Joint Estimation of Gaussian Graphical Models for Multiclass and Multilevel Data

Liang Shan

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
Feng Guo
George R. Terrell

May 2, 2016
Blacksburg, Virginia

KEYWORDS: Bias Correction; Gaussian graphical model; Heterogeneous classes; Joint adaptive graphical lasso; Joint estimation; Multilevel network; Precision matrix; Unbalanced multi-class.

Joint Estimation of Gaussian Graphical Models for Multiclass and Multilevel Data

Liang Shan

(ACADEMIC ABSTRACT)

Gaussian graphical model has been a popular tool to investigate conditional dependency between random variables by estimating sparse precision matrices. The estimated precision matrices could be mapped into networks for visualization. For related but different classes, jointly estimating networks by taking advantage of common structure across classes can help us better estimate conditional dependencies among variables. Furthermore, there may exist multilevel structure among variables; some variables are considered as higher level variables and others are nested in these higher level variables, which are called lower level variables. In this dissertation, we made several contributions to the area of joint estimation of Gaussian graphical models across heterogeneous classes: the first is to propose a joint estimation method for estimating Gaussian graphical models across unbalanced multi-classes, whereas the second considers multilevel variable information during the joint estimation procedure and simultaneously estimates higher level network and lower level network.

For the first project, we consider the problem of jointly estimating Gaussian graphical models across unbalanced multi-class. Most existing methods require equal or similar sample size among classes. However, many real applications do not have similar sample sizes. Hence, in this dissertation, we propose the joint adaptive graphical lasso, a weighted $\boldsymbol{L}_1$ penalized approach, for unbalanced multi-class problems. Our joint adaptive graphical lasso approach combines information across classes so that their common characteristics can be shared during the estimation process. We also introduce regularization into the adaptive term so that the unbalancedness of data is taken into account. Simulation studies show that our approach performs better than existing methods in terms of false positive rate, accuracy, Mathews correlation coefficient, and false discovery rate. We demonstrate the advantage of our approach using liver cancer data set.

For the second one, we propose a method to jointly estimate the multilevel Gaussian graphical models across multiple classes. Currently, methods are still limited to investigate a single level conditional dependency structure when there exists the multilevel structure among variables. Due to the fact that higher level variables may work together to accomplish certain tasks, simultaneously exploring conditional dependency structures among higher level variables and among lower level variables are of our main interest. Given multilevel data from heterogeneous classes, our method assures that common structures in terms of the multilevel conditional dependency are shared during the estimation procedure, yet unique structures for each class are retained as well. Our proposed approach is achieved by first introducing a higher level variable factor within a class, and then common factors across classes. The performance of our approach is evaluated on several simulated networks. We also demonstrate the advantage of our approach using breast cancer patient data.

Joint Estimation of Gaussian Graphical Models for Multiclass and Multilevel Data

Liang Shan

(PUBLIC ABSTRACT)

A network can be represented by nodes and edges between nodes. For instance, a gene network could tell how genes are interacting with each other. In this dissertation, we contribute to the network science literature by providing two approaches to discover network structures. More specifically, our goal is to find common and unique structures of networks across heterogeneous classes.

For the first topic, we consider the scenario where heterogenous classes have unbalanced data. For instance, certain type of disease might be rare, so healthy people group and normal people group may have very unbalanced sample sizes. By our first approach, we keep the majority class from dominating the final result, and we are able to find differences and common structure among classes in terms of network.

For the second topic, we consider the scenario where data are of multilevel structure. For instance, a gene pathway is composed of a series of genes to work together for a particular cellular or physical function. Moreover, pathways are not isolated, they actually interact with each other. Via our second approach, we may simultaneously discover pathway network (how pathways are interacting with one another) and gene networks within pathways (how genes interact with one another within each pathway), which are called the multilevel network. The common structure and unique structures across heterogeneous classes can be discovered in terms of the multilevel network.

Dedication

To my heavenly father, for His everlasting and never-failing love and guidance!

Acknowledgments

My deepest gratitude is to my advisor, Dr. Inyoung Kim. I cannot express how grateful I am to have her as my advisor. She set me a great example of what an excellent teacher should be: she really devoted herself to offering great courses and setting aside enough time to meet her students' needs. During my PhD career, she gave me enough guidance and freedom, and lead me to enter into the wonderland of Statistics. Her support, patience, and trust helped me overcome many hard situations and finish the dissertation. She is more than an advisor, but a life-long mentor.

I would also like to thank my committee members, Dr. Xinwei Deng, Dr. Feng Guo, and Dr. George Terrell for their time, help, and valuable inputs. Thank you to Dr. Birch, for accepting me into this wonderful program and offering me so many great opportunities to work as a teacher, a collaborator, and a researcher. Thank you to Dr. Vance, for introducing LISA to my life and inspiring my passion to collaborate with non-statisticians in an efficient way. Thank you to the faculty and staff from the Department of Statistics for all the support.

Last but not least, I would like to thank my precious family. Thank you to my parents, my grandparents, and my parents-in-law, for their sacrifice and continuous support to me. Thank you to my little ones John and Abigail, for making my PhD life as colorful as a rainbow. Thank you to my beloved husband Zhilei Qiao, for his love, endurance, encouragement and accompany.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 Background

1.1.1 Gaussian graphical model

In mathematics, a graph is composed of nodes and edges between nodes, where edges could be directed, undirected, or bidirected. In recent years, graphical models are becoming popular in investigating networks. For instance, a gene network that is composed of genes and connections among genes can be visualized by a graph, where genes are represented by nodes and connections are represented by edges.

Under the assumption of multivariate Gaussian distribution, a graphical model is called a Gaussian graphical model, where edges are undirected. The main idea to infer a graph from a set of variables of certain samples is to estimate a sparse precision matrix, elements of which indicate conditional dependency between pairs of variables. That is, if the (i, j)th element in precision matrix is 0, variables i and j are conditionally independent, otherwise,

they are dependent given all other variables. For example, in a gene network, genes i and j are unconnected if the (i, j)th element in precision matrix is 0; they are connected if the (i, j)th element is not 0.

One natural way to estimate precision matrix is to obtain the maximum likelihood estimator (MLE). However, MLE can hardly generate exact 0's in the estimated precision matrix, which gives us no clue on conditional dependency among variables. Moreover, under high-dimensional settings where the number of variables is larger than or equal to the number of samples, MLE is ill defined. There have been a number of studies proposed to have a sparse estimate of a precision matrix. The idea of setting elements of precision matrix to zero was proposed by Dempster (1972). He also provided rules and algorithms illustrated by a simple sample data. However, his approach is computationally expensive except for very low-dimensional settings. Meinshausen and Bühlmann (2006) proposed neighborhood selection to estimate sparse precision matrix for high-dimensional settings. They first fit a regression model using the Lasso for each variable by treating all other variables as predictors. Then, if either the regression coefficient of variables i on j or that of variables j on i is nonzero, the (i, j)th element in precision matrix is estimated to be nonzero. Yuan and Lin (2007), Friedman et al. (2008), and Rothman et al. (2008) studied penalized likelihood approaches with $\boldsymbol{L}_1$ penalty and estimated penalized MLE using different algorithms. By doing this, model selection and parameter estimation are simultaneously achieved. Yuan and Lin (2007) used the determinant maximization (MAXDET) algorithm. Friedman et al. (2008) took advantage of the clockwise coordinate descent approach and developed the graphical lasso (Glasso) algorithm which is remarkably fast. Rothman et al. (2008) derived the optimization algorithm utilizing Cholesky decomposition and the local quadratic approximation, and finally produced the sparse estimator that is permutation invariant. Nonetheless, it has been shown that the LASSO penalty produces biases in regression. To correct biases, the

Smoothly Clipped Absolute Deviation (SCAD) penalty and the adaptive LASSO penalty were proposed by Fan and Li (2001) and by Zou (2006), respectively. Fan et al. (2009) employed the two penalties above in precision matrix estimation and solved the bias problem.

1.1.2 Joint estimation of Gaussian graphical models

Sometimes, we may have observations from heterogeneous classes, and their corresponding true precision matrices should have some differences. Therefore, assuming that they all come from the same multivariate normal distribution is inappropriate. On the other hand, classes are related to each other in certain ways, so network structures of different classes may have something in common. For instance, patients with different types of diabetes (Type 1 diabetes, Type 2 diabetes, and gestational diabetes) may have different gene network structures, but parts of the structures may be exactly the same due to the fact that they are all diabetes patients. In this situation, it is inappropriate to estimate the precision matrix by viewing all the observations as one group, since it ignores the distinctions among classes. Separately estimating precision matrices with respect to each class fails to take advantage of the common structure among classes. Therefore, jointly estimating precision matrices across multiple classes will take advantage of using information across classes, so that common structure is estimated more precisely than separate estimation, and unique structures are able to be found as well. Guo et al. (2011) proposed jointly estimate precision matrices for different classes by reparameterizing their off-diagonal elements to be multiples of a common factor across categories and a unique factor for each category. Their method could be solved by iterative weighted Glasso (Friedman et al., 2008). In addition, Danaher et al. (2014) used generalized fused lasso or group lasso as the penalty and employed the alternating directions method of multipliers (ADMM) algorithm to solve the optimization problem.

1.2 Motivation

In reality, it is not rare to have unbalanced data. For instance, certain types of cancer are rarely found, so the numbers of samples for that type of cancer and the normal population are very unbalanced. In these scenarios, the majority class could easily dominate estimation results when precision matrices are estimated jointly. However, neither of Guo et al. (2011) and Danaher et al. (2014) has considered the problem of unbalanced data from heterogeneous classes, and we are motivated to fill that gap in the first project.

On the other hand, in many scientific and engineering applications, we may have multilevel data structure: some variables are considered as higher level variables and others are nested in these higher level variables, which are called lower level variables. For instance, a gene pathway is composed of a series of genes to work together for a particular cellular or physical function. In that scenario, pathways are the higher level variables and genes within a pathway are the lower level variables. Moreover, higher level variables are not independent. For example, pathways are not isolated. Instead, they work together to accomplish certain tasks. Therefore, simultaneously exploring conditional dependency structures among higher level variables and among lower level variables are of interest, so that the multilevel network composed of higher level network and lower level network can be generated. Nevertheless, neither Guo et al. (2011) or Danaher et al. (2014) took into account the multilevel data structure when investigating the conditional dependency: they only concentrate on estimating single level precision matrices jointly across classes. To the best of our knowledge, no research so far has considered estimating multilevel network jointly across classes, and we are motived to fill that gap in the second project.

1.3 Overview

The rest of the dissertation is organized as follows. In Chapter 2, we propose the joint adaptive graphical lasso, a weighted $\boldsymbol{L}_1$ penalized approach, to jointly estimate Gaussian graphical models for unbalanced multi-classes. Our joint adaptive graphical lasso approach combines information across classes so that their common characteristics can be shared during the estimation process. We also introduce regularization into the adaptive term so that the unbalancedness of data is taken into account. Simulation studies show that our approach performs better than existing methods in terms of false positive rate, accuracy, Mathews correlation coefficient, and false discovery rate. We demonstrate the advantage of our approach using liver cancer data set. In Chapter 3, we propose a method to jointly estimate the multilevel Gaussian graphical models across multiple classes. Our proposed approach is achieved by first introducing a higher level variable factor within a class, and then common factors across classes. The performance of our approach is evaluated on several simulated networks. We also demonstrate the advantage of our approach using breast cancer patient data. In Chapter 4, we give a general review on the contributions of this dissertation, as well as discuss directions for future research.

Chapter 2

Joint Estimation of Multiple Gaussian Graphical Models across Unbalanced Classes

2.1 Introduction

In mathematics, a graph is composed of nodes and edges between nodes, where edges can be directed, undirected, or bidirected. In recent years, graphical models have become popular in investigating networks. For instance, a gene network that is composed of genes and connections among genes can be illustrated by a graph where genes are represented by nodes and connections are represented by edges. Under the multivariate Gaussian distribution assumption, a graphical model is called a Gaussian graphical model where edges are undirected. The main idea of inferring a graph from a set of variables of certain samples is to identify an inverse covariance matrix (or precision matrix), elements of which indicate conditional dependency between pairs of variables. Specifically, if the (i, j)th element in precision matrix

is 0, variables i and j are conditionally independent, otherwise, they are dependent given all other variables. To illustrate using a gene network again, if the (i, j)th element in a precision matrix is 0, genes i and j are unconnected. Otherwise, they are connected.

One natural way to estimate precision matrix is to obtain a maximum likelihood estimator (MLE). However, an MLE can hardly generate exact 0's in the estimated precision matrix, which gives us no clues about conditional dependency among variables. Moreover, under high-dimensional settings where number of variables is larger than or equal to number of samples, MLE is ill defined. There have been a number of studies proposed to get a sparse estimate of a precision matrix. Related ideas dates back to Dempster (1972), who suggested the idea of setting elements of a precision matrix to zero and provided rules and algorithms illustrated by a simple sample data. However, their approach is computationally expensive except for very low-dimensional settings. Meinshausen and Bühlmann (2006) proposed neighborhood selection to estimate sparse precision matrices for high-dimensional settings. They firstly fit a regression model using the Lasso for each variable, treating all other variables as predictors. Then, if either the regression coefficient of variables i on j or that of variables j on i is nonzero, the (i, j)th element in the precision matrix is estimated to be nonzero. Yuan and Lin (2007), Friedman et al. (2008) and Rothman et al. (2008) studied penalized likelihood approaches with $\boldsymbol{L}_1$ penalty and estimated penalized MLE using different algorithms. By doing this, model selection and parameter estimation were simultaneously achieved. Yuan and Lin (2007) used the determinant maximization (MAXDET) algorithm. Friedman et al. (2008) took advantage of the clockwise coordinate descent approach and developed the graphical lasso (Glasso) algorithm, which is remarkably fast. Rothman et al. (2008) derived the optimization algorithm using Cholesky decomposition and the local quadratic approximation, and produced the sparse estimator that is permutation invariant. Nonetheless, it has been shown that the LASSO penalty produces

biases in regression. To correct biases, the Smoothly Clipped Absolute Deviation (SCAD) penalty and the adaptive LASSO penalty were proposed by Fan and Li (2001) and Zou (2006), respectively. Fan et al. (2009) employed the two penalties above in precision matrix estimation and solved the bias problem.

However, all of these approaches ignore the fact that observations may come from different classes. Since the true precision matrix may have some differences among classes, assuming that they all come from the same multivariate normal distribution is inappropriate. On the other hand, classes are related to each other in certain ways, so network structures of different classes may have something in common. For instance, patients with different types of diabetes (Type 1 diabetes, Type 2 diabetes, and gestational diabetes) may have different gene network structures, but parts of the structures may be exactly the same due to the fact that they are all diabetes patients. In this situation, it is inappropriate to estimate the precision matrix by viewing all the observations as one group, since it ignores the distinctions among classes. Separately estimating precision matrices with respect to each class fails to take advantage of the common structure among classes. Therefore, jointly estimating precision matrices across multiple classes will take advantage of using information across classes, so that common structure is estimated more precisely than separate estimation, and unique structures are able to be found as well. Guo et al. (2011) proposed jointly estimate precision matrices for different classes by reparameterizing their off-diagonal elements to be multiples of a common factor across categories and a unique factor for each category. Their method could be solved by iterative weighted Glasso (Friedman et al., 2008). In addition, Danaher et al. (2014) used generalized fused lasso or group lasso as the penalty and employed the alternating directions method of multipliers (ADMM) algorithm to solve the optimization problem. Nevertheless, neither of them considered the problem of unbalanced data, which is pretty common in many real applications. For instance, certain types of cancer are rarely

found, so the numbers of samples for that type of cancer and the normal population are very unbalanced. In those scenarios, the majority class could easily dominate estimation results when precision matrices are estimated jointly.

Therefore, the goal of this chapter is to propose joint Gaussian graphical method for unbalanced multiclass, which jointly estimates precision matrices across multiple unbalanced classes with the penalized graphical model approach, so that common structures are estimated more precisely than separate estimations, and unique structures are capable of being discovered.

This chapter is organized as follows. In Section 2.2, we propose our weighted penalized likelihood approach. In Section 2.3, we conduct simulation studies to compare our method with the existing methods. In Section 2.4, we apply our approaches to the liver cancer data set analyzed by Chen et al. (2002) and de Souto et al. (2008). Section 2.5 contains concluding remarks.

2.2 Joint Adaptive Graphical Lasso Approach

In this section, we first explain our model in Section 2.2.1 and then describe our joint adaptive graphical lasso (JAGL) approach in Section 2.2.2.

2.2.1 Unbalanced multiclass Gaussian graphical models

Suppose we have M heterogeneous classes with p variables, where $M \geq 2$. The mth class is expressed as a $n_m \times p$ matrix, which is denoted as X^m where $m = 1, \ldots, M$. Each row of X^m corresponds to an observation, and each column corresponds to a variable. Let $\underline{x}_i^m = (x_{i,1}^m, \ldots, x_{i,p}^m)$ be the ith row of X^m, $i = 1, \ldots, n_m$. With this notation, we write X^m

as the follows:

$$X^m = \begin{bmatrix} x^m_{1,1} & \cdots & x^m_{1,p} \\ \vdots & \ddots & \vdots \\ x^m_{n_m,1} & \cdots & x^m_{n_m,p} \end{bmatrix} = \begin{bmatrix} \underline{x}^m_1 \\ \vdots \\ \underline{x}^m_{n_m} \end{bmatrix} m = 1, \ldots, M.$$

Unbalanced Multiclass Gaussian graphical models assumes that the following two:

(i) Within each class m, $\underline{x}^m_1, \ldots, \underline{x}^m_{n_m} \in \mathbb{R}^p$ are i.i.d MN $[\underline{\mathbf{0}}, (\Omega^m)^{-1}]$, where the precision matrix,

$$\Omega^m = \begin{bmatrix} \omega^m_{1,1} & \cdots & \omega^m_{1,p} \\ \vdots & \ddots & \vdots \\ \omega^m_{p,1} & \cdots & \omega^m_{p,p} \end{bmatrix}$$

is symmetric positive definite. Here $\omega^m_{i,j}$ represents the ith row and jth column element of class m precision matrix Ω^m.

(ii) Observations from different classes are independent from each other.

Let $S^m = (1/n_m)(X^m)^T(X^m)$ denote the empirical covariance matrix of X^m. Based on (i) and (ii), the maximum likelihood approach estimates the precision matrices $\Omega^1, \Omega^2, \ldots, \Omega^M$ by maximizing the log-likelihood (l) of the whole data set, which is

$$\max_{(\Omega^m)_{m=1}^M} l(\Omega^1, \Omega^2, \ldots, \Omega^K) \propto \frac{1}{2} \sum_{m=1}^{M} n_m [\log|\Omega^m| - \text{trace}(S^m \Omega^m)]. \tag{2.1}$$

Solving (2.1) gives the maximum likelihood estimates $(S^1)^{-1}, (S^2)^{-1}, \ldots, (S^M)^{-1}$. However, the usual MLE can hardly generate exact $0'$s in the estimated precision matrix, which gives us no clues about conditional independence among variables. Moreover, under high-dimensional

settings where the number of variables is larger than or equal to the number of samples, MLE is ill-defined.

2.2.2 The joint adaptive graphical Lasso

The joint adaptive graphical lasso (JAGL) is proposed in order to obtain sparse estimates of precision matrices across unbalanced multi-classes jointly. That is, JAGL is to achieve a weighted $\boldsymbol{L}1$ penalized estimator by solving

$$\min_{(\Omega^m)_{m=1}^M} \sum_{m=1}^{M} n_m[\text{trace}(\mathrm{S}^m\Omega^m) - \log|\Omega^m|] + \lambda \sum_{i\neq j} \frac{1}{|(1-\pi_m)\hat{t}_{i,j} + (\pi_m)\hat{s}^m_{i,j}|^r} (\sum_{m=1}^{m} |\omega^m_{i,j}|). \quad (2.2)$$

Here $\lambda > 0$, $r > 0$, $0 \leq \pi_m \leq 1$. $\hat{t}_{i,j}$ is the precision matrix estimated by pooling all the observations into one class, and $\hat{s}^m_{i,j}$ is obtained by estimating precision matrix separately for each class m.

Objective function (2.2) can be decomposed into M individual optimization problems:

$$\begin{aligned} \Omega^m &= \arg\min_{\Omega^m} [\text{trace}(\mathrm{S}^m\Omega^m) - \log|\Omega^m|] + \frac{\lambda}{n_m} \sum_{i\neq j} \frac{1}{|(1-\pi_m)\hat{t}_{i,j} + (\pi_m)\hat{s}^m_{i,j}|^r} |\omega^m_{i,j}| \\ &= \arg\min_{\Omega^m} [\text{trace}(\mathrm{S}^m\Omega^m) - \log|\Omega^m|] + \lambda_m \sum_{i\neq j} \frac{1}{|(1-\pi_m)\hat{t}_{i,j} + (\pi_m)\hat{s}^m_{i,j}|^r} |\omega^m_{i,j}|. \end{aligned} \quad (2.3)$$

Note that the penalty term for element $\omega^m_{i,j}$ is composed of two parts, where the former is λ_m, and the latter is $\frac{1}{|(1-\pi_m)\hat{t}_{i,j}+(\pi_m)\hat{s}^m_{i,j}|^r}$, which indicates the shrinkage introduced by common and unique structures of precision matrices. When $\pi_m = 1$, JAGL reduces to the adaptive graphical lasso proposed by Fan et al. (2009) for each individual class m. In addition, in simulation and real data application, we choose $r = 0.5$. The tuning parameters λ and π_m's are well-defined because π_m is bounded, i.e. $0 \leq \pi_m \leq 1$.

2.2.3 Algorithm

The algorithm for solving the proposed JAGL is consisted of the following two Steps:

Step 1 Initialize the precision matrices as $(S + vI_p)^{-1}, (S^1 + v_1 I_p)^{-1}, (S^2 + v_2 I_p)^{-1}, \ldots,$and $(S^m + v_K I_p)^{-1}$, where I_p is the identity matrix and the constants $v, v_1, v_2, \ldots,$ and v_M are chosen to guarantee positive definite initial values;

Step 2 Update $\hat{\Omega}^m$ by (2.3) using the weighted Glasso (Friedman et al., 2008).

2.2.4 Tuning parameters selection

The tuning parameters λ and π_m's in (2.3) controls the sparsity of the estimator. We select them using the Bayesian Information Criterion (BIC), defined as:

$$\text{BIC}(\lambda, \pi_1, \ldots, \pi_M) \;=\; \sum_{m=1}^{M} \{n_m[\text{trace}(S^m \hat{\Omega}^m_{\lambda,\pi_m}) - \log|\hat{\Omega}^m_{\lambda,\pi_m}|] + df_m \log(n_m)\},$$

where $df_m = \#\{(i,j) : i < j, \hat{\omega}^m_{i,j} \neq 0\}$.

2.2.5 Asymptotic properties

In this section, we provide theoretical justifications of the oracle properties of the proposed JAGL. Some of the assumptions are listed as in Section 2.2.1. Here, p is assumed to be fixed, and we study the asymptotic properties of our penalized estimates with JAGL as the sample size $n \to \infty$.

Proposition 1. *Let us consider our JAGL with the weight specified by* $\tau^m_{i,j} = \frac{1}{|(1-\pi_m)\hat{t}_{i,j} + (\pi_m)\hat{s}^m_{i,j}|^r}$ *for some* $\gamma > 0$ *and* $0 \leq \pi_m \leq 1$. *If* $\hat{T} = (\hat{t}_{ij})_{1 \leq i,j \leq p}$ *and* $\hat{S}^m = (\hat{s}^m_{i,j})_{1 \leq i,j \leq p}$ *are both* a_{n_m}-

consistent estimators of Ω^m*, then the denominator of* $\tau_{i,j}^m$ *is also* a_{n_m}*-consistent estimator of* Ω^m*. That is,* $a_{n_m}\{[(1-\pi_m)\hat{t}_{i,j} + (\pi_m)\hat{s}_{i,j}^m] - \Omega^m\} = O_p(1)$.

Proposition 1 can be easily proved by Slutsky's theorem.

Theorem 1. *When* $\sqrt{n_m}{\lambda_m}' = O_p(1)$*,* ${\lambda_m}'\sqrt{n_m}{a_{n_m}}^\gamma \to \infty$*, and* $\hat{T} = (\hat{t}_{ij})_{1\le i,j\le p}$ *and* $\hat{S}^m = (\hat{s}_{i,j}^m)_{1\le i,j\le p}$ *are both* a_{n_m}*-consistent estimators of* Ω^m *as* $n_m \to \infty$*, the oracle property also holds for the JAGL penalty with weights specified by* $\tau_{i,j}^m = \frac{1}{|(1-\pi_m)\hat{t}_{i,j}+(\pi_m)\hat{s}_{i,j}^m|^r}$ *for some* $\gamma > 0$.

When $\pi_m = 1$, the proposed JAGL reduces to the adaptive graphical lasso proposed by Fan et al. (2009) for each individual class m. Hence, the proof exactly follows of the proposition in Fan et al. (2009). When $0 \le \pi_m < 1$, given that $\hat{T} = (\hat{t}_{ij})_{1\le i,j\le p}$ and $\hat{S}^m = (\hat{s}_{i,j}^m)_{1\le i,j\le p}$ are both a_{n_m}-consistent estimators of Ω^m as $n_m \to \infty$, $\{(1-\pi_m)\hat{t}_{i,j} + (\pi_m)\hat{s}_{i,j}^m\}$ is also a a_{n_m}-consistent estimator of Ω^m, as $n_m \to \infty$ by Proposition 1. Hence the proof can be done by following the proposition in Fan et al. (2009) as well. Detailed proof is omitted here.

2.3 Simulation

We conduct simulation to understand the performance of our JAGL under several cases of stimulated networks. We consider two simulation studies: one is under the scenario where $M = 2$ classes for two types of network structures (chain network and scale-free network) described in Section 2.3.1 and the other is under the scenario where $M = 3$ classes for two types of network structures (chain network and scale-free network) described in Section 2.3.2, respectively. The performance of our method is compared with others among different settings of sample sizes, dimensionality, and degree of heterogeneity. We compare our JAGL with three other methods in terms of several evaluation metrics defined in Section 2.3.1.2 and Section 2.3.2.2. The methods that we made comparisons to are the followings:

- JAGL: Our joint adaptive graphical lasso.
- GLS: Method that treats the classes separately and estimate precision matrix individually for each class using graphical lasso.
- GLT: Method that combines all the observations together and estimate a single precision matrix using graphical lasso.
- JGL: Guo et al. (2011)'s joint graphical lasso.

