FLEXIBLE GRAPH-BASED LEARNING WITH APPLICATIONS TO GENETIC DATA ANALYSIS

Jianyu Liu

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2019

Approved by:

Yufeng Liu

Shankar Bhamidi

Wei Sun

Quoc Tran-Dinh

Kai Zhang

ABSTRACT

JIANYU LIU: Flexible Graph-based Learning with Applications to Genetic Data Analysis
(Under the direction of Yufeng Liu)

With the abundance of increasingly complex and high dimensional data in many scientific disciplines, graphical models have become an extremely useful statistical tool to explore data structures. In this dissertation, we study graphical models from two perspectives: i) to enhance supervised learning, classification in particular, and ii) graphical model estimation for specific data types. For classification, the optimal classifier is often connected with the feature structure within each class. In the first project, starting from the Gaussian population scenario, we aim to find an approach to utilize the graphical structure information of the features in classification. With respect to graphical models, many existing graphical estimation methods have been proposed based on a homogeneous Gaussian population. Due to the Gaussian assumption, these methods may not be suitable for many typical genetic data. For instance, the gene expression data may come from individuals of multiple populations with possibly distinct graphical structures. Another instance would be the single cell RNA-sequencing data, which are featured by substantial sample dependence and zero-inflation. In the second and the third project, we propose multiple graphical model estimation methods for these scenarios respectively. In particular, two dependent count-data graphical models are introduced for the latter case. Both numerical and theoretical studies are performed to demonstrate the effectiveness of these methods.

ACKNOWLEDGEMENTS

This dissertation would not have been completed without the great support of people who stood by me during my five years at UNC. I would like to thank all of them.

Firstly, I would like to express my sincere appreciation and gratitude to my advisor Professor Yufeng Liu for his continuous support of my Ph.D. research and career, for his patience, motivation, and immense knowledge. His guidance helped me throughout my research and writing of this thesis. He has always been supporting and encouraging me to make my own choices. I could not imagine having a better advisor and mentor for my Ph.D. study. I would also like to thank Professor Wei Sun for collaborating with me on the exciting projects and providing a lot of valuable suggestions on my thesis.

I would like to convey my sincere thanks to the rest of my thesis committee: Professor Shankar Bhamidi, Professor Quoc Tran-Dinh, and Professor Kai Zhang, for their time, support, guidance, and insightful comments on my dissertation.

Last but not least, I am very grateful to my friends, my parents, and my wife for supporting me spiritually throughout writing this thesis and my life in general.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

CIG	Conditional independence graph
DAG	Directed acyclic graph
EBIC	Extended Bayesian information criterion
GSLDA	Graph-based linear discriminant analysis
LDA	Linear discriminant analysis
QDA	Quadratic discriminant analysis
scRNA-seq	Single-cell RNA-sequencing
$\mathcal{G}$	An undirected graph
$\mathcal{D}$	A directed graph
$\mathcal{N}_j$	The neighborhood of predictor j in a graph
$\mathbb{R}^n$	Set of n-dimensional real valued vectors
$\mathbb{S}^n_{++}$	Set of n-dimensional positive definite matrices
$\mathbf{a}_{-i}$	The vector after removing the i-th element of the vector $\mathbf{a}$
$\|\mathbf{a}\|_2$	The ℓ_2 norm of the vector $\mathbf{a}$
$\|\mathbf{a}\|_\infty$	$\max\{\|a_1\|, \ldots, \|a_n\|\}$ if $\mathbf{a}$ is an n-dimensional vector
$\mathbf{A}_{i,\cdot}$	The i-th row vector of the matrix $\mathbf{A}$
$\mathbf{A}_j$	The j-th column vector of the matrix $\mathbf{A}$
$\mathbf{A}_{-j}$	The matrix after removing the j-th column of the matrix $\mathbf{A}$
$\|\|\mathbf{A}\|\|_\infty$	$\max_{1\le i\le k} \sum_{j=1}^m \|A_{ij}\|$ if $\mathbf{A}$ is a $k \times m$ matrix
$\|\|\|\mathbf{A}\|\|\|_\infty$	$\max_{i,j} \|A_{ij}\|$ if $\mathbf{A}$ is a $k \times m$ matrix
$\|\mathbf{A}\|_F$	$\sqrt{\sum_{i=1}^k \sum_{j=1}^m A_{ij}^2}$ if $\mathbf{A}$ is a $k \times m$ matrix

CHAPTER 1

Introduction

Machine learning is a very important area for scientific research. Many machine learning techniques are shown to be very powerful for various applications. In this dissertation, we investigate several new graph-based machine learning methods. In our first project, we propose a new graph-based classification technique. For our second and third projects, we study new approaches for graph estimation.

In this chapter, we provide some background knowledge and literature review on machine learning techniques. In Section 1.1, we introduce the linear discriminant analysis and its extensions in high dimensions. In Section 1.2, we briefly review graphical models, including Gaussian graphical models, count-data graphical models, and directed acyclic graphs (DAGs), and their estimation.

1.1 Linear Discriminant Analysis and Its High Dimensional Extensions

Classification is a typical supervised learning problem with categorical response variables. Classification problems are commonly seen in practice. There are many existing classification techniques in the literature; see Bishop (2006); Hastie et al. (2009) for a comprehensive review. Among various existing methods, linear discriminant analysis (LDA) has a long history and remains an important tool in the standard classification toolbox. LDA can be viewed as a rule for a classification problem of two Gaussian populations with a common covariance matrix. Despite its seemingly strong assumptions, LDA often works well in practice, especially for low-dimensional problems (Hand et al., 2006). It mimics the Bayes' rule and has a simple closed form which only involves the within-class sample covariance matrix and group averages. Given these estimates, the original formulation for the discriminant vector of LDA is computed as the product of the inverse within-class sample covariance matrix and the mean difference vector. Thus, standard LDA can be computed and implemented easily in the traditional low-dimensional setting. LDA also has interpretations be-

yond the Gaussian model. In particular, the same formulation can be obtained from the Fisher's discriminant analysis problem (Fisher, 1936), the optimal scoring problem (Hastie et al., 1994), and linear regression (Hastie et al., 2009).

Despite the usefulness of LDA, it needs to be adapted when the dimension of features is high. For example, the form of standard LDA is only valid when the sample covariance matrix is invertible. Moreover, as the dimension grows, the errors in the sample covariance and group means accumulate and consequently LDA can become increasingly unstable (Fan and Fan, 2008; Shao et al., 2011). To address this problem, a number of LDA extensions have been proposed for high-dimensional scenarios.

