.02)	0.95(0.01)
GGL	1.00(0.10)	1.00(0.09)	0.92(0.06)	0.97(0.08)
GLasso	0.54(0.11)	0.58(0.07)	1.00(0.10)	0.72(0.07)
BMGGM1	0.98(0.08)	0.97(0.10)	1.00(0.07)	0.98(0.10)
BMGGM2	0.88(0.03)	0.83(0.04)	0.99(0.09)	0.92(0.09)

Table 2.3: Comparison of five methods in terms of four measures with standard error (SE) over 100 simulated runs under AR(2) network case; GLasso=graphical lasso (Friedman et al, 2008); GGL=group graphical lasso (Danaher et al, 2014); FGL=fused graphical lasso (Danaher et al, 2014); BMGGM1=our proposed method with thresholding; BMGGM2=our proposed method with testing.

Chapter 3

Gaussian Process Selections in Semiparametric Multi-Kernel Machine Regression for Multi-Pathway Analysis

3.1 Introduction

In recent years, set-based analyses have attracted extensive interest. Take pathway-based analyses as an example: because of fast developing genomic technologies, it is possible to explore its profound impact on human biology and drug development. Pathways are sets of genes that serve a particular cellular or physiological function. Set-based analysis for identifying a set of related elements has the ability to detect a subtle change in expression level which could not be found using element-based analysis (Mootha et al., 2003). In drug

analysis, it would be more of interest to target a specific pathway, rather than identifying individual genes as therapeutic targets.

One possible way to estimate overall set effect is via the kernel machine method, as it is a powerful nonparametric statistical learning model to learn unknown function spaces, especially for high-dimensional data. A number of methods have been developed for testing the overall set effect. Liu et al. (2007) proposed a flexible framework by connecting the kernel machine with a linear fixed model that estimate fixed effects and set effects simultaneously. They developed a score test for identifying overall set effects. This method was further extended to predict disease in the context of survival analysis (Cai et al., 2011). On the other hand, Chen et al. (2011) developed a method to test set effects under the Bayesian framework. Kim et al. (2012) considered both Bayes-factor-based method and resampling-based method to identify significant sets.

Testing for the overall set effect has been well-studied. Set-based analyses are especially of interest to identify the specific promising signals (e.g. genetic biomarkers) that could have huge impacts on target. In reality, it is common that most of the sets are noises, while only a few signal sets can justify the target outcome (e.g. status of disease). However, the standard formulation for estimating set effect is based on a myopic strategy, which assumes only one set at a time. Hence, this single set-based analysis has some limitations. First, the score test proposed by Liu et al. (2007) requires large samples to obtain asymptotic distribution of test statistic, which is not realistic in high-dimensional data. A present-day data can involve limited samples but multiple or even thousands of sets that potentially explain the outcome of interest. Secondly, since the outcome is affected by multiple sets in common, it is inappropriate to model set by marginal analysis. Estimating set effects based on a single set ignores the fact sets interact with each other and thus results in many false positives or false negatives. Hence, to overcome these limitations on single set-based

analysis, we propose Gaussian multi-kernel machines approach for multi set-based analyses and also derive a computationally attractive algorithm to that is able not only to predict target but also to identify multiple sets that relate to the outcome of interest. We model the unknown high-dimension functions of multi-sets via multi-Gaussian kernel machines to consider the possibility that elements within the same set interact with each other. Hence, our set selection can be considered as Gaussian process selection.

In this chapter, we propose Gaussian process selection under the Bayesian variable selection framework. This technique models the unknown high-dimension functions of multi-sets via multi-Gaussian kernel machines. Our approaches allow prior knowledge for structural sets by imposing an Ising prior. The method is solved by variational Bayes algorithm.

Bayesian variable selection is usually implemented by Markov chain Monte Carlo (MCMC) method. By applying MCMC, one can focus on the subsets of signal sets rather than searching over 2^p combinations of all sets. However, the MCMC method can be hard to carry out in this work since the dimension of the parameter is high. This makes things worse when some MCMC methods require integrating over a large number of unknown parameters. Thus, it is extremely hard to calculate the posterior probability of set inclusion. One may avoid this problem by implementing Metropolis-Hastings but will still end up computationally inefficient MCMC method. Moreover, a degeneracy problem will also arise when we sample in a high-dimensional space. Specifically, the sequential MCMC sampling algorithm fails after a few steps because most of the sets will be excluded from the model. To solve this, an efficient variational algorithm for Bayesian variable selection was proposed by Carbonetto and Stephens (2012) where they update variational approximations for the hyperparameters with importance weights. Our algorithm follows the similar procedures, but has the following key differences: 1) we update the posterior probabilities of inclusion of set effects rather than those of fixed effects, which is much trickier; 2) We assess the full conditional of γ_m

rather than posterior marginal inclusion probabilities.

To illustrate our method, we present simulation studies with high-dimensional data where the sample size is smaller than the number of elements. We apply the proposed approach and algorithm with data that consists of noisy sets and signal sets. We start the first prior by Bernoulli distribution assuming sets are independent of each other. Then, we add prior knowledge for structural sets via Ising distribution and find the improved accuracy of signal sets detection. We also extend the simulation to a more realistic scenario where some sets have elements in common, that is, some sets are overlapped.

In this chapter, we make several novel contributions: (1) develop a semiparametric multi-kernel machine regression framework that catches the signal sets via on variance rather than mean function; (2) introduce a Bayesian variable selection method that is able to detect multiple significant set effects simultaneously instead of doing model selection by some myopic strategy; (3) implement a fast variational Bayesian algorithm to fit the proposed model and apply it to correlated high-dimensional data. Selecting a subset of Gaussian processes is very difficult and by far we are not aware of any prior work on the relevant problem.

The rest of this chapter is organized as follows. In Chapter 3.2, we present Gaussian process selection problem. Chapter 3.3 develops variational Bayes algorithm for its solution. In Chapter 3.4, we illustrate the performance of our Gaussian process selection approach under the semiparametric multi-kernel machine regression in a simulation study. Lastly, our concluding remarks are presented in Chapter 3.6.

3.2 Gaussian Process Selection

We propose our Gaussian process selection under semiparametric multi-kernel machine learning Model. We model the unknown high-dimension functions of multi-sets via multi-Gaussian kernel machines. Our Gaussian process selection approach is developed using Bayesian hierarchical model framework.

3.2.1 Semiparametric Multi-Kernel Machine Learning Model

Suppose we have n observations. Let $\mathbf{y}$ denote a $n \times 1$ vector representing outcome of interest, $\mathbf{X}$ denote a $n \times q$ matrix of fixed covariates. Suppose that we have $M(\geq 2)$ set measurements, $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$ and p_m denotes the dimension of mth set, where $\mathbf{Z}_m$ is a $n \times p_m$ matrix of elements. The continuous outcome $\mathbf{y}$ is modeled by a linear combination of both $\mathbf{X}$ and $\mathbf{Z}$s with the following expression,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + h_1(\mathbf{Z}_1) + \ldots + h_M(\mathbf{Z}_M) + \boldsymbol{\epsilon}, \tag{3.1}$$

where $\boldsymbol{\beta}$ is a $q \times 1$ vector of coefficients, $h(\mathbf{Z})$s are unknown smooth nonparametric functions, which belong to Hilbert functional space $\mathcal{H}$ by an arbitrary kernel function $\mathbf{K}(.,.)$, and $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I})$.

The variable selection problem can be formulated as identifying which set effects $h_1(\cdot), \ldots, h_M(\cdot)$ are zeros. The "variable" in the context of "variable selection" is actually a function that follows Gaussian process. That is, $h_m(\mathbf{Z}_m) \sim GP\{0, \tau_m\mathbf{K}_m(\cdot,\cdot)\}$. If τ_m is zero, set effect $h_m(\cdot)$ become zero. Thus it is referred as "Gaussian process selection". Unlike the typical method that selects signals via mean function, we would like to select signals via variance component (matrix).

To accomplish Gaussian process selection, we use indicator variable γ_m to indicate whether or not the effect of mth set is included in the model. The variance caused by each set is scaled by the overall noise level τ_m. The τ_m depends on the indicator γ_m and has mixture prior with inverse-gamma density. If $\gamma_m = 0$, then $\tau_m = 0$ so that $h_m(.) = 0$.

Mathematically, we can write this prior as:

$$h_m(.) \sim \begin{cases} \text{GP}\{0, \tau_m \mathbf{K}_m(.,.)\}, & \text{if } \gamma_m = 1 \\ 0, & \text{if } \gamma_m = 0, \end{cases}$$

$$\tau_m \sim \begin{cases} \text{IG}(a_\tau, b_\tau), & \text{if } \gamma_m = 1 \\ 0, & \text{if } \gamma_m = 0. \end{cases}$$

Each kernel function $\mathbf{K}_m(.,.)$ lies on some functional Hilbert space $\mathcal{H}$, which will be fully discussed in Chapter 3.2.2.

This semiparametric model can be easily represented as a Bayesian Gaussian selection regression. We formulate our Bayesian hierarchical model by treating each fixed unknown parameters as random ones:

$$\begin{aligned} y|\mathbf{X}, h(\mathbf{Z}_1), \ldots, h(\mathbf{Z}_M) &\sim N\{\mathbf{X}\boldsymbol{\beta} + \sum_m h_m(\mathbf{Z}_M), \sigma^2\mathbf{I}\}, \\ \boldsymbol{\beta} &\sim N(0, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}), \\ \sigma^2 &\sim \text{IG}(a_{\sigma^2}, b_{\sigma^2}), \\ h_m(.) &\sim (1-\gamma_m)\cdot\delta_0 + \gamma_m \cdot \text{GP}\{0, \tau_m \mathbf{K}_m(.,.)\}, \\ \tau_m &\sim (1-\gamma_m)\cdot\delta_0 + \gamma_m \cdot \text{IG}(a_\tau, b_\tau), \end{aligned} \tag{3.2}$$

where δ_0 denotes a point mass function at zero.

3.2.2 Nonparametric Functions in Hilbert Space $\mathcal{H}$

To model the smooth functions $h_1(\boldsymbol{z}_{i1}), \ldots, h_M(\boldsymbol{z}_{iM})$ $(i = 1, \ldots, n)$ with unknown structure and provide procedure to test the overall effect of each $h_m(\boldsymbol{z}_{im})$ $(m = 1, \ldots, M)$ (i.e. set effects) respectively, as well as to test each element effect in the presence of set effects and possible element interactions and set-set interactions with missing information, we proposed to model $h_m(\boldsymbol{z}_{im})$'s using Gaussian process to avoid potential power loss compared with modeling $h_m(\boldsymbol{z}_{im})$'s parametrically. We present the details in the this chapter.