2.3.1 Simulation study under $M = 2$

2.3.1.1 Simulation settings

We consider two classes $M = 2$. They have $n_1 = 100$ and $n_2 = 50$, respectively, to have unbalanced sample sizes. We evaluate the effect of dimension p on the performance of our JAGL by varying p between 30, 50, 80 and 100. We further conduct simulations by changing $n_1 = 50$, $n_2 = 25$, and varying p between 15, 25, 40 and 50.

We generate $p \times 1$ vector $\mathbf{x}_i^m$ from multivariate normal distribution, MN $(\underline{0}, (\Omega^m)^{-1})$, where $m = 1, 2$ and $i = 1, \ldots, n_m$. The precision matrix Ω^m is generated using two cases of simulated networks. One is a chain network, and the other is a scale-free network. The chain network is corresponding to a tridiagonal precision matrix. It was used in Fan et al. (2009). Scale-free networks were thought to be prevalent in the World Wide Web, social networks, businesses, and biology networks. In scale-free networks, there are several hub nodes with many links though most nodes only have a few connections. The distribution of node linkages follows a power law degree distribution. In other words, the probability that

a certain node was connected to m other nodes is proportional to $1/m^{\gamma}$, where $2 < \gamma < 3$ (Barabási and Bonabeau, 2003). The precision matrix Ω^m is generated in the following ways:

- Chain network: The covariance matrix for chain network is $\Sigma^m = (\sigma^m_{i,j})$ where the (i,j)th element $\sigma^m_{i,j} = \exp(-|s_i - s_j|)$, $s_1 < s_2 < \cdots < s_p$, and $s_i - s_{i-1} \overset{i.i.d.}{\sim} \text{Unif}(0.5, 1)$, $i = 2, \ldots, p$. We then have precision matrix $\Omega^m = \Sigma^{m(-1)}$. In this way, the two precision matrices can share the same pattern of zeros and nonzeros, but the values of the nonzero elements may be different. In addition, we add heterogeneity between the two classes as well. For $m = 1$, we generate the precision structure as we described before. However, for $m = 2$, we randomly select $(1 - \rho) \times p$ tridiagonal element of the generated structure to be the same and then reset the rest of the tridiagonal element blocks to be the identity matrix. We set $\rho = (0, 1/4, 1/2)$. As the value ρ increases, the heterogeneity between the two classes gradually increases. This simulation procedure ensures that both structures have chain networks. And also, by taking the inverse of precision matrices, we can obtain covariance matrices and generate observations out of MN $(\underline{0}, (\Omega^m)^{-1})$ for $m = 1, 2$.

- Scale-free network: The scale-free networks are composed of five equally sized unconnected subnetworks, each with a power law degree distribution. For $m = 1, 2$, each subnetwork is generated by using the Barabási-Albert algorithm (Barabási and Albert, 1999). In this procedure, we make sure the generated subnetworks for $m = 1$ and $m = 2$ are the same in terms of network structure, but differ in the values of the nonzero elements. We then convert the scale-free networks to corresponding precision matrices and ensure their positive definiteness by enforcing the diagonal dominant. Next, the precision matrix Ω^m is constructed by diagonally joining the five sub-precision matrices together. Likewise, we add heterogeneity between the two classes as follows: for $m = 1$, we keep the structure as generated above, and for $m = 2$, we start from the generated

structure above and replace $\rho \times 5$ sub-diagonal blocks with identity matrices. As with the chain network, we use the value of ρ as 0, 1/4, 1/2 to guarantee the heterogeneity between the two classes gradually increases. Moreover, this simulation procedure ensures that both networks are composed of scale-free networks. By taking the inverse of precision matrices, we can obtain covariance matrices and generate observations out of MN $(\underline{0}, (\Omega^m)^{-1})$ for $m = 1, 2$.

2.3.1.2 Evaluation metrics

We generate 100 sets of observations from the simulated precision matrices (or network structures) described in Section 2.3.1.1. The methods comparisons mentioned in the beginning of Section 2.3 are evaluated in terms of two sets of measurements: evaluating information losses and providing accuracy of estimation of the precision matrix structure. We further explain them in this section.

The first set of evaluation metrics are with respect to information loss, including entropy loss (EL) and Frobenius loss (FL) which are defined as

$$\begin{aligned} EL_m &= \text{trace}[(\Omega^m)^{-1}\hat{\Omega}^m] - \log[|(\Omega^m)^{-1}\hat{\Omega}^m|] - p; \\ FL_m &= \frac{\|\Omega^m - \hat{\Omega}^m\|_F^2}{\|\Omega^m\|_F^2}; \\ EL &= \frac{1}{M}\sum_{m=1}^{M} EL_m; \\ FL &= \frac{1}{M}\sum_{m=1}^{M} FL_m. \end{aligned} \tag{2.4}$$

The second set of evaluation metrics are with respect to how precision matrix structures (zeros and non-zeros) are estimated, including false positive rate (FPR), accuracy (ACC),

Matthews Correlation Coefficient (MCC), and false discovery rate (FDR). They are calculated in the following ways. First we define notations:

$$\begin{aligned} FP_m &= \sum_{1\le i<j\le p} I(\omega_{i,j}^m = 0, \hat{\omega}_{i,j}^m \neq 0); \\ TP_m &= \sum_{1\le i<j\le p} I(\omega_{i,j}^m \neq 0, \hat{\omega}_{i,j}^m \neq 0); \\ FN_m &= \sum_{1\le i<j\le p} I(\omega_{i,j}^m \neq 0, \hat{\omega}_{i,j}^m = 0); \\ TN_m &= \sum_{1\le i<j\le p} I(\omega_{i,j}^m = 0, \hat{\omega}_{i,j}^m = 0). \end{aligned} \tag{2.5}$$

Using the above notations, we obtain the following measurements:

$$\begin{aligned} FPR_m &= \frac{FP_m}{FP_m + TN_m}; \\ ACC_m &= \frac{TP_m + TN_m}{p(p-1)/2}; \\ MCC_m &= \frac{TP_m \times TN_m - FP_m \times FN_m}{\sqrt{(TP_m + FP_m)(TP_m + FN_m)(TN_m + FP_m)(TN_m + FN_m)}}; \\ FDR_m &= \frac{FP_m}{FP_m + TP_m}. \end{aligned} \tag{2.6}$$

We then calculate the average to compare the overall evaluation across classes in terms of different measurements.

$$\begin{aligned} FPR &= \frac{1}{M}\sum_{m=1}^{M} FPR_m; \\ ACC &= \frac{1}{M}\sum_{m=1}^{M} ACC_m; \\ MCC &= \frac{1}{M}\sum_{m=1}^{M} MCC_m; \\ FDR &= \frac{1}{M}\sum_{m=1}^{M} FDR_m. \end{aligned} \tag{2.7}$$

2.3.1.3 Simulation results

We summarize the simulation results by looking at the evaluation metrics' means over the 100 replicates. We observe that overall simulation results for chain networks under $n_m = (100, 50)$ and $n_m = (50, 25)$ are pretty similar. For scale-free networks, the results are also similar to each other. Hence, in the dissertation, we show the results for chain networks under $n_m = (100, 50)$ and the results for scale-free networks under $n_m = (50, 25)$ for illustration purposes. For chain networks with $n_m = (100, 50)$, the best performed evaluation metrics (FPR, ACC, MCC, FDR) are displayed in Figures 2.1-2.2. For scale-free network under $n_m = (50, 25)$, the best performed evaluation metrics (FPR, ACC, MCC, FDR) are summarized in Figures 2.3-2.4. We compare our proposed JAGL with others from three perspectives: information loss, precision matrix structure estimation, and computing efficiency.

First, in terms of information losses, including EL and FL as defined in equation (2.4), JAGL is always better or comparable to JGL regardless of network structure, sample size (n_m), dimensionality (p), and degree of heterogeneity (ρ). When $p \leq min(n_1, n_2)$, GLS performs better than or comparable to our JAGL approach. GLT is the best in terms of EL and FL. Ours is comparable to GLT.

Second, in terms of precision matrix structure estimation, including FPR, ACC, MCC, and FDR as defined in equation (2.7), we have the following observations:

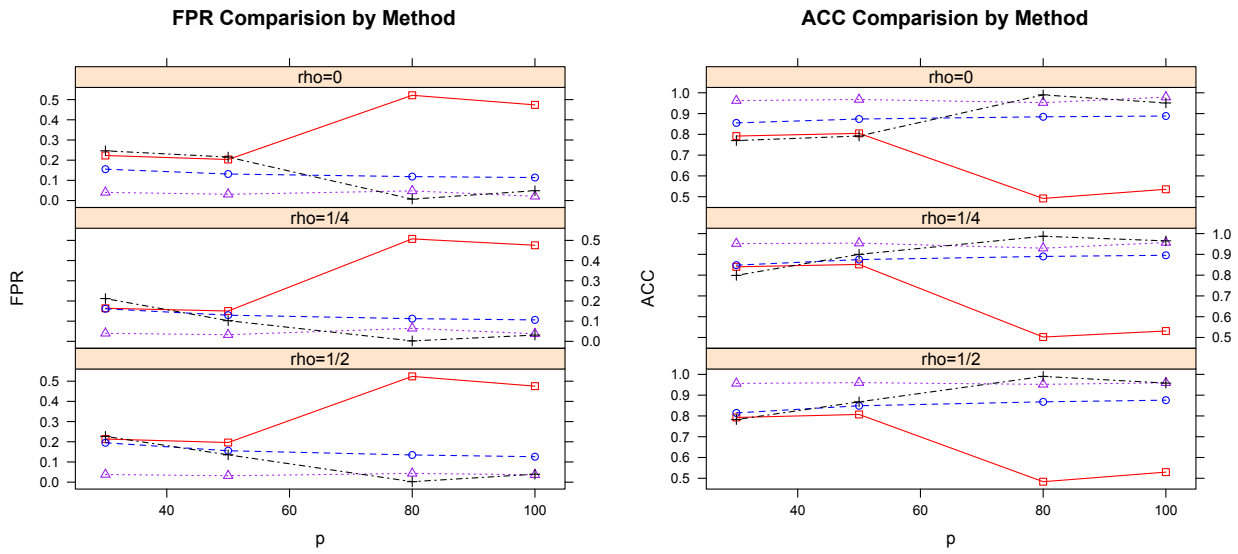

Figure 2.1: FPR and ACC comparison among the four methods for chain network in terms of different p and ρ under $n_m = (100, 50)$; GLS=method that treats the classes separately and estimate precision matrix individually for each class; GLT=method that combines all the observations together and estimate a single precision matrix; JAGL=our joint adaptive graphical lasso; JGL=Guo et al. (2011)'s joint method; GLS is represented by rectangular points and solid lines with red color; GLT is represented by circular points and dashed lines with blue colors; JAGL is represented by triangular points and dotted lines with purple color; JGL is represented by plus points and dash dot lines with black color.

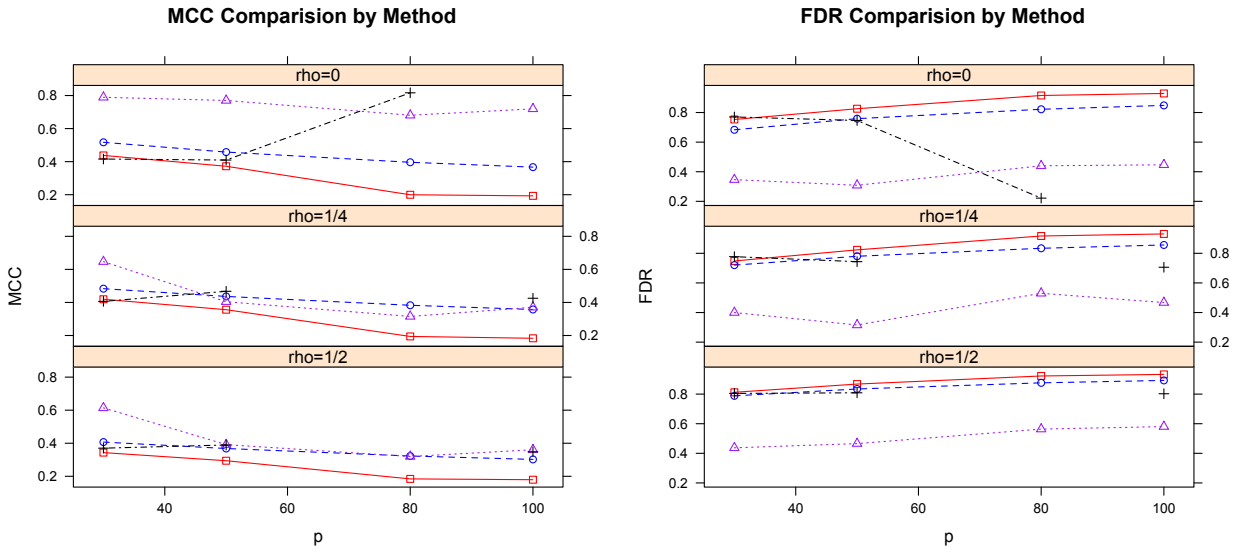

Figure 2.2: MCC and FDR comparison among the four methods for chain network in terms of different p and ρ under $n_m = (100, 50)$; GLS=method that treats the classes separately and estimate precision matrix individually for each class; GLT=method that combines all the observations together and estimate a single precision matrix; JAGL=our joint adaptive graphical lasso; JGL=Guo et al. (2011)'s joint method; GLS is represented by rectangular points and solid lines with red color; GLT is represented by circular points and dashed lines with blue colors; JAGL is represented by triangular points and dotted lines with purple color; JGL is represented by plus points and dash dot lines with black color.

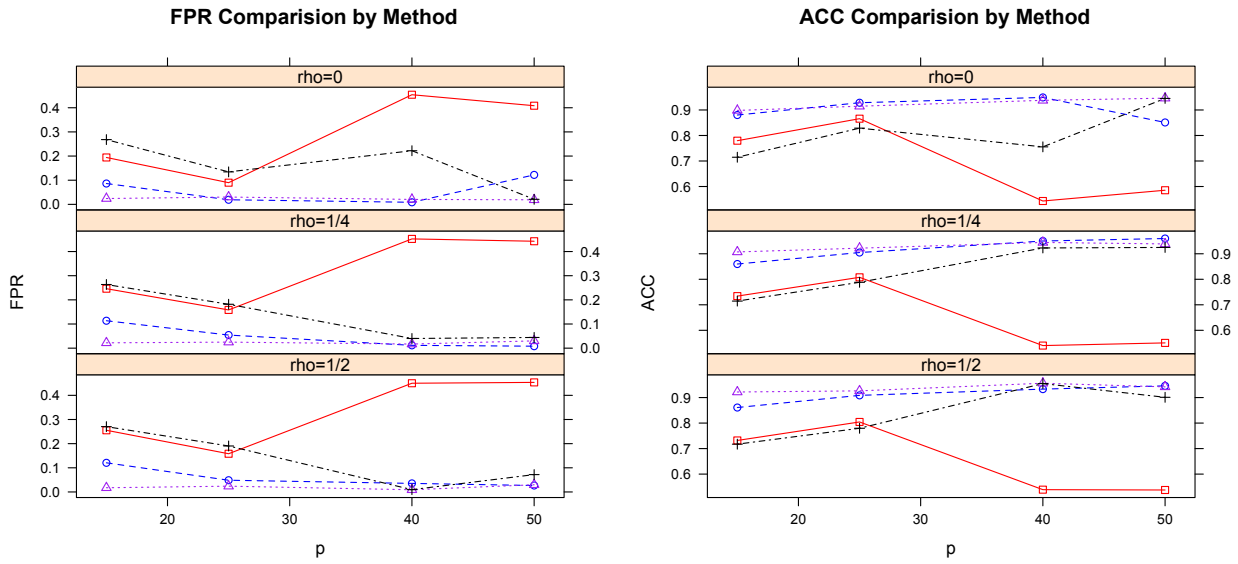

Figure 2.3: FPR and ACC comparison among the four methods for scale-free network in terms of different p and ρ under $n_m = (50, 25)$; GLS=method that treats the classes separately and estimate precision matrix individually for each class; GLT=method that combines all the observations together and estimate a single precision matrix; JAGL=our joint adaptive graphical lasso; JGL=Guo et al. (2011)'s joint method; GLS is represented by rectangular points and solid lines with red color; GLT is represented by circular points and dashed lines with blue colors; JAGL is represented by triangular points and dotted lines with purple color; JGL is represented by plus points and dash dot lines with black color.

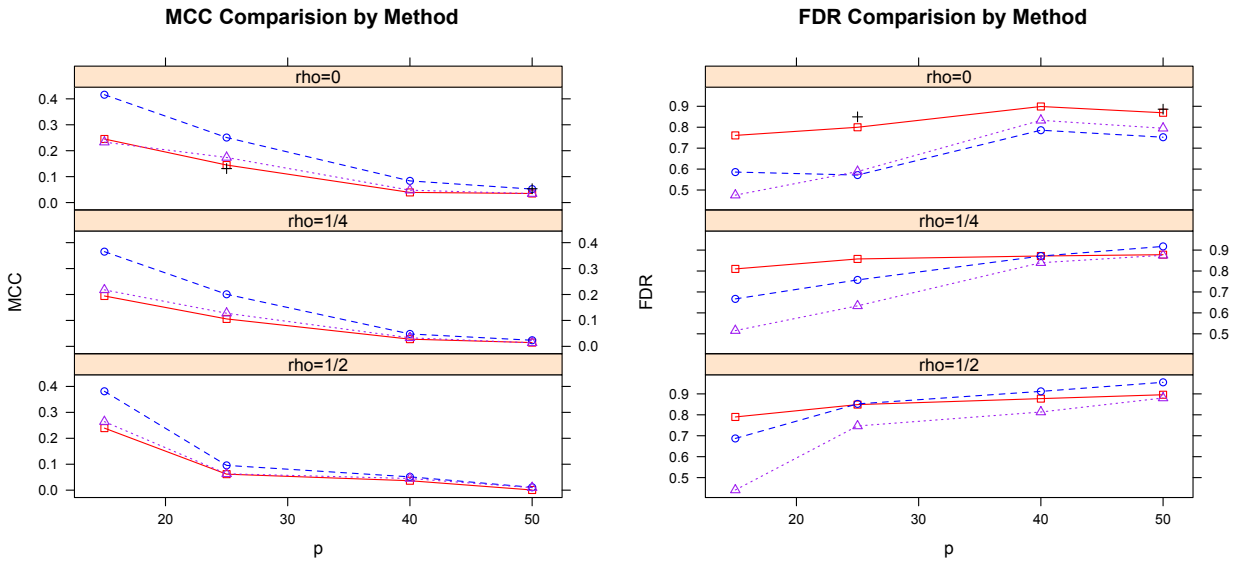

Figure 2.4: MCC and FDR comparison among the four methods for scale-free network in terms of different p and ρ under $n_m = (50, 25)$; GLS=method that treats the classes separately and estimate precision matrix individually for each class; GLT=method that combines all the observations together and estimate a single precision matrix; JAGL=our joint adaptive graphical lasso; JGL=Guo et al. (2011)'s joint method; GLS is represented by rectangular points and solid lines with red color; GLT is represented by circular points and dashed lines with blue colors; JAGL is represented by triangular points and dotted lines with purple color; JGL is represented by plus points and dash dot lines with black color.

We see similar trends of FPR and ACC, that is, JAGL is almost always the lowest/highest in terms of FPR/ACC regardless of network structure, sample size (n_m), dimensionality (p), and degree of heterogeneity (ρ). JGL starts to perform similarly as JAGL when p increases and ρ increases. Generally, GLT performs next to JAGL and GLS performs the worst.

Regarding MCC and FDR, regardless of network structure, sample size (n_m) and dimensionality (p), JAGL almost always provides the highest/lowest MCC/FDR or is comparable to the best method, especially as ρ increases. An interesting note is that for MCC and FDR, JGL easily produces NA values (both FP_k and TP_k in equation (2.6) are equal to 0) in many simulation scenarios since it easily shrinks every off-diagonal element to 0.

Third, in terms of computing efficiency, JAGL is always faster than JGL and GLS, regardless of network structure, sample size (n_m), dimensionality (p), and degree of heterogeneity (ρ). In addition, JAGL is almost as efficient as GLT in most simulated scenarios. The programs were coded in R, and all timings were carried out on a intel Xeon 3.00GHz processor.

In summary, from the three perspectives, our JAGL is the best or close to the best in terms of FPR, ACC, MCC, and FDR, regardless of network type, sample size, dimensionality, and ρ. What is more, JAGL outperforms JGL and GLS in terms of information loss in most cases and in terms of computing efficiency, only performs slightly worse than GLT.

2.3.2 Simulation study under $M = 3$

2.3.2.1 Simulation settings

We consider three classes ($M = 3$), where sample sizes are $n_1 = 500$, $n_2 = 250$, and $n_3 = 100$ respectively. The dimensionality p is set to be 50. Again, we generate $p \times 1$ vector $\mathbf{x}_i^m$ from multivariate normal distribution, MN $(\underline{0}, (\Omega^m)^{-1})$, where $m = 1, 2, 3$ and $i = 1, \ldots, n_m$. The

precision matrices Ω^m's common structure are corresponding to chain networks. Here is how we generate Ω^m's: First, the common structure of precision matrices Ω^m's are generated the same way as described in Section 2.3.1.1. Then, heterogeneity among the classes are introduced by introducing ρ ($\rho = 0, 1/4, 3/4, 5/4$): For each $\Omega^{(m)}$ ($m = 1, 2, 3$), a pair of symmetric zero elements is randomly picked, and it was replaced with a value uniformly generated from the $[-1, -0.5] \bigcup [0.5, 1]$ interval. This procedure is repeated ρT times, where T is the number of common links on the higher level network ($T = p - 1$), and ρ is the ratio of the number of individual links to the number of common links. As ρ increases, the number of unique links increases compared with that of the common links. Aside from considering $M = 3$ unbalanced classes, another major difference between the two simulations is that, the heterogeneity is introduced by removing common links in simulation study under $M = 2$, whereas the heterogeneity is introduced by adding unique links to the common links in simulation study under $M = 3$.

2.3.2.2 Evaluation metrics

In addition to the evaluation metrics mentioned in Section 2.3.1.2, we would like to evaluate how common structures (including common zeros and common non-zeros) are estimated as compared to other methods. The common zero prediction rate (CZPR), the common non-zero prediction rate (CNPR), and the common structure prediction rate (CPR) are defined as:

$$\begin{aligned} CZPR &= \frac{\sum_{1\le i<j\le p} I(\sum_{m=1}^{M} \omega_{i,j}^m = 0, \sum_{m=1}^{M} \hat{\omega}_{i,j}^m = 0)}{\sum_{1\le i<j\le p} I(\sum_{m=1}^{M} \omega_{i,j}^m = 0)}; \\ CNPR &= \frac{\sum_{1\le i<j\le p} I(\bigcap_{m=1}^{M} \omega_{i,j}^m \neq 0, \bigcap_{m=1}^{M} \hat{\omega}_{i,j}^m \neq 0)}{\sum_{1\le i<j\le p} I(\bigcap_{m=1}^{M} \omega_{i,j}^m \neq 0)}; \\ CPR &= \frac{1}{2}(CZPR + CNPR); \end{aligned} \tag{2.8}$$

2.3.2.3 Simulation results

First, we look at the previous best performed evaluation metrics (FPR, ACC, MCC, FDR), which are summarized and displayed in Tables 2.1 and Figures 2.5-2.6. In addition, the information loss, the sparsity level, and the common structure estimation of the estimated precision matrices are displayed in Tables 2.2 and Figure 2.7.

Regarding FPR, ACC, MCC, and FDR, JAGL is the best when $\rho = 0$. When $\rho > 0$, JAGL is the second best or comparable to the best. Regarding information loss, those methods are comparable to one another. In terms of the sparsity of the estimated network, JAGL is comparable to GLS and it generates a sparser estimate of network than JGL. Finally, in terms of the common structure estimation, JAGL behaves the best.

Table 2.1: Simulation results for chain network in terms of different ρ under $n_m = (500, 250, 100)$ and $p = 50$; FPR=False Positive Rate, ACC=Accuracy, MCC=Matthews Correlation Coefficient, and FDR=False Discovery Rate; GLS=Method that treats the classes separately and estimate precision matrix individually for each class, JAGL=our joint adaptive graphical lasso, JGL=Guo et al. (2011)'s joint method.

ρ	method		FPR	ACC	FDR	MCC
0	GLS	Mean	0.198	0.810	0.809	0.389
	JAGL	Mean	0.045	0.957	0.443	0.718
	JGL	Mean	0.187	0.821	0.794	0.406
0.25	GLS	Mean	0.015	0.966	0.213	0.580
	JAGL	Mean	0.015	0.975	0.205	0.749
	JGL	Mean	0.096	0.901	0.626	0.510
0.75	GLS	Mean	0.007	0.949	0.110	0.457
	JAGL	Mean	0.024	0.948	0.236	0.592
	JGL	Mean	0.102	0.882	0.615	0.436
1.25	GLS	Mean	0.005	0.926	0.089	0.354
	JAGL	Mean	0.017	0.930	0.192	0.481
	JGL	Mean	0.113	0.858	0.638	0.367

Table 2.2: Simulation results for chain network in terms of different ρ under $n_m = (500, 250, 100)$ and $p = 50$; EL=Entroy Loss, FL=Frobenius Loss, Number of Zeros=average number of estimated 0's in the precision matrices, CZPR=Common Zero Prediction Rate, CNPR=Common Non-zero Prediction Rate, CPR=Common Structure Prediction Rate; GLS=Method that treats the classes separately and estimate precision matrix individually for each class, JAGL=our joint adaptive graphical lasso, JGL=Guo et al. (2011)'s joint method.