Existing high-dimensional LDA methods in the literature can be roughly divided into two categories, plug-in approaches and direct approaches. A plug-in approach tackles high-dimensional problems by using regularized estimates for the within-class covariance matrix and group means. For example, the naive Bayes method, or the independence rule, treats the covariance matrix as diagonal. Bickel and Levina (2004) showed that it outperforms LDA with the Moore–Penrose pseudoinverse covariance matrix when the dimension grows faster than the sample size. To further reduce the instability of LDA, Tibshirani et al. (2002) additionally used shrunken estimates of group means. Fan and Fan (2008) showed that, even under the independence feature assumption, naive Bayes can be as bad as random guessing due to error accumulation in group means. They resolved this issue by reducing the dimension via feature screening. In contrast to these independence rules, Shao et al. (2011) assumed sparsity of the covariance matrix and the mean difference vector, and used thresholded estimates to construct a sparse LDA classifier. It was shown to be asymptotically optimal under certain conditions. All of these methods adopt the original formulation of LDA by calculating some improved estimates of the covariance matrix and group means. Thus, some strong assumptions on the covariance matrix and the group means need to be imposed for the resulting LDA rule.

In contrast to the plug-in methods, direct approaches aim at estimating the discriminant vector $\boldsymbol{\beta}$ directly. Since LDA can also be obtained from some risk minimization problems, it can be extended to high-dimensional scenarios via these formulations with regularization on $\boldsymbol{\beta}$. For example, Wu et al. (2009) considered the Fisher's discriminant analysis and proposed an ℓ_1-penalized version for dimension reduction. The corresponding problem has a piece-wise linear solution path

which can be computed efficiently. Witten and Tibshirani (2011) also used Fisher's discriminant analysis formulation for a general K-class problem with a general regularization term. Clemmensen et al. (2011) proposed the optimal scoring formulation with the ℓ_1-penalty. Following the idea of minimizing the misclassification rates, Fan et al. (2012) proposed a method closely related to the method by Wu et al. (2009) and directly computed the misclassification rate of the classifier. Mai et al. (2012) took advantage of the regression formulation and estimated the discriminant vector of LDA by solving a Lasso-type problem, which was shown to have the same solution path as the method of Wu et al. (2009) and the method of Clemmensen et al. (2011) when $K = 2$; see Mai and Zou (2013). Using a different idea for direct estimation, Cai and Liu (2011) formulated a linear programming problem to estimate $\boldsymbol{\beta}$ and showed that the error rate of the estimated classifier is close to the Bayes rule under certain conditions. Compared to plug-in approaches, these methods estimate LDA directly and the assumptions can be less stringent since only the sparsity of the discriminant vector of LDA is assumed (Cai and Liu, 2011).

Both plug-in and direct methods can work well for certain practical problems. However, these methods do not utilize the feature structure information when available. In practice, features are often correlated with some structure. Such structure can usually be represented by an undirected graph $\mathcal{G}$. Connected features may work together and thus be effective or not effective simultaneously for classification. For instance, in the diagnosis of a disease using genetic information, genes are naturally grouped by their functions or gene pathways. Relevant genes tend to contribute or not contribute to the disease together. Moreover, when the population in consideration is Gaussian, the conditional independence graph, or the Gaussian graphical model, often represents a natural structure. By considering such structure information, we are likely to be able to construct a better classifier.

1.2 Graphical Models and Their Estimation

Generally, a graph consists of nodes and edges, directed or undirected, between nodes. In graphical models, a node often represents a random variable, while an edge between two nodes indicates the (conditional) dependence or correlation between the corresponding variables. By constructing a graphical model, researchers can investigate the feature structure in data and visualize

the graph. In the following subsections, we consider Gaussian graphical models and directed acyclic graphs respectively.

1.2.1 The Gaussian Graphical Model and Its Extensions

Among various graphical models, the Gaussian graphical model is possibly the most popular one for its simplicity and easy interpretation. For a multivariate Gaussian population, denoted as $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the Gaussian graphical model is defined as the conditional independence graph, that is, each node represents a variable and two nodes are connected if they are conditionally dependent given the remaining nodes. It has been shown that the Gaussian graphical estimation is equivalent to finding the nonzero elements of the precision matrix, namely the inverse covariance matrix (Yuan and Lin, 2007). Moreover, in this scenario two variables are conditionally dependent if and only if one is in the partial regression model of the other with a nonzero coefficient.

Based on the above observations, many methods have been proposed for Gaussian graphical model estimation. For example, Meinshausen and Bühlmann (2006) proposed to estimate the graph by nodewise Lasso regression and showed the selection consistency. Similar works include Yuan (2010); Luo and Chen (2014a). Yuan and Lin (2007) and Friedman et al. (2008) proposed graphical Lasso that estimates the precision matrix via penalized likelihood. Cai et al. (2011) took advantage of the inversion relationship between the precision matrix and the covariance matrix and estimated a sparse precision matrix directly. Liu et al. (2009) and Liu et al. (2012) further generalized Gaussian graphical models for non-Gaussian data by non-paranormal transformation.

1.2.2 Count Data Graphical Models

Gaussian and non-paranormal graphical models often work well when the random variables or their continuous transformation are approximately Gaussian distributed. However, there are increasingly more scenarios in which this assumption is not true. For example, the Gaussian graphical model is typically not suitable for count data.

There is a rich literature on general count-data graphical models. For count data with binary values, Ravikumar et al. (2010) proposed to estimate Ising models by nodewise ℓ_1-penalized logistic regression. Yang et al. (2012a) and Chen et al. (2014) proposed a class of exponential family

graphical models, of which the Poisson graphical model is a special case. But the model only allows negative dependence among the features. A number of approaches have been proposed to address this issue. Yang et al. (2013) proposed to remove such a restriction by modifying the base measure of the multivariate Poisson distribution. Allen et al. (2013) proposed a neighborhood selection approach that estimates the neighborhood of each node via penalized Poisson regression. See Inouye et al. (2017) for a comprehensive review. Recently, the Poisson-logNormal model has attracted a lot of attention for count-data graphical modeling. A variety of methods have been proposed for model estimation (Choi et al., 2017; Wu et al., 2018a; Sinclair and Hooker, 2017; Chiquet et al., 2018). However, none of them is feasible for high-dimensional modeling due to computational burdens.

Despite progresses on count-data graphical models, these existing models may not be appropriate for some new data applications. In particular, the single cell RNA sequencing (scRNA-seq) techniques provide a way to study the gene expressions of each cell in a biological sample. However, there often exists substantial dependence among the cells and inflated zeros in the data. Thus new graphical models are needed to fit scRNA-seq data.