Let $\mathcal{H}_{\mathcal{M}}$ denote a Hilbert kernel function space generated by a positive definite kernel function $\mathbf{K}_m(.,.)$ and $h_m(z) \in \mathcal{H}_{\mathcal{M}}$ for all $m = 1, \ldots, M$. By Mercer's theorem, for any $h(z) \in \mathcal{H}_{\mathcal{M}}$, $h(z)$ can be represented as $h(z) = \sum_{b=1}^{B} \omega_b \phi_b(z) = \boldsymbol{\phi}(x)^\top \boldsymbol{\omega}$ (primal representation), and $h(z)$ can also be written as $h(z) = \sum_{l=1}^{L} \zeta_l k(z_l^*, z)$ (dual representation), where $\boldsymbol{\phi}(z) = l\{\phi_1(z), \ldots, \phi_B(z)r\}^\top$ is a set of orthogonal basis functions, and $\boldsymbol{\omega}$ and $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_L)^\top$ are vectors of coefficients, respectively.

Let $\mathbf{K}_m$ be the m^{th} kernel matrix associated with $\mathcal{H}_{\mathcal{M}m}$, $\boldsymbol{hz_m} = \{h_m(\boldsymbol{z_{im}})\}_{i=1,\ldots,n} \in \mathcal{H}_{\mathcal{M}m}$ and $\mathcal{H}_{\mathcal{M}1} \cup \ldots \cup \mathcal{H}_{\mathcal{M}M} = \mathcal{H}_{\mathcal{M}}$. $\boldsymbol{hz_m}$ is modeled using $\mathrm{GP}_m = \mathrm{GP}(\mathbf{0},\ \mathbf{K}_m)$ for all m's. Let M be the kernel matrix associated with $\mathbf{K}_m$ $(\in \mathcal{H}_{\mathcal{M}})$ and $\mathbf{P}$ be the off-diagonal matrices

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_1 & \ldots & \mathbf{P} \\ & \ddots & \\ \mathbf{P} & \ldots & \mathbf{K}_M \end{pmatrix},$$

where each $\mathbf{K}_m$ accounts for the correlations among elements within the m^{th} set for $m = 1, \ldots, M$; If $\mathbf{P}$ are zero-matrices, then all M sets are do not have hidden associations other than shared elements. $\boldsymbol{hz} = (\boldsymbol{hz_1}^\top, \ldots, \boldsymbol{hz_M}^\top)^\top$ can be modeled using $\mathrm{GP}(\mathbf{0}, \mathbf{K})$, i.e. $\boldsymbol{hz} \sim$

GP($\mathbf{0}$, $\mathbf{P}$).

Therefore, we model the set effect through a flexible function $h_m(.)$ to consider the possibility that elements within the same set interact with each other. Specifically, the set effect caused by $h_m(.)$ is modeled via an inverse covariance matrix of random elements by using the kernel method.

The kernel matrix $\mathbf{K}_m(.)$ plays a key role here which considers linear or nonlinear expression effect within a particular set. To achieve this, we map the element expression information to a higher-dimensional space via the kernel. Intuitively, the kernel matrix $\mathbf{K}_m(.)$ represents "dissimilarity" between each sample and can be used to encode complex biological information. $\mathbf{K}_m(.)$ must be symmetric and positive definite. One of most popular kernel functions (Liu et al., 2007), (Kim et al., 2012), (Cai et al., 2011) is Gaussian kernel. In our study, we also model h_m via a Gaussian kernel.

If we consider correlation $\mathbf{\Sigma}$ among $\mathbf{z}$, Gaussian kernel can be defined by

$$\mathbf{K}(\mathbf{z}, \mathbf{z}') = \exp\{-\frac{1}{2}(\mathbf{z} - \mathbf{z}')^\top \mathbf{\Sigma}^{-1}(\mathbf{z} - \mathbf{z}')\}.$$

It is common to use diagonal $\mathbf{\Sigma}^{-1}$, which leads to primal representation of $h(\mathbf{Z})$ by using finite basis functions. In this case, the kernel becomes

$$\mathbf{K}(\mathbf{z}, \mathbf{z}') = \exp(\frac{-\|\mathbf{z} - \mathbf{z}'\|^2}{\rho}),$$

where $\|\mathbf{z} - \mathbf{z}'\|^2 = \sum_{j=1}^{p}(z_j - z_{j'})^2$ and ρ is defined as characteristic length scale.

However, in our study, $\mathbf{\Sigma}^{-1}$ is considered as the sparse matrix instead of the diagonal matrix. This is because the diagonal matrix assumes that each element with set works as an individual while sparse matrix considers the interaction between elements. For specific, we learn the

$\mathbf{\Sigma}^{-1}$ by some graphical model methods which are able to encode the conditional dependence relationship. The intuition is that a set effect can be passed through a covariance matrix by Gaussian process.

The kernel matrix $\mathbf{K}_m(.)$ can also serve as a trick to reduce the dimension. Calculating inverse covariance matrix $\mathbf{\Sigma}_M^{-1}$ can be computationally demanding when p_m is large. But applying the kernel methods can dramatically reduce the computational complexity to calculating the kernel $\mathbf{K}_m(.)$ which is $n \times n$ matrix.

3.2.3 Ising Prior Linking sets

We employ Ising priors to incorporate the structure of sets that encourages the inclusions of related sets into our analysis. Typically, the latent variable for each set is assumed to have independent Bernoulli prior. Incorporating prior knowledge for set structures would help stochastic search of the set spaces (Li and Zhang, 2010).

Let the inclusion of sets $1, \ldots, M$ denoted by the binary vector $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_M\}$. The distribution for Ising prior, $\boldsymbol{\gamma}$, is given by

$$p(\boldsymbol{\gamma}) = \exp\{\mathbf{a}'\boldsymbol{\gamma} + \boldsymbol{\gamma}'\mathbf{B}\boldsymbol{\gamma} - C(\mathbf{a}, \mathbf{B})\},$$

where $\mathbf{a} = \{a, \ldots, a\}$ is a $M \times 1$ vector, $\mathbf{B}$ is a $M \times M$ symmetric matrix, and $C(\mathbf{a}, \mathbf{B})$ is the normalizing constant

$$C(\mathbf{a}, \mathbf{B}) = \log\{\sum_{\boldsymbol{\gamma} \in \{0,1\}^M} \exp(\mathbf{a}'\boldsymbol{\gamma} + \boldsymbol{\gamma}'\mathbf{B}\boldsymbol{\gamma})\}.$$

Using this prior, the conditional probability of γ_m has the simple form,

$$p(\gamma_m|\gamma_{-m}) = \frac{\exp\{\gamma_i(a + b\sum_{j\in\text{-m}}\gamma_j)\}}{1+\exp\{(a + b\sum_{j\in\text{-m}}\gamma_j)\}}.$$

Ising prior has two hyperparameters. The hyperparameter, $\mathbf{a}$, controls the sparsity of γ. The similarity matrix $\mathbf{B}$ represents the pairwise relatedness of the sets, where the diagonal entries are set to 0 and off-diagonal are non-zero. The entry of $\mathbf{B}$, $b_{m,m'}$, indicates the prior belief on the associated between the pairs of neighbor sets (m, m'). The higher $b_{m,m'}$, the higher the possibility these two sets will end up the same inclusion status in the model. If $\mathbf{B} = \mathbf{0}$, then it is equivalent to independent Bernoulli priors. In other words, If $\mathbf{B} = \mathbf{0}$, then the sets are independent of each other. In Chapter 3.4, we will provide the performance of our Gaussian process selection approach by using independent Bernoulli priors and Ising priors.

3.2.4 Gaussian Process Selection under Generalized Linear Model

Our Gaussian process selection regression model is also flexible enough to be applied to the natural exponential family. In practice, it is more often to have binary data (e.g. clinical case-control data) or counting data (e.g. high-throughput sequencing data). We can generalize our approach to the generalized linear model with link function $g(\cdot)$,

$$g\{\mathbb{E}(\mathbf{y})\} = \mathbf{X}\boldsymbol{\beta} + h_1(\mathbf{Z}_1) + \ldots + h_M(\mathbf{Z}_M).$$

We briefly showcase a semiparametric logistic regression and Poisson regression with these two types of data. We fit the following semiparametric logistic regression by logit link

function:

$$\log \frac{\mathbb{E}(\mathbf{y})}{1-\mathbb{E}(\mathbf{y})} = \mathbf{X}\boldsymbol{\beta} + h_1(\mathbf{Z}_1) + \ldots + h_M(\mathbf{Z}_M), \tag{3.3}$$

where the left hand side denotes the canonical link function for the Bernoulli distribution of $\mathbf{y}$ and the right side is the same as that of equation (3.1). Alternatively, one can build a binary classification model based on probit regression (Kim et al., 2012).

Poisson regressions are also of interest for count data. Similarly, we only need to replace the logit link function in equation (3.3) with log link function:

$$\log \mathbb{E}(\mathbf{y}) = \mathbf{X}\boldsymbol{\beta} + h_1(\mathbf{Z}_1) + \ldots + h_M(\mathbf{Z}_M).$$

3.3 Methodology

In this chapter, we derive a variational inference algorithm to implement our Gaussian process selection under semiparametric multi-kernel machine learning model. First, we briefly review deterministic approximate inference algorithms based on variational inference and provide the specification to our models.

3.3.1 Overview of Variational Inference

Let $\boldsymbol{\theta}$ denote the set of all parameters. Variational inference works on the ideas of using a tractable $q(\boldsymbol{\theta})$ to approximate true posterior $p(\boldsymbol{\theta})$, where $q(\boldsymbol{\theta})$ taks a fully factorized approximation of the form $q(\boldsymbol{\theta}) = \prod_j q(\theta_j)$, where j is the index of jth parameter. One can decompose the marginal distribution $p(\mathbf{y})$ by

$$\log p(\mathbf{y}) = \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{D})} d\boldsymbol{\theta} + \int q(\boldsymbol{\theta}) \log \frac{p(\mathbf{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta},$$

where the first term is the Kullback-Leibler (KL) divergence, KL$[q||p](\geq 0)$ and the second term is called the *free energy*, denoted as $F(q, \mathbf{y})$ which provides lower bound for the log marginal distribution $\log p(y)$. One can optimize the lower bound for each $q(\theta_j)$ sequentially, while fixing other parameters $\boldsymbol{\theta}_{-j}$. To maximize the lower bound, one only needs to calculate

$$\mathop{\mathbb{E}}_{j \neq j'} \{\log p(\mathbf{y}, \boldsymbol{\theta})\} = \int \log p(\mathbf{y}, \boldsymbol{\theta}) \prod_{j \neq j'} q_j d\theta_j. \tag{3.4}$$

3.3.2 Variational Inference for Solving Gaussian Process Selection

We then apply the variational inference to our semiparametric multi-kernel machine learning model (3.2) using Bayesian hierarchical framework. An efficient variational algorithm for Bayesian variable selection was proposed by Carbonetto and Stephens (2012) where they update variational approximations for the hyperparameters with importance weights. Our algorithm follows the similar procedures, but has the following key differences: 1) we update the posterior probabilities of inclusion of set effects rather than those of fixed effects, which is much tricker; 2) We assess the full conditional of γ_m rather than posterior marginal inclusion probabilities.