ρ	method		EL	FL	Number of Zeros	CZPR	CNPR	CPR
0	GLS	Mean	49.08	0.08	942.65	0.52	1.00	0.76
	JAGL	Mean	47.86	0.03	1123.00	0.88	1.00	0.94
	JGL	Mean	48.07	0.02	956.24	0.56	1.00	0.78
0.25	GLS	Mean	49.15	0.08	1169.51	0.95	0.04	0.50
	JAGL	Mean	48.33	0.05	1160.95	0.96	0.58	0.77
	JGL	Mean	48.19	0.04	1061.24	0.76	0.71	0.73
0.75	GLS	Mean	49.05	0.08	1184.91	0.98	0.00	0.49
	JAGL	Mean	48.34	0.05	1149.96	0.93	0.37	0.65
	JGL	Mean	48.28	0.05	1050.90	0.74	0.48	0.61
1.25	GLS	Mean	48.83	0.07	1195.89	0.99	0.00	0.49
	JAGL	Mean	48.38	0.05	1163.60	0.95	0.18	0.57
	JGL	Mean	48.28	0.05	1036.58	0.72	0.38	0.55

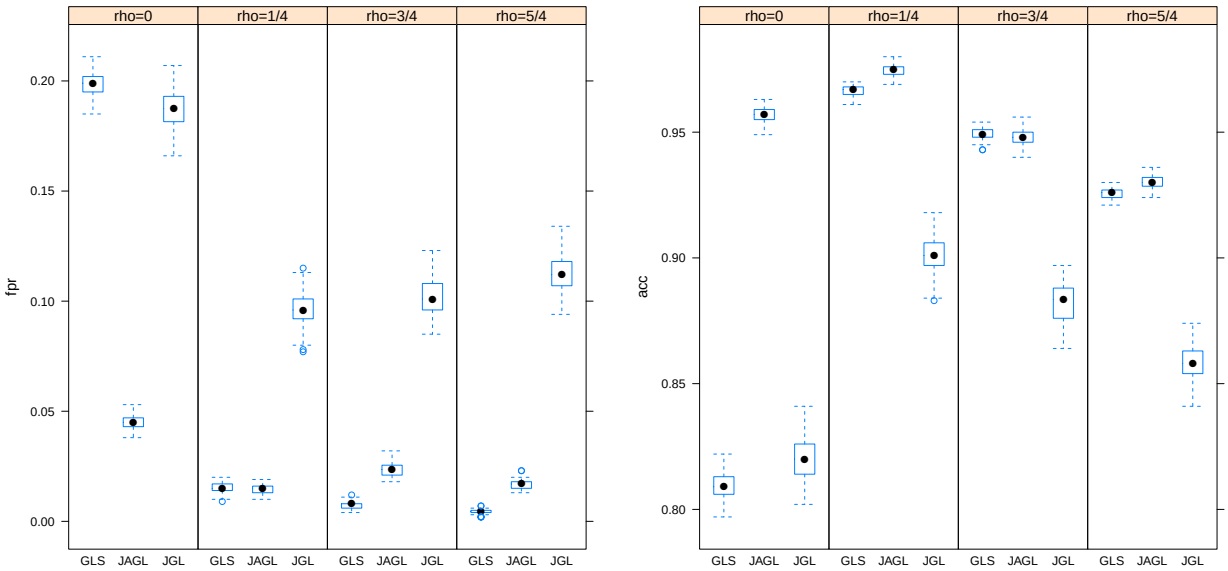

Figure 2.5: FPR and ACC comparison among the three methods for chain network in terms of different ρ under $n_m = (500, 250, 100)$ and $p = 50$; GLS=method that treats the classes separately and estimate precision matrix individually for each class; JAGL=our joint adaptive graphical lasso; JGL=Guo et al. (2011)'s joint method.

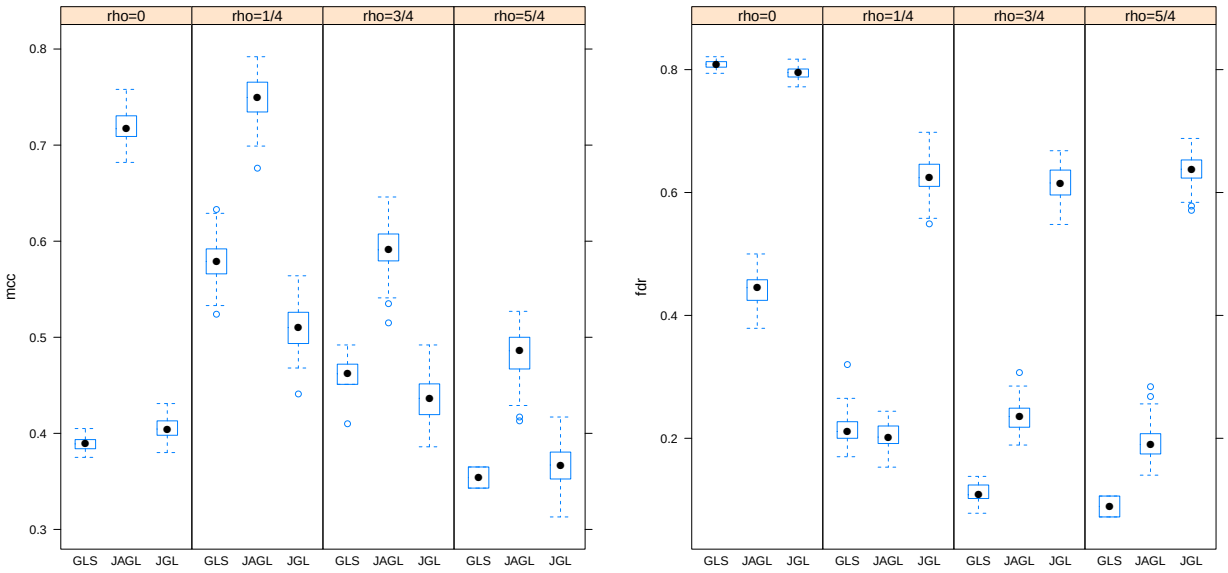

Figure 2.6: MCC and FDR comparison among the three methods for chain network in terms of different ρ under $n_m = (500, 250, 100)$ and $p = 50$; GLS=method that treats the classes separately and estimate precision matrix individually for each class; JAGL=our joint adaptive graphical lasso; JGL=Guo et al. (2011)'s joint method.

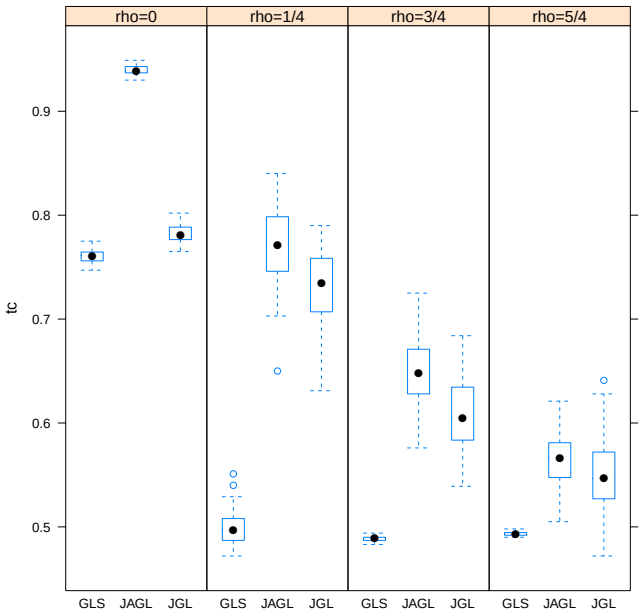

Figure 2.7: CPR comparison among the three methods for chain network in terms of different ρ under $n_m = (500, 250, 100)$ and $p = 50$; GLS=method that treats the classes separately and estimate precision matrix individually for each class; JAGL=our joint adaptive graphical lasso; JGL=Guo et al. (2011)'s joint method.

2.4 Application

In this section, we apply the Joint Adaptive Graphical Lasso (JAGL) to the liver cancer data (Chen et al., 2002; de Souto et al., 2008) described in Section 2.4.1. We then evaluate its performance and make some conclusions in Section 2.4.2.

2.4.1 The liver cancer data

Hepatocellular carcinoma (HCC) is one of the most common causes of death worldwide, especially in East Asia and sub-Saharan Africa. Moreover, the incidence of HCC is growing in recent years in North America and most of Europe (Venook et al., 2010). Chen et al. (2002) conducted a research to take a closer look at the differences in gene expression patterns between HCC tissues and those seen in normal livers. Using hierarchical clustering, $3,180$ genes with the greatest variation in their expression between HCC samples and normal samples were grouped into eight clusters, where each cluster was composed of a set of genes with similar functionality. Later on, de Souto et al. (2008) used 35 cancer gene expression data sets to compare seven different clustering methods and four proximity measures. Chen et al. (2002) is one of the data sets that was selected. In their comparative study, Chen et al. was further reduced from $22,699$ genes to 85 genes by the filter procedure, thereby restricting it to 104 HCC samples and 75 normal ones.

We started from the data sets that were used in de Souto et al. (2008) that contains 85 genes among 104 HCC samples and 75 normal samples. We first used permutation test where the class labels of the samples were permuted $500,000$ times. For each permutation, two-sample Welch t statistics were computed for each gene. Any gene for which the p-value was less than 0.05 was considered to be potentially and differently expressed among groups. By doing this, the number of gene were further reduced to 53. We then standardized them to have

mean 0 and standard deviation 1 within each class.

2.4.2 Application of JAGL to the liver cancer data

For classifying liver tissue samples, we randomly divide the data into training sets and testing sets of sizes 160 and 19, separately, and repeat the process 100 times. To assure that the class unbalance is are similar across the training and testing sets, we used a stratified sampling. Each time we randomly selected 93 subjects from HCC samples and 67 subjects from normal samples so that the training set consists of those 160 subjects. The remaining subjects were used as the testing set.

We then estimated the precision matrix using our JAGL. The quadratic discriminant analysis (QDA) is used for classification. The QDA assumes the normalized gene expression data in class-m follows MN $(\underline{\mu}^m, (\Omega^m)^{-1})$, where $m = 1, 2$. The quadratic discriminant scores for observation $\underline{x}$ are defined as:

$$\delta_m(\underline{x}) = -\frac{1}{2}\log|(\hat{\Omega}^m)^{-1}| - \frac{1}{2}(\underline{x} - \hat{\underline{\mu}}^m)'\hat{\Omega}^m(\underline{x} - \hat{\underline{\mu}}^m) + \log\hat{\pi}_k,$$

where $\hat{\pi}_m = n_m/n$, $\hat{\underline{\mu}}^m = (\sum_{i\in class-m}\underline{x}_i)/n_m$, $m = 1, 2$ and $i = 1, \ldots, n_m$. If $\delta_1(\underline{x}) > \delta_2(\underline{x})$, we say observation $\underline{x}$ belongs to class 1. Vice versa.

To evaluate the classification performance of JAGL, we used accuracy (ACC), defined as the sum of true positives and true negatives over the number of total observations in the testing set. Over 100 repetitions, the ACC for JAGL is 0.6, which shows that JAGL performs well in classification.

For exploring networks among genes, we used all the observations (the 104 HCC samples and 75 normal ones) across the 53 normalized genes. JAGL estimated 88 shared links between

HCC and normal tissue networks, while 213 links only existed in the HCC network and 189 links only existed in the normal tissue network. The results are displayed in Figures 2.8-2.9. From the estimated networks, we observed several interesting findings.

First, KLK10 (Gene5655) was linked to POMT1 (Gene10585), Iap3rb9 (Gene15643), In(1)24Rk-d (Gene16223), and Tpi-rs7 (Gene21999) in HCC samples, but it was connected to a totally different set of genes (including Gene5763, Gene11404, Gene18395, Gene18565, Gene18961, Gene20682, and Gene23605) in normal liver tissue samples. This may infer that gene KLK10 acts differently from normal when HCC is developed. The result could be potentially supported by Lu et al. (2009), who concluded that KLK10 were often found to be hypermethylated in HCC and may be used as potential markers for clinical application.

Second, AMPD1 (Gene270) was connected with LTβR (Gene 4055) and NR0B2 (Gene23957) in HCC samples. However, those connections break in normal liver tissue samples. AMP-deaminase, which consists of independent genes (AMPD1, AMPD2, and AMPD3), is the enzyme catalyzes AMP. The reaction catalyzed by AMP-deaminase constitutes rate-limiting steps of adenine nucleotide catabolism and plays an important roles in cellular energy metabolism (Kaminsky and Kosenko, 2010). It was indicated that the expression levels of AMPD genes in HCC samples are significantly higher than that of normal liver samples, resulting in significant higher specific activity of AMP-deaminase in HCC tumors (Szydowska and Roszkowska, 2008). In addition, Haybaeck et al. (2009) reported that LT signaling was critically involved in HCC development, and blocking LTβR signaling might become a helpful approach in treating HBV- or HCV-induced chronic hepatitis. However, the association between AMPD1 and LTβR is still under exploration. It was also found that the NR0B2 gene expression level was decreased in the development of HCC, regardless of the species and underlying disease present in patients (He et al., 2008). However, the association between AMPD1 and LTβR and that between AMPD1 and NR0B2 are still under exploration.

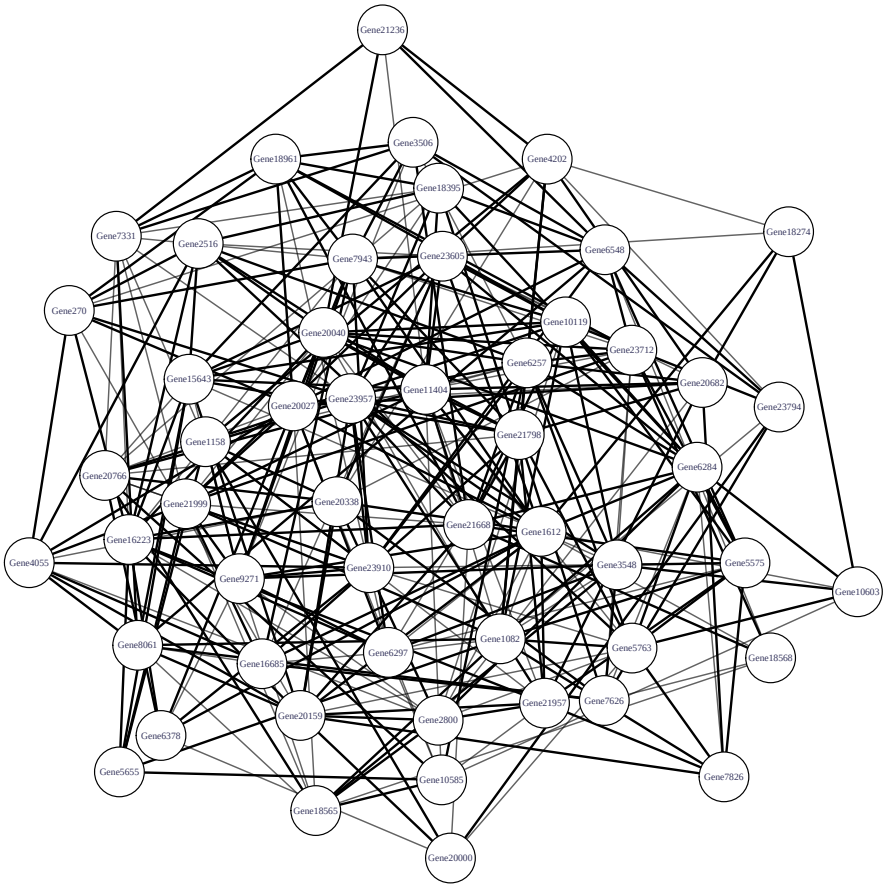

Figure 2.8: Gene network for HCC samples. The thin light lines are the gene connections that are present in both classes, while the thick dark lines are the connections that only belong to HCC samples.

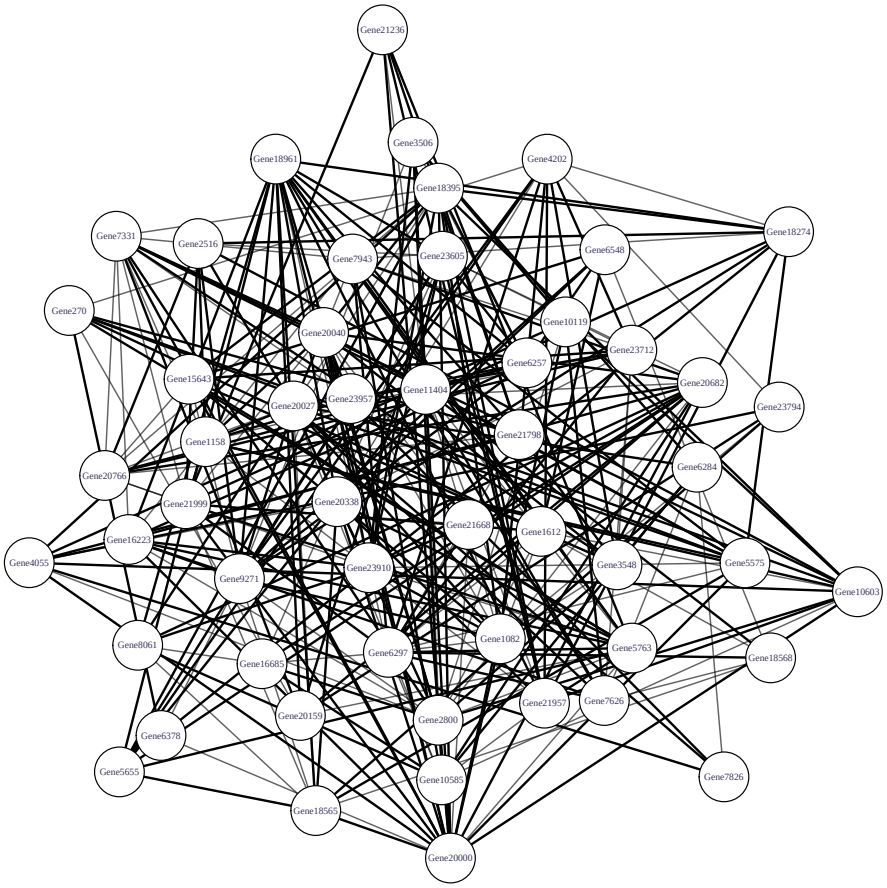

Figure 2.9: Gene network for normal liver tissue samples. The thin light lines are the gene connections that are present in both classes, while the thick dark lines are the connections that only belong to normal liver tissue samples.

2.5 Discussion

In this chapter, we propose the Joint Adaptive Graphical Lasso, a weighted $\boldsymbol{L}_1$ penalized approach, for unbalanced multiclass problems. Our Joint Adaptive Graphical Lasso approach combines information across classes so that their common characteristics can be shared during the estimation process. We also introduce regularization into the adaptive term so that not only the tuning parameters for every class are different, but also the tuning parameters for each element within a class. By doing it this way, we are able to prevent the majority class from dominating the estimated precision matrix result. Our approach is more flexible than the approach of Guo et al. (2011) because their tuning parameters for every class are exactly the same. Simulation studies show that our approach performs better than existing methods or close to the best method in terms of false positive rate, accuracy, Mathews Correlation Coefficient, and false discovery rate. We demonstrate the advantages of our approach using real data.

Our current study only provided the asymptotic properties where $n \to \infty$ and p is fixed. Hence, we need to further develop the asymptotic properties of our approach when $n/log(p) \to \infty$ in a future research. Last but not least, although some of networks are identified to distinguish HCC using our approach, they need to be further validated biologically.

Acknowledgements

This study was partially supported by grants from the National Science Foundation (CNS-096480 and CNS-1115839).

Chapter 3

Joint Estimation of the Multilevel Gaussian Graphical Models across Multiple Classes

3.1 Introduction

In mathematics, a graph is composed of nodes and edges between nodes, where edges could be directed, undirected, or bidirected. In recent years, graphical models are becoming popular in investigating networks. For instance, a gene network that is composed of genes and connections among genes can be visualized by a graph, where genes are represented by nodes and connections are represented by edges. Under the assumption of multivariate Gaussian distribution, a graphical model is called a Gaussian graphical model, where edges are undirected. The main idea to infer a graph from a set of variables of certain samples is to estimate a sparse precision matrix, elements of which indicate conditional dependency between pairs of variables. That is, if the (i, j)th element in precision matrix is 0, variables i

and j are conditionally independent, otherwise, they are dependent given all other variables. For example, in a gene network, genes i and j are unconnected if the (i, j)th element in precision matrix is 0; they are connected if the (i, j)th element is not 0.

One natural way to estimate precision matrix is to obtain the maximum likelihood estimator (MLE). However, MLE can hardly generate exact 0's in the estimated precision matrix, which gives us no clue on conditional dependency among variables. Moreover, under high-dimensional settings where the number of variables is larger than or equal to the number of samples, MLE is ill defined. There have been a number of studies proposed to have a sparse estimate of a precision matrix. The idea of setting elements of precision matrix to zero was proposed by Dempster (1972). He also provided rules and algorithms illustrated by a simple sample data. However, his approach is computationally expensive except for very low-dimensional settings. Meinshausen and Bühlmann (2006) proposed neighborhood selection to estimate sparse precision matrix for high-dimensional settings. They first fit a regression model using the Lasso for each variable by treating all other variables as predictors. Then, if either the regression coefficient of variables i on j or that of variables j on i is nonzero, the (i, j)th element in precision matrix is estimated to be nonzero. Yuan and Lin (2007), Friedman et al. (2008), and Rothman et al. (2008) studied penalized likelihood approaches with $\boldsymbol{L}_1$ penalty and estimated penalized MLE using different algorithms. By doing this, model selection and parameter estimation are simultaneously achieved. Yuan and Lin (2007) used the determinant maximization (MAXDET) algorithm. Friedman et al. (2008) took advantage of the clockwise coordinate descent approach and developed the graphical lasso (Glasso) algorithm which is remarkably fast. Rothman et al. (2008) derived the optimization algorithm utilizing Cholesky decomposition and the local quadratic approximation, and finally produced the sparse estimator that is permutation invariant. Nonetheless, it has been shown that the LASSO penalty produces biases in regression. To correct biases, the

Smoothly Clipped Absolute Deviation (SCAD) penalty and the adaptive LASSO penalty were proposed by Fan and Li (2001) and by Zou (2006), respectively. Fan et al. (2009) employed the two penalties above in precision matrix estimation and solved the bias problem.

However, all of these approaches ignore the fact that observations may come from different classes. Since the true precision matrix may have some differences among classes, assuming they all come from the same multivariate normal distribution is inappropriate. On the other hand, classes are related to each other in certain ways, so precision matrices (or the mapped network structures) of different classes may have something in common. For instance, patients with different types of diabetes (Type 1 diabetes, Type 2 diabetes, and gestational diabetes) may have different gene network structures, but part of the structures may be exactly the same due to the fact that patients are all of diabetes. In this situation, it is inappropriate to estimate precision matrix by viewing all observations as from one class, since it ignores the distinctions among classes; separately estimating precision matrices with respect to each class fails to take advantage of common structure among classes. Therefore, a joint estimation of precision matrices across multiple classes will take advantage of information across classes, so that common structure is estimated more precisely and unique structures are identified as well. Guo et al. (2011) proposed jointly estimate precision matrices for different classes by reparameterizing off-diagonal elements of precision matrices to be a multiplication of a common factor across classes and a unique factor for each class. Their method could be solved by the iterative weighted Glasso (Friedman et al., 2008). On the other hand, Danaher et al. (2014) used generalized fused lasso or group lasso as the penalty, and employed alternating directions method of multipliers (ADMM) algorithm to solve the optimization problem.

However, both Guo et al. (2011) and Danaher et al. (2014) are still limited to investigate the conditional dependency when there is multilevel data structure. This multilevel structure

means that some variables are considered as higher level variables and others are nested in these higher level variables, which are called lower level variables. For instance, a gene pathway is composed of a series of genes to work together for a particular cellular or physical function. In that scenario, pathways are the higher level variables and genes within a pathway are the lower level variables. Furthermore, higher level variables are not independent. For example, pathways are not isolated. Instead, they work together to accomplish certain tasks. Therefore, simultaneously exploring conditional dependency structures among higher level variables and among lower level variables are of interest, so that the multilevel network composed of higher level network and lower level network can be generated.

In this chapter, we are dealing with data that are of multilevel variables across multiple classes. We consider the scenarios where common conditional dependency structure among variables are of two levels: higher level and lower level. For instance, for the diabetes example we illustrated above, the common multilevel network structure across the three classes (Type 1 diabetes, Type 2 diabetes, and gestational diabetes) are composed of common pathway network and common gene network within pathway. Hence, we propose jointly estimating the multilevel Gaussian graphical models across multiple classes, by sharing the common multilevel conditional dependency structure during the estimation procedure. Previous research only concentrate on estimating single level precision matrices jointly across classes (Guo et al., 2011; Danaher et al., 2014). To the best of our knowledge, no research so far has considered estimating multilevel network jointly across classes. We summarized all the terms and definitions in this chapter in Table 3.1.

The rest of the chapter is organized as follows. In Section 3.2, we propose the multilevel Gaussian graphical model, followed by the joint estimation method (JMGGM). Section 3.3 describes the algorithm for the proposed method (JMGGM). Asymptotic properties are given in Section 3.4. Section 3.5 shows the simulation results with chain-chain network and chain-

scale free network, and summarizes the advantages of our method. In Section 3.6, we apply our proposed method to the multilevel gene and pathway data of the breast cancer patients. Section 3.7 contains concluding remarks.