1.2.3 Directed Acyclic Graphs and Skeleton Estimation

There are many different types of graphical models. Besides undirected graphs, there is another type of graphs, namely the directed graph. One can tell from the name that they differ in the edge type: edges in a directed graph usually have directions. These directions sometimes may imply causal relationships, which makes directed graphs particularly suitable for causal inference.

A directed acyclic graph (DAG) is a directed graph of which the edges do not form any directed cycle. Statistically, a DAG is a model that describes conditional dependence of random variables. A directed edge $l \to j$ indicates that j is dependent on l given any subset of the remaining nodes. On the other hand, the absence of an edge indicates that the two nodes are marginally independent, or conditionally independent given some subset of the remaining nodes. By removing the direction of each edge in a DAG, the resulting undirected graph is the *skeleton* of the DAG. Estimation of a DAG is often infeasible due to lack of interventional data. Instead, the DAG skeleton can be estimated using observational data. Given the DAG skeleton, one can orient as many edges as

possible to form a completed partially directed graph (CPDAG) using a set of deterministic rules (Pearl, 2009). We will focus on skeleton estimation in Chapter 3.

There are various approaches to estimate a DAG or its skeleton. A typical search-and-score approach searches for a skeleton that maximizes the regularized likelihood or the posterior probability (Heckerman et al., 1995; Friedman and Koller, 2003). These methods can quickly become computationally infeasible for problems with thousands of variables. An alternative solution for skeleton estimation is constraint-based approaches, which are often more efficient computationally. Among these constraint-based methods, the PC algorithm (Spirtes et al., 2000; Kalisch and Bühlmann, 2007; Colombo and Maathuis, 2014), which is based on conditional independence tests, has been shown to be consistent in high dimension. There are also hybrid methods that combine the search-and-score and constraint-based approaches (Tsamardinos et al., 2006; Schmidt et al., 2007; Han et al., 2016). Nandy et al. (2015) showed that some hybrid methods with greedy search over the constrained DAG space are consistent under high dimensional settings. In a recent paper, Ha et al. (2016) proposed a two-stage skeleton estimation method called PenPC, which first estimates a conditional independence graph by neighborhood selection, and then estimates the skeleton by a modified PC algorithm.

In genetic study, the biological samples are often collected from individuals of multiple populations. Each population can have a unique DAG structure while these DAGs may have a significant overlap. Since the population labels are often unknown, typical approaches for DAG (skeleton) estimation are to cluster or classify the samples into several groups and estimate a common DAG for all populations or a unique DAG for each population. However, these approaches either neglect the differences or do not take advantage of the similarities among the DAGs.

1.3 New Contributions and Outline

In Sections 1.1 and 1.2, we discuss the possibility of enhancing supervised learning with graphical structure information of features as well as scenarios in which existing graph estimation methods may not be appropriate. Motivated by these problems, in this dissertation, we propose several new approaches involving graphical models. In particular, the following chapters of the dissertation are organized as follows.

- In Chapter 2, we introduce a new high-dimensional LDA technique, namely graph-based sparse LDA (GSLDA), that utilizes the graph structure among the features. In particular, we use the regularized regression formulation for penalized LDA techniques, and propose to impose a structure-based sparse penalty on the discriminant vector $\boldsymbol{\beta}$. The graph structure can be either given or estimated from the training data. Moreover, we explore the relationship between the within-class feature structure and the overall feature structure. Based on this relationship, we further propose a variant of our proposed GSLDA to utilize unlabeled data effectively, which can be abundant in the semi-supervised learning setting. With the new regularization, we can obtain a sparse estimate of $\boldsymbol{\beta}$ and more accurate and interpretable classifiers than many existing methods. Both the selection consistency of $\boldsymbol{\beta}$ estimation and the convergence rate of the classifier are established, and the resulting classifier has an asymptotic Bayes error rate. Finally, we demonstrate the competitive performance of the proposed GSLDA on both simulated and real data studies.

- In Chapter 3, we consider the high-dimensional skeleton estimation with heterogeneous observational data. A two-step approach is proposed to jointly estimate the DAG skeletons of multiple populations while the population origin of each sample may or may not be labeled. In particular, our method allows a probabilistic soft label for each sample, which can be easily computed and often leads to more accurate skeleton estimation than hard labels. Compared with separate estimation of skeletons for each population, our method is more accurate and robust to labeling errors. We study the estimation consistency for our method, and demonstrate its performance using simulation studies in different settings. Finally, we apply our method to analyze gene expression data from breast cancer patients of multiple cancer subtypes.

- In Chapter 4, we focus on the graphical modeling for single cell RNA-sequencing (scRNA-seq) data. We investigate the characteristics of scRNA-seq data, especially the zero-inflation of data and the dependence among cells. By taking these features into account, we propose a Poisson-logNormal graphical model to capture gene interactions. To handle excessive zeros in scRNA-seq data, we further extend our model to a Hurdle-logNormal graphical model. The two graphical models account for the unique characteristics of scRNA-seq data and are shown

to produce better graph estimation for scRNA-seq data. Both models are computationally efficient so that they work for datasets with thousands of genes. We demonstrate the advantages of our proposed models against existing methods using both simulated examples and real data applications.

CHAPTER 2

Graph-based Sparse Linear Discriminant Analysis for High Dimensional Data

2.1 Introduction

Linear discriminant analysis often has good performance in low dimensional classification problems. Though it has difficulties when directly applied to high dimensional problems, its extensions as mentioned in Section 1.1.1 can work well for certain practical problems. However, these methods do not utilize the feature structure information when available. In practice, features are often correlated with some structure. Such structure can usually be represented by an undirected graph $\mathcal{G}$. Connected features may work together and thus be effective or not effective simultaneously for classification. For instance, in the diagnosis of a disease using genetic information, genes are naturally grouped by their functions or gene pathways. Relevant genes tend to contribute or not contribute to the disease together. Moreover, when the population in consideration is Gaussian, the conditional independence graph, or Gaussian graphical model, often represents a natural structure. By considering such structure information, we are likely to be able to construct a better classifier. For regression problems, there are some methods that utilize the graph structure in the literature; see, e.g., Bondell and Reich (2008); Pan et al. (2010); Zhu et al. (2013); Kim et al. (2013). For example, Li and Li (2008) proposed a penalty on the coefficient difference of each pair of connected features. Yang et al. (2012b) used pairwise ℓ_∞ penalties on relevant features to encourage simultaneous inclusion and exclusion. Based on the decomposition of the regression coefficient vector, Yu and Liu (2016) proposed a node-wise penalty. In particular, the regularization term is the summation of penalties over all nodes rather than all edges. Compared to pairwise penalties, the node-wise penalty is better motivated and computationally efficient. More recently, Zhao and Shojaie (2016) proposed new inference methods for such graph-constrained estimation.