Since the priors are chosen to be conjugate to the likelihoods, we can compute the exact posteriors in closed forms. This leads to a straightforward variational Bayesian algorithm.

The set of all parameters $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \sigma^2, \mathbf{h}, \boldsymbol{\tau}, \boldsymbol{\gamma}\}$ is restricted to take the form

$$q(\boldsymbol{\theta}) = q(\boldsymbol{\beta}, \sigma^2, \mathbf{h}, \boldsymbol{\tau}, \boldsymbol{\gamma}) = q_\beta(\boldsymbol{\beta}) q_{\sigma^2}(\sigma^2) \times \prod_{m=1}^{M} q_h(\mathbf{h}_m) q_\tau(\tau_m) q(\gamma_m). \tag{3.5}$$

The updates on the right-hand side of (3.5) have two components. The first component approximates the posterior distribution of ($\boldsymbol{\beta}$, σ^2) from fix covariates. The second component

estimates the posterior distribution of (h_m, τ_m, γ_m) for each set coefficient. The approximate posteriors can be factorized by applying mean-field assumption. The update of approximate posterior can be obtained by minimizing the Kullback-Leibler divergence (Bishop, 2006) via equation (3.4). This yields the following coordinate descent updates (see Appendix for details):

- Update $q_{\boldsymbol{\beta}}(\boldsymbol{\beta})$:

$$\log q_{\boldsymbol{\beta}}(\boldsymbol{\beta}) = -\frac{1}{2}\boldsymbol{\beta}^\top\{\frac{\mathbf{X}^\top\mathbf{X}}{\mathbb{E}(\sigma^2)} + \Sigma_{\boldsymbol{\beta}}^{-1}\}\boldsymbol{\beta} - \frac{2\boldsymbol{\beta}^\top\mathbf{X}^\top\{\mathbf{y} - \mathbb{E}(\sum_m h_m)\}}{\mathbb{E}(\sigma^2)} + \text{const.} \tag{3.6}$$

  Hence $q_{\boldsymbol{\beta}}(\boldsymbol{\beta}) = N(\boldsymbol{\mu}_{\boldsymbol{\beta}}^N, \mathbf{V}_{\boldsymbol{\beta}}^N) = N(\mathbf{K}_{\boldsymbol{\beta}}^{-1}M_{\boldsymbol{\beta}}, \mathbf{K}_{\boldsymbol{\beta}}^{-1})$, where

$$\mathbf{K}_{\boldsymbol{\beta}} = \frac{\mathbf{X}^\top\mathbf{X}}{\mathbb{E}(\sigma^2)} + \Sigma_{\boldsymbol{\beta}}^{-1}, \quad M_{\boldsymbol{\beta}} = \frac{\mathbf{X}^\top\{\mathbf{y} - \mathbb{E}(\sum_m h_m)\}}{\mathbb{E}(\sigma^2)}.$$

- Update $q_{\sigma^2}(\sigma^2)$:

$$\begin{aligned}\log q_{\sigma^2}(\sigma^2) &= \mathbb{E}_{\boldsymbol{\beta},h_1,\ldots,h_m}\{\log p(\mathbf{y}|\mathbf{X}\beta + \sum_m h_m, \sigma^2) + \log p(\sigma^2)\} + \text{const} \\ &= -(a + \frac{n}{2} - 1)\log\sigma^2 - (b + \frac{A}{2})\frac{1}{\sigma^2} + \text{const.}\end{aligned} \tag{3.7}$$

  Hence, $q_{\sigma^2}(\sigma^2) = \text{IG}(a_{\sigma^2}^N, b_{\sigma^2}^N)$, where

$$a_{\sigma^2}^N = a + \frac{n}{2} \quad b_{\sigma^2}^N = b + \frac{A}{2}.$$

Here A can be expressed as

$$\begin{aligned}
A &= \mathbb{E}_{h_1,\ldots,h_m}[\mathrm{Tr}\{\mathbf{X}^\top \mathbb{V}(\boldsymbol{\beta})\mathbf{X}\} + \mathbb{E}(\boldsymbol{\beta})^\top \mathbf{X}^\top \mathbf{X}\, \mathbb{E}(\boldsymbol{\beta}) - 2\,\mathbb{E}(\boldsymbol{\beta}^\top)\mathbf{X}^\top(\mathbf{y} - \sum_m h_m) + \\
&\quad (\mathbf{y} - \sum_m h_m)^\top(\mathbf{y} - \sum_m h_m)] \\
&= \mathrm{Tr}\{\mathbf{X}^\top \mathbb{V}(\boldsymbol{\beta})\mathbf{X}\} + \mathbb{E}(\boldsymbol{\beta})^\top \mathbf{X}^\top \mathbf{X}\, \mathbb{E}(\boldsymbol{\beta}) - 2\,\mathbb{E}(\boldsymbol{\beta}^\top)\mathbf{X}^\top(\mathbf{y} - \sum_m h_m) + \\
&\quad \mathbf{y}^\top\mathbf{y} - 2\mathbf{y}^\top \mathbb{E}(\sum_m h_m) + \mathbb{E}(\sum_m h_m^\top h_m),
\end{aligned}$$

where

$$\begin{aligned}
\mathbb{E}(\sum_m h_m^\top h_m) &= \mathbb{E}(h_1 + \ldots + h_m)^\top(h_1 + \ldots + h_m) \\
&\quad \sum_m \{\mathbb{E}(h_m^\top)\,\mathbb{E}(h_m) + \mathrm{Tr}(\mathbb{V}_m)\} + 2(\sum_m h_m^\top h_{m'}).
\end{aligned}$$

- Update $q_{\gamma_m}(\gamma_m)$:

  To calculate posterior odds of γ_m, we evaluate likelihood odds and prior odds respectively. We can update for likelihood odds as

  $$\frac{\Pr(\mathbf{y}|\gamma_m = 1, \boldsymbol{\gamma}_{(-k)}, \boldsymbol{\beta}, \sigma^2)}{\Pr(\mathbf{y}|\gamma_m = 0, \boldsymbol{\gamma}_{(-k)}, \boldsymbol{\beta}, \sigma^2)} = \frac{\mathbb{E}[\exp\{-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \sum_{m=1}^M h_m)^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \sum_{m=1}^M h_m)\}]}{\mathbb{E}[\exp\{-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \sum_{j\in -k}^M h_m)^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \sum_{j\in -k}^M h_m)\}]},$$

  and update for prior odds as

  $$p(\gamma_m|\gamma_{-m}) = \frac{\exp\{\gamma_m(a + \sum_{m'\in\text{-m}} b_{mm'}\gamma_{m'})\}}{1 + \exp\{(a + \sum_{m'\in\text{-m}} b_{mm'}\gamma_{m'})\}}.$$

  We then update posterior odds as

  $$O_m = \frac{\Pr(\mathbf{y}|\gamma_m = 1, \boldsymbol{\gamma}_{(-k)}, \boldsymbol{\beta}, \sigma^2)}{\Pr(\mathbf{y}|\gamma_m = 0, \boldsymbol{\gamma}_{(-k)}, \boldsymbol{\beta}, \sigma^2)} \cdot \frac{p(\gamma_m = 1|\boldsymbol{\gamma}_{-m})}{p(\gamma_m = 0|\boldsymbol{\gamma}_{-m})}.$$

Hence we have

$$\mathbb{E}(\gamma_m) = \frac{O_m}{1+O_m}. \tag{3.8}$$

- Update $q_{\tau_m}(\tau_m)$:

  If $\gamma_m = 0$, then $q_{\tau_m}(\tau_m) = 0$; If $\gamma_m = 1$, then

$$\begin{aligned} \log q_{\tau_m}(\tau_m) &= \mathbb{E}_{h_m}\{\log p(h_m) + \log p(\tau_m)\} + \text{const} \\ &\quad - (a + \frac{n}{2} - 1)\log\tau_m - (b + \frac{B}{2})\frac{1}{\tau_m} + \text{const}, \end{aligned} \tag{3.9}$$

  where $B = \mathbb{E}\{h_m^\top(\mathbf{K}_m)^{-1}h_m)\} = \text{Tr}(\mathbf{K}_m^{-1}\,\mathbb{V}_m) + \mathbb{E}(h_m^\top)\mathbf{K}_m^{-1}E(h_m)$. Hence, $q_{\tau_m}(\tau_m) = \text{IG}(a_{\sigma^2}^N, b_{\sigma^2}^N)$, where

$$a_{\tau_m}^N = a + \frac{n}{2} \quad b_{\tau_m}^N = b + \frac{B}{2}.$$

  Since it is standard inverse Gamma density, one can easily show $\mathbb{E}(\tau_m) = b_{\tau_m}^N/a_{\tau_m}^N - 1$.

- Update $q_{h_m}(h_m)$:

  If $\gamma_m = 0$, then $q_{\tau_m}(\tau_m) = 0$; If $\gamma_m = 1$, then

$$\begin{aligned} \log q_{h_m}(h_m) &= -\frac{1}{2}h_m^\top\{\frac{1}{\mathbb{E}(\sigma^2)} + \mathbb{E}(\tau_m)\mathbf{K}_m\}^{-1}h_m \\ &\quad - 2h_m^\top\{\mathbf{y} - \mathbf{X}\,\mathbb{E}(\boldsymbol{\beta}) - \frac{\mathbb{E}(\sum_{-m} h_{-m})}{\mathbb{E}(\sigma^2)}\}. \end{aligned} \tag{3.10}$$

  Hence, $q_{h_m}(h_m) = N(\boldsymbol{\mu}_{h_m}^N, \mathbf{V}_{h_m}^N) = N(\mathbf{K}_{h_m}^{-1}M_{h_m}, \mathbf{K}_{h_m}^{-1})$, where uncomment the following and make some changes

The procedures to implement our Gaussian selection methods are summarized in Algorithm

2.

Algorithm 2 Gaussian Process Selection by Variational Bayesian

```
Initialize {β, σ², h_m, τ_m, γ_m} for m = 1, ..., M.
Select prior.
for each iteration do:
    Update β by the equation (3.6).
    Update σ² by the equation (3.7).
    for each set m = 1, ..., M do
        Update h_m by the equation (3.10).
        Update τ_m by the equation (3.9).
        Update γ_m by the equation (3.8).
    end for
end for
Return: {E_β, E_σ², E_h_m, E_τ_m, E_γ_m}.
```

3.4 Simulation

In this chapter, we conduct two simulation studies to understand the performance of our Gaussian process selection approach. We evaluate the accuracy of Bayesian variational approximation using two priors: one is independent Bernoulli prior and the other is Ising prior. We also study the effect of varying parameters on the performance of Gaussian process selection.