Table 3.1: Terms and Definitions

Term	**Definition**
"Lower level variable" (Variable)	Variables that are nested in a higher level variable (E.g., Gene)
"Higher level variable" (Group)	A set of variables that serves a particular function (E.g., Pathway)
"Lower level network"	Connections among the set of variables within a higher level variable (E.g., Gene network within a pathway)
"Higher level network"	Connections among a set of higher level variables (E.g., Pathway network)
"Multilevel network"	Higher level network and lower level network
"Heterogeneous classes"	Classes differ, yet share common characteristics (E.g., White vs Non-white breast cancer patients)
"Joint estimation of multilevel network"	Common multilevel network structure is shared among heterogeneous classes

3.2 The Joint Estimation Method for the Multilevel Gaussian Graphical Model

3.2.1 Problem set-up

Suppose we have M heterogeneous classes with p variables, where $M \geqslant 2$. Furthermore, we know that the p variables are in K pre-specified groups, denoted by $P_1, \ldots, P_K$. The

groups are called higher-level variables and the variables within a higher-level variable are called lower-level variables, so that we have the so called "multilevel variables". There are p_k variables within the kth group P_k and $\sum_{k=1}^{K} p_k = p$. The mth class contains n_m observations $(\underline{x}_1^m, \ldots, \underline{x}_{n_m}^m)$, where $\underline{x}_i^m = (x_{i,1}^m, \ldots, x_{i,p}^m)$, $i = 1, \ldots, n_m$.

Our assumptions are:

1. Within each class m, $\underline{x}_1^m, \ldots, \underline{x}_{n_m}^m \in \mathbb{R}^p$ are independent and identically distributed samples from MN $(\underline{0}, (\Omega^{(m)})^{-1})$, where

$$\Omega^{(m)} = \begin{bmatrix} \omega_{1,1}^{(m)} & \cdots & \omega_{1,p}^{(m)} \\ \vdots & \ddots & \vdots \\ \omega_{p,1}^{(m)} & \cdots & \omega_{p,p}^{(m)} \end{bmatrix}$$

   and $\Omega^{(m)}$ is symmetric positive definite.

2. Observations from different classes are independent of each other.

We develop our approach under these assumptions which have been used for Gaussian graphical models (Yuan and Lin, 2007; Friedman et al., 2008; Rothman et al., 2008; Guo et al., 2011; Danaher et al., 2014). We note that for the mean is equal to 0 assumption, we can center the observations along each variable within each class. For the multivariate normality assumption, it can be tested using tests such as Mardia's test, Henze-Zirkler's test and Royston's test (Korkmaz et al., 2014). We may also use some graphical tools such as chi-square Q-Q, perspective and contour plots to check the multivariate normality assumption.

3.2.2 The multilevel Gaussian graphical model

Let the conditional correlations among variables in the kth and $k^{'}$th groups within class m be written as a p_k by $p_{k'}$ sub-block precision matrix $\Omega^{(m)}_{kk'}$, which is

$$\Omega^{(m)}_{kk'} = \begin{bmatrix} \omega^{kk'(m)}_{1,1} & \omega^{kk'(m)}_{1,2} & \cdots & \omega^{kk'(m)}_{1,p_{k'}} \\ \omega^{kk'(m)}_{2,1} & \omega^{kk'(m)}_{2,2} & \cdots & \omega^{kk'(m)}_{2,p_{k'}} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{kk'(m)}_{p_k,1} & \omega^{kk'(m)}_{p_k,2} & \cdots & \omega^{kk'(m)}_{p_k,p_{k'}} \end{bmatrix}.$$

Specifically, when $k = k'$, $\Omega^{(m)}_{kk}$ shows the conditional correlations among the p_k variables within the kth group given all the other variables, so the network structure of the p_k variables within the kth group could be inferred.

Now we consider that conditional correlations among variables in class m are coming from two layers, with the first layer indicating groups' (higher level variables) contribution and the second indicating variables from groups' (lower level variables) contribution. Consequently, we reparameterize the sub-block precision matrix $\Omega^{(m)}_{kk'}$ by introducing the "higher level factor" whose parameter is denoted as $\theta^{(m)}_{kk'}$ and the "lower level factor" whose parameter is denoted as $\gamma^{kk'(m)}_{i,j}$, respectively. Then, the partial correlation between the ith variable in the kth group and the jth variable in the k'th group in class m can be written as $\omega^{kk'(m)}_{i,j} = \theta^{(m)}_{kk'}\gamma^{kk'(m)}_{i,j}$. One constraints for the decomposition is $\theta^{(m)}_{kk'} \geqslant 0$, $1 \leqslant k, k' \leqslant K$, so that the sign of $\omega^{kk'(m)}_{i,j}$ is consistent with $\gamma^{kk'(m)}_{i,j}$. The following constraints helps reserve symmetry of precision matrix $\Omega^{(m)}$: $\theta^{(m)}_{kk'} = \theta^{(m)}_{k'k}$, $\gamma^{kk'(m)}_{i,j} = \gamma^{k'k(m)}_{j,i}$, and $\theta^{(m)}_{kk} = 1$ (i.e. $\omega^{kk(m)}_{i,j} = \gamma^{kk(m)}_{i,j}$).

Therefore, the precision matrix for class m is written as

$$\Omega^{(m)} = \begin{bmatrix} \Omega_{11}^{(m)} & \Omega_{12}^{(m)} & \cdots & \Omega_{1K}^{(m)} \\ \Omega_{21}^{(m)} & \Omega_{22}^{(m)} & \cdots & \Omega_{2K}^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \Omega_{K1}^{(m)} & \Omega_{K2}^{(m)} & \cdots & \Omega_{KK}^{(m)} \end{bmatrix} = \begin{bmatrix} \theta_{11}^{(m)}\Gamma_{11}^{(m)} & \theta_{12}^{(m)}\Gamma_{12}^{(m)} & \cdots & \theta_{1K}^{(m)}\Gamma_{1K}^{(m)} \\ \theta_{21}^{(m)}\Gamma_{21}^{(m)} & \theta_{22}^{(m)}\Gamma_{22}^{(m)} & \cdots & \theta_{2K}^{(m)}\Gamma_{2K}^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{K1}^{(m)}\Gamma_{K1}^{(m)} & \theta_{K2}^{(m)}\Gamma_{K2}^{(m)} & \cdots & \theta_{KK}^{(m)}\Gamma_{KK}^{(m)} \end{bmatrix},$$

where

$$\Gamma_{kk'}^{(m)} = \begin{bmatrix} \gamma_{1,1}^{kk'(m)} & \gamma_{1,2}^{kk'(m)} & \cdots & \gamma_{1,p_{k'}}^{kk'(m)} \\ \gamma_{2,1}^{kk'(m)} & \gamma_{2,2}^{kk'(m)} & \cdots & \gamma_{2,p_{k'}}^{kk'(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{p_k,1}^{kk'(m)} & \gamma_{p_k,2}^{kk'(m)} & \cdots & \gamma_{p_k,p_{k'}}^{kk'(m)} \end{bmatrix}$$

and we write

$$\Theta^{(m)} = \begin{bmatrix} \theta_{11}^{(m)} & \theta_{12}^{(m)} & \cdots & \theta_{1K}^{(m)} \\ \theta_{21}^{(m)} & \theta_{22}^{(m)} & \cdots & \theta_{2K}^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{K1}^{(m)} & \theta_{K2}^{(m)} & \cdots & \theta_{KK}^{(m)} \end{bmatrix}.$$

In the multilevel Gaussian graphical model, $\theta_{kk'}^{(m)}$ represents the "higher level factor" contributing to $\omega_{i,j}^{kk'(m)}$: if $\theta_{kk'}^{(m)} = 0$, then $\Omega_{kk'}^{(m)} = \mathbf{0}$, which means variables in group k are conditionally independent of those in group k', and thus we say group k and group k' are conditionally independent. Otherwise, we say the two groups are conditionally dependent.

Meanwhile, $\gamma_{i,j}^{kk'(m)}$ indicates the "lower level factor" contributing to $\omega_{i,j}^{kk'(m)}$, so if $\theta_{kk'}^{(m)} \neq 0$, the conditional dependency between the ith variable in the kth group and the jth variable in the k'th group is further affected by the "lower level factor" $\gamma_{i,j}^{kk'(m)}$.

3.2.3 Extension of the multilevel Gaussian graphical model

Given the fact that a lower level variable may belong to multiple higher level variables, our multilevel Gaussian graphical model can be also applicable and extended to the scenario where adjacent groups have overlapped variables, and non-adjacent groups have no overlaps.

Suppose the p variables are in K pre-specified groups, denoted by $P_1, \ldots, P_K$, and the number of genes in group k is p_k. Let G_k represents the set of indices of the variables in the kth group. Instead of having overlapped variables appear multiple times in Ω (target of the estimation), the overlaps are put in between groups where they belong to. Let us illustrate how to determine interactions among and within groups using the following Example 1.

Example 1. *Let us consider* 3 *groups, denoted as* P_1*,* P_2*, and* P_3*; Each group contains* 5 *variables:* P_1*,* P_2*, and* P_3 *have variables* $g_1 - g_5$*,* $g_4 - g_8$*, and* $g_7 - g_{11}$*, respectively. Therefore,* P_1 *and* P_2 *have overlapped variables* g_4 *and* g_5*,* P_2 *and* P_3 *have overlapped variables* g_7 *and* g_8*, but* P_1 *and* P_3 *have nothing in common.*

Therefore, in Example 1, we have $K = 3$, $p_k = 5$, $G_1 = \{1, 2, 3, 4, 5\}$, $G_2 = \{4, 5, 6, 7, 8\}$,

$G_3 = \{7, 8, 9, 10, 11\}$. The precision matrix is written as

$$\Omega = \begin{array}{c} \\ g_1 - g_3 \\ g_4 - g_5 \\ g_6 \\ g_7 - g_8 \\ g_9 - g_{11} \end{array} \begin{array}{c} \begin{array}{ccccc} g_1 - g_3 & g_4 - g_5 & g_6 & g_7 - g_8 & g_9 - g_{11} \end{array} \\ \begin{pmatrix} \Omega_{11} & \Omega_{12} & \Omega_{13} & \Omega_{14} & \Omega_{15} \\ & \Omega_{22} & \Omega_{23} & \Omega_{24} & \Omega_{25} \\ & & \Omega_{33} & \Omega_{34} & \Omega_{35} \\ & & & \Omega_{44} & \Omega_{45} \\ & & & & \Omega_{55} \end{pmatrix} \end{array},$$

where Ω_{12} captures the conditional correlations between variables in P_1 unique and those in $P_1\&P_2$ common ; $\Omega_{13}\&\Omega_{14}$ captures that between P_1 unique and P_2 unique; $\Omega_{23}\&\Omega_{24}$ captures that between $P_1\&P_2$ common and P_2 unique.

Our decision rules for the conditional dependency among the 3 groups are defined as the following rules:

- For groups with no overlaps such as P_1 and P_3:
  - If $(\Omega_{14} = \mathbf{0}) \bigcap (\Omega_{15} = \mathbf{0}) \bigcap (\Omega_{24} = \mathbf{0}) \bigcap (\Omega_{25} = \mathbf{0})$, we say P_1 and P_3 are conditionally independent because all the variables in P_1 are conditionally independent of those in P_3.
  - Otherwise, we say P_1 and P_3 are conditionally dependent.
- For adjacent groups with overlaps such as P_1 and P_2:
  - If $(\Omega_{13} \neq \mathbf{0}) \bigcup (\Omega_{14} \neq \mathbf{0})$, we say P_1 and P_2 are conditionally dependent because at least one variable in P_1 unique are conditionally dependent with at least one variable in P_2 unique.

- If $(\Omega_{13} = \mathbf{0}) \bigcap (\Omega_{14} = \mathbf{0})$, $(\Omega_{12} \neq \mathbf{0}) \bigcap ((\Omega_{23} \neq \mathbf{0}) \bigcup (\Omega_{24} \neq \mathbf{0}))$, we say P_1 and P_2 are conditionally dependent because the common variables are conditionally dependent with at least one variable in both groups.
- Otherwise, P_1 and P_2 are conditionally independent.

On the other hand, our decision rules for determining lower level networks are defined as the follows:

- For variables in P_1, $(\Omega_{11}\&\Omega_{12}\&\Omega_{22})$ tells the conditional dependency among variables in P_1 $(G_1 = \{1, 2, 3, 4, 5\})$;
- For variables in P_2: $(\Omega_{22}\&\Omega_{23}\&\Omega_{24}\&\Omega_{33}\&\Omega_{34}\&\Omega_{44})$ tells the conditional dependency among variables in P_2 $(G_2 = \{4, 5, 6, 7, 8\})$;
- For variables in P_3: $(\Omega_{44}\&\Omega_{45}\&\Omega_{55})$ tells the conditional dependency among variables in P_3 $(G_3 = \{7, 8, 9, 10, 11\})$.

Therefore, our multilevel Gaussian graphical model can be easily applicable with our decision rule.

3.2.4 The joint estimation method

The elements in $\Omega^{(m)}$ are decomposed into a "higher level factor" and a "lower level factor" within class m. When multiple multilevel Gaussian graphical models share some common structure, estimating them jointly will take advantage of the common structure across classes, and thus estimation accuracy could be improved. Therefore, we reparameterize the higher level factor $\theta_{kk'}^{(m)}$, as $\theta_{kk'}^{(m)} = \alpha_{kk'}\beta_{kk'}^{(m)}$ $(1 \leqslant k \neq k' \leqslant K,\ 1 \leqslant m \leqslant M)$ and the lower level factor $\gamma_{i,j}^{kk(m)}$, as $\gamma_{i,j}^{kk(m)} = \iota_{i,j}^{(kk)}\rho_{i,j}^{kk(m)}$ $(1 \leqslant k \leqslant K,\ 1 \leqslant i, j \leqslant p_k,\ 1 \leqslant m \leqslant M)$. Similarly,

the constraints for the first decomposition are $\alpha_{kk'} \geqslant 0$, $\alpha_{kk'} = \alpha_{k'k}$ and $\beta^{(m)}_{kk'} = \beta^{(m)}_{k'k}$ $(1 \leqslant k \neq k' \leqslant K,\ 1 \leqslant m \leqslant M)$, and those for the second are $\iota^{(kk)}_{i,j} \geqslant 0$, $\iota^{(kk)}_{i,j} = \iota^{(kk)}_{j,i}$ and $\rho^{kk(m)}_{i,j} = \rho^{kk(m)}_{j,i} (1 \leqslant k \leqslant K,\ 1 \leqslant i \neq j \leqslant p_k,\ 1 \leqslant m \leqslant M)$ and $\iota^{(kk)}_{i,i} = 1$ (i.e. $\gamma^{kk(m)}_{i,i} = \rho^{kk(m)}_{i,i}$).

In the decomposition, for the higher level Gaussian graphical model, $\alpha_{kk'}$ indicates the common higher level structure across classes and $\beta^{(m)}_{kk'}$ reflects the unique higher level strcture for class m. Likewise, for the lower level Gaussian graphical model, $\iota^{(kk)}_{i,j}$ shows the common lower level structure within group k among the M classes and $\rho^{kk(m)}_{i,j}$ reflects the unique lower level structure within pathway k for class m.

By adding the information of common structure into the model, we have

$$\Omega^{(m)}_{kk'} = \theta^{(m)}_{kk'}\Gamma^{(m)}_{kk'} = \alpha_{kk'}\beta^{(m)}_{kk'}\Gamma^{(m)}_{kk'}$$

so that

$$\Omega^{(m)} = \begin{bmatrix} \theta^{(m)}_{11}\Gamma^{(m)}_{11} & \theta^{(m)}_{12}\Gamma^{(m)}_{12} & \cdots & \theta^{(m)}_{1K}\Gamma^{(m)}_{1K} \\ \theta^{(m)}_{21}\Gamma^{(m)}_{21} & \theta^{(m)}_{22}\Gamma^{(m)}_{22} & \cdots & \theta^{(m)}_{2K}\Gamma^{(m)}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \theta^{(m)}_{K1}\Gamma^{(m)}_{K1} & \theta^{(m)}_{K2}\Gamma^{(m)}_{K2} & \cdots & \theta^{(m)}_{KK}\Gamma^{(m)}_{KK} \end{bmatrix} = \begin{bmatrix} \Gamma^{(m)}_{11} & \alpha_{12}\beta^{(m)}_{12}\Gamma^{(m)}_{12} & \cdots & \alpha_{1K}\beta^{(m)}_{1K}\Gamma^{(m)}_{1K} \\ \alpha_{21}\beta^{(m)}_{21}\Gamma^{(m)}_{21} & \Gamma^{(m)}_{22} & \cdots & \alpha_{2K}\beta^{(m)}_{2K}\Gamma^{(m)}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{K1}\beta^{(m)}_{K1}\Gamma^{(m)}_{K1} & \alpha_{K2}\beta^{(m)}_{K2}\Gamma^{(m)}_{K2} & \cdots & \Gamma^{(m)}_{KK} \end{bmatrix}$$

By denoting the Schur-Hadamard product of two equal size matrices I and P as $I \circ P$ and

defining matrices A, $I^{(kk)}$, and $P^{kk(m)}$ as

$$A = \begin{bmatrix} 1 & \alpha_{12} & \cdots & \alpha_{1K} \\ \alpha_{21} & 1 & \cdots & \alpha_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{K1} & \alpha_{K2} & \cdots & 1 \end{bmatrix}$$

$$I^{(kk)} = \begin{bmatrix} 1 & \iota_{1,2}^{(kk)} & \cdots & \iota_{1,p_k}^{(kk)} \\ \iota_{2,1}^{(kk)} & 1 & \cdots & \iota_{2,p_k}^{(kk)} \\ \vdots & \vdots & \ddots & \vdots \\ \iota_{p_k,1}^{(kk)} & \iota_{p_k,2}^{(kk)} & \cdots & 1 \end{bmatrix} \quad P^{kk(m)} = \begin{bmatrix} 1 & \rho_{1,2}^{kk(m)} & \cdots & \rho_{1,p_k}^{kk(m)} \\ \rho_{2,1}^{kk(m)} & 1 & \cdots & \rho_{2,p_k}^{kk(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{p_k,1}^{kk(m)} & \rho_{p_k,2}^{kk(m)} & \cdots & 1 \end{bmatrix},$$

we can express $\Gamma_{kk}^{(m)}$ using the Schur-Hadamard product and define $\Gamma_{\beta kk'}^{(m)}$ as the follows:

$$\begin{aligned} \Gamma_{kk}^{(m)} &= I_{kk} \circ P_{kk}^{(m)}; \\ \beta_{kk'}^{(m)} \Gamma_{kk'}^{(m)} &= \Gamma_{\beta kk'}^{(m)}. \end{aligned}$$

Given the decomposition and notations described above, we propose the following penalized log-likelihood, with the objective function denoted as Q_1:

$$\begin{aligned} \min_{\{A, \Gamma_{\beta kk'}^{(m)}, I_{kk}, P_{kk}^{(m)}\}_{m=1}^{M}} \quad & \sum_{m=1}^{M} \quad n_m [\text{trace}(\mathrm{S}^{(m)} \Omega^{(m)}) - \log|\Omega^{(m)}|] \\ & + \quad \eta_1 \sum_{k \neq k'} \alpha_{kk'} + \eta_2 \sum_{k \neq k'} \sum_{m=1}^{M} |\Gamma_{\beta kk'}^{(m)}|_1 \\ & + \quad \eta_3 \sum_{k=1}^{K} \sum_{1 \leqslant i \neq j \leqslant p_k} \iota_{i,j}^{kk} \end{aligned}$$

$$+ \quad \eta_4 \sum_{k=1}^{K} \sum_{1 \leqslant i \neq j \leqslant p_k} \sum_{m=1}^{M} |\rho_{i,j}^{kk(m)}|. \tag{3.1}$$

Objective function (3.1) could be reduced to an equivalent problem with two tuning parameters, with the objective function denoted as Q_2:

$$\begin{aligned} \min_{\{A,\Gamma_{\beta kk'}^{(m)},I_{kk},P_{kk}^{(m)}\}_{m=1}^{M}} \quad & \sum_{m=1}^{M} \quad n_m[\text{trace}(\mathrm{S}^{(m)}\Omega^{(m)}) - \log|\Omega^{(m)}|] \\ & + \quad \sum_{k \neq k'} \alpha_{kk'} + \eta_{12} \sum_{k \neq k'} \sum_{m=1}^{M} |\Gamma_{\beta kk'}^{(m)}|_1 \\ & + \quad \sum_{k=1}^{K} \sum_{1 \leqslant i \neq j \leqslant p_k} \iota_{i,j}^{kk} \\ & + \quad \eta_{34} \sum_{k=1}^{K} \sum_{1 \leqslant i \neq j \leqslant p_k} \sum_{m=1}^{M} |\rho_{i,j}^{kk(m)}|, \end{aligned} \tag{3.2}$$

where $\eta_{12} = \eta_1\eta_2$ and $\eta_{34} = \eta_3\eta_4$.

The equivalence of minimizing Q_1 and minimizing Q_2 is shown in Lemma 1.

Lemma 1:

Let $\{\hat{A}^*, \hat{\Gamma}_{\beta kk'}^{(m)*}, \hat{I}_{kk}^*, \hat{P}_{kk}^{(m)*}\}$ *be a local minimizer of* $Q1$*, then there exists a local minimizer* $\{\hat{A}^{**}, \hat{\Gamma}_{\beta kk'}^{(m)**}, \hat{I}_{kk}^{**}, \hat{P}_{kk}^{(m)**}\}$ *of* $Q2$*, such that* $\hat{\alpha}_{kk'}^{*}\hat{\Gamma}_{\beta kk'}^{(m)*} = \hat{\alpha}_{kk'}^{**}\hat{\Gamma}_{\beta kk'}^{(m)**}$ *and* $\hat{\iota}_{i,j}^{kk*}\hat{\rho}_{i,j}^{kk(m)*} = \hat{\iota}_{i,j}^{kk**}\hat{\rho}_{i,j}^{kk(m)**}$*. Similarly, let* $\{\hat{A}^{**}, \hat{\Gamma}_{\beta kk'}^{(m)**}, \hat{I}_{kk}^{**}, \hat{P}_{kk}^{(m)**}\}$ *be a local minimizer of* $Q2$*, then there exists a local minimizer* $\{\hat{A}^*, \hat{\Gamma}_{\beta kk'}^{(m)*}, \hat{I}_{kk}^*, \hat{P}_{kk}^{(m)*}\}$ *of* $Q1$*, such that* $\hat{\alpha}_{kk'}^{*}\hat{\Gamma}_{\beta kk'}^{(m)*} = \hat{\alpha}_{kk'}^{**}\hat{\Gamma}_{\beta kk'}^{(m)**}$ *and* $\hat{\iota}_{i,j}^{kk*}\hat{\rho}_{i,j}^{kk(m)*} = \hat{\iota}_{i,j}^{kk**}\hat{\rho}_{i,j}^{kk(m)**}$.

Proof:

Suppose $\{\hat{A}^*, \hat{\Gamma}_{\beta kk'}^{(m)*}, \hat{I}_{kk}^*, \hat{P}_{kk}^{(m)*}\}$ is a local minimizer of $Q1$. We would like to prove $\{\hat{A}^{**} = \eta_1\hat{A}^*, \hat{\Gamma}_{\beta kk'}^{(m)**} = \hat{\Gamma}_{\beta kk'}^{(m)*}/\eta_1, \hat{I}_{kk}^{**} = \eta_3\hat{I}_{kk}^*, \hat{P}_{kk}^{(m)**} = \hat{P}_{kk}^{(m)*}/\eta_3\}$ is a local minimizer of $Q2$. It could be shown that $Q1(A, \Gamma_{\beta kk'}^{(m)}, I_{kk}, P_{kk}^{(m)}, \eta_1, \eta_2, \eta_3, \eta_4) = Q2(\eta_1 A, \Gamma_{\beta kk'}^{(m)}/\eta_1, \eta_3 I_{kk}, P_{kk}^{(m)}/\eta_3, \eta_{12}, \eta_{34})$, and the rest of the proof can be easily shown by following the definition of local minimizer.

The proof is similar for the other direction.

Furthermore, we formulate (3.2) as follows, with the objective function denoted as Q_3:

$$\begin{aligned}\min_{\{\Omega_{kk'}^{(m)},\Gamma_{kk}^{(m)}\}_{m=1}^{M}} \quad & \sum_{m=1}^{M} \quad n_m[\text{trace}(\mathrm{S}^{(m)}\Omega^{(m)}) - \log|\Omega^{(m)}|] \\ & + \quad \lambda_1 \sum_{k\neq k'} (\sum_{m=1}^{M} \sum_{\substack{1\leqslant i\leqslant p_k \\ 1\leqslant j\leqslant p_{k'}}} |\omega_{i,j}^{kk'(m)}|)^{1/2} \\ & + \quad \lambda_2 \sum_{k=1}^{K} \sum_{1\leqslant i\neq j\leqslant p_k} (\sum_{m=1}^{M} |\gamma_{i,j}^{kk(m)}|)^{1/2}. \end{aligned} \tag{3.3}$$

where $\lambda_1 = 2\sqrt{\eta_{12}}$ and $\lambda_2 = 2\sqrt{\eta_{34}}$.

Similarly, it can be proved that minimizing Q_2 and minimizing Q_3 are equivalent by Lemma 2.

Lemma 2:

Let $\{\hat{\Omega}_{kk'}^{(m)**}, \hat{\Gamma}_{kk}^{(m)**}\}$ *be a local minimizer of* $Q3$*, then there exists a local minimizer* $\{\hat{A}^{**}, \hat{\Gamma}_{\beta kk'}^{(m)}{}^{**}, \hat{I}_{kk}^{**}, \hat{P}_{kk}^{(m)**}\}$ *of* $Q2$*, such that* $\hat{\Omega}_{kk'}^{(m)**} = \hat{\alpha}_{kk'}^{**}\hat{\Gamma}_{\beta kk'}^{(m)}{}^{**}$ *and* $\hat{\Gamma}_{kk}^{(m)**} = \hat{I}_{kk}^{**} \circ \hat{P}_{kk}^{(m)**}$*. Likewise, let* $\{\hat{A}^{**}, \hat{\Gamma}_{\beta kk'}^{(m)}{}^{**}, \hat{I}_{kk}^{**}, \hat{P}_{kk}^{(m)**}\}$ *be a local minimizer of* $Q2$*, then there exists a local minimizer* $\{\hat{\Omega}_{kk'}^{(m)**}, \hat{\Gamma}_{kk}^{(m)**}\}$ *of* $Q3$*, such that* $\hat{\Omega}_{kk'}^{(m)**} = \hat{\alpha}_{kk'}^{**}\hat{\Gamma}_{\beta kk'}^{(m)}{}^{**}$ *and* $\hat{\Gamma}_{kk}^{(m)**} = \hat{I}_{kk}^{**} \circ \hat{P}_{kk}^{(m)**}$.