Despite great progress for regression problems, much less research has been done for classification problems. Structured penalties such as group Lasso and fused Lasso have been employed in

classification methods (Meier et al., 2008; Witten and Tibshirani, 2011), but they are not applicable to a general sparse graph structure among predictors. Zhang et al. (2013) considered logistic regression with a combination of ℓ_1 penalty and pairwise ℓ_2 difference penalty. Min et al. (2018) generalized the regularization and provided a unified algorithm. However, both methods may also suffer from too much computational burden in high dimensions. Very recently, Wu et al. (2018b) proposed an unsupervised graph-based variable screening method for general problems.

In this chapter, we propose a new method, called graph-based sparse LDA (GSLDA), that exploits the graphical structure of features. GSLDA estimates LDA in high dimensions directly by solving a convex optimization problem. Similar to the sparse regression method in Yu and Liu (2016), we incorporate the graph structure through a node-wise penalty. In the presence of an underlying feature structure, the new method outperforms existing high-dimensional LDA methods by utilizing the structure directly. As a key component, the graphical structure can be either given or estimated from the training data. In addition, we investigate the relationship between the within-class inverse covariance matrix and overall inverse covariance matrix. Based on these findings, we propose a variant of GSLDA that can utilize unlabeled data, which are often much more accessible than labeled data. We name this variant as the semi-supervised GSLDA. Selection consistency is shown for the estimated discriminant vector. Moreover, we show that the misclassification rate of our classifier converges to the Bayes error rate at a fast rate under certain conditions. Numerical studies are used to demonstrate the performance of this method. In particular, the semi-supervised GSLDA enjoys higher classification accuracy than the original GSLDA method in most cases. This reveals the potential advantages of using unlabeled data in classification problems.

The rest of the chapter is organized as follows. In Section 2.1, we review some existing high-dimensional LDA methods, and introduce our motivations and formulations of our proposed methods. Section 2.3 focuses on graph estimation and the implementation of GSLDA. In particular, graph estimation methods are discussed for both GSLDA and its variant. In Section 2.4, theoretical justification is provided for our method. Sections 2.5 and 2.6 demonstrate the performance of GSLDA by simulated examples and real data studies respectively. We conclude this chapter with some discussion in Section 2.7. Proofs of the theoretical results are provided in the Appendix A.

2.2 Methodology

In this section, we first review LDA and construct a relationship between $\boldsymbol{\beta}$ and the graph structure of features in Section 2.2.1, based on which GSLDA is proposed. We also explain how to estimate the graph structure when it is not directly available and discuss the connections of our methods with several existing classification methods. In Section 2.2.2, we investigate the overall graph structure of the features and consider a variant of GSLDA which can efficiently utilize unlabeled data.

2.2.1 Motivation and formulation of GSLDA

We first discuss the problem setting and introduce some notations. Given the training dataset $\{(\mathbf{x}_1, g_1), \ldots, (\mathbf{x}_n, g_n)\}$ where for each $i \in \{1, \ldots, n\}$, $\mathbf{x}_i \in \mathbb{R}^p$ is the feature vector and $g_i \in \{1, 2\}$ is the class label. A linear classifier $g_{\beta_0,\boldsymbol{\beta}}$ is defined as follows. For any $\mathbf{x} \in \mathbb{R}^p$, $g_{\beta_0,\boldsymbol{\beta}}(\mathbf{x}) = 1$ if $\beta_0 + \mathbf{x}^\top \boldsymbol{\beta} > 0$ and 2 otherwise. In particular, we consider the standard setting of the two-class LDA. That is, the binary label G takes 1 with probability π_1 and 2 with probability $\pi_2 = 1 - \pi_1$ and the feature vector $\mathbf{X}$ has a conditional Gaussian distribution, i.e., $\mathbf{X}|(G = k) \sim \mathcal{N}(\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma})$ for $k \in \{1, 2\}$. Under this setting, the Bayes classifier $g_{\beta_0^*,\boldsymbol{\beta}^*}$ is specified by

$$\boldsymbol{\beta}^* = \boldsymbol{\Sigma}^{-1}\boldsymbol{\delta} \quad \text{and} \quad \beta_0^* = -(\boldsymbol{\mu}^{(1)} + \boldsymbol{\mu}^{(2)})^\top \boldsymbol{\beta}^*/2 + \ln(\pi_1/\pi_2), \tag{2.1}$$

where $\boldsymbol{\delta} = \boldsymbol{\mu}^{(1)} - \boldsymbol{\mu}^{(2)}$. By replacing $\boldsymbol{\Sigma}$ and $\boldsymbol{\delta}$ in (2.1) with their sample estimates, we have the LDA classifier with $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\delta}}$. Typically, we take $\hat{\boldsymbol{\Sigma}} = (n_1\mathbf{S}^{(1)} + n_2\mathbf{S}^{(2)})/(n-2)$ and $\hat{\boldsymbol{\delta}} = \bar{\mathbf{x}}^{(1)} - \bar{\mathbf{x}}^{(2)}$, where n_k, $\bar{\mathbf{x}}^{(k)}$, and $\mathbf{S}^{(k)}$ denote respectively the sample size, mean, and covariance matrix for group k. Note that this formulation is valid only when $n > p$. In high-dimensional problems or when $n \leq p$, there are various extensions of LDA that either use the formulation with shrunken estimates of $\boldsymbol{\Sigma}$ and $\boldsymbol{\delta}$ or find a direct estimation of $\boldsymbol{\beta}$; see (Tibshirani et al., 2002; Shao et al., 2011; Cai et al., 2011; Mai et al., 2012; Fan et al., 2012). Here we focus on the direct estimation approach.

Inspired by the regression formulation of LDA (Hastie et al., 2009), Mai et al. (2012) proposed the direct sparse discriminant analysis (DSDA) method to estimate $\boldsymbol{\beta}$ by solving the Lasso problem

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + \lambda ||\boldsymbol{\beta}||_1,$$

where $y_i = n/n_1$ if $g_i = 1$ and $-n/n_2$ if $g_i = 2$. It was shown that DSDA gives the same solution path as the method in Mai and Zou (2013); Wu et al. (2009). Compared to plug-in approaches, the DSDA estimates $\boldsymbol{\beta}$ directly in high dimensions and the assumptions are less stringent. However, it is unclear how we can utilize any structure information among features with the method or other high-dimensional LDA methods.