3.4.1 Simulation Setup

We consider the semiparametric multi-kernel machine learning model (3.1),

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + h_1(\mathbf{Z}_1) + \ldots + h_M(\mathbf{Z}_M) + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I})$.

We set $n = 100$ observations, $q = 2$ fixed covariates, and $M = 10$ sets. We consider $p_m = 20$ elements for each set. Since few of sets would contribute to predictive power, we consider only two sets out of 10 sets are significant. Accordingly, the true indicators $\boldsymbol{\gamma} = \{\mathbf{0}_{8\times1}, \mathbf{1}_{2\times1}\}$ and the true $\boldsymbol{\tau} = \{\mathbf{0}_{8\times1}, \mathbf{2}_{2\times1}\}$.

The true regression coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2)^\top = (1, 10)^\top$ and $\mathbf{X} = 3\cos(\mathbf{Z}_{10}) + \mathbf{u}$ with $\mathbf{u}$ being independent of $\mathbf{Z}_{10}$ and following $N(0, \mathbf{I})$, where $\mathbf{Z}_{10}$ is the 10th set effect.

We consider $\mathbf{Z}_m \sim N(\mathbf{0}, \boldsymbol{\Sigma}_m^{-1})$. We will explain how to generate $\mathbf{Z}_m$ and $\boldsymbol{\Sigma}_m^{-1}$:

- $\mathbf{Z}_m$ are generated using the following ways:
  - If the set effect by mth set is significant, where $m = 9,\ 10$,

    Step 1: Sample elements $\mathbf{Z}_m \sim N(\mathbf{0}, \boldsymbol{\Sigma}_m^{-1})$, where $\boldsymbol{\Sigma}_m^{-1}$ will be generated using scale-free networks. We will further explain shortly.

    Step 2: Use Gaussian Kernel to reduce dimension from Chapter 3.2.2 to obtain $\mathbf{K}_m(.,.)$;

    Step 3: Generate set effects $h_m \sim \mathrm{GP}\{0, \tau_m \mathbf{K}_m(.,.)\}$.
  - If the set effect by mth set is not significant, where $m = 1, \ldots, 8$
    - Step 1: Sample elements $\mathbf{Z}_m \sim N(\mathbf{0}, 10^{-4}\mathbf{I})$;
    - Step 2: Set set effects h_m to $\mathbf{0}$, which means that nonparametric function $h_m(.)$ does not depend on $\mathbf{Z}_m$.

  As stated in Chapter 3.2.2, the kernel matrix describes the dissimilarity between the samples. The lower the entries of the matrix, the more similar they are.

  This is intended to mimic the structure of biological networks. For a scale-free network, the number of edges originate from a particular node and follow a power law distribution with a certain degree. That is the degree d distribution $p(d) \propto d^{-\alpha}$, where

α is some prefixed parameter (Barabási and Albert, 1999). We then generate $\mathbf{\Sigma}_m^{-1}$ by using Gaussian graphical model. Specifically, if data has joint Gaussian distribution, $\mathbf{z_m} \sim N(0, \mathbf{\Sigma}_m)$, where the inverse covariance matrix, $\mathbf{\Sigma}_m^{-1}$, indicates conditional dependency between elements. In other words, the zeros in the inverse covariance matrix $\mathbf{\Sigma}_m^{-1}$ correspond to zeros in the adjacency matrix. We implemented this generation by following steps described in Danaher et al. (2014):

Step 1: convert the network structure to corresponding adjacency matrix (i.e. $(0, 1)$ matrix with zeros on its diagonal);

Step 2: replace entries of 1 with other nonzero entries ($\sim$ uniform$\{0.5, 1\}$)

Step 3: make the matrix positive definite by dividing each off-diagonal entry by the sum of the absolute values of the off-diagonal entries in its row, and averaging the matrix with its transpose.

Scenario 1: all sets do not have overlapped elements. Elements within each set are generated independently each other.

Scenario 2: sets have overlapped elements. We fit the model when some significant and non-significant overlap. To represent this scenario, we created an augmented precision matrix that contains both signal and noise. In specific, we generate the non-significant and significant precision matrix same way as described in Chapter (3.4.1) and add them to an augmented precision matrix on its diagonal block.

3.4.2 Gaussian Process Selection with Two Priors

We compare the performance of our Gaussian process selection (referred as "GPS") using two priors of γs: independent Bernoulli prior and Ising prior. We refer two approaches to

GPSB and GPSI:

- **GPSB**: Gaussian process selection with Bernoulli prior. This approach fits the model without modeling the associations between sets. This is equivalent to setting $a = 0, \mathbf{B} = 0$.

- **GPSI**: Gaussian process selection with Ising prior. This approach fits the model with prior knowledge of structural set via Ising priors. From Chapter 3.2.2, vector $\mathbf{a} = a * \mathbf{1}$. For similarity matrix $\mathbf{B}$, we can incorporate prior by giving high correlation coefficients to the pairs of connected sets and low correlation coefficients to the pairs of unconnected sets. For example, $b_{m,m'} = 0.9$ if the sets are m and m' are included in the model and $b_{m,m'} = 0.1$ otherwise. In this way, the Ising prior forces the inclusion of edges by sharing similar information.

The prior knowledge can be defined by biologists in practice, but in this chapter, we estimate the matrix $\mathbf{B}$ by the data. Specifically, for a given pair of sets m and m', we calculate the correlation for each variable and take the average of this quantity. The parameter a can be set based on the empirical suggestion by Li and Zhang (2010).

3.4.3 Evaluation Metrics

Following the standard variable selection practice, beyond the parameter estimation, we particularly assess the performance of Gaussian process selection regression model by focusing on the posterior probability of inclusion, γ_m. The key measure of the model should reflect the predictive power, not only the ability to include the significant sets but also exclude the non-significant sets.

A number of metrics are worth assessing: true positive rate (TPR), true negative rate (TNR),

and accuracy (ACC). To calculate these metrics, we first define false positive (FP), true positive (TP), false negative (FN), and true negative (TN) for γ_m:

$$\begin{aligned}
\mathrm{TP}_m &= \mathbf{I}\{\gamma_m = 1, \hat{\gamma}_m = 1\};\\
\mathrm{FP}_m &= \mathbf{I}\{\gamma_m = 0, \hat{\gamma}_m = 1\};\\
\mathrm{TN}_m &= \mathbf{I}\{\gamma_m = 0, \hat{\gamma}_m = 0\};\\
\mathrm{FN}_m &= \mathbf{I}\{\gamma_m = 1, \hat{\gamma}_m = 0\}.
\end{aligned}$$

We also introduce precision and recall which are widely used in information retrieval and binary classification. The intuition is that the precision measures "how many selected sets are relevant" and the recall measures "how many relevant sets are selected". We can follow the definition:

$$\begin{aligned}
\text{Precision} &= \frac{1}{M}\sum_{m=1}^{M}\left(\frac{\mathrm{TP}_m}{\mathrm{TP}_m + \mathrm{FP}_m}\right);\\
\text{Recall} &= \frac{1}{M}\sum_{m=1}^{M}\left(\frac{\mathrm{TP}_m}{\mathrm{TP}_m + \mathrm{FN}_m}\right).
\end{aligned}$$

In the binary classification problem, F1-measure accounts for both precision and recall, by taking the harmonic average of these two terms:

$$F_1 = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}},$$

which is equivalent to:

$$F_1 = \frac{2 \cdot \sum_{m=1}^{M} \mathrm{TP}_m}{2 \cdot \sum_{m=1}^{M} \mathrm{TP}_m + \sum_{m=1}^{M} \mathrm{FN}_m + \sum_{m=1}^{M} \mathrm{FP}_m}.$$

3.4.4 Sensitivity Analysis of Gaussian Graphical Model

In this subchapter, we discuss the reason we apply Gaussian graphical model to decode the inverse covariance matrix $\mathbf{\Sigma}_m^{-1}$. Our experiments showed there is not a significant improvement by using Gaussian graphical model in terms of selecting significant set effects. This means our Gaussian Process Selection is not sensitive to the graph of the inverse covariance matrix. This also means less computation work as calculating inverse covariance matrix takes a lot of time, especially we have a large dimension of a set. Therefore, it is pretty safe to use settings proposed by Liu et al. (2007) which is equivalent to using an identity matrix of $\mathbf{\Sigma}_m$ but one has to tune the parameter τ.

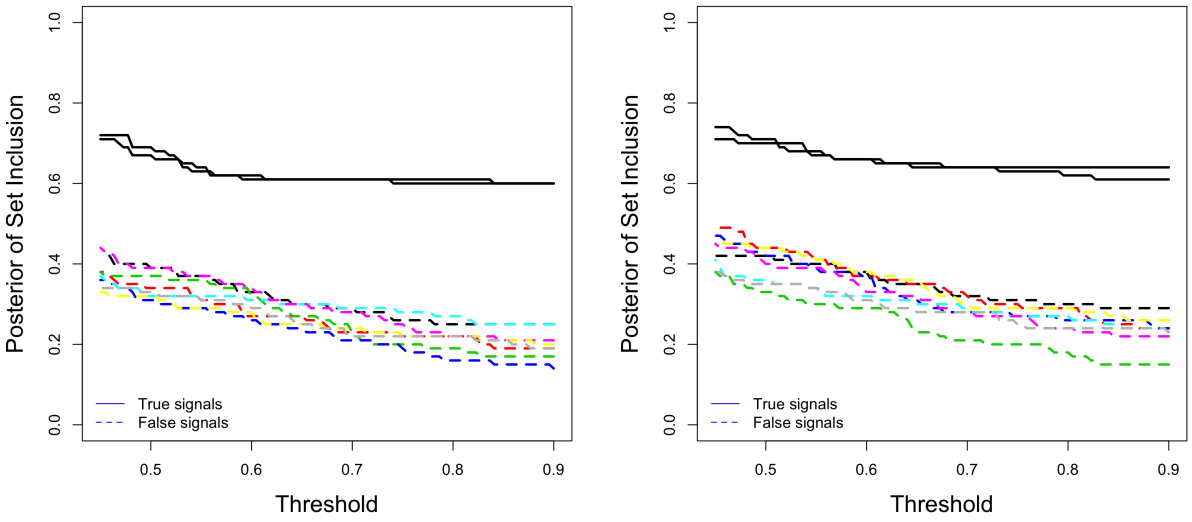
We do the sensitivity analysis by a toy example. We generate data based on Chapter 3.4.1 in scenario 1 with Ising priors. Then we show the result in Table 3.1 and Figure 3.1. It is obvious that there is not a significant improvement by using Gaussian Graphical Model in terms of selecting significant set effects.