Proof:

The key step is to write $\hat{\alpha}_{kk'}^{**} = \sqrt{\eta_{12}\sum_{m=1}^{M}|\hat{\Omega}_{kk'}^{(m)}{}^{**}|_1}$ and $\hat{\Gamma}_{\beta kk'}^{(m)}{}^{**} = \hat{\Omega}_{kk'}^{(m)**}/\sqrt{\eta_{12}\sum_{m=1}^{M}|\hat{\Omega}_{kk'}^{(m)}{}^{**}|_1}$, where $|\hat{\Omega}_{kk'}^{(m)}{}^{**}|_1 = \sum_{\substack{1\leqslant i\leqslant p_k \\ 1\leqslant j\leqslant p_{k'}}} |\hat{\omega}_{i,j}^{kk'(m)**}|$. Similarly, we may write $\hat{\iota}_{i,j}^{kk**} = \sqrt{\eta_{34}\sum_{m=1}^{M}|\hat{\gamma}_{i,j}^{kk(m)**}|}$ and $\hat{\rho}_{i,j}^{kk(m)**} = \hat{\gamma}_{i,j}^{kk(m)**}/\sqrt{\eta_{34}\sum_{m=1}^{M}|\hat{\gamma}_{i,j}^{kk(m)**}|}$.

The solution of (3.3) could be obtained from using an interactive approach based on local linear approximation (Zou and Li, 2008). By lettng $\omega_{i,j}^{kk'(m)(t)}$ and $\gamma_{i,j}^{kk(m)(t)}$ denote the

estimates from the tth iteration, we could the following approximation solutions:

$$\begin{aligned}(\sum_{m=1}^{M}\sum_{\substack{1\leqslant i\leqslant p_k\\1\leqslant j\leqslant p_{k'}}}|\omega_{i,j}^{kk'(m)}|)^{1/2} &\approx \frac{\sum_{m=1}^{M}\sum_{\substack{1\leqslant i\leqslant p_k\\1\leqslant j\leqslant p_{k'}}}|\omega_{i,j}^{kk'(m)}|}{(\sum_{m=1}^{M}\sum_{\substack{1\leqslant i\leqslant p_k\\1\leqslant j\leqslant p_{k'}}}|\omega_{i,j}^{kk'(m)^{(t)}}|)^{1/2}};\\ (\sum_{m=1}^{M}|\gamma_{i,j}^{kk(m)}|)^{1/2} &\approx \frac{\sum_{m=1}^{M}|\gamma_{i,j}^{kk(m)}|}{(\sum_{m=1}^{M}|\gamma_{i,j}^{kk(m)^{(t)}}|)^{1/2}}.\end{aligned}$$

Therefore, (3.3) could be decomposed into M individual optimization problems at the $(t+1)$th iteration:

$$\begin{aligned}\min_{\{\Omega_{kk'}^{(m)},\Gamma_{kk}^{(m)}\}} \quad & n_m[\text{trace}(\mathrm{S}^{(m)}\Omega^{(m)})-\log|\Omega^{(m)}|]\\ &+\lambda_1\textstyle\sum_{k\neq k'}\frac{\sum_{\substack{1\leqslant i\leqslant p_k\\1\leqslant j\leqslant p_{k'}}}|\omega_{i,j}^{kk'(m)}|}{(\sum_{m=1}^{M}\sum_{\substack{1\leqslant i\leqslant p_k\\1\leqslant j\leqslant p_{k'}}}|\omega_{i,j}^{kk'(m)^{(t)}}|)^{1/2}}\\ &+\lambda_2\textstyle\sum_{k=1}^{K}\sum_{1\leqslant i\neq j\leqslant p_k}\frac{|\gamma_{i,j}^{kk(m)}|}{(\sum_{m=1}^{M}|\gamma_{i,j}^{kk(m)^{(t)}}|)^{1/2}}.\end{aligned}\tag{3.4}$$

Therefore, the solution of (3.4) could be efficiently solved using the weighted Glasso algorithm by Friedman et al. (2008).

3.2.5 Tuning parameters selection

Notation $|\cdot|$ and trace$(\cdot)$ are used to represent the determinant and the trace of a matrix, respectively. The tuning parameters λ_1 and λ_2 in (3.3) controls the sparsity of multilevel Gaussian graphical models individually. We choose them using the Bayesian Information Criterion (BIC), defined as:

$$\text{BIC}(\lambda_1,\lambda_2)=\sum_{m=1}^{M}\{n_m[\text{trace}(S^{(m)}\hat{\Omega}_{\lambda_1,\lambda_2}^{(m)})-\log|\hat{\Omega}_{\lambda_1,\lambda_2}^{(m)}|]+df_m\log(n_m)\},$$

where $df_m = \#\{(i,j) : i < j, \hat{\omega}_{i,j}^{(m)} \neq 0\}$, and $\hat{\Omega}_{\lambda_1,\lambda_2}^{(m)}$ is the $\hat{\Omega}^{(m)}$ when we impose tuning parameters λ_1 and λ_2.

Therefore we finalize the proposed joint estimation method for the multilevel Gaussian graphical models across multiple classes, denoted as JMGGM.

3.3 Algorithm for JMGGM

Our joint estimation algorithm for solving (3.3) can be proceeded to the following steps:

Step 1 Initialize $\hat{\Omega}^{(m)} = (\hat{\Sigma}^{(m)} + v^{(m)} I_p)^{(-1)}$ for all $m = 1, \cdots, M$. I_p is the identity matrix and the constant $v^{(m)}$ is chosen to guarantee $\hat{\Omega}^{(m)} = (\hat{\Sigma}^{(m)} + v^{(m)} I_p)$ is positive definite.

Step 2 Update $\hat{\Omega}^{(m)}$ by (3.4) for all $m = 1, \cdots, M$ using the graphical lasso algorithm.

Step 3 Repeat Step2 until convergence is attained or until stability is broken.
Note that we say stability is broken when either $(\sum_{m=1}^{M} \sum_{\substack{1 \leqslant i \leqslant p_k \\ 1 \leqslant j \leqslant p_{k'}}} |\omega_{i,j}^{kk'(m)^{(t)}}|)^{1/2}$ or $(\sum_{m=1}^{M} |\gamma_{i,j}^{kk(m)^{(t)}}|)^{1/2}$ are less than $1e-10$. In addition, the criteria for checking convergence is that the average absolute estimate change over the precision matrix is less than a threshold, say $1e-4$.

The initialization in Step1 is achieved by selecting the right $v^{(m)}$ so that $\hat{\Sigma}^{(m)} + v^{(m)} I_p$ is strictly diagonal dominant, and thus nonsingular and positive definite. For example, one can set them as $\hat{\Sigma}^{(m)} = \{\hat{\sigma}_{i,j}^{(m)}\}_{1 \leqslant i,j \leqslant p}$, and $v^{(m)} = \max\{\sum_{j=1}^{p} |\hat{\sigma}_{1,j}^{(m)}| - 2|\hat{\sigma}_{1,1}^{(m)}|, \sum_{j=1}^{p} |\hat{\sigma}_{2,j}^{(m)}| - 2|\hat{\sigma}_{2,2}^{(m)}|, \cdots, \sum_{j=1}^{p} |\hat{\sigma}_{p,j}^{(m)}| - 2|\hat{\sigma}_{p,p}^{(m)}|\} + 1$. In addition, we also note that we need to have $\lambda_2 < \lambda_1$, because the shrinkage speed for $\gamma_{i,j}^{kk(m)}$ is much faster than that of $\omega_{i,j}^{kk'(m)}$.

3.4 Asymptotic Properties

In this section, we provide the asymptotic properties of the joint estimation method of the multilevel Gaussian graphical model (JMGGM): consistency and sparsity. Our asymptotic properties are developed under the scenario where M is fixed, both n and p are going to infinity, and the tuning parameters are going to zeros at a certain rate which will be described in detail in this section. First we define some necessary notations. We then introduce the regularity conditions for the true precision matrices $(\Omega_0^{(1)}, \Omega_0^{(2)}, \cdots, \Omega_0^{(M)})$, where

$$\Omega_0^{(m)} = \begin{bmatrix} \Omega_{0,11}^{(m)} & \Omega_{0,12}^{(m)} & \cdots & \Omega_{0,1K}^{(m)} \\ \Omega_{0,21}^{(m)} & \Omega_{0,22}^{(m)} & \cdots & \Omega_{0,2K}^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \Omega_{0,K1}^{(m)} & \Omega_{0,K2}^{(m)} & \cdots & \Omega_{0,KK}^{(m)} \end{bmatrix}$$

Let $S_m = \{(i,j) : 1 \leqslant k \neq k' \leqslant K, \omega_{i,j}^{kk'(m)} \neq 0\}$ be the set of indices of all nonzero elements in the off diagonal block matrices $\Omega_{kk'}^{(m)}$s, and let $S = S_1 \bigcup S_2 \bigcup \cdots \bigcup S_M$. Denote $T_m = \{(i,j) : i \neq j, \omega_{i,j}^{kk(m)} \neq 0\}$ to be the set of indices of all nonzero off-diagonal elements in the diagonal block matrices $\Omega_{kk}^{(m)}$s and $T = T_1 \bigcup T_2 \bigcup \cdots \bigcup T_M$ be the union of them. Let $q = |S| + |T|$, sum of cardinalities of S and T. Furthermore, let $\|\cdot\|_F$ and $\|\cdot\|$ be the Frobenius norm and the 2-norm of matrices, respectively.

Suppose the following regularity conditions hold:

Condition 1: There exist constant τ_1, τ_2 such that for all $p \geqslant 1$ and $m = 1, \cdots, M$, $0 < \tau_1 < \lambda_{min}(\Omega_0^{(m)}) \leqslant \lambda_{max}(\Omega_0^{(m)}) < \tau_2 < \infty$, where $\lambda_{min}(\Omega_0^{(m)})$ and $\lambda_{max}(\Omega_0^{(m)})$ denote the minimal and maximal eigenvalues of $\Omega_0^{(m)}$.

Condition 2: There exists a constant $\tau_3 > 0$, such that $\min\limits_{m=1,\cdots,M} \min\{ \min\limits_{(i,j)\in S_m} |\omega_{0,i,j}^{kk'(m)}|, \min\limits_{(i,j)\in T_m} |\omega_{0,i,j}^{kk(m)}| \} \geqslant \tau_3$.

Theorem 2. *(CONSISTENCY)*

Under the regularity conditions, assume that $(p+q)(\log p)/n = o(1)$ *and* $\Lambda_1\{(\log p)/n\}^{1/2} \leqslant \lambda_2 < \lambda_1 \leqslant \Lambda_2\{(1+p/q)(\log p)/n\}^{1/2}$ *for some positive constants* Λ_1 *and* Λ_2*. Then there exists a local minimizer of* $(\hat{\Omega}^{(m)})_{m=1}^{M}$ *of* (3.3) *such that*

$$\sum_{m=1}^{M} \|\hat{\Omega}^{(m)} - \Omega_0^{(m)}\|_F = O_p[\{(p+q)(\log p)/n\}^{1/2}].$$

Theorem 3. *(SPARSISTENCY)*

Under the assumptions of Theorem 2, assume that $\sum_{m=1}^{M} \|\hat{\Omega}^{(m)} - \Omega_0^{(m)}\|^2 = O_p(\eta_n)$*, where* $\eta_n \to 0$*,* $\{(\log p)/n\}^{1/2} + {\eta_n}^{1/2} = O(\lambda_1)$*, and* $\{(\log p)/n\}^{1/2} + {\eta_n}^{1/2} = O(\lambda_2)$*. Then with probability tending to 1, the local minimizer* $(\hat{\Omega}^{(m)})_{m=1}^{M}$ *in Theorem 2 satisfies* $\omega_{i,j}^{kk'(m)} = 0$ *for all* $(i,j) \in S_m^c$ *and* $\omega_{i,j}^{kk(m)=0}$ *for all* $(i,j) \in T_m^c$*,* $m = 1, \cdots, M$*.*

The proofs of Theorem 2 and Theorem 3 follow closely those in Guo et al. (2011). The main difference is that we consider a higher level variable information, and impose a uniform tuning parameter $\frac{\lambda_2}{(\sum_{m=1}^{M} \sum_{\substack{1 \leqslant i \leqslant p_k \\ 1 \leqslant j \leqslant p_{k'}}} |\omega_{i,j}^{kk'(m)^{(t)}}|)^{1/2}}$ on all the elements in the off-diagonal block matrices $\Omega_{kk'}^{(m)}$. As long as the tuning parameter is well defined, and the regularity conditions are consistent with the rest of off-diagonal elements of $\Omega^{(m)}$, the proofs are similar to Guo et al. (2011) and are straightforward.

3.5 Simulation

3.5.1 Simulation settings

We conduct simulation to evaluate the performance of our JMGGM under two types of common multilevel network. Both of the higher level networks are chain networks, and the

lower level networks are either chain network or scale-free network. The chain-chain network is denoted as CCN and the chain-scalefree network is denoted as CSN. The left panel of Figure 3.1 shows the common multilevel network CCN and the right panel displays CSN.

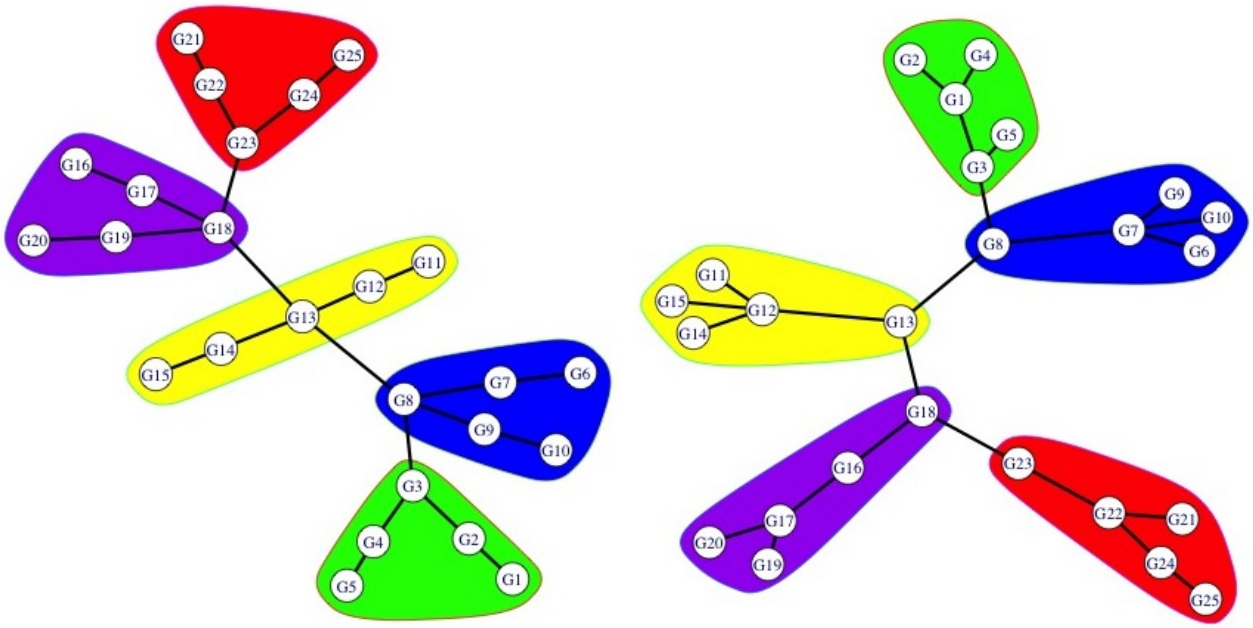

Figure 3.1: The simulated multilevel networks. The left penal shows a chain-chain network (CCN), and the right penal represents a chain-scalefree network (CSN).

We consider $K = 5$ groups (the higher level variable), where each group has $p_k = 10$ variables (the lower level variable), so that the total number of variables is $p = 50$, and $M = 3$ heterogeneous classes, where each class has three sets of sample size: $(n_1, n_2, n_3) = (200, 200, 200)$, $(n_1, n_2, n_3) = (100, 100, 100)$, and $(n_1, n_2, n_3) = (50, 50, 50)$.

Independent and identically distributed samples $\underline{x}_1^m, \ldots, \underline{x}_{n_m}^m \in \mathbb{R}^p$ are generated from MN $(\underline{0}, (\Omega^{(m)})^{-1})$, where $m = 1, 2, 3$. The general procedure of simulating the multilevel network structure $(\Omega^{(m)})$ consists of two steps: one is to generate common structure and the other is to add heterogeneity. We summarize how to generate the multilevel network structure for chain-chain network in the following steps:

Step 1 Generate the common multilevel network structure using Step1.1-1.2:

Step 1.1 The common higher level network - chain network is generated. That is, tridiagonal matrix $\Theta^{(m)} = (\theta^{(m)}_{kk'})$ is simulated for class m, so that the $\Theta^{(m)}$s share the same pattern of zeros and non-zeros, but the values of the non-zero elements may be different among classes. The inverse of $\Theta^{(m)} = (\theta^{(m)}_{kk'})$ is generated as follows: the (i,j)th element is $\exp(-|s_i - s_j|)$, $s_1 < s_2 < \cdots < s_K$, and $s_i - s_{i-1} \overset{i.i.d.}{\sim} \text{Unif}(0.5, 1)$, $i = 2, \ldots, K$. Taking the inverse of the generated matrix will give us tridiagonal matrix $\Theta^{(m)} = (\theta^{(m)}_{kk'})$.

Step 1.2 The lower level network - chain network is generated. Basically, $\Gamma^{(m)}_{kk}$ for class m and group k is generated, so that we have matrices $\Gamma^{(m)}_{kk}$ corresponding to chain network structure. As for $\Gamma^{(m)}_{kk'}$s, which corresponds to the conditional correlation between variables in group k and that in group k' for class m, we set all of them to be diagonal matrices.

Step 2 Add heterogeneity:

We add heterogeneity among the classes by introducing ρ ($\rho = 0, 1/4, 3/4$) - the heterogeneity at the higher level network. For each $\Theta^{(m)}$ ($m = 1, 2, 3$), a pair of symmetric zero elements is randomly picked, and it was replaced with a value uniformly generated from the $[-1, -0.5] \bigcup [0.5, 1]$ interval. This procedure is repeated ρT times, where T is the number of common links on the higher level network ($T = K - 1$), and ρ is the ratio of the number of individual links to the number of common links.

For Chain-Scale-free network, the simulation procedures are the same except for that in Step 1.2, we generate scale-free network using the Barabási-Albert algorithm (Barabási and Albert, 1999) for the lower level network.

3.5.2 Evaluation metrics

We generate 100 replicates of $M = 3$ classes for each network described in Section 3.5.1. Our method is evaluated in terms of two layers of measurements: one layer is on the higher level network and the other is on the lower level network.

On the higher level network, the first evaluation metric is "group connection degree bias (GCDBias)", which is defined as:

$$\begin{aligned} GCD_{kk'}^{(m)} = \frac{\#nonzero(\Omega_{kk'}^{(m)})}{p_k \times p_{k'}}; \hat{GCD}_{kk'}^{(m)} = \frac{\#nonzero(\hat{\Omega}_{kk'}^{(m)})}{p_k \times p_{k'}}; \\ GCDBias_{kk'}^{(m)} = |GCD_{kk'}^{(m)} - \hat{GCD}_{kk'}^{(m)}|; \\ GCDBias = \frac{1}{M}\sum_{m=1}^{M} GCDBias^{(m)} = \frac{1}{M}\sum_{m=1}^{M}\sum_{k\neq k'} GCDBias_{kk'}^{(m)}. \end{aligned} \tag{3.5}$$

The second set of evaluation metrics are in terms of how the higher level network structure (zeros and non-zeros) is estimated, including false positive rate (FPR), true positive rate (TPR), accuracy (ACC), and false discovery rate (FDR). We first define false positive (FP), true positive (TP), false negative (FN), and true negative (TN) for $\hat{\Theta}^{(m)}$ as follows:

$$\begin{aligned} FP_m &= \sum_{1\leq k<k'\leq K} I(\theta_{k,k'}^{(m)} = 0, \hat{\theta}_{k,k'}^{(m)} \neq 0); \\ TP_m &= \sum_{1\leq k<k'\leq K} I(\theta_{k,k'}^{(m)} \neq 0, \hat{\theta}_{k,k'}^{(m)} \neq 0); \\ FN_m &= \sum_{1\leq k<k'\leq K} I(\theta_{k,k'}^{(m)} \neq 0, \hat{\theta}_{k,k'}^{(m)} = 0); \\ TN_m &= \sum_{1\leq k<k'\leq K} I(\theta_{k,k'}^{(m)} = 0, \hat{\theta}_{k,k'}^{(m)} = 0); \end{aligned}$$

and then calculate the overall FPR, TPR, FDR, and ACC across classes, defined as:

$$\begin{aligned} FPR &= \frac{1}{M}\sum_{m=1}^{M} FPR_m = \frac{1}{M}\sum_{m=1}^{M} \frac{FP_m}{FP_m + TN_m}; \\ TPR &= \frac{1}{M}\sum_{m=1}^{M} TPR_m = \frac{1}{M}\sum_{m=1}^{M} \frac{TP_m}{TP_m + FN_m}; \\ FDR &= \frac{1}{M}\sum_{m=1}^{M} FDR_m = \frac{1}{M}\sum_{m=1}^{M} \frac{FP_m}{FP_k + TP_m}; \\ ACC &= \frac{1}{M}\sum_{m=1}^{M} ACC_m = \frac{1}{M}\sum_{m=1}^{M} \frac{TP_m + TN_m}{K(K-1)/2}. \end{aligned} \tag{3.6}$$

Regarding the lower level network, two sets of evaluation metrics are assessed. The first set is about the information losses, including entropy loss (EL) and Frobinus loss (FL), which are defined as:

$$\begin{aligned} EL &= \frac{1}{KM}\sum_{k=1}^{K}\sum_{m=1}^{M} EL_{kk}^{(m)} = \frac{1}{KM}\sum_{k=1}^{K}\sum_{m=1}^{M} \{\text{trace}[(\Omega_{kk}^{(m)})^{-1}\hat{\Omega}_{kk}^{(m)}] - \log[|(\Omega_{kk}^{(m)})^{-1}\hat{\Omega}_{kk}^{(m)}|] - p_k\}; \\ FL &= \frac{1}{KM}\sum_{k=1}^{K}\sum_{m=1}^{M} FL_{kk}^{(m)} = \frac{1}{KM}\sum_{k=1}^{K}\sum_{m=1}^{M} \{\|\Omega_{kk}^{(m)} - \hat{\Omega}_{kk}^{(m)}\|_F^2 / \|\Omega_{kk}^{(m)}\|_F^2\}. \end{aligned} \tag{3.7}$$

The second set of evaluation metrics are in terms of how the lower level network structure (zeros and non-zeros) is estimated, including FPR, TPR, ACC, and FDR. Again, FP, TP, FN and TN for $\Omega_{kk}^{(m)}$ are firstly defined as:

$$
\begin{aligned}
FP^{kk(m)} &= \sum_{1\le i<j\le p_k} I(\omega_{i,j}^{kk(m)} = 0, \hat{\omega}_{i,j}^{kk(m)} \neq 0);\\
TP^{kk(m)} &= \sum_{1\le i<j\le p_k} I(\omega_{i,j}^{kk(m)} \neq 0, \hat{\omega}_{i,j}^{kk(m)} \neq 0);\\
FN^{kk(m)} &= \sum_{1\le i<j\le p_k} I(\omega_{i,j}^{kk(m)} \neq 0, \hat{\omega}_{i,j}^{kk(m)} = 0);\\
TN^{kk(m)} &= \sum_{1\le i<j\le p_k} I(\omega_{i,j}^{kk(m)} = 0, \hat{\omega}_{i,j}^{kk(m)} = 0).
\end{aligned}
$$

We then calculate the overall FPR, TPR, FDR, ACC, and MCC across all the M classes and K groups defined as:

$$
\begin{aligned}
FPR &= \frac{1}{KM}\sum_{m=1}^{M}\sum_{k=1}^{K} FPR^{kk(m)} = \frac{1}{KM}\sum_{m=1}^{M}\sum_{k=1}^{K} \frac{FP^{kk(m)}}{FP^{kk(m)} + TN^{kk(m)}};\\
TPR &= \frac{1}{KM}\sum_{m=1}^{M}\sum_{k=1}^{K} TPR^{kk(m)} = \frac{1}{KM}\sum_{m=1}^{M}\sum_{k=1}^{K} \frac{TP^{kk(m)}}{TP^{kk(m)} + FN^{kk(m)}};\\
FDR &= \frac{1}{KM}\sum_{m=1}^{M}\sum_{k=1}^{K} FDR^{kk(m)} = \frac{1}{KM}\sum_{m=1}^{M}\sum_{k=1}^{K} \frac{FP^{kk(m)}}{FP^{kk(m)} + TP^{kk(m)}};\\
ACC &= \frac{1}{KM}\sum_{m=1}^{M}\sum_{k=1}^{K} ACC^{kk(m)} = \frac{1}{KM}\sum_{m=1}^{M}\sum_{k=1}^{K} \frac{TP^{kk(m)} + TN^{kk(m)}}{p_k(p_k-1)/2}.
\end{aligned}
\tag{3.8}
$$

3.5.3 Simulation results

We compare our method with other two joint estimation methods (Guo et al., 2011; Danaher et al., 2014) in terms of the evaluation metrics described in Section 3.5.2. The following three approaches are compared:

- JMGGM: Our proposed method - the joint estimation of multilevel Gaussian graphical model;
- JMGM: Guo et al. (2011)'s joint estimation method;

- FGL: Danaher et al. (2014)'s joint estimation method.

We note that these two methods-JMGM and FGL do not consider multilevel networks structure, they only jointly estimate lower level Gaussian graphical models across classes.