Assume that there is some structure among the features. In particular, we consider the case where the structure can be represented by a graph, denoted as $\mathcal{G}$. There are methods that effectively use the graph structure in regression problems. For example, Li and Li (2008) used the penalty

$$\sum_{(j,\ell)\in\mathcal{G}} \left(\beta_j/\sqrt{d_j} - \beta_\ell/\sqrt{d_\ell}\right)^2,$$

where d_j denotes the neighborhood size of feature j, to encourage close coefficients for connected features. Yang et al. (2012b) employed pairwise ℓ_∞ penalty for connected features, i.e., $\sum_{(j,\ell)\in\mathcal{G}} \max\{|\beta_j|, |\beta_\ell|\}$, so their coefficients can be estimated zero or nonzero simultaneously. Recently, Yu and Liu (2016) proposed a node-wise penalty

$$P_{\mathcal{G},\boldsymbol{\tau}}(\boldsymbol{\beta}) = \min_{\sum_{j=1}^{p} \mathbf{v}^{(j)} = \boldsymbol{\beta},\ \operatorname{supp}(\mathbf{v}^{(j)}) \subseteq \mathcal{N}^{(j)}} \sum_{j=1}^{p} \tau_j ||\mathbf{v}^{(j)}||_2$$

based on the decomposition of regression coefficient vector $\boldsymbol{\beta} = (X)^{-1}\operatorname{cov}(X, Y)$. In contrast to these developments for regression problems, little work has been done for classification problems.

We propose our method formulation based on a decomposition of $\boldsymbol{\beta}^*$, the discriminant vector of Bayes' rule. Denote $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ the within-class precision matrix and $\boldsymbol{\delta} = \boldsymbol{\mu}^{(1)} - \boldsymbol{\mu}^{(2)}$ the group mean difference. We can decompose the discriminant vector $\boldsymbol{\beta}^*$ in (2.1) as

$$\boldsymbol{\beta}^* = \boldsymbol{\Omega}\boldsymbol{\delta} = \sum_{j=1}^{p} \delta_j \boldsymbol{\omega}_j, \tag{2.2}$$

where $\boldsymbol{\omega}_j$ is the jth column of $\boldsymbol{\Omega}$. Recall that the support of $\boldsymbol{\Omega}$ in fact forms a conditional correlation graph of features X. In this way, the optimal discriminant vector is linked to the Gaussian graph structure of the features. We use a toy example for demonstration. In a 3-dimensional LDA setting, assume $\omega_{23} = \omega_{32} = 0$, then $\boldsymbol{\beta}^* = \boldsymbol{\Omega}\boldsymbol{\delta} = (\delta_1\omega_{11} + \delta_2\omega_{21} + \delta_3\omega_{31}, \delta_1\omega_{12} + \delta_2\omega_{22}, \delta_1\omega_{13} + \delta_3\omega_{33})^\top$. See Figure A.1 in the Appendix for a graphical demonstration of the decomposition.

Denote the graph corresponding to $\boldsymbol{\Omega}$ as $\mathcal{G}$, and the neighborhood of feature $j \in \{1, \ldots, p\}$ as $\mathcal{N}^{(j)}$. Replacing $\delta_j\boldsymbol{\omega}_j$ by $\mathbf{v}^{(j)}$, then $\boldsymbol{\beta}^* = \mathbf{v}^{(1)} + \cdots + \mathbf{v}^{(p)}$, where $\mathbf{v}^{(j)}$ is either $\mathbf{0}$ (when $\delta_j = 0$) or with a support $\text{supp}(\mathbf{v}^{(j)}) = \mathcal{N}^{(j)}$ when $\delta_j \neq 0$. Instead of estimating $\boldsymbol{\beta}^*$ itself, we can estimate $\mathbf{v}^{(j)}$'s. Moreover, the decomposition (2.2) motivates a natural regularization on $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(p)}\}$, viz.

$$\sum_{j=1}^{p} \tau_j \, ||\mathbf{v}^{(j)}||_2,$$

in which $\text{supp}(\mathbf{v}^{(j)}) \subseteq \mathcal{N}^{(j)} = \text{supp}(\boldsymbol{\omega}_j)$ and the τ_js are positive weights. Note that the group ℓ_2 penalty on $\mathbf{v}^{(j)}$ encourages a group sparsity effect, i.e., $\mathbf{v}^{(j)}$ is estimated as $\mathbf{0}$ or a sparse vector with support $\mathcal{N}^{(j)}$, which matches the decomposition (2.2). In this formulation, the τ_js are weights for the group regularization. In particular, the larger τ_j is, the more likely $\mathbf{v}^{(j)}$ is estimated as $\mathbf{0}$. Similar to the group Lasso (Yuan and Lin, 2006), we can take

$$\tau_j = \sqrt{|\mathcal{N}^{(j)}|}/|\hat{\delta}_j|,$$

where $\hat{\delta}_j = \bar{x}_j^{(1)} - \bar{x}_j^{(2)}$.

We need to apply this regularization to a risk minimization framework of LDA to formulate our method. The regression formulation is an appropriate one due to its simplicity and convenience for theoretical analysis. By combining the formulation with the group regularization, we can estimate $\{\mathbf{v}_1, \ldots, \mathbf{v}_p\}$ by

$$(\hat{\beta}_0, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_p) = \underset{\beta_0,\, \mathbf{v}_1, \ldots, \mathbf{v}_p}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \mathbf{x}_i^\top \sum_{j=1}^{p} \mathbf{v}^{(j)} \right)^2 + \lambda \sum_{j=1}^{p} \tau_j ||\mathbf{v}^{(j)}||_2, \tag{2.3}$$

where $\text{supp}(\mathbf{v}^{(j)}) \subseteq \mathcal{N}^{(j)}$ for all $j \in \{1, \ldots, p\}$. Then $\boldsymbol{\beta}$ is estimated as $\hat{\mathbf{v}}_1 + \cdots + \hat{\mathbf{v}}_p$. Furthermore, from the perspective of $\boldsymbol{\beta}$ estimation, the formulation is equivalent to

$$(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) = \underset{\beta_0, \boldsymbol{\beta}}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + \lambda ||\boldsymbol{\beta}||_{\mathcal{G}, \boldsymbol{\tau}}, \tag{2.4}$$

where

$$||\boldsymbol{\beta}||_{\mathcal{G}, \boldsymbol{\tau}} = \min_{\sum_{j=1}^{p} \mathbf{v}^{(j)} = \boldsymbol{\beta},\ \text{supp}(\mathbf{v}^{(j)}) \subseteq \mathcal{N}^{(j)}} \sum_{j=1}^{p} \tau_j ||\mathbf{v}^{(j)}||_2 \tag{2.5}$$