Method	Precision	Recall	F1
GPSI + $\hat{\mathbf{\Sigma}}_{GGM}^{-1}$	0.64(0.34)	0.70(0.26)	0.59(0.20)
GPSI + $\hat{\mathbf{\Sigma}}_{\boldsymbol{I}}^{-1}$	0.67(0.36)	0.70(0.31)	0.60(0.23)

Table 3.1: Sensitivity analysis of Gaussian process selection by using Gaussian graphical model with fixed threshold t of 0.6 on 100 simulated runs; GPSI=Gaussian process selection with Ising prior; Precision $= \frac{1}{M}\sum_{m=1}^{M}(\frac{\mathrm{TP}_m}{\mathrm{TP}_m+\mathrm{FP}_m})$; Recall $= \frac{1}{M}\sum_{m=1}^{M}(\frac{\mathrm{TP}_m}{\mathrm{TP}_m+\mathrm{FN}_m})$; $F_1 = 2 \cdot \frac{\mathrm{Recall}\cdot\mathrm{Precision}}{\mathrm{Recall}+\mathrm{Precision}}$.

3.4.5 Simulation Results

We apply our approaches with two priors under two simulation scenarios. We estimate the posterior probability of inclusion for each set along with other parameters.



(a) (b)

Figure 3.1: Sensitivity analysis of Gaussian process selection: the profile curves of the selection probability of GSPI under scenario I; (a) GPSI + $\hat{\boldsymbol{\Sigma}}_{GGM}^{-1}$ (b) GPSI + $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{I}}^{-1}$; GPSI=Gaussian process selection with Ising prior.

Estimation of Fixed Effects

The point estimators are represented in Table 3.2. The results suggest a good estimation accuracy of $\boldsymbol{\beta}$, which are very close to the true values. For these three particular cases, the estimation of $\boldsymbol{\beta}$ is unbiased no matter which case. However, one may observe that the estimation of σ is rather biased. This is reasonable because the set effects of sets are unobserved. We further discuss this issue in Chapter 3.4.5.

Estimation of Set Effects

The choice of threshold, t, the probability of posterior inclusion, plays a key role as it separates the significant set from non-significant one. It is defined as follow:

Scenario	Method	n	p_m	β_0	β_1	σ^2
1	GPSB	50	10	0.92 (0.19)	10.06 (0.05)	1.89 (0.40)
	GPSI			1.02 (0.16)	9.93 (0.21)	1.97 (0.25)
2	GPSB			1.02 (0.12)	10.00 (0.09)	1.84 (0.19)
	GPSI			0.99 (0.06)	9.99 (0.03)	1.93 (0.30)
1	GPSB	50	20	1.00 (0.08)	9.92 (0.01)	1.97 (0.46)
	GPSI			1.00 (0.25)	9.95 (0.19)	1.89 (0.37)
2	GPSB			0.98 (0.05)	9.99 (0.09)	1.84 (0.25)
	GPSI			0.98 (0.03)	9.95 (0.09)	1.87 (0.24)
1	GPSB	100	10	0.92 (0.09)	9.97 (0.03)	2.23 (0.08)
	GPSI			1.07 (0.02)	9.96 (0.10)	1.97 (0.28)
2	GPSB			0.95 (0.02)	9.98 (0.04)	2.03 (0.29)
	GPSI			0.96 (0.12)	9.94 (0.03)	1.96 (0.26)
1	GPSB	100	20	0.95 (0.11)	9.93 (0.06)	2.11 (0.28)
	GPSI			1.03 (0.03)	9.89 (0.05)	1.97 (0.30)
2	GPSB			0.96 (0.06)	10.02 (0.12)	2.00 (0.28)
	GPSI			1.00 (0.04)	10.06 (0.02)	1.94 (0.26)

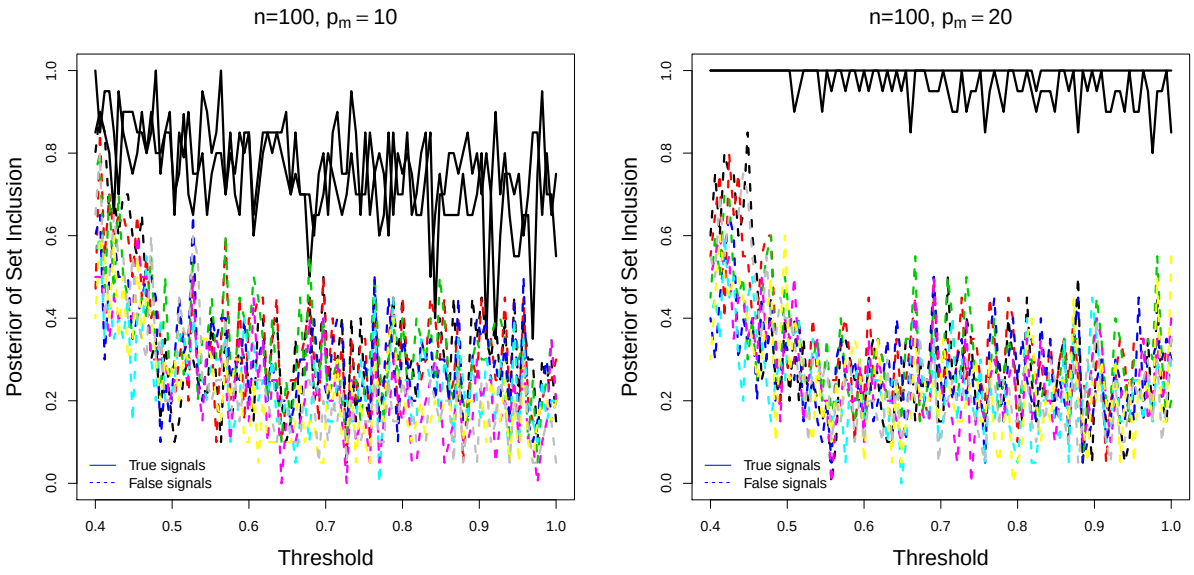
Table 3.2: Simulation results of estimated regression coefficients with standard errors in Gaussian process selection regression model over 100 simulated runs under different settings of sample size n and set dimension p_m; The true parameters are $\beta_0 = 1, \beta_1 = 10, \sigma^2 = 1$; GPSB=Gaussian process selection with Bernoulli prior; GPSI=Gaussian process selection with Ising prior.

$$\hat{\gamma}_m = \begin{cases} 1, & \text{if } p(\gamma_m|.) \geq t, \\ 0, & \text{if } p(\gamma_m|.) < t. \end{cases}$$

We investigate the performance of the overall performance of Gaussian process selection by varying thresholds.

Figure 3.2 and 3.3 displays selection probability of 10 sets against different thresholds under scenario 1. It shows us a general idea where an optimal threshold can be found. We can observe that when the threshold is larger than 0.5, the profile curves of selection probabilities of signal sets and non-signal sets are well separated. The choice of the threshold makes sense because set effects are unobserved from different sources and the algorithm might overestimate some of the noisy sets and underestimate some of the signal sets. It also leads to a high true positive rate (or recall) and a low true negative rate (or precision). Interesting

facts are that this phenomenon is mitigated when the dimension of each set increases. These figures demonstrate that the gap of selection probability for noise and signals is more obvious, as the p_m increases. So is the mean of posterior of inclusion.



(a) (b)

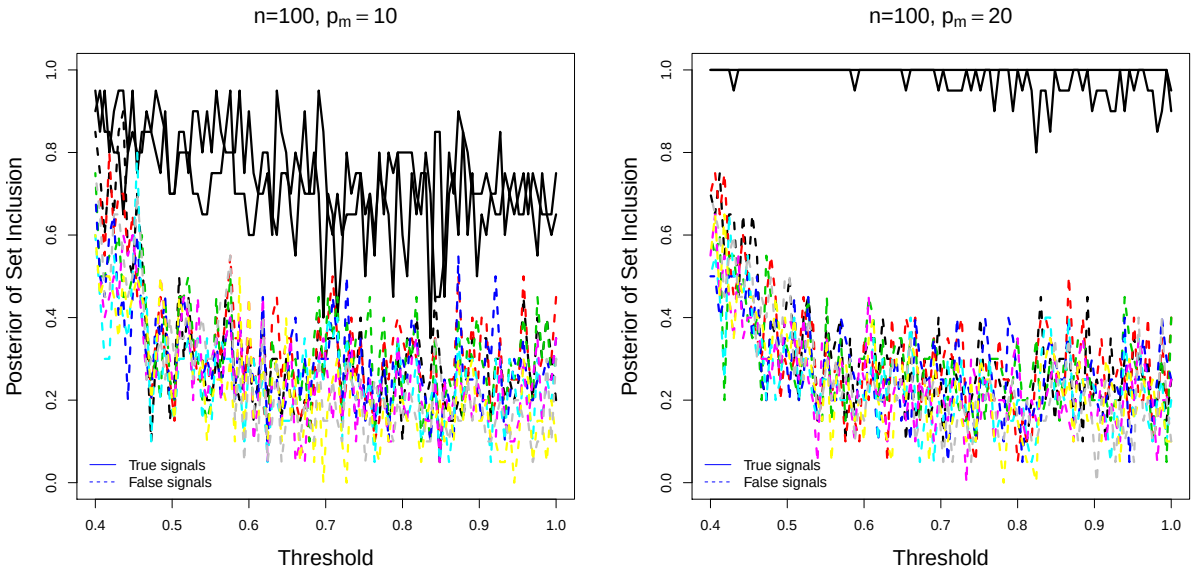
Figure 3.2: The profile curves of the selection probability of GSPB (a) (b) under scenario I. The probability of inclusion is calculated for each threshold based on 100 simulated data sets; The posteriors of set inclusion using GSPB with specific specific threshold t; GPSB=Gaussian process selection with Bernoulli prior; GPSI=Gaussian process selection with Ising prior.

Similar patterns of GPSI and GPSB can be found under scenario 2 (see, Figures 3.4-3.5).

Performance of Gaussian Process Selection with Two Priors

We study the performance of Gaussian process selection under two scenario by different sample sizes.