We observe that the overall simulation results for Chain-Chain network and Chain-Scale-free network are similar to each other in terms of evaluations metrics. Hence, in the dissertation, we only provide the results for Chain-Scale-free network under different sample sizes and different heterogeneity levels. Evaluation for the higher level networks with $(n_1, n_2, n_3) = (200, 200, 200)$, $(n_1, n_2, n_3) = (100, 100, 100)$, and $(n_1, n_2, n_3) = (50, 50, 50)$ are summarized in Tables 3.2-3.4, whereas that for the lower level networks are displayed in Tables 3.5-3.7. Under each measurement, the best two methods are bolded. The best performed evaluation metrics for the estimated higher level network are displayed in Figures 3.2-3.7, while those for the lower level networks are shown in Figures 3.8-3.11. All the other tables and figures under different simulation settings are provided in the Supplementary Material.

Table 3.2: Simulation results for CSN higher level network when $n_m = 200$. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

ρ	method		FPR	TPR	FDR	ACC	GCDBias	Number of Zeros
0	FGL	Mean	1.0000	**1.0000**	0.6000	0.4000	0.0863	0.0000
		S.D.	0.0000	0.0000	0.0000	0.0000	0.0028	0.0000
	JMGGM	Mean	**0.0300**	0.9908	**0.0347**	**0.9783**	**0.0093**	**5.8567**
		S.D.	0.0477	0.0262	0.0537	0.0301	0.0009	0.3080
	JMGM	Mean	**0.8767**	**1.0000**	**0.5646**	**0.4740**	**0.0232**	**0.7400**
		S.D.	0.0764	0.0000	0.0238	0.0458	0.0014	0.4583
0.25	FGL	Mean	1.0000	**1.0000**	0.5000	0.5000	0.0889	0.0000
		S.D.	0.0000	0.0000	0.0000	0.0000	0.0033	0.0000
	JMGGM	Mean	**0.0280**	0.9527	**0.0238**	**0.9623**	**0.0096**	**5.0967**
		S.D.	0.0404	0.0477	0.0338	0.0324	0.0009	0.3007
	JMGM	Mean	**0.9080**	**1.0000**	**0.4735**	**0.5460**	**0.0259**	**0.4600**
		S.D.	0.0720	0.0000	0.0217	0.0360	0.0014	0.3601
0.75	FGL	Mean	1.0000	**1.0000**	0.3000	0.7000	0.0897	0.0000
		S.D.	0.0000	0.0000	0.0000	0.0000	0.0028	0.0000
	JMGGM	Mean	**0.0311**	0.7700	**0.0142**	**0.8297**	**0.0079**	**4.5167**
		S.D.	0.0549	0.0704	0.0256	0.0530	0.0009	0.5089
	JMGM	Mean	**0.8878**	**0.9962**	**0.2742**	**0.7310**	**0.0272**	**0.3633**
		S.D.	0.0941	0.0146	0.0229	0.0293	0.0016	0.3074

Table 3.3: Simulation results for CSN higher level network when $n_m = 100$. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

ρ	method		FPR	TPR	FDR	ACC	GCDBias	Number of Zeros
0	FGL	Mean	1.0000	**1.0000**	0.6000	0.4000	0.0427	0.0000
		S.D.	0.0000	0.0000	0.0000	0.0000	0.0025	0.0000
	JMGGM	Mean	**0.3528**	0.9867	**0.3182**	**0.7830**	**0.0083**	**3.9367**
		S.D.	0.1252	0.0329	0.0904	0.0753	0.0012	0.7723
	JMGM	Mean	**0.9972**	**1.0000**	**0.5993**	**0.4017**	**0.0368**	**0.0167**
		S.D.	0.0122	0.0000	0.0032	0.0073	0.0026	0.0730
0.25	FGL	Mean	1.0000	**1.0000**	0.5000	0.5000	0.0448	0.0000
		S.D.	0.0000	0.0000	0.0000	0.0000	0.0026	0.0000
	JMGGM	Mean	**0.3640**	0.9587	**0.2542**	**0.7973**	**0.0088**	**3.3867**
		S.D.	0.1384	0.0508	0.0743	0.0684	0.0015	0.7867
	JMGM	Mean	**0.9987**	**1.0000**	**0.4996**	**0.5007**	**0.0387**	**0.0067**
		S.D.	0.0094	0.0000	0.0026	0.0047	0.0031	0.0469
0.75	FGL	Mean	1.0000	**1.0000**	0.3000	0.7000	0.0432	0.0000
		S.D.	0.0000	0.0000	0.0000	0.0000	0.0025	0.0000
	JMGGM	Mean	**0.3378**	0.8228	**0.1367**	**0.7747**	**0.0076**	**3.2267**
		S.D.	0.1602	0.0824	0.0619	0.0716	0.0014	0.7841
	JMGM	Mean	**0.9989**	**0.9995**	**0.2999**	**0.7000**	**0.0384**	**0.0067**
		S.D.	0.0111	0.0048	0.0028	0.0047	0.0031	0.0469

Table 3.4: Simulation results for CSN higher level network when $n_m = 50$. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

ρ	method		FPR	TPR	FDR	ACC	GCDBias	Number of Zeros
0	FGL	Mean	0.9972	**1.0000**	0.5993	0.4017	0.0331	0.0167
		S.D.	0.0122	0.0000	0.0032	0.0073	0.0029	0.0730
	JMGGM	Mean	**0.3383**	0.7917	**0.3651**	**0.7137**	**0.0041**	**4.8033**
		S.D.	0.1142	0.1082	0.0928	0.0777	0.0009	0.8423
	JMGM	Mean	**0.8872**	**0.9917**	**0.5696**	**0.4643**	**0.0144**	**0.7100**
		S.D.	0.0826	0.0251	0.0262	0.0500	0.0018	0.5117
0.25	FGL	Mean	0.9973	**1.0000**	0.4993	0.5013	0.0344	0.0133
		S.D.	0.0131	0.0000	0.0036	0.0066	0.0029	0.0656
	JMGGM	Mean	**0.3600**	0.7700	**0.3007**	**0.7050**	**0.0044**	**4.3500**
		S.D.	0.1337	0.1048	0.0852	0.0703	0.0009	0.9737
	JMGM	Mean	**0.8833**	**0.9947**	**0.4670**	**0.5557**	**0.0148**	**0.6100**
		S.D.	0.0800	0.0182	0.0273	0.0427	0.0017	0.3937
0.75	FGL	Mean	0.9978	**0.9986**	0.2998	**0.6997**	0.0328	0.0167
		S.D.	0.0156	0.0082	0.0042	0.0075	0.0031	0.0730
	JMGGM	Mean	**0.3122**	0.6205	**0.1617**	0.6407	**0.0038**	**4.7200**
		S.D.	0.1392	0.0956	0.0709	0.0768	0.0008	0.8092
	JMGM	Mean	**0.8711**	**0.9714**	**0.2752**	**0.71867**	**0.01407**	**0.58667**
		S.D.	0.1091	0.0338	0.0284	0.0403	0.0016	0.4051

Table 3.5: Simulation results for CSN lower level network when $n_m = 200$. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

ρ	method		FPR	TPR	FDR	ACC	EL	FL	Number of Zeros
0	FGL	Mean	0.1806	**0.3320**	0.6760	0.7219	**0.0846**	**0.0164**	35.5093
		S.D.	0.0169	0.0430	0.0372	0.0154	0.0062	0.0012	0.7469
	JMGGM	Mean	**0.0214**	**0.0875**	**0.4708**	**0.8004**	0.1227	0.0231	**43.4420**
		S.D.	0.0066	0.0258	0.0853	0.0067	0.0087	0.0015	0.3595
	JMGM	Mean	**0.0212**	0.0867	**0.4597**	**0.8004**	**0.0916**	**0.0177**	**43.4560**
		S.D.	0.0066	0.0258	0.0900	0.0067	0.0067	0.0012	0.3620
0.25	FGL	Mean	0.1822	**0.3376**	0.6778	0.7218	**0.0836**	**0.0161**	35.4020
		S.D.	0.0179	0.0410	0.0409	0.0171	0.0062	0.0012	0.7123
	JMGGM	Mean	**0.0226**	**0.0885**	**0.5446**	**0.7996**	0.1281	0.0239	**43.3893**
		S.D.	0.0075	0.0235	0.1052	0.0072	0.0093	0.0016	0.3582
	JMGM	Mean	**0.0223**	0.0876	**0.5439**	**0.7997**	**0.0942**	**0.0181**	**43.4087**
		S.D.	0.0074	0.0231	0.1052	0.0072	0.0067	0.0012	0.3513
0.75	FGL	Mean	0.1739	**0.2950**	0.6970	0.7199	**0.0795**	**0.0153**	36.0860
		S.D.	0.0166	0.0350	0.0360	0.0151	0.0068	0.0013	0.6688
	JMGGM	Mean	**0.0220**	**0.0634**	**0.5917**	**0.7951**	0.1178	0.0219	**43.6360**
		S.D.	0.0055	0.0200	0.0118	0.0056	0.0096	0.0016	0.2840
	JMGM	Mean	**0.0219**	0.0630	**0.5973**	**0.7951**	**0.0899**	**0.0171**	**43.6453**
		S.D.	0.0056	0.0201	0.0275	0.0057	0.0078	0.0014	0.2820

Table 3.6: Simulation results for CSN lower level network when $n_m = 100$. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

ρ	method		FPR	TPR	FDR	ACC	EL	FL	Number of Zeros
0	FGL	Mean	0.0790	**0.1424**	**0.6885**	0.7653	**0.1384**	**0.0265**	40.8740
		S.D.	0.0120	0.0237	0.0586	0.0103	0.0096	0.0018	0.5002
	JMGGM	Mean	**0.0655**	**0.1161**	0.6908	**0.7709**	0.1713	0.0315	**41.5980**
		S.D.	0.0122	0.0260	0.0753	0.0109	0.0120	0.0020	0.5020
	JMGM	Mean	**0.0640**	0.1135	**0.6907**	**0.7715**	**0.1408**	**0.0271**	**41.6760**
		S.D.	0.0119	0.0261	0.0765	0.0107	0.0096	0.0018	0.4933
0.25	FGL	Mean	0.0803	**0.1425**	**0.6848**	0.7643	**0.1401**	**0.0265**	40.8260
		S.D.	0.0116	0.0310	0.0615	0.0104	0.0116	0.0022	0.5374
	JMGGM	Mean	**0.0659**	**0.1198**	**0.6885**	**0.7712**	0.1766	0.0322	**41.5480**
		S.D.	0.0121	0.0314	0.0704	0.0106	0.0140	0.0024	0.5606
	JMGM	Mean	**0.0642**	0.1171	0.6898	**0.7720**	**0.1427**	**0.0272**	**41.6333**
		S.D.	0.0122	0.0303	0.0730	0.0105	0.0113	0.0022	0.5572
0.75	FGL	Mean	0.0747	**0.1207**	**0.7034**	0.7644	**0.1308**	**0.0246**	41.2247
		S.D.	0.0123	0.0319	0.0629	0.0105	0.0114	0.0021	0.5787
	JMGGM	Mean	**0.0639**	**0.1010**	0.7157	**0.7691**	0.1640	0.0298	**41.7920**
		S.D.	0.0117	0.0270	0.0710	0.0097	0.0134	0.0023	0.5303
	JMGM	Mean	**0.0627**	0.0993	**0.7143**	**0.7697**	**0.1352**	**0.0255**	**41.8500**
		S.D.	0.0113	0.0273	0.0721	0.0095	0.0117	0.0022	0.5205

Table 3.7: Simulation results for CSN lower level network when $n_m = 50$. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

ρ	method		FPR	TPR	FDR	ACC	EL	FL	Number of Zeros
0	FGL	Mean	0.0666	**0.0935**	0.7294	0.7655	**0.2307**	**0.0459**	41.7627
		S.D.	0.0099	0.0252	0.0791	0.0085	0.0239	0.0059	0.4594
	JMGGM	Mean	**0.0216**	**0.0344**	*NaN*	**0.7896**	0.2652	0.0497	**43.9140**
		S.D.	0.0065	0.0155	*NA*	0.0052	0.0282	0.0059	0.3046
	JMGM	Mean	**0.0213**	0.0341	*NaN*	**0.7898**	**0.2422**	**0.0465**	**43.9287**
		S.D.	0.0063	0.0155	*NA*	0.0051	0.0256	0.0057	0.3009
0.25	FGL	Mean	0.0689	**0.0993**	0.7300	0.7647	**0.2315**	**0.0454**	41.6247
		S.D.	0.0129	0.0274	0.0804	0.0111	0.0220	0.0052	0.5496
	JMGGM	Mean	**0.0226**	**0.0384**	*NaN*	**0.7896**	0.2706	0.0500	**43.8400**
		S.D.	0.0069	0.0184	*NA*	0.0066	0.0244	0.0047	0.2981
	JMGM	Mean	**0.0223**	0.0377	*NaN*	**0.7897**	**0.2463**	**0.0465**	**43.8580**
		S.D.	0.0069	0.0187	*NA*	0.0066	0.0227	0.0048	0.3043
0.75	FGL	Mean	0.0643	**0.0853**	0.7698	0.7656	**0.2247**	**0.0444**	41.9167
		S.D.	0.0115	0.0252	0.0626	0.0104	0.0248	0.0059	0.4792
	JMGGM	Mean	**0.0208**	**0.0318**	*NaN*	**0.7897**	0.2593	0.0481	**43.9647**
		S.D.	0.0074	0.0153	*NA*	0.0067	0.0273	0.0056	0.3028
	JMGM	Mean	**0.0205**	0.0315	*NaN*	**0.7899**	**0.2384**	**0.0452**	**43.9773**
		S.D.	0.0074	0.0154	*NA*	0.0066	0.0258	0.0056	0.3012

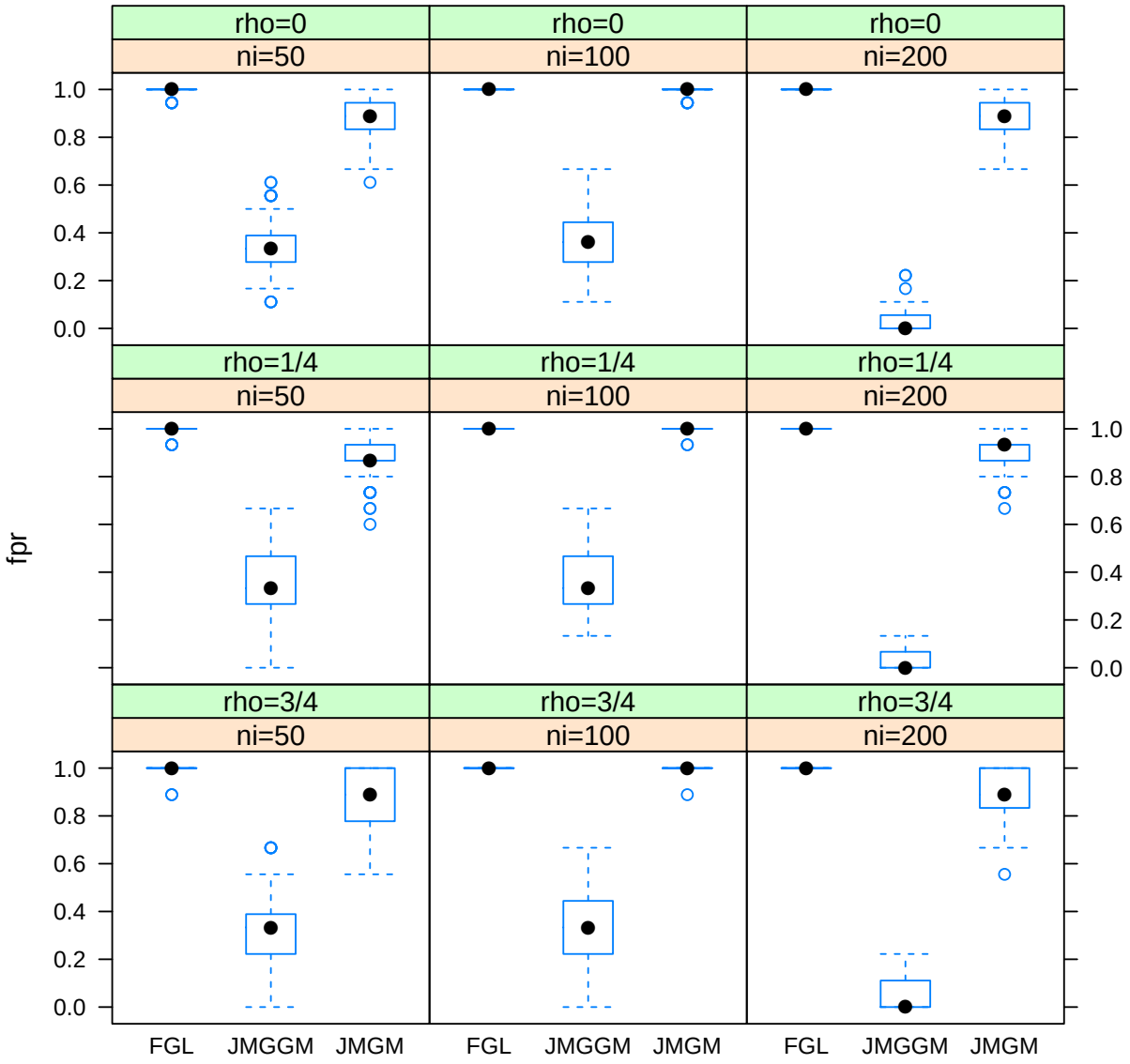

Figure 3.2: FPR Comparison for CSN higher level network. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

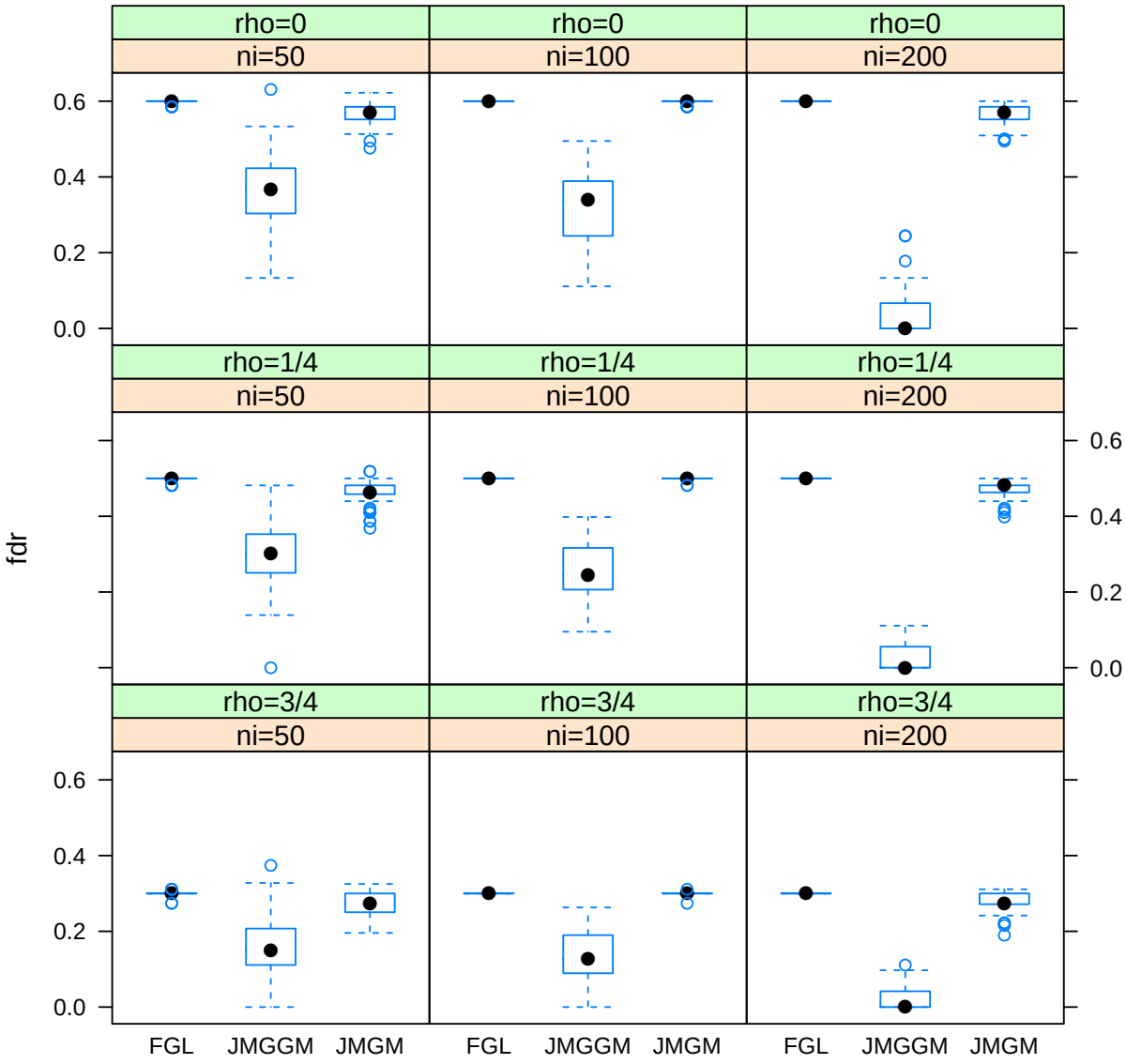

Figure 3.3: FDR Comparison for CSN higher level network. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

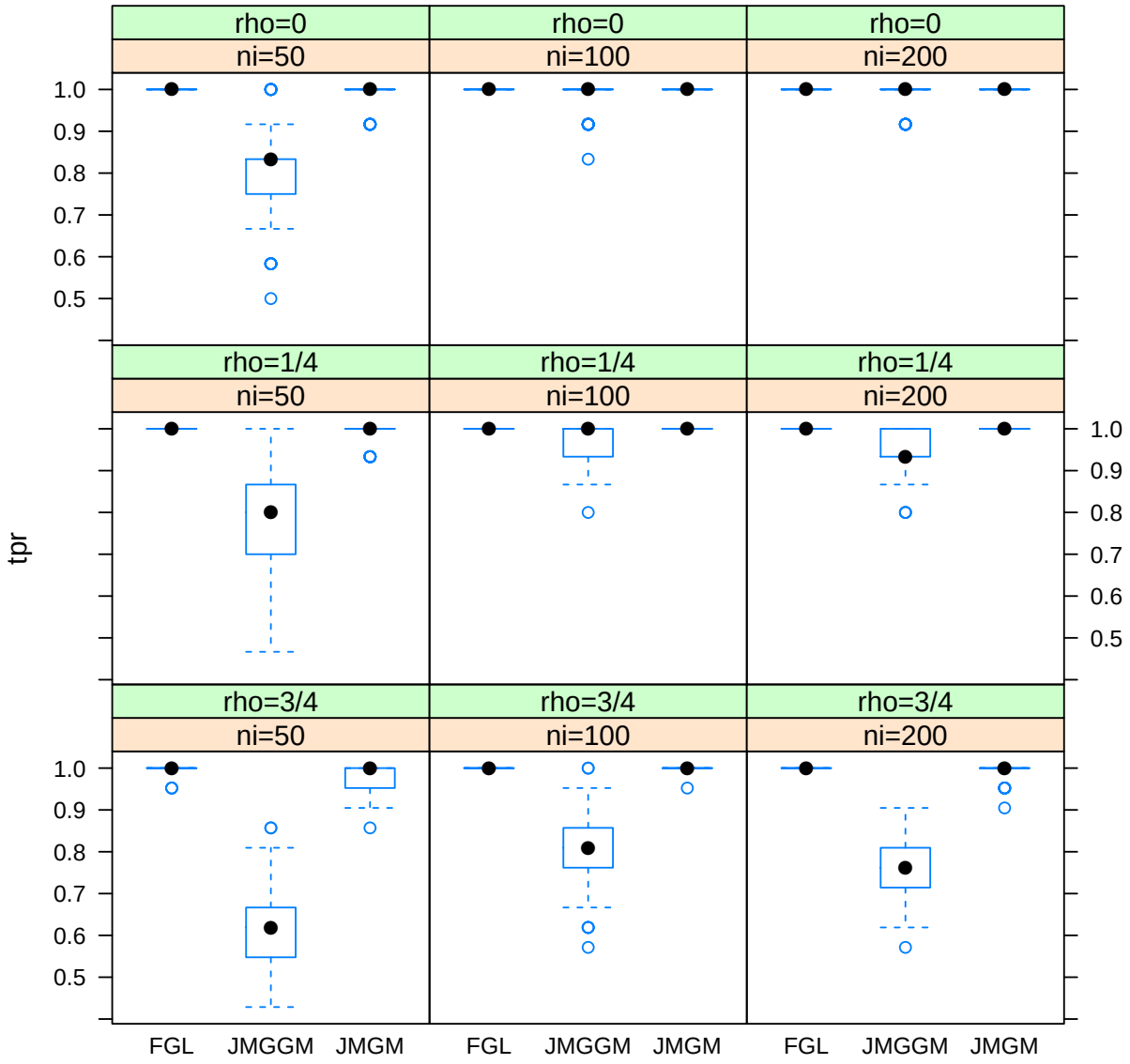

Figure 3.4: TPR Comparison for CSN higher level network. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