can be viewed as a structured regularization on $\boldsymbol{\beta}$; see Obozinski et al. (2011). Since the regularization is specified by the graph $\mathcal{G}$, we call the method graph-based sparse LDA (GSLDA). Although we use the same squared loss function as in Mai et al. (2012), our method focuses on utilizing the graph structure of features in $\boldsymbol{\beta}^*$ estimation. We use the estimator $\hat{\boldsymbol{\beta}}$ from (2.4) for the discriminant vector $\boldsymbol{\beta}$. With respect to β_0, the estimator from (2.4) may not be a good choice for the classification problem due to the regression formulation. To solve this problem, we adopt a similar approach by Mai et al. (2012) and estimate it by

$$\hat{\beta}_0 = -(\bar{\mathbf{x}}^{(1)} + \bar{\mathbf{x}}^{(2)})^\top \hat{\boldsymbol{\beta}}/2 + \ln(n_1/n_2) \hat{\boldsymbol{\beta}}^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\beta}} / (\hat{\boldsymbol{\delta}}^\top \hat{\boldsymbol{\beta}}).$$

While the GSLDA method is motivated from the discriminant vector decomposition (2.2), the decomposition of $\boldsymbol{\beta}^*$ is not restricted to this form only. Therefore, the graph structure $\mathcal{G}$ used in our method is not restricted to the conditional independence graph. We will present another decomposition of $\boldsymbol{\beta}^*$ in Section 2.2. In fact, any graph structure of features satisfying our assumptions in Section 2.4.1 can be possibly used. When the structure information is available, e.g., the gene pathways in genetic studies, we can construct a graph $\mathcal{G}$ using the gene pathway information. If the graph is not available, we can estimate it based on the training data. There are many methods for estimation of Gaussian graphical models, including the neighborhood selection (Meinshausen and Bühlmann, 2006), the graphical Lasso (Yuan and Lin, 2007; Friedman et al., 2008), and the CLIME (Cai et al., 2011). We will discuss them further in Section 2.3. In summary, GSLDA can be implemented in two steps: (i) graph construction and (ii) direct estimation of $\boldsymbol{\beta}$ via solving formulation (2.4).

The formulation (2.4) is closely related to the regression method proposed in Yu and Liu (2016). However, both the problem setting and the motivation of our methods are different. In our problem, the response y is a binary variable and the features are from a mixed population. Although our formulation also uses the squared loss as in regression, the "error" $y_i - \mathbf{x}_i^\top \boldsymbol{\beta}$ has a very different interpretation and distribution. In particular, the distribution of $y_i - \mathbf{x}_i^\top \boldsymbol{\beta}$ depends on $\mathbf{x}_i$. These issues bring unique challenges for the theoretical analysis of GSLDA. Although there are some classification methods that also utilize predictor structure, such as logistic regression with group Lasso penalty (Meier et al., 2008) and LDA with fused Lasso penalty (Witten and Tibshirani, 2011), these methods do not utilize a general graph structure.

Depending on the feature structure, there are special cases in which GSLDA is closely connected with existing sparse LDA methods. For example, if we use an empty graph $\mathcal{G}$ with no edge at all, the regularization (2.5) simplifies to $\tau_1|\beta_1| + \cdots + \tau_p|\beta_p|$. Then, formulation (2.4) becomes an adaptive Lasso type problem, viz.

$$\underset{\beta_0,\, \boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \tau_j |\beta_j|.$$

When all penalty weights τ_j take value 1, the GSLDA is equivalent to the DSDA method in Mai et al. (2012). When the graph $\mathcal{G}$ consists of K disjoint complete subgraphs, denoted as $\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(K)}$, then the regularization (2.5) simplifies to $\tau^{(1)}||\boldsymbol{\beta}_{G^{(1)}}||_2 + \cdots + \tau^{(K)}||\boldsymbol{\beta}_{G^{(K)}}||_2$ where $\tau^{(k)} = \min_{j \in G^{(k)}} \tau_j$ and $G^{(k)}$ is the index set of predictors involved in the subgraph $\mathcal{G}^{(k)}$. In this case, GSLDA becomes a variant of DSDA with the group Lasso penalty, i.e.,

$$\underset{\beta_0,\, \boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{k=1}^{K} \tau^{(k)} ||\boldsymbol{\beta}_{G^{(k)}}||_2.$$

For a general graph $\mathcal{G}$, our method is different from the existing ones.

Remark 1. While we are mainly concerned with binary classification in this chapter, there are many scenarios with more than two classes (Liu and Yuan, 2011; Zhang and Liu, 2013; Zhang et al., 2016). Our GSLDA method can also be extended to the multi-class case. For example, consider a

formulation of K-class sparse LDA proposed in Mai et al. (2015), viz.

$$(\hat{\boldsymbol{\theta}}_2, \ldots, \hat{\boldsymbol{\theta}}_K) = \underset{\boldsymbol{\theta}_2,\ldots,\boldsymbol{\theta}_K}{\operatorname{argmin}} \sum_{k=2}^{K} \{\boldsymbol{\theta}_k^\top \hat{\boldsymbol{\Sigma}} \boldsymbol{\theta}_k / 2 - (\bar{\mathbf{x}}^{(k)} - \bar{\mathbf{x}}^{(1)})^\top \boldsymbol{\theta}_k\} + \lambda \sum_{j=1}^{p} \|\boldsymbol{\theta}_{\cdot j}\|_2,$$

where $\boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K$ are discriminant vectors and $\boldsymbol{\theta}_{\cdot j} = (\theta_{2j}, \ldots, \theta_{Kj})^\top$ for $j \in \{1, \ldots, p\}$. The resulting discriminant rule is $\hat{g} = \operatorname{argmax}_k \{\hat{\boldsymbol{\theta}}_k^\top (\mathbf{x} - \bar{\mathbf{x}}^{(k)}/2) + \ln \hat{\pi}_k\}$ where $\hat{\boldsymbol{\theta}}_1 = \mathbf{0}$ and $\hat{\pi}_k$ is the proportion of class k in the sample. We can take advantage of a similar formulation with the graph-based regularization $\lambda \|\boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\|_{\mathcal{G},\boldsymbol{\tau},\text{grouped}}$, where

$$\|\boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\|_{\mathcal{G},\boldsymbol{\tau},\text{grouped}} = \underset{\sum_{j=1}^{p} \mathbf{v}_k^{(j)} = \boldsymbol{\theta}_k, \operatorname{supp}(\mathbf{v}_k^{(j)}) \subseteq \mathcal{N}^{(j)}}{\operatorname{argmin}} \sum_{j=1}^{p} \tau_j \|(\mathbf{v}_2^{(j)\top}, \ldots, \mathbf{v}_K^{(j)\top})^\top\|_2.$$

This formulation can be solved in a way similar to the binary GSLDA. Nevertheless, we do not pursue this direction in the thesis so we can focus on core ideas of the GSLDA.