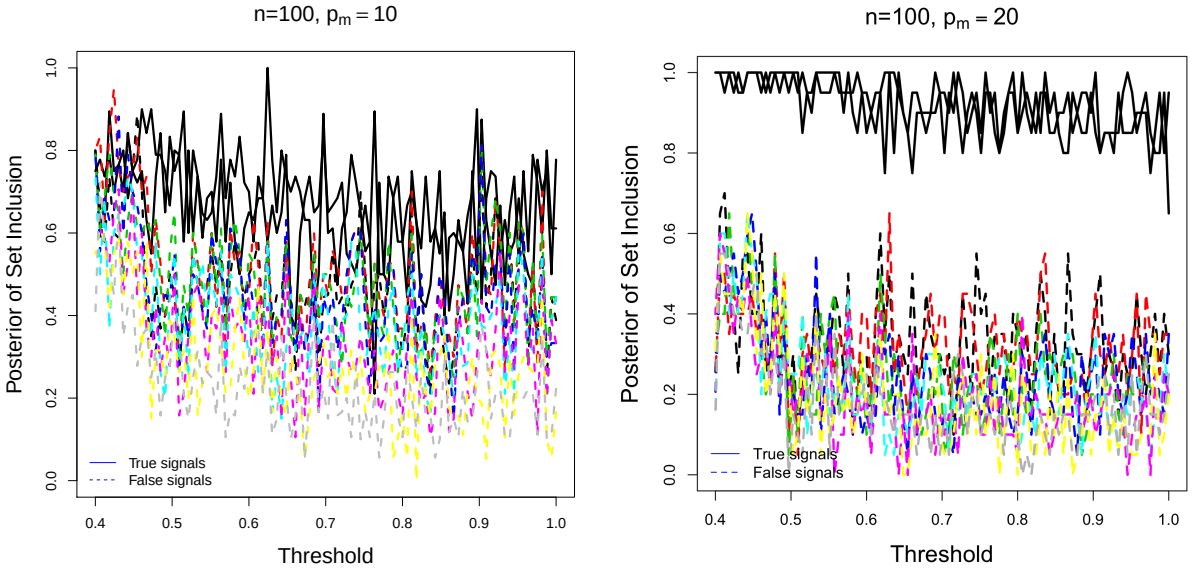
Figure 3.6 and 3.7 display the performance of our Gaussian process selection in terms of F1 measure. One can observe that when the sample size is large ($n = 100$), the results from these two cases seem very close to each other. However, when we have a small sample size ($n = 50$),



(a) (b)

Figure 3.3: The profile curves of the selection probability of GSPI (a) (b) under scenario I. The probability of inclusion is calculated for each threshold based on 100 simulated data sets; The posteriors of set inclusion using GSPI with specific specific threshold t; GPSB=Gaussian process selection with Bernoulli prior; GPSI=Gaussian process selection with Ising prior.

the Gaussian process selection with Bernoulli prior is outperformed by Ising priors. More specifically, we found with Bernoulli prior, one may end up over-estimating the set effects which lead to committing type I error, which is also shown in Chapter 3.4.5. However, one can avoid this by adding network structure via Ising priors. Again, these accumulated effects sometimes can only explain a small portion of the variance of the phenotype. However, the estimation of $\boldsymbol{\gamma}$ greatly improves as the structural set prior knowledge is introduced. This also makes the estimation of σ^2 less biased under Scenario 2.



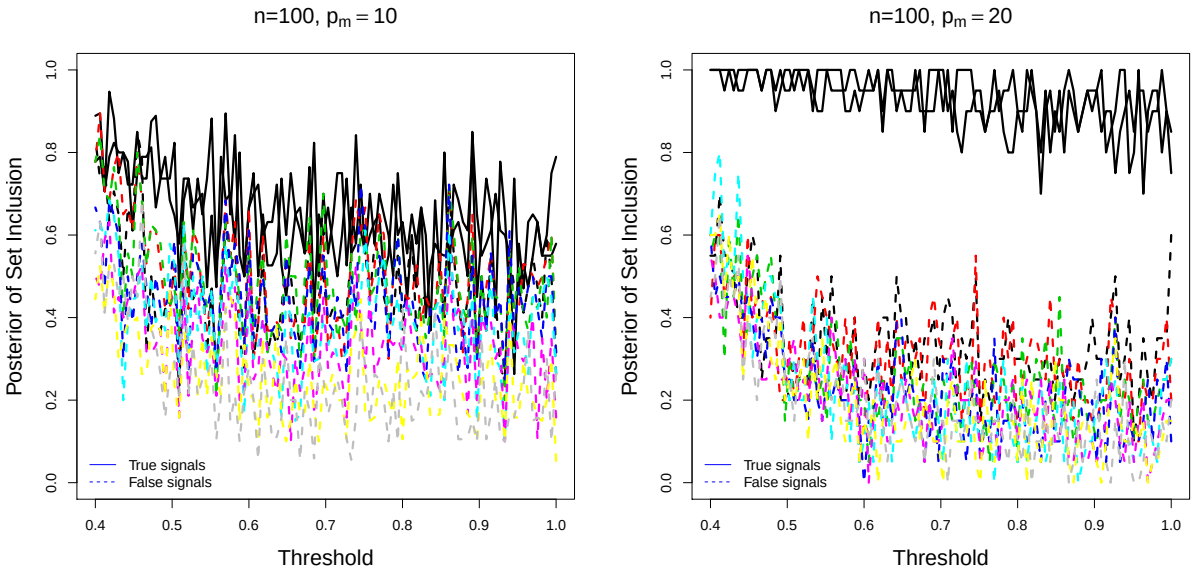
(a) (b)

Figure 3.4: The profile curves of the selection probability of GSPB (a) (b) under scenario II. The probability of inclusion is calculated for each threshold based on 100 simulated data sets; The posteriors of set inclusion using GSPB with specific specific threshold t; GPSB=Gaussian process selection with Bernoulli prior; GPSI=Gaussian process selection with Ising prior.

3.5 Application: a Type II Diabetes Genomics Data

In this chapter, we applied our Gaussian process selection regression model 3.1 to a real microarray expression data set on type II diabetes from Mootha et al. (2004) and (Pang et al., 2015) and evaluate the accuracy of finding significant random effects models. We then evaluate the performance of Gaussian process selection and compare it with the other method (Fang et al., 2018) that was applied in the same dataset.

We studied a total of 278 pathways consisting of 128 KEGG pathways and 150 curated pathways. The KEGG pathway database is a collection of curated pathways representing our knowledge on the molecular interaction and reaction networks for metabolism, genetic information processing, environmental information processing, cellular processes, and human



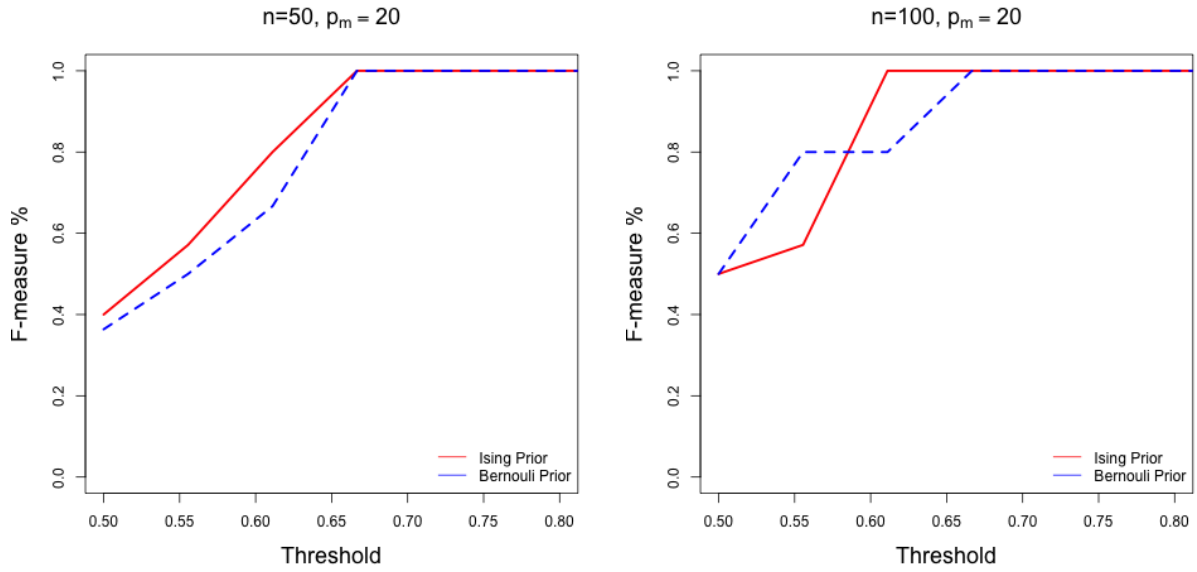
(a) (b)

Figure 3.5: The profile curves of the selection probability of GSPI (a) (b) under scenario II. The probability of inclusion is calculated for each threshold based on 100 simulated data sets; The posteriors of set inclusion using GSPI with specific specific threshold t; GPSB=Gaussian process selection with Bernoulli prior; GPSI=Gaussian process selection with Ising prior.

diseases. The 150 curated pathways were constructed from known biological experiments by Mootha and colleagues. Our interest in pathways is to understand the specific pathway effect rather than the effects of individual genes.

In our analysis, we considered a continuous outcome, where $\mathbf{Y}$ is the log-transformed glucose level. Let $\mathbf{X_{n\times 2}}$ be the body mass index (BMI) and $\mathbf{Z}_m$ be the $p_m \times n$ gene expression levels within each pathway. This data contains $n = 35$ samples, $p_\beta = 2$ clinical covariates and $M = 251$ pathways with the number of genes p_{τ_m} for each ranging from 3 to 543. All the variables are continuous. Our goal is to identify important pathways that affect the glucose level related to diabetes after adjusting for the BMI effect.

We normalize all the genes with zero means and one standard deviation for pathways. Then, we randomly selected five pathways to apply our Gaussian process selection regression model



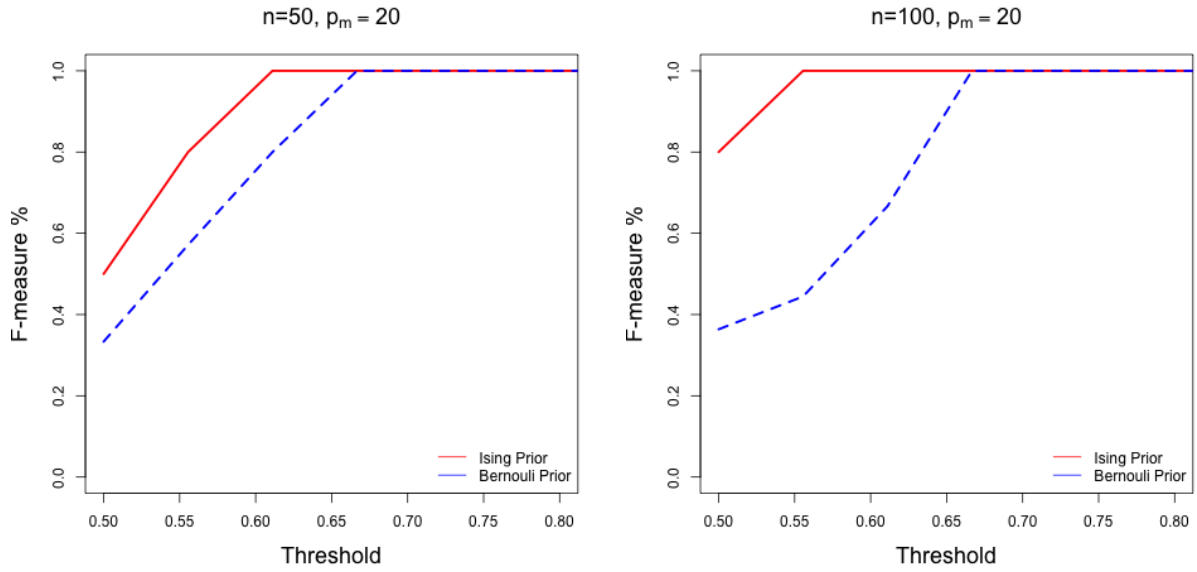
(a) (b)

Figure 3.6: F_1 measure against varying thresholds wtih two priors under scenario 1 where sets are independent when sample size $n = 50$ (a) and $n = 100$ (b); $F_1 = \frac{2 \cdot \sum_{m=1}^{M} \mathrm{TP}_m}{2 \cdot \sum_{m=1}^{M} \mathrm{TP}_m + \sum_{m=1}^{M} \mathrm{FN}_m + \sum_{m=1}^{M} \mathrm{FP}_m}$, where $M = 10$.

with 10,000 runs. This means our final results include some randomness as the fewer pathway included, the more accurate we are able to identify random set effects caused by pathways. We calculate the posterior mean of inclusion of each pathway after the algorithm converges and then select the significant pathways. We carried out the algorithm with Ising priors.