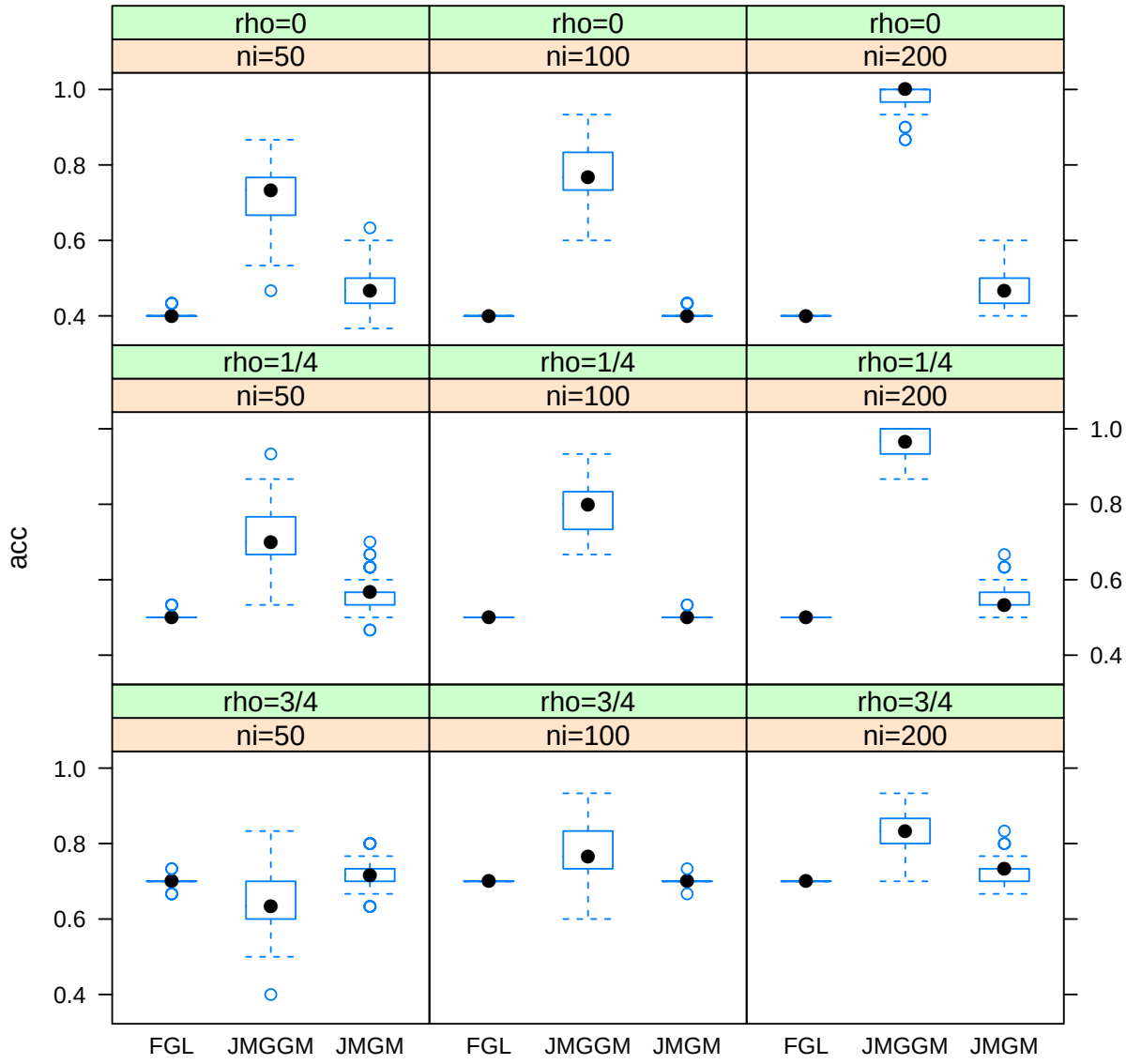

Figure 3.5: ACC Comparison for CSN higher level network. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

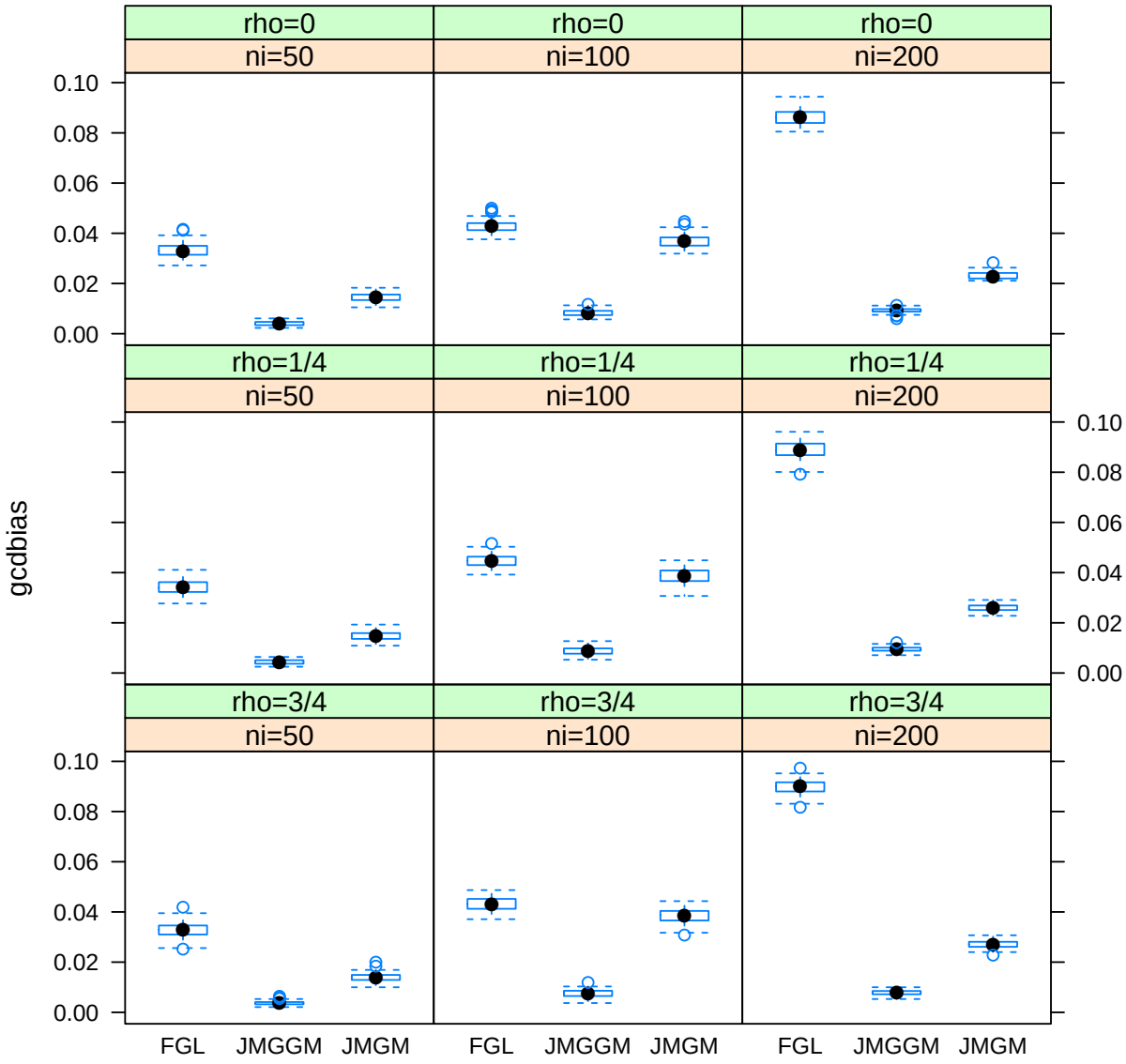

Figure 3.6: GCDBias Comparison for CSN higher level network. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

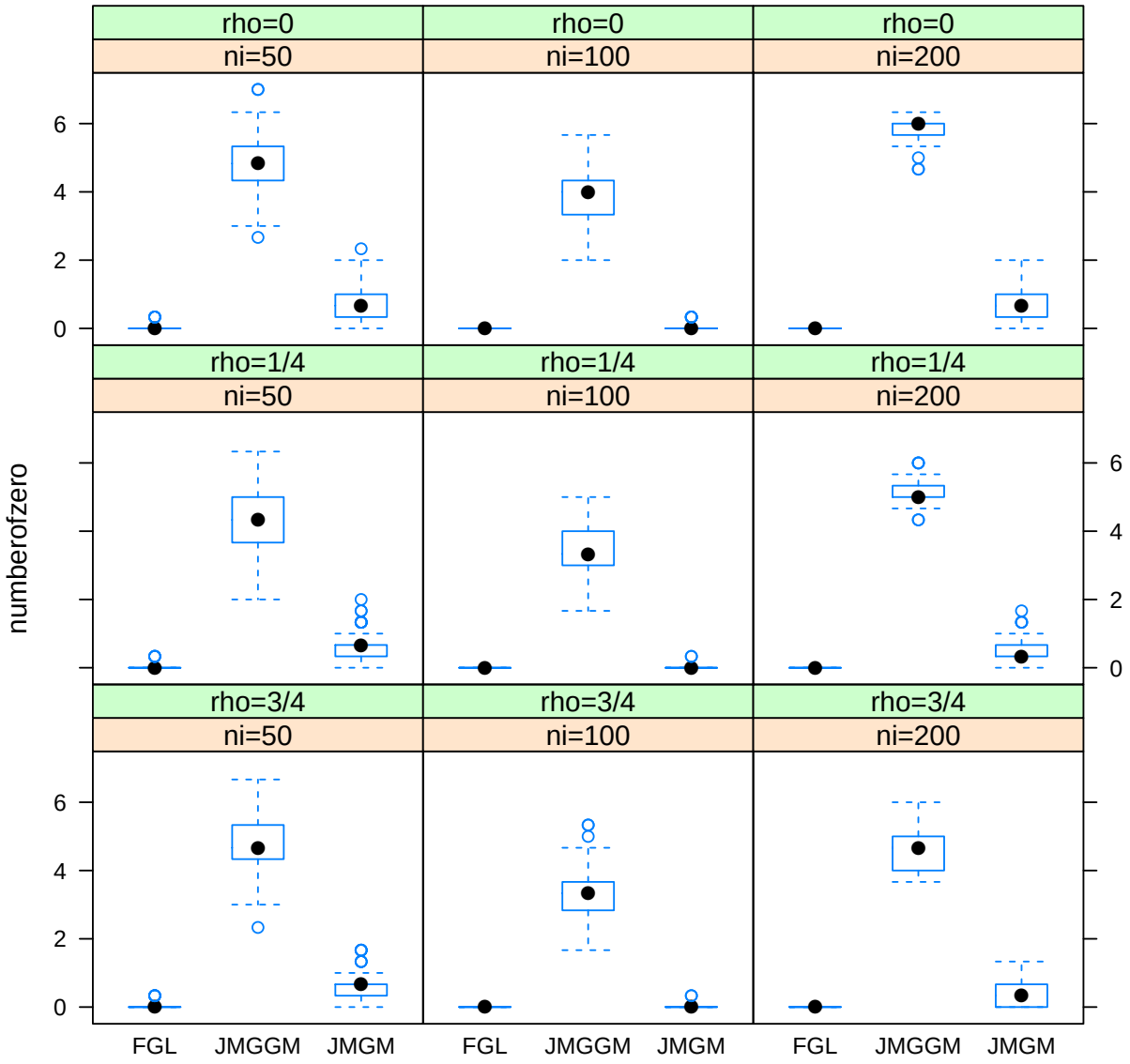

Figure 3.7: Sparsity Comparison for CSN higher level network. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

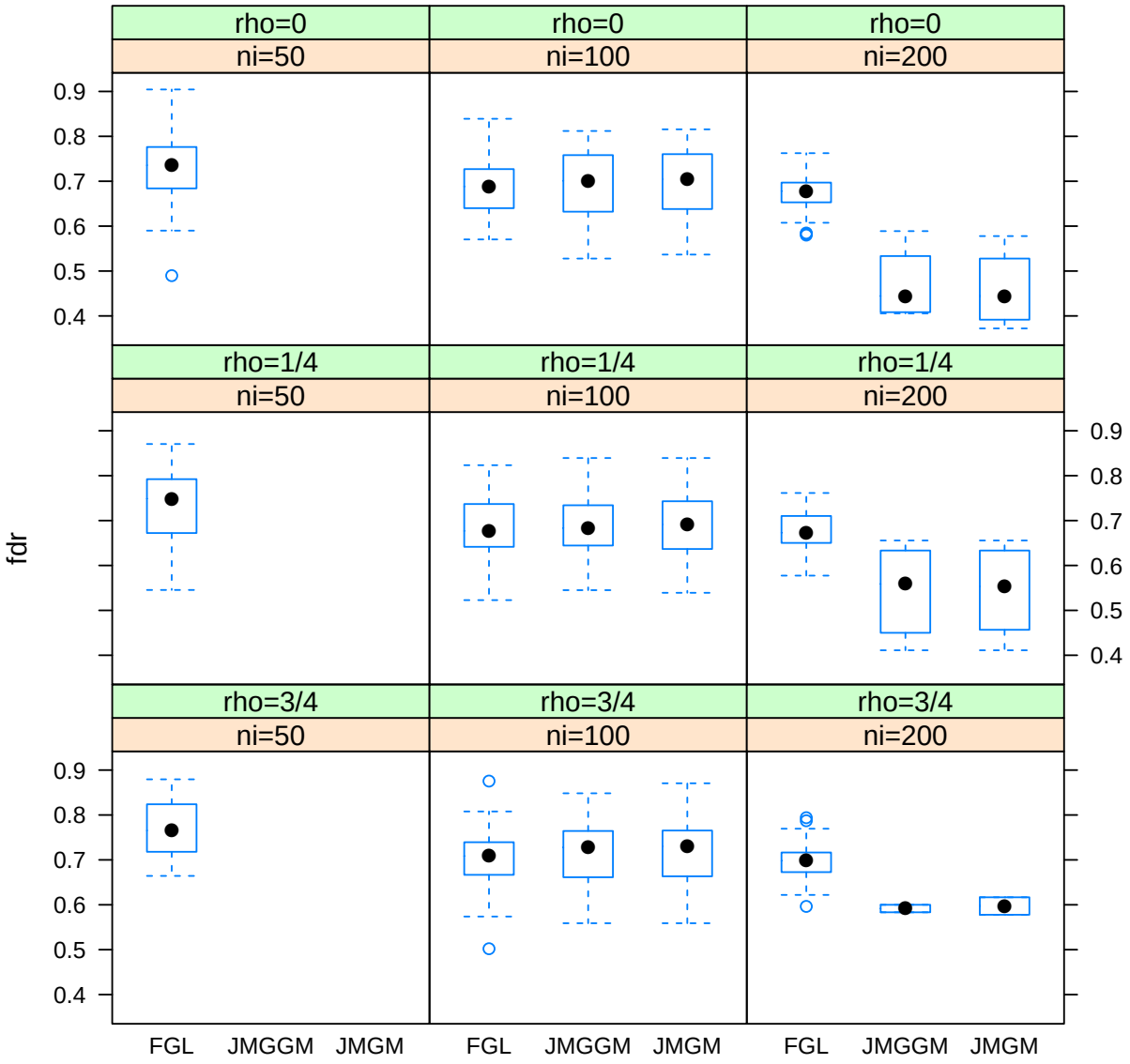

Figure 3.8: FDR Comparison for CSN lower level network. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

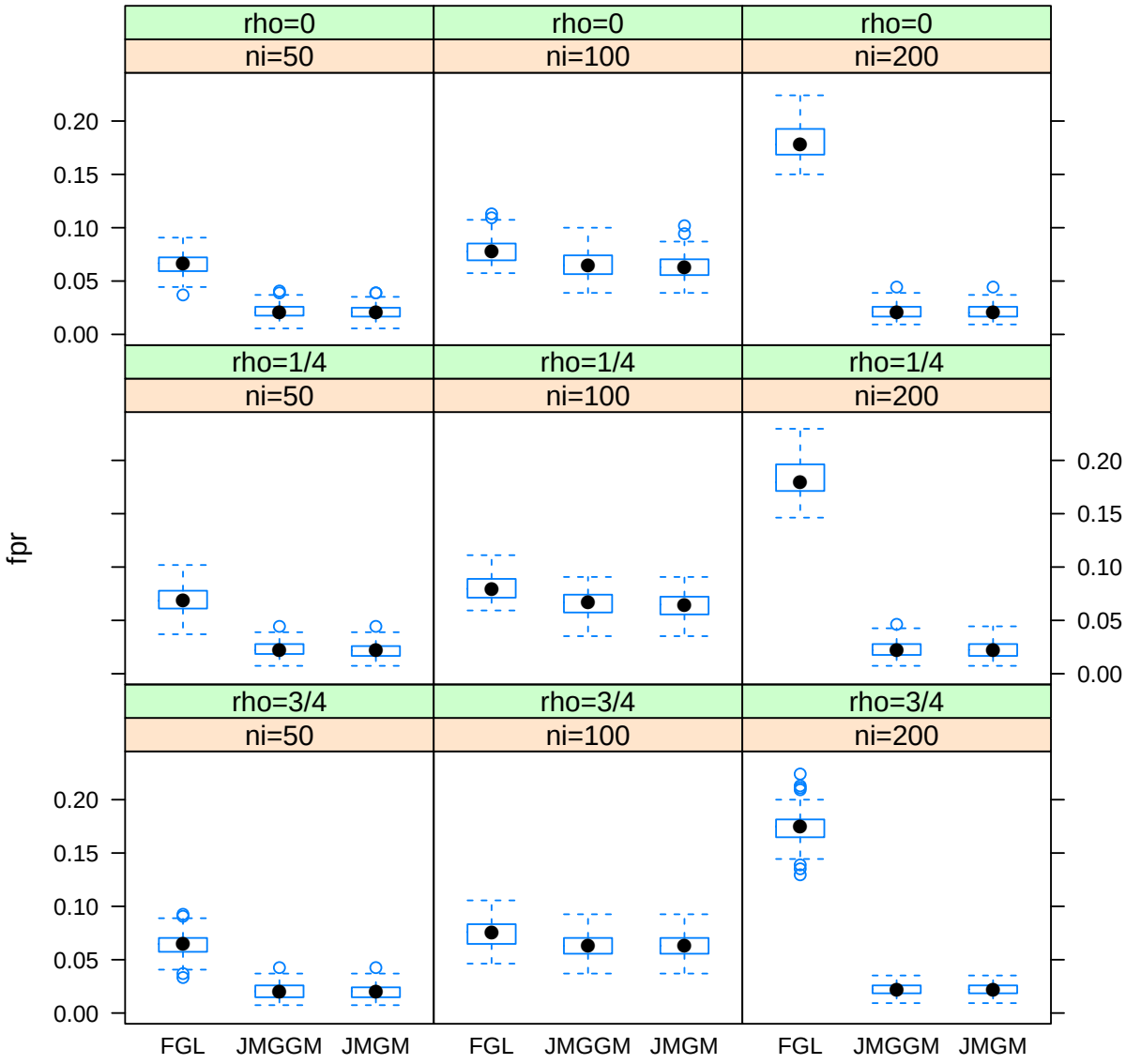

Figure 3.9: FPR Comparison for CSN lower level network. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

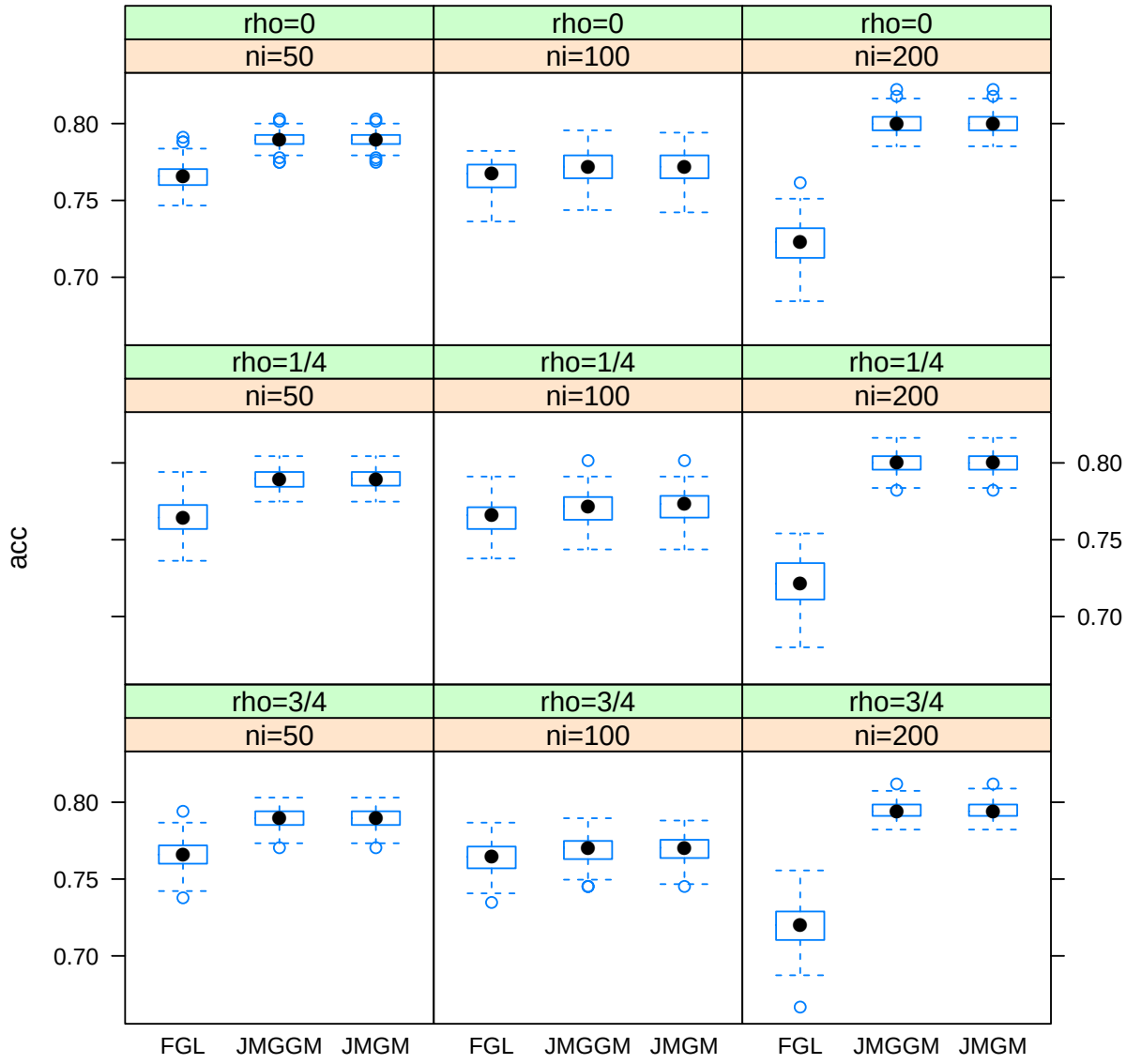

Figure 3.10: ACC Comparison for CSN lower level network. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

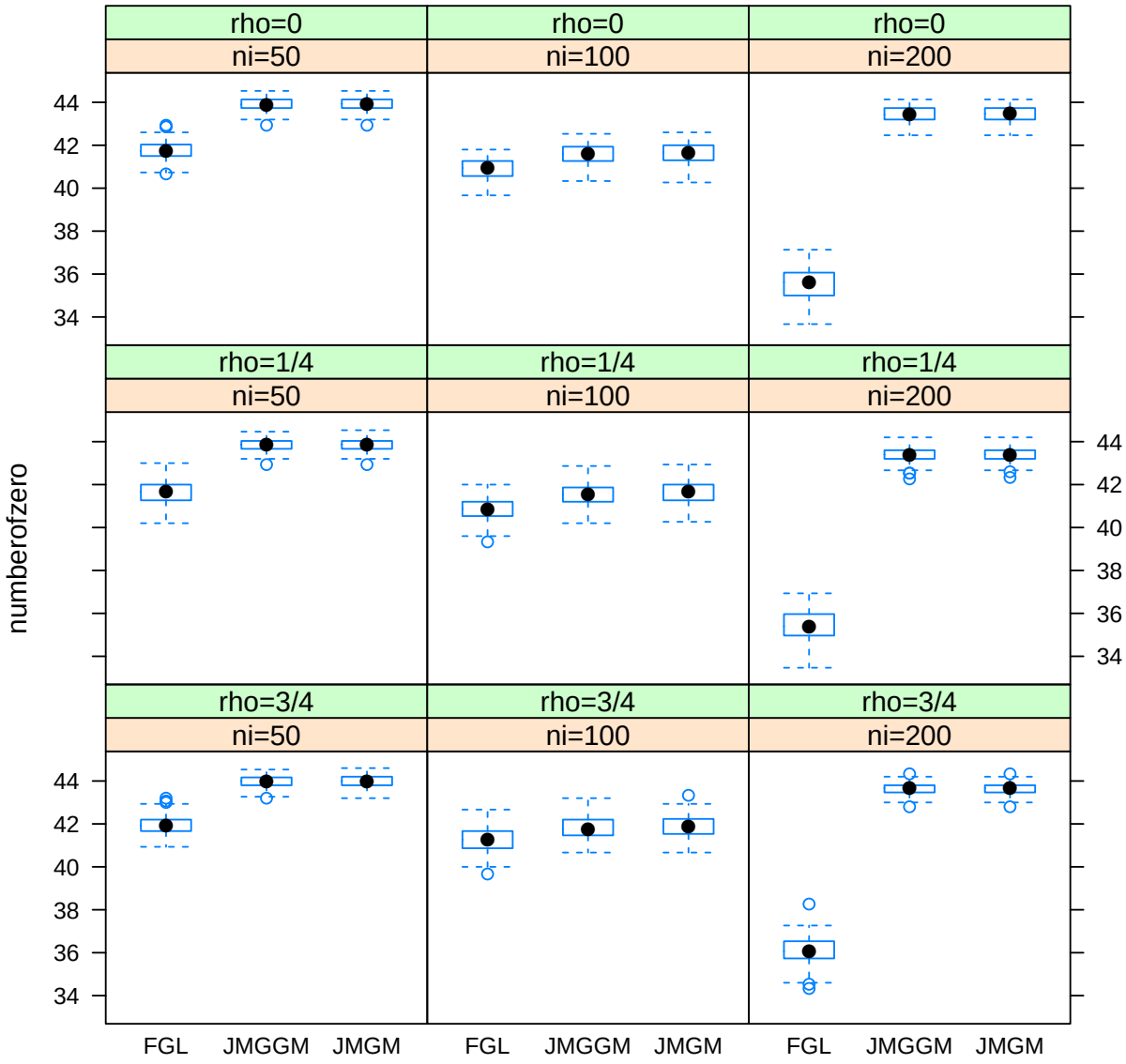

Figure 3.11: Sparsity Comparison for CSN lower level network. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method; FGL = Danaher et al. (2014)'s joint estimation method.