2.2.2 Semi-supervised GSLDA

With recent advances in graphical estimation (Meinshausen and Bühlmann, 2006; Yuan and Lin, 2007; Cai et al., 2011), we can estimate $\mathcal{G}$ for the GSLDA based on the training data when the graph structure is unknown. However, as the dimension p increases, we expect the selection error to accumulate. When the dimension is much larger than the sample size, the graph estimate of GSLDA can be almost random. We use a toy example in Figure 2.1 to illustrate this phenomenon. In the setting of standard LDA, we set weights $\pi_1 = \pi_2 = 0.5$, and group means $\boldsymbol{\mu}^{(1)} = (0.5, \ldots, 0.5, 0, \ldots, 0)^\top$ and $\boldsymbol{\mu}^{(2)} = (-0.5, \ldots, -0.5, 0, \ldots, 0)^\top$, which only differ in the first 10 features. To specify the graph structure, $\boldsymbol{\Omega}$ is generated from an AR(5) model, i.e., $\Omega_{jj} = c, \Omega_{j\ell} = -0.5$ if $1 \leq |j - \ell| \leq 5$ and 0 otherwise, where $c > 0$ is a scalar such that the eigenvalues of $\boldsymbol{\Omega}$ are between 0 and 1. We standardize $\boldsymbol{\Omega}$ so that $\operatorname{diag}(\boldsymbol{\Omega}) = \mathbf{1}$ and define in-class covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Omega}^{-1}$. Let the sample size n be 50 and p vary from 10 to 200. We estimate the graph by SR-SLasso (Luo and Chen, 2014a) with extended BIC for tuning. For each setting, we repeat the procedure 100 times and evaluate the accuracy of graph estimation by false positive rate (FPR) and false negative rate (FNR). Figure 2.1 summarizes the performance of graph estimation for varying dimensions.

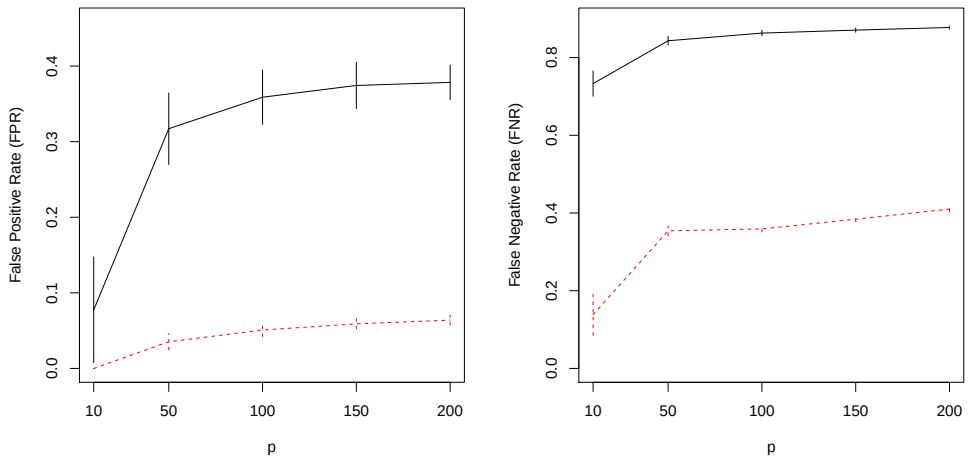

Figure 2.1: Performance evaluation of graph estimation for varying dimensions. The black solid lines are for graph estimation based on a labeled dataset of size 50; the red dashed lines are for graph estimation based on an unlabeled dataset of size 1000; vertical segments indicate the standard deviations of FPR or FNR of 100 repetitions.

As shown in Figure 2.1, the graph estimation using only labeled data deteriorates quickly as the dimension increases. Note that the structured penalty in (2.5) encourages the coefficients of all features in a neighborhood to be nonzero together as long as some of them is useful for classification. Inaccurate graph estimation can reduce the accuracy and the interpretability of GSLDA.

Compared to labeled data, unlabeled data can be more accessible in many applications. For example, in the handwritten digit recognition problem discussed in Section 2.6.2, we can easily obtain a large number of images of different digits. However, it can be expensive to label these images by corresponding digits. As a result, many semi-supervised methods try to utilize the unlabeled data to improve the classification accuracy (Pan and Shen, 2007; Cai et al., 2007). In this thesis, we focus on using unlabeled data for the graph construction when available. The following proposition studies the relationship between the within-class inverse covariance matrix and the overall one.

Proposition 1. *Assume X comes from a mixture of two populations with a common covariance matrix $\mathbf{\Sigma}$. The weight and the expectation of population $k \in \{1, 2\}$ is π_k and $\boldsymbol{\mu}^{(k)}$. Denote the mean difference of the two populations $\boldsymbol{\mu}^{(1)} - \boldsymbol{\mu}^{(2)}$ as $\boldsymbol{\delta}$. We denote $\tilde{\mathbf{\Sigma}} = (X)$ the overall covariance matrix of the population mixture and $\tilde{\mathbf{\Omega}} = \tilde{\mathbf{\Sigma}}^{-1}$ the overall precision matrix. Then $\tilde{\mathbf{\Sigma}} = \mathbf{\Sigma} + \pi_1\pi_2\boldsymbol{\delta}\boldsymbol{\delta}^\top$ and $\tilde{\mathbf{\Omega}} = \mathbf{\Omega} - c\boldsymbol{\beta}^*\boldsymbol{\beta}^{*\top}$, where $\boldsymbol{\beta}^* = \mathbf{\Omega}\boldsymbol{\delta}$ and $c = 1/\{(\pi_1\pi_2)^{-1} + \boldsymbol{\delta}^\top\mathbf{\Omega}\boldsymbol{\delta}\}$.*