We display the results of the top 50 significant pathways in Figure 3.8. Table 3.4 ranked ascendingly in terms of the posterior of inclusion to detect pathway effect by Gaussian process selection regression.

We compared the output of our Gaussian process selection regression model to the other methods (Fang et al., 2018) that was applied in the same dataset. The method selects important variables for recovering sparsity in nonadditive nonparametric models which is essentially univariate analysis. Among the 20 pathways with the most significant overall



(a) (b)

Figure 3.7: F_1 measure against varying thresholds wtih two priors under scenario 2 where signal sets are overlapped with noisy sets when sample size $n = 50$ (a) and $n = 100$ (b); $F_1 = \frac{2\cdot\sum_{m=1}^{M}\mathrm{TP}_m}{2\cdot\sum_{m=1}^{M}\mathrm{TP}_m+\sum_{m=1}^{M}\mathrm{FN}_m+\sum_{m=1}^{M}\mathrm{FP}_m}$, where $M = 10$.

pathway effect, we discovered that significant pathway effects detected by both methods are $\{8, 101, 103, 133, 144, 151, 158, 172, 228, 229, 230, 236, 271\}$.

Based on the view of the biological literature, pathways $\{230, 228, 172\}$ are known to Type II diabetes. Pathway 230 is the OXPHOS HG-U133A probes pathway. In patients with type II diabetes, genes involved in oxidative phosphorylation were reported to be coordinated with fasting hyperglycemia in the livers (Misu et al., 2007). The transcription levels of the genes involved in oxidative phosphorylation mechanisms are consistently lower in diabetics than in controls (Mootha et al., 2003),(Misu et al., 2007). Pathway 228 is involved in oxidative phosphorylation and is known to be related to diabetes (Mootha et al., 2003), (Misu et al., 2007). This pathway is associated with a process of human cell respiration (and generally eukaryotes); it contains co-regulated genes across different tissues and is associated with the

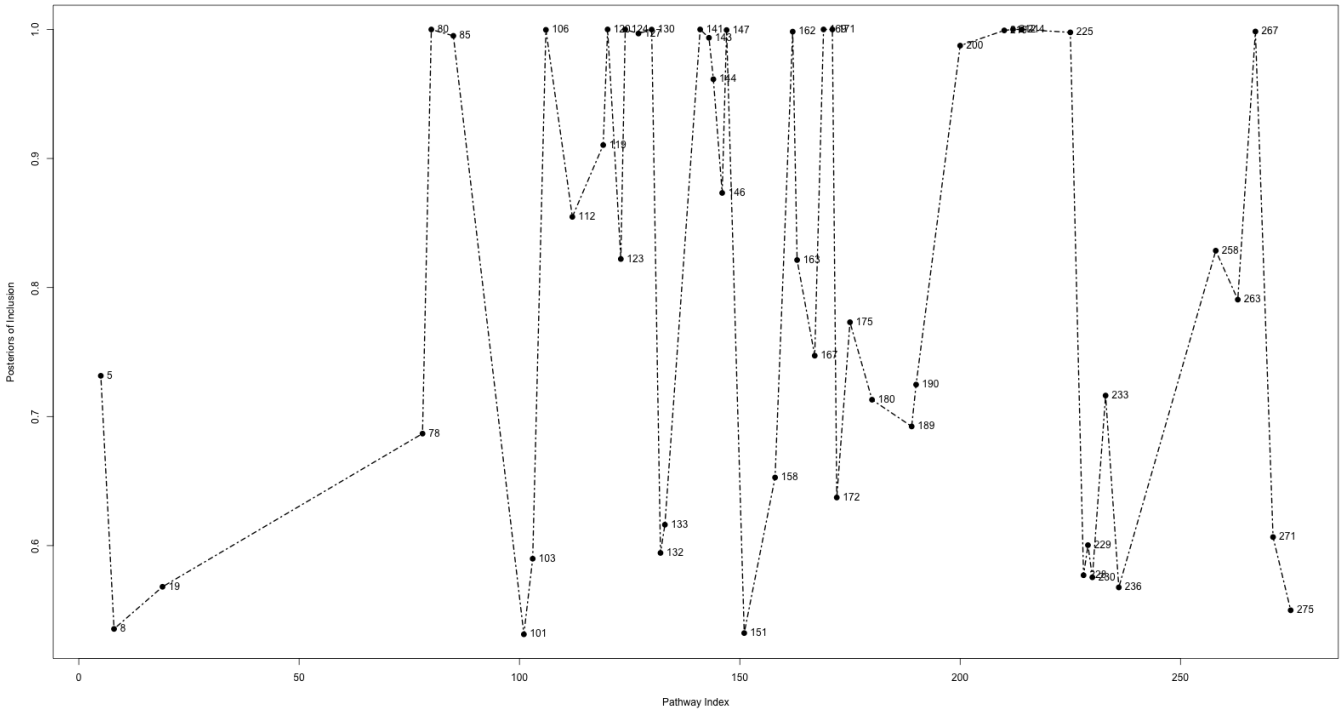


Figure 3.8: Posteriors of inclusion for top 50 significant pathways related to Diabetes II in Type II diabetes genetic pathway application data.

disposal of insulin/glucose. It is related with ATP synthesis, an energy transfer pathway. Pathway 172 is 23 the MAP00530 Aminosugars metabolism pathway. Aminosugars (or glucosamine) have no effect on blood glucose levels, glucose metabolism or insulin sensitivity in healthy subjects, people with diabetes or people with impaired glucose tolerance at any oral dose level. (Simon et al., 2011).

3.6 Discussion

We have proposed a Gaussian process selection approach via a semiparametric multi-kernel model framework for studying the effects of fix covariates and multiple sets. We employ a variational Bayesian algorithm to Gaussian process selection problem and provide closed-form solutions for updates of parameters. Our Gaussian process selection approach is suitable for high-dimensional data, as it can reduce the dimension of data by kernel machine learning. One can also incorporate prior knowledge for structural sets via Ising prior, making it feasible to apply in genetic data analysis.

Our Gaussian process selection approach can achieve good accuracy in terms of overall F1 measure. This model also provides an interpretation through the posterior probabilities of inclusions.

This work is novel in several aspects. First, the model is developed on the Bayesian variable selection framework that catches the signals via on variance rather than mean function. Second, it is able to detect multiple significant set effects simultaneously. Third, our approach is developed by using the variational Bayesian instead of MCMC. Compared to MCMC, variational inference based method is faster, deterministic and easy to determine when to stop.

There are several possibilities for future work. For instance, one could optimize the proposed variational Bayesian method in terms of speed or convergence. In fact, a relevant approach has been taken in recent work (Wang et al., 2016) where the parameters are updated batch-wise instead of component-wise. This leads to a fast and scalable algorithm.

A `Julia` package implementing Gaussian process selection regression are available on author's `Github` repository.

Pathway ID	Name	Posterior of Inclusion
214	MAPK signaling pathway	1.00
169	MAP00520 Nucleotide sugars metabolism	1.00
171	MAP00522 Erythromycin biosynthesis	1.00
120	MAP00040 Pentose and glucuronate interconversions	0.99
124	MAP00061 Fatty acid biosynthesis path 1	0.99
141	MAP00253 Tetracycline biosynthesis	0.99
130	MAP00130 Ubiquinone biosynthesis	0.99
80	Fatty acid biosynthesis	0.99
210	MAP03030 DNA polymerase	0.99
267	Tetrachloroethene degradation	0.99
225	Nucleotide sugars metabolism	0.99
212	MAP03090 Type II secretion system	0.99
127	MAP00072 Synthesis and degradation of ketone bodies	0.99
106	Inositol metabolism	0.99
147	MAP00300 Lysine biosynthesis	0.97
162	MAP00471 D Glutamine and D glutamate metabolism	0.97
200	MAP00790 Folate biosynthesis	0.95
144	MAP00272 Cysteine metabolism	0.95
85	Folate biosynthesis	0.95
143	MAP00271 Methionine metabolism	0.94
258	Sphingophospholipid biosynthesis	0.94
146	MAP00290 Valine leucine and isoleucine biosynthesis	0.93
119	MAP00031 Inositol metabolism	0.92

Table 3.3: Posterior of Inclusion for top 30 pathway significant in the pathway effect Gaussian process selection regression model using Ising priors in Type II diabetes genetic pathway application data; The number of pathway included is 5.

Pathway ID	Name	Posterior of Inclusion
123	MAP00053 Ascorbate and aldarate metabolism	0.82
5	Alkaloid biosynthesis I	0.82
163	MAP00472 D Arginine and D ornithine metabolism	0.82
112	KET HG-U133A probes	0.81
167	MAP00511 N Glycan degradation	0.77
175	MAP00533 Keratan sulfate biosynthesis	0.76
263	Synthesis and degradation of ketone bodies	0.75
19	Benzoate degradation via hydroxylation	0.72
190	MAP00632 Benzoate degradation	0.72
233	Pentose and glucuronate interconversions	0.72
180	MAP00580 Phospholipid degradation	0.71
189	MAP00631 1 2 Dichloroethane degradation	0.69
78	Ethylbenzene degradation	0.68
158	MAP00430 Taurine and hypotaurine metabolism	0.65
172	MAP00530 Aminosugars metabolism	0.63
133	MAP00190 Oxidative phosphorylation	0.61
271	Tyrosine metabolism	0.60
229	Oxidative phosphorylation	0.60
132	MAP00150 Androgen and estrogen metabolism	0.59
103	Histidine metabolism	0.58
228	Oxidation Phosphorylation	0.57
230	OXPHOS HG-U133A probes	0.57
19	Benzoate degradation via hydroxylation	0.56
236	Phenylalanine metabolism	0.56
275	Valine, leucine and isoleucine biosynthesis	0.54
8	Aminoacyl-tRNA biosynthesis	0.53
151	MAP00350 Tyrosine metabolism	0.53
101	Glyoxylate and dicarboxylate metabolism	0.53

Table 3.4: (Continued)Posterior of Inclusion for top 30 pathway significant in the pathway effect Gaussian process selection regression model using Ising priors in Type II diabetes genetic pathway application data; The number of pathway included is 5.