Specifically, for the higher level network estimation, JMGGM discovered more disconnections, and it obtains the lowest FPR, FDR, and GCDBias in all situations. In addition, its ACC is the highest and TPR is comparable to others when sample sizes are adequate or heterogeneity is moderate. For the lower level network estimation, it can be shown that JMGGM is comparable to the others in terms of information losses (EL and FL). Regarding network structure estimation, the sparsity level of that generated by JMGGM and JMGM are similar and they are higher than FGL. Moreover, it is comparable to JMGM and better than FGL in FPR, FDR, ACC.

In summary, our simulation results suggest that for the higher level network, JMGGM generates sparser networks, and it outperforms JMGM and FGL in terms of FPR, FDR, ACC, and GCDBias. For the lower level network, JMGGM performs better or similarly as JMGM and FGL in terms of all measured aspects. In addition, our JMGGM has the ability to control the sparsity on different levels individually, which is more flexible than JMGM and FGL since they only consider the lower level.

3.6 Real Data Analysis

In this section, we apply our JMGGM to the breast cancer gene expression data as described in Section 3.6.1. We then describe the application results in Section 3.6.2.

3.6.1 The gene expression data of white and nonwhite breast cancer patients

In the United States, it was reported that White women were slightly more likely to develop breast cancer than African American, Hispanic, and Asian women (Chlebowski et al., 2005).

Keenan et al. (2015) identified genomic differences between breast tumors of African American and that of white women, which may contribute to racial disparity in breast cancer death rate.

Hence our main question of interest is to explore the genetic difference among racial groups in terms of the multilevel network, which contains pathway network (how pathways interact with one another) and gene network within pathways (how genes interact with one another within a pathway). It is known that gene based analysis is not able to detect subtle change in expression level of individual genes (Mootha et al., 2003). Instead, pathway based analysis could catch that because genes within a pathway usually change coordinately. Furthermore, exploring how genes interact with one another within a pathway could help reveal the processes involved, and thus driver genes may be identified. Pathways are not isolated, they interact with one another either through shared genes or via regulatory mechanisms (Huang and Li, 2010; Liu et al., 2012; Dutta et al., 2012; Ponzoni et al., 2014; Creixell et al., 2015).

The human breast cancer data set was collected from the University of Texas M.D. Anderson Cancer Center (Shi et al., 2010), which contains 22283 genes expression measurements across 176 white patients and 102 non-white patients. Furthermore, the genes were mapped into 1320 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 proposed method - JMGGM to the top 5 pathways which are differently expressed between white and non-white breast cancer patients. These pathways are statistically significant using hybrid omnibus test (Xu, 2014), and they are P_{956} (REACTOME_DOWNSTREAM_SIGNAL_TRANSDUCTION), P_{943} (REACTOME_SIGNALING_BY_PDGF), P_{113} (KEGG_FOCAL_ADHESION), P_{141} (KEGG_REGULATION_OF_ACTIN_CYTOSKELETON), and P_{772} (REACTOME_SIGNALING_BY_FGFR_IN_DISEASE). Thus in this data set we have $M = 2$ and $K = 5$. The number of genes in the 5 pathways are $p_{P_{956}} = 3$, $p_{P_{943}} = 22$, $p_{P_{113}} = 101$,

$p_{P_{141}} = 111$, and $p_{P_{772}} = 30$ individually, and the total number of unique genes across the 5 pathways is 197. The selected pathways are ordered in the sequence of P_{956}, P_{943}, P_{113}, P_{141}, and P_{772}, so that adjacent pathways have overlapped genes and non-adjacent pathways have no overlaps. For any pair of adjacent pathways, overlapped genes are laid out in between the pathways. We then standardize the gene expression data to have mean 0 and standard deviation 1 within each class.

3.6.2 Application results

To explore the multilevel network, we used all the observations (176 white patients and 102 non-white patients) across the normalized genes from the first five pathways (P_{956}, P_{943}, P_{113}, P_{141}, P_{772}).

On the pathway level, four links for non-white patients and three links for white patients are identified via our method. Three out of the four links are common, while the link between P_{943} and P_{141} only occurs for non-white patients. We could clearly identify the difference from Figure 3.12, which demonstrate the estimated multilevel networks by our method. Especially, when we look into the details about the connection between P_{943} (REACTOME_SIGNALING_BY_PDGF) and P_{141} (KEGG_REGULATION_OF_ACTIN_CYTOSKELETON), it was found that was basically from connections between genes $COL4A4$ and $PIP4K2C$ and between genes $COL4A4$ and $FGF14$. According to the Broad Institute TCGA Genome Data Analysis Center (2013), these genes all belong to the top ranked mutated gene sets for breast cancer patients. Wang et al. (2015) identified $PIP4K2C$ to be one of the biomarkers to detect Basal-like breast cancer (BLBC). Moreover, it was suggested that $FGF14 - AS2$ is involved in breast cancer progress and may act as a tumor suppressor gene (Jia et al., 2013; Yang et al., 2016). On the other hand, other methods all ignore pathway information,

so no pathway network could be inferred from them.

On the gene network within pathway level, Table 3.8 displays the number of detected gene connections within each pathway via our method and Guo et al. (2011)'s method, and it can be shown that both of the two methods generate similar amount of gene connections within each pathway, and the generated gene networks within pathways are sparse. We only compare our result with that generated by Guo et al. (2011) because it shows from the simulation study in Section 3.5 that Guo et al. (2011)'s method generally performs better than that of Danaher et al. (2014).

Table 3.8: Summary of the generated gene network within each pathway by JMGGM and JMGM. JMGGM = Our proposed method - the joint estimation of multilevel Gaussian graphical model; JMGM = Guo et al. (2011)'s joint estimation method.

Class	Pathway	# of Genes	# of Possible Links	# of Links Detected by Method	
				JMGGM	JMGM
Nonwhite	P956	3	3	0	0
	P943	22	231	34	32
	P113	101	5050	400	386
	P141	111	6105	491	475
	P772	30	435	64	36
White	P956	3	3	0	0
	P943	22	231	38	36
	P113	101	5050	319	353
	P141	111	6105	355	430
	P772	30	435	37	30

To evaluate how the estimated precision matrices performs in the classification of breast cancer patients, we do cross validation. Basically, we randomly divide the data into training set and testing set of sizes 263 and 15, individually, and repeat the process 100 times. To assure that the class unbalance is are similar across the training and testing sets, we used a stratified sampling. Each time we randomly selected 167 subjects from the white breast cancer patients group and 96 from the nonwhite breast cancer patients group, so that the training set consists of those 263 patients. The remaining subjects were used as the testing

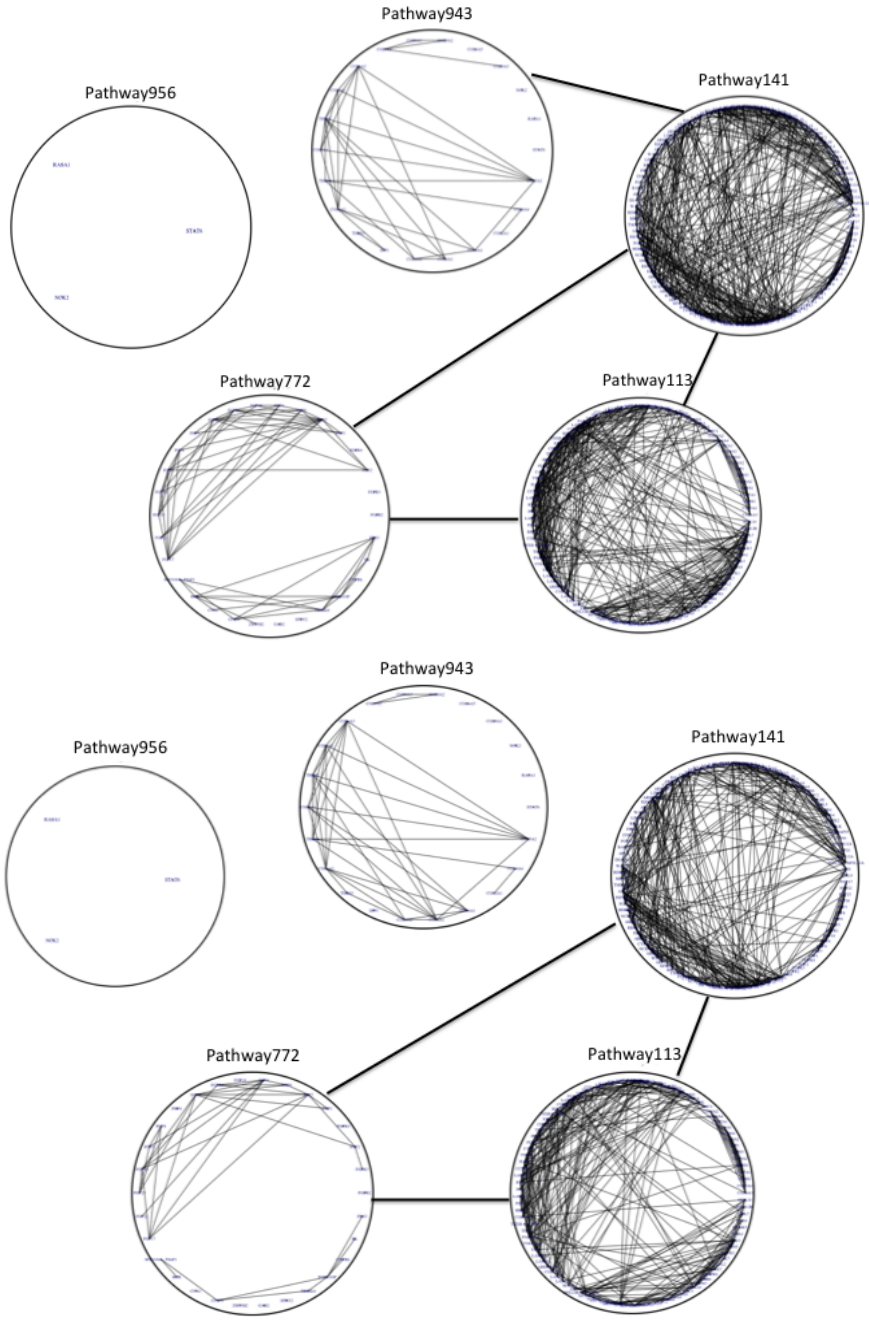

Figure 3.12: Estimated multilevel networks for nonwhite and white breast cancer patients by JMGGM - our proposed method. Big circles represent pathways and connections among big circles represent pathway network. Connections within big circles represent gene network within pathways. The upper penal represents the estimated multilevel network for nonwhite breast cancer patients, and the right represents that for white breast cancer patients.

set.

We estimated the precision matrices using our proposed method - JMGGM. The quadratic discriminant analysis (QDA) is used for classification. The QDA assumes the normalized gene expression data in class m follows MN $(\underline{\mu}^m, (\Omega^{(m)})^{-1})$, where $m = 1, 2$. The quadratic discriminant scores for observation $\underline{x}$ are defined as:

$$\delta_m(\underline{x}) = -\frac{1}{2}\log|(\hat{\Omega}^{(m)})^{-1}| - \frac{1}{2}(\underline{x} - \underline{\hat{\mu}}^{(m)})'\hat{\Omega}^{(m)}(\underline{x} - \underline{\hat{\mu}}^{(m)}) + \log\hat{\pi}_m,$$

where $\hat{\pi}_m = n_m/n$, $\underline{\hat{\mu}}^m = (\sum_{i \in class-m} \underline{x}_i)/n_m$, $m = 1, 2$ and $i = 1, \ldots, n_m$. If $\delta_1(\underline{x}) > \delta_2(\underline{x})$, we say observation $\underline{x}$ belongs to class 1.

Given the estimated multilevel network as displayed in Figure 3.12, we could see that the major difference between the two groups is whether there is a connection between P_{943} and P_{141}. Therefore, we only keep the genes in this two pathways ($p = 133$), and conduct the cross validation again. We calculate accuracy (ACC) which is defined as:

$$ACC = \frac{\text{number of true negatives} + \text{number of true positives}}{\text{number of samples in the testing set}}.$$

The estimated ACC was 0.61 based on the means of the 100 replications, with all the 5 pathways and with only P_{943} and P_{141}, separately. Therefore, we could see that dimension reduction is achieved by our method, and the classification performance keeps robust using the precision matrices estimates from our method.

3.7 Discussion

In this chapter, we have proposed a method to jointly estimate the multilevel Gaussian graphical model across classes. A sparse estimate of the precision matrix could be mapped into a sparse higher level network and a sparse lower level network. On the other hand, common structure in terms of the multilevel network is being shared during the estimation procedure, so that the common multilevel network is estimated more precisely, and the unique multilevel structures are retained as well. Current joint estimation methods only consider a single level network structure, and sparse higher level network could rarely be estimated. However, our method has the flexibility of controlling sparsity with regard to both the higher level and the lower level. Simulation studies show that our approach outperforms other methods under a set of simulated networks. Real data application also shows the advantage of our method.

We note that in our simulation, the higher level network is chain network. We may further perform simulation study to explore the scenario where the higher level network is scale-free network. In the real data application, we found some pathway networks and gene networks within pathways which were never detected from other methods. These results need to be further investigated to assess the biological meanings and need to be validated experimentally as the ultimate test of our proposal methods.

Our current model applies to a specific overlapping scenario, where only adjacent groups have overlaps and non-adjacent groups have nothing in common. We may think of dealing with more general cases in our future research. One possible idea is to employ the hierarchical lasso idea (Zhou and Zhu, 2010) in the context of multivariate regression, so that we may have an approximation to the exact optimization problem regarding the multilevel Gaussian graphical models. Secondly, tuning parameters' selection is still an open question for esti-

mating graphical models, while current proposed methods only consider selection criterion for a few simple cases (Foygel and Drton, 2010). Considering that and data's hierarchical structure, we would like to develop more appropriate model selection criterion or address the problem under Bayesian framework. Thirdly, extending our method to the scenario where variables are discrete will allow us to estimate networks for discrete variables. For instance, Allen and Liu (2012) proposed a method to estimate Poisson graphical models, so that networks based on gene sequencing data could be estimated. Adding pathway information, we may use counts of sequencing reads for genes to infer gene network and pathway network, which may provide another perspective to look at the conditional dependencies among genes and pathways. Moreover, our model is based on the assumption that there is common network structure among classes regarding the multilevel network, how to validate those assumptions deserves further effort. Lastly, we currently provide estimates of precision matrices, making inferences based on our estimates will be our next step.

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

Gaussian graphical model is becoming a popular tool to investigate conditional dependency between random variables by estimating sparse precision matrices. The estimated precision matrices could be mapped into networks for visualization. For heterogeneous classes, joint estimation of Gaussian graphical models could take advantage of common structure across classes, so that the common structure is estimated more accurately, and yet unique structures are retained as well. Furthermore, there may exist multilevel structure among variables; some variables are considered as higher level variables and others are nested in these higher level variables, which are called lower level variables. In this dissertation, we made several contributions to the area of joint estimation of Gaussian graphical models across heterogeneous classes: we have firstly proposed a joint estimation method for estimating Gaussian

graphical models across unbalanced multi-classes, whereas the second considered multilevel variable information during the joint estimation procedure and simultaneously estimates higher level network and lower level network.

In Chapter 2, we considered the problem of jointly estimating Gaussian graphical models across unbalanced multi-class. Most existing methods require equal or similar sample size among classes. However, many real applications do not have similar sample sizes. Our joint adaptive graphical lasso approach combines information across classes so that their common characteristics can be shared during the estimation process. We also introduce regularization into the adaptive term so that not only the tuning parameters for every class are different, but also the tuning parameters for each element within a class. By doing it this way, we are able to prevent the majority class from dominating the estimated precision matrix result. Our approach is more flexible than the approach of Guo et al. (2011) because their tuning parameters for every class are exactly the same. Simulation studies show that our approach performs better than existing methods in terms of false positive rate, accuracy, Mathews correlation coefficient, and false discovery rate. Moreover, we confirmed through simulation study that common structure is estimated more accurately than separate estimation and Guo et al. (2011)'s joint estimation method. We demonstrate the advantage of our approach using liver cancer data set analyzed by Chen et al. (2002) and de Souto et al. (2008).

In Chapter 3, we proposed a method to jointly estimate the multilevel Gaussian graphical models across multiple classes. Currently, methods are still limited to investigate a single level conditional dependency structure when there exists the multilevel structure among variables. Due to the fact that higher level variables may work together to accomplish certain tasks, simultaneously exploring conditional dependency structures among higher level variables and among lower level variables are of our main interest. Given multilevel data from heterogeneous classes, our method assures that common structures in terms of the multilevel

conditional dependency are shared during the estimation procedure, yet unique structures for each class are retained as well. Our proposed approach is achieved by first introducing a higher level variable factor within a class, and then common factors across classes. The performance of our approach is evaluated on several simulated networks: chain-chain network and chain-scale free network. Our simulation results suggest that for the higher level network, our method generates sparser networks, and it outperforms the method of Guo et al. (2011) and Danaher et al. (2014) in terms of false positive rate, false discovery rate, accuracy, and group connection degree bias. For the lower level network, our performs better or similarly as Guo et al. (2011) and Danaher et al. (2014) in terms of all measured aspects. In addition, our method has the ability to control the sparsity on different levels individually, which is more flexible than Guo et al. (2011) and Danaher et al. (2014) since they only consider a single level. We also demonstrate the advantage of our approach using the breast cancer patient data: our method is capable of discovering both sparse pathway network and sparse gene network within pathways across heterogeneous classes. Because of the sparsity of the pathway network, we may indirectly accomplish dimension reduction, and the classification performance based on QDA keeps robust.

4.2 Future Work

The first topic - joint estimation of Gaussian graphical models across unbalanced classes can be further extended in the following ways:

- Our current study only provided the asymptotic properties where $n \to \infty$ and p is fixed. Hence, we need to further develop the asymptotic properties of our approach when $n/log(p) \to \infty$ in a future research.

- Tuning parameters selection for Gaussian graphical model estimation is an open question. It becomes even harder when we have large p small n, and the network structure is very complex (e.g. Scale-free network). Current proposed methods only consider selection criterion for a few simple cases. For instance, Foygel and Drton (2010) developed the extended BIC criterion for Gaussian graphical models by adding a term to describe model complexity. However, the added term cannot tell the difference between chain network and scale free network when they have the same number of links.

- In our current research, we applied the adaptive LASSO penalty to correct the bias. However, one drawback of the adaptive LASSO penalty is the requirement of a consistent initial estimate, which is really hard to get for large p small n case (Fan et al., 2009). Applying the SCAD penalty (Fan and Li, 2001) in the joint estimation procedure is an alternative that are worthy of trying.

The second topic - joint estimation of the multilevel Gaussian graphical models across heterogeneous classes can be further extended in the following ways:

- In our simulation, the higher level network is chain network. We may further perform simulation study to explore the scenario where the higher level network is more complex, for instance, scale-free network.

- In the real data application, we found some pathway networks and gene networks within pathways which were never detected from other methods. These results need to be further investigated to assess the biological meanings and need to be validated experimentally as the ultimate test of our proposal methods.

- Our current model applies to a specific overlapping scenario, where only adjacent groups have overlaps and non-adjacent groups have nothing in common. We may

think of dealing with more general cases in our future research. One possible idea is to employ the hierarchical lasso idea (Zhou and Zhu, 2010) in the context of multivariate regression, so that we may have an approximation to the exact optimization problem regarding the multilevel Gaussian graphical models.

- Tuning parameters' selection is still an open question for estimating graphical models, while current proposed methods only consider selection criterion for a few simple cases (Foygel and Drton, 2010). Considering the data's multilevel structure, we may further developing BIC criterion or Extended BIC criterion that incorporates the multilevel structure, and we may address the problem under Bayesian framework.

- Extending our method to the scenario where variables are discrete will allow us to estimate networks for discrete variables. For instance, Allen and Liu (2012) proposed a method to estimate Poisson graphical models, so that networks based on gene sequencing data could be estimated. Adding pathway information, we may use counts of sequencing reads for genes to infer gene network and pathway network, which may provide another perspective to look at the conditional dependencies among genes and pathways.

- Our model is based on the assumption that there is common network structure among classes regarding the multilevel network, how to validate those assumptions deserves further effort. One possible approach is that we may develop some measurements like heterogeneity, and make inference on heterogeneity based on our estimates, and choose appropriate estimation method based on the inference accordingly. For instance, if the heterogeneity is not significantly different from 0, that may indicate the network structures across classes are quite similar to one another, thereafter, pooling all the observations into one group and estimate one precision matrix will suffice.

- Currently, we only provide estimates of precision matrices, but how to make inference based on our estimates will be our next step.

Bibliography

Allen, G. I. and Liu, Z. (2012). A log-linear graphical model for inferring genetic networks from high-throughput sequencing data. In *Bioinformatics and Biomedicine (BIBM), 2012 IEEE International Conference on*, pages 1–6.

Barabási, A.-L. and Albert, R. (1999). Emergence of Scaling in Random Networks. *Science*, 286:509–512.

Barabási, A.-L. and Bonabeau, E. (2003). Scale-Free Networks. *Scientific American*, pages 50–59.

Broad Institute TCGA Genome Data Analysis Center (2013). Breast Invasive Carcinoma (Primary solid tumor cohort) - 21 April 2013: Mutation Analysis (MutSig v2.0).

Chen, X. et al. (2002). Gene expression patterns in human liver cancers. *Molecular biology of the cell*, 13(6):1929–1939.

Chlebowski, R. T. et al. (2005). Ethnicity and breast cancer: Factors influencing differences in incidence and outcome. *Journal of the National Cancer Institute*, 97(6):439–447.

Creixell, P. et al. (2015). Pathway and network analysis of cancer genomes. *Nature Methods*, 12(7):615–621.

BIBLIOGRAPHY

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(2):373–397.

de Souto, M. C., Costa, I. G., de Araujo, D. S., Ludermir, T. B., and Schliep, A. (2008). Clustering cancer gene expression data: a comparative study. *BMC bioinformatics*, 9(1):1.

Dempster, A. P. (1972). Covariance Selection. *Biometrics*, 28(1):157–175.

Dutta, B., Wallqvist, A., and Reifman, J. (2012). Pathnet: a tool for pathway analysis using topological information. *Source code for biology and medicine*, 7(1):1.

Fan, J., Feng, Y., and Wu, Y. (2009). Network exploration via the adaptive LASSO and SCAD penalties. *Annals of Applied Statistics*, 3(2):521–541.

Fan, J. and Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Foygel, R. and Drton, M. (2010). Extended bayesian information criteria for gaussian graphical models. In *Advances in Neural Information Processing Systems*, pages 604–612.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

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

Haybaeck, J. et al. (2009). A Lymphotoxin-Driven Pathway to Hepatocellular Carcinoma. *Cancer Cell*, 16:295–308.

BIBLIOGRAPHY

He, N., Park, K., Zhang, Y., Huang, J., Lu, S., and Wang, L. (2008). Epigenetic Inhibition of Nuclear Receptor Small Heterodimer Partner Is Associated With and Regulates Hepatocellular Carcinoma Growth. *Gastroenterology*, 134:793–802.

Huang, Y. and Li, S. (2010). Detection of characteristic sub pathway network for angiogenesis based on the comprehensive pathway network. *BMC bioinformatics*, 11 Suppl 1:S32.

Jia, H., Lipovich, L., Ju, D., and Kosir, M. A. (2013). Identifying long noncoding rnas in breast cancer microarrays. *Cancer Research*, 73(8 Supplement):1842–1842.

Kaminsky, Y. and Kosenko, E. (2010). AMP deaminase and adenosine deaminase activities in liver and brain regions in acute ammonia intoxication and subacute toxic hepatitis. *Brain Research*, 1311:175–181.

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(31):3621–3627.

Korkmaz, S., Goksuluk, D., and Zararsiz, G. (2014). MVN: An R package for assessing multivariate normality. *The R Journal*, 6(2013):151–162.

Liu, K., Liu, Z., Hao, J., Chen, L., and Zhao, X. (2012). Identifying dysregulated pathways in cancers from pathway interaction networks. *BMC Bioinformatics*, 13(1):126.

Lu, C. et al. (2009). Aberrant dna methylation profile and frequent methylation of klk10 and oxgr1 genes in hepatocellular carcinoma. *Genes, Chromosomes and Cancer*, 48(12):1057–1068.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462.

BIBLIOGRAPHY

Mootha, V. K. et al. (2003). PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature genetics*, 34(3):267–273.

Ponzoni, I., Nueda, M. J., Tarazona, S., Götz, S., Montaner, D., Dussaut, J. S., Dopazo, J., and Conesa, A. (2014). Pathway network inference from gene expression data. *BMC systems biology*, 8(2):1.

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

Shi, L. et al. (2010). The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. *Nature biotechnology*, 28:827–38.

Szydowska, M. and Roszkowska, A. (2008). Expression patterns of AMP-deaminase isozymes in human hepatocellular carcinoma (HCC). *Molecular and Cellular Biochemistry*, 318:1–5.

Venook, A. P., Papandreou, C., Furuse, J., and Ladrón De Guevara, L. (2010). The Incidence and Epidemiology of Hepatocellular Carcinoma: A Global and Regional Perspective. *The Oncologist*, 15:5–13.

Wang, J., Figueroa, J. D., Wallstrom, G., Barker, K., Park, J. G., Demirkan, G., Lissowska, J., Anderson, K. S., Qiu, J., and LaBaer, J. (2015). Plasma autoantibodies associated with basal-like breast cancers. *Cancer Epidemiology Biomarkers and Prevention*, 24(9):1332–1340.

Xu, Y. (2014). *Frequentist-Bayesian Hybrid Tests in Semi-parametric and Non-parametric Models with Low/High-Dimensional Predictor*. PhD dissertation, Virginia Polytechnic Institute and State University, Department of Statistics.

BIBLIOGRAPHY

Yang, F., Liu, Y., Dong, S., Ma, R., Bhandari, A., Zhang, X., and Wang, O. (2016). A novel long non-coding RNA FGF14-AS2 is correlated with progression and prognosis in breast cancer. *Biochemical and Biophysical Research Communications*, 470(3):479–483.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35.

Zhou, N. and Zhu, J. (2010). Group variable selection via a hierarchical lasso and its oracle property. *Statistics and Its Interface*, 3(4):557–574.

Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics*, 36(4):1509–1533.