As a remark, we do not require any specific distribution for the populations in Proposition 1, while $\boldsymbol{\beta}^*$ is the optimal discriminant vector if both classes are Gaussian populations. The overall precision matrix $\tilde{\boldsymbol{\Omega}}$ is sparse if both $\boldsymbol{\Omega}$ and $\boldsymbol{\beta}^*$ are sparse, and its support forms the conditional correlation graph of the mixed population. Moreover, we have $\tilde{\boldsymbol{\Omega}}\boldsymbol{\delta} = (1 - c\boldsymbol{\beta}^{*\top}\boldsymbol{\delta})\boldsymbol{\beta}^* \propto \boldsymbol{\beta}^*$. In our problem, a decomposition of the optimal discriminant vector analogous to (2.2) using $\tilde{\boldsymbol{\Omega}}$ can be written as

$$\boldsymbol{\beta}^* = \xi \sum_{j=1}^{p} \delta_j \tilde{\mathbf{w}}_j,$$

where ξ is a positive scalar and $\tilde{\mathbf{w}}_j$ is the jth column of $\tilde{\boldsymbol{\Omega}}$. Therefore, the Bayes classifier can be connected to the graph structure of the mixed population through the new decomposition. Define the graph corresponding to the support of $\tilde{\boldsymbol{\Omega}}$ as $\tilde{\mathcal{G}}$. Following the same rationale of GSLDA, we can formulate another estimator of $\boldsymbol{\beta}$ based on the overall graph structure, viz.

$$(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) = \underset{\beta_0, \boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + \lambda ||\boldsymbol{\beta}||_{\tilde{\mathcal{G}}, \tilde{\boldsymbol{\tau}}}, \tag{2.6}$$

where $||\boldsymbol{\beta}||_{\tilde{\mathcal{G}}, \tilde{\boldsymbol{\tau}}}$ is defined in (2.5) and $\tilde{\boldsymbol{\tau}}$ adapts to $\tilde{\mathcal{G}}$ as in (2.4). The only difference between (2.6) and (2.4) is which graph structure we use. When unlabeled data are abundant, the estimated graph $\tilde{\mathcal{G}}$ can be more accurate and thus the new formulation may provide better classification. We name the formulation (2.6) as semi-supervised GSLDA. Similar to the original GSLDA, the semi-supervised variant also has two steps: (i) graph estimation based on all available data and (ii) direct estimation of $\boldsymbol{\beta}$ by solving formulation (2.6).

Both versions of GSLDA need to estimate a graph when no prior graph structure is given. But there is a major difference: unlike $\mathcal{G}$ in (2.4), the graph $\tilde{\mathcal{G}}$ in (2.6) is not for a Gaussian population but a Gaussian mixture. As we will see in Section 2.3, likelihood-based estimation such as graphical Lasso would be too complicated to implement. Instead, we can still use neighborhood selection. In fact, in regressing the feature X_j on the other features X_{-j}, the coefficient vector corresponds to the conditional correlations between X_j and other features regardless of the distribution of the features, as stated by the following lemma.

Lemma 1. *For any random vector* $X = (X_1, \ldots, X_p)^\top \sim F$*, assume we have finite second-order moments and denote* $\tilde{\boldsymbol{\mu}} = \mathrm{E}_F(X)$*,* $\tilde{\boldsymbol{\Sigma}} = \mathrm{E}_F\{(X - \tilde{\boldsymbol{\mu}})(X - \tilde{\boldsymbol{\mu}})^\top\}$ *and* $\tilde{\boldsymbol{\Omega}} = \tilde{\boldsymbol{\Sigma}}^{-1}$*. Then for any* $j, \ell \in \{1, \ldots, p\}$*,*

(i) $\tilde{\omega}_{j\ell}$*, the* (j,ℓ)*th element of* $\tilde{\boldsymbol{\Omega}}$*, is 0 if and only if* X_j *and* X_ℓ *are conditionally uncorrelated, i.e.,* $\mathrm{cov}(X_j, X_\ell | X_{-\{j,\ell\}}) = 0$*, where* $X_{-\{j,\ell\}}$ *denotes all features other than* X_j *and* X_ℓ*;*

(ii) $\tilde{\omega}_{j\ell}$ *is* 0 *if and only if* $\gamma_\ell^{(j)} = 0$*, where* $\gamma_\ell^{(j)}$ *is the coefficient of* X_ℓ *in the regression of* X_j *on* X_{-j}*.*

This lemma is closely related to the results in Meinshausen and Bühlmann (2006). According to Lemma 1, the graph based on the inverse covariance matrix always corresponds to the conditional correlation structure. As long as variable selection consistency of the regression is guaranteed, neighborhood selection methods are valid for graph estimation. Figure 2.1 also shows the performance of graph estimation based on a large unlabeled dataset under the same settings. We can observe that the estimation still performs well when the dimension increases.

Remark 2. In practice, we generally use all available data, including both unlabeled and labeled data, in the first step of semi-supervised GSLDA. Note that even without unlabeled data, the method is still applicable. If we use neighborhood selection for graph estimation, then the error variance of the jth node-wise regression is $(X_j | X_{-j}) = 1/\tilde{\omega}_{jj} = 1/(\omega_{jj} - c\beta_j^{*2})$ by Proposition 1. In contrast, when using the labels as in the original GSLDA, the error variance is $(X_j | X_{-j}, G) = 1/\omega_{jj} < 1/(\omega_{jj} - c\beta_j^{*2})$. Therefore, the semi-supervised GSLDA has better graph estimation only when unlabeled data are abundant. When there are relatively little unlabeled data, the original GSLDA is more advantageous.

2.3 Graph estimation and method implementation

If the feature structure is given from prior knowledge, the graph can be directly constructed by assigning edges between related features. Otherwise, we need to estimate the graph based on training data. In particular, when unlabeled data are available, we can also use that to estimate the graph and implement semi-supervised GSLDA. In this section, we first discuss specific graph

estimation methods for GSLDA. Then we introduce algorithms to solve formulation (2.4) as well as some strategies for efficient implementation.

2.3.1 Graph estimation

There have been extensive studies on graphical model estimation (Yuan and Lin, 2007; Friedman et al., 2008; Meinshausen and Bühlmann, 2006; Cai et al., 2011; Voorman et al., 2013; Chen et al., 2014). As we discussed in Section 2.2.2, the graph estimation based on labeled and unlabeled data are different to some extent. Next we discuss them separately. Given labeled data, the likelihood conditional on the labels becomes

$$(2\pi)^{-pn/2}|\mathbf{\Omega}|^{n/2}\exp\left\{-\frac{1}{2}\sum_{g_i=1}(\mathbf{x}_i-\boldsymbol{\mu}^{(1)})^\top\mathbf{\Omega}(\mathbf{x}_i-\boldsymbol{\mu}^{(1)})-\frac{1}{2}\sum_{g_i=2}(\mathbf{x}_i-\boldsymbol{\mu}^{(2)})^\top\mathbf{\Omega}(\mathbf{x}_i-\boldsymbol{\mu}^{(2)})\right\}.$$

Similar to the graphical Lasso, we can estimate $\mathbf{\Omega}$ by minimizing ℓ_1 penalized log-likelihood, i.e.,

$$\underset{\boldsymbol{\mu}