Chapter 4

Summary and Future Research

Major conclusions and contributions of this dissertation are summarized in this chapter and possible future research areas are introduced.

4.1 Summary

In Chapter 2, we have proposed the Bayesian multiple Gaussian graphical models, a framework for learning networks structure of multiple groups that could be possibly connected to each other. Given a data set that has a hierarchical structure, this model is able to uncover the network structure for set and element simultaneously. Furthermore, our approach can evaluate the Bayes factor to decide the class membership for each class. We design and employ a simple Gibbs sampling scheme to solve the Bayesian multiple Gaussian graphical models. Our method is also robust to missing data. This is because the Gibbs sampler is able to retain the partial information in each iteration having missing data.

Our method is comparable to the existing method in terms of accuracy of adjacency matrix

estimation. Our method also provides a good statistical interpretation through posterior probabilities for each parameter, because our generative model is more explainable and interpretable over deterministic models. We define the model by explaining the generative mechanism in a top-down fashion 2.3.

In the first project, we make several novel contributions: (1) development of a Bayesian multiple Gaussian graphical models that are able to reveal set networks and element networks simultaneously; (2) establishment of a Bayesian factor approach to evaluate class membership for the observations from Gaussian graphical model; (3) estimation of the model by using Bayesian rather than frequentist approach. This Bayesian formulation is fully model-based, thus has nice probabilistic interpretations. Clustering data for the Gaussian graphical model is very difficult and only a few works focus on the relevant problem so far.

In Chapter 3, we have proposed a Gaussian process selection approach via a semiparametric multi-kernel model framework for studying the effects of fix covariates and multiple sets. We employ a variational Bayesian algorithm to Gaussian process selection problem and provide closed-form solutions for updates of parameters. Our Gaussian process selection approach is suitable for high-dimensional data, as it can reduce the dimension of data by kernel machine learning. One can also incorporate prior knowledge for structural sets via Ising prior, making it feasible to apply in genetic data analysis.

Our Gaussian process selection approach can achieve good accuracy in terms of overall F1 measure. This model also provides an interpretation through the posterior probabilities of inclusions. This may not for every scenario, so we encourage users to evaluate the method by applying it to the other type of data. However, the algorithm is able to evaluate models even users introduce new random effects.

This work is novel in several aspects. First, the model is developed on the Bayesian variable

selection framework that catches the signals via on variance rather than mean function. Second, it is able to detect multiple significant set effects simultaneously. Third, our approach is developed by using variational Bayesian instead of MCMC. Compared to MCMC, variational inference based method is faster, deterministic and easy to determine when to stop.

4.2 Future Research

For Chapter 2, there are several potentials for future work. For instance, when it comes to clustering the observations, one could discuss the properties of the Bayesian method in terms of convergence. There are rooms for improving clustering accuracy. One may also work on developing a fast and scalable Bayesian algorithm to apply to larger datasets. Another possibility is to extend our model to allow discrete data. Finally, we can extend our method when the multivariate Gaussian assumption is violated or there are outliers in the data.

For Chapter 3, there are several possibilities for future work as well. For instance, one could optimize the proposed variational Bayesian method in terms of speed or convergence. In fact, a relevant approach has been taken in recent work (Wang et al., 2016) where the parameters are updated batch-wise instead of component-wise. This leads to a fast and scalable algorithm.

Bibliography

Banerjee, O., Ghaoui, L. E., and dAspremont, A. (2008), "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *Journal of Machine Learning Research*, 9, 485–516.

Barabási, A.-L., and Albert, R. (1999), "Emergence of scaling in random networks," *Science*, 286, 509–512.

Barbieri, M. M., and Berger, J. O. (2004), "Optimal predictive model selection," *The Annals of Statistics*, 32, 870–897.

Bishop, C. M. (2006), *Pattern recognition and machine learning,* springer.

Cai, T., Tonini, G., and Lin, X. (2011), "Kernel machine approach to testing the significance of multiple genetic markers for risk prediction," *Biometrics*, 67, 975–986.

Carbonetto, P., and Stephens, M. (2012), "Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies," *Bayesian Analysis*, 7, 73–108.

Chen, H., and Sharp, B. M. (2004), "Content-rich biological network constructed by mining PubMed abstracts," *BMC Bioinformatics*, 5, 147.

Chen, Y., Stingo, F., Tadesse, M., and Vannucci, M. (2011), "Incorporating biological information in Bayesian models for the selection of pathways and genes," *The Annals of Applied Statistics*, 5, 1978–2002.

Chlebowski, R. T., Chen, Z., Anderson, G. L., Rohan, T., Aragaki, A., Lane, D., Dolan, N. C., Paskett, E. D., McTiernan, A., and Hubbell, F. A. (2005), "Ethnicity and breast cancer: factors influencing differences in incidence and outcome," *Journal of the National Cancer Institute*, 97, 439–448.

Danaher, P., Wang, P., and Witten, D. M. (2014), "The joint graphical lasso for inverse covariance estimation across multiple classes," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76, 373–397.

Dempster, A. P. (1972), "Covariance selection," *Biometrics*, 28, 157–175.

Fan, J., and Li, R. (2001), "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, 96, 1348–1360.

Fang, Z., Kim, I., and Jung, J. (2018), "Semiparametric Kernel-Based Regression for Evaluating Interaction Between Pathway Effect and Covariate," *Journal of Agricultural, Biological and Environmental Statistics*, 23, 129–152.

Friedman, J., Hastie, T., and Tibshirani, R. (2008), "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, 9, 432–441.

George, E. I., and McCulloch, R. E. (1993), "Variable selection via Gibbs sampling," *Journal of the American Statistical Association*, 88, 881–889.

Guo, J., Levina, E., Michailidis, G., and Zhu, J. (2011), "Joint estimation of multiple graphical models," *Biometrika*, 98, 1–15.

Keenan, T., Moy, B., Mroz, E. A., Ross, K., Niemierko, A., Rocco, J. W., Isakoff, S., Ellisen, L. W., and Bardia, A. (2015), "Comparison of the genomic landscape between primary breast cancer in African American versus white women and the association of racial differences with tumor recurrence," *Journal of Clinical Oncology*, 33, 3621.

Kim, I., Liu, Y., and Zhao, H. (2007), "Bayesian methods for predicting interacting protein pairs using domain information," *Biometrics*, 63, 824–833.

Kim, I., Pang, H., and Zhao, H. (2012), "Bayesian semiparametric regression models for evaluating pathway effects on continuous and binary clinical outcomes," *Statistics in Medicine*, 31, 1633–1651.

Kuo, L., and Mallick, B. (1998), "Variable selection for regression models," *Sankhyā: The Indian Journal of Statistics, Series B*, 60, 65–81.

Li, F., and Zhang, N. R. (2010), "Bayesian variable selection in structured high-dimensional covariate spaces with applications in genomics," *Journal of the American Statistical Association*, 105, 1202–1214.

Liu, D., Lin, X., and Ghosh, D. (2007), "Semiparametric Regression of Multidimensional Genetic Pathway Data: Least-Squares Kernel Machines and Linear Mixed Models," *Biometrics*, 63, 1079–1088.

Marlin, B. M., and Murphy, K. P. (2009), Sparse Gaussian graphical models with unknown block structure,, in *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, 705–712.

Meinshausen, N., and Bühlmann, P. (2006), "High-dimensional graphs and variable selection with the lasso," *The Annals of Statistics*, 43, 1436–1462.

Misu, H., Takamura, T., Matsuzawa, N., Shimizu, A., Ota, T., Sakurai, M., Ando, H., Arai, K., Yamashita, T., and Honda, M. (2007), "Genes involved in oxidative phosphorylation are coordinately upregulated with fasting hyperglycaemia in livers of patients with type 2 diabetes," *Diabetologia*, 50, 268–277.

Mootha, V. K., Handschin, C., Arlow, D., Xie, X., Pierre, J. S., Sihag, S., Yang, W., Altshuler, D., Puigserver, P., and Patterson, N. (2004), "Errα and Gabpa/b specify PGC-1α-dependent oxidative phosphorylation gene expression that is altered in diabetic muscle," *Proceedings of the National Academy of Sciences of the United States of America*, 101, 6570–6575.

Mootha, V. K., Lindgren, C. M., Eriksson, K.-F., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E., Ridderstråle, M., and Laurila, E. (2003), "PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes," *Nature Genetics*, 34, 267.

Pang, H., Kim, I., and Zhao, H. (2015), "Random effects model for multiple pathway analysis with applications to type II diabetes microarray data," *Statistics in Biosciences*, 7, 167–186.

Peng, J., Wang, P., Zhou, N., and Zhu, J. (2009), "Partial correlation estimation by joint sparse regression models," *Journal of the American Statistical Association*, 104, 735–746.

Peterson, C., Stingo, F. C., and Vannucci, M. (2015), "Bayesian inference of multiple Gaussian graphical models," *Journal of the American Statistical Association*, 110, 159–174.

Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008), "Sparse permutation invariant covariance estimation," *Electronic Journal of Statistics*, 2, 494–515.

Rue, H., and Held, L. (2005), *Gaussian Markov random fields: theory and applications,* CRC press.

Shi, L., Campbell, G., Jones, W. D., Campagne, F., Wen, Z., Walker, S. J., Su, Z., Chu, T.-M., Goodsaid, F. M., and Pusztai, L. (2010), "The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models," *Nature Biotechnology*, 28, 827–838.

Simon, R., Marks, V., Leeds, A., and Anderson, J. (2011), "A comprehensive review of oral glucosamine use and effects on glucose metabolism in normal and diabetic individuals," *Diabetes/Metabolism Research and Reviews*, 27, 14–27.

Wang, H. (2012), "Bayesian graphical lasso models and efficient posterior computation," *Bayesian Analysis*, 7, 867–886.

Wang, J., Liang, F., and Ji, Y. (2016), "An ensemble EM algorithm for Bayesian variable selection," *arXiv preprint arXiv:1603.04360*, .

Yuan, M., and Lin, Y. (2007), "Model selection and estimation in the Gaussian graphical model," *Biometrika*, 94, 19–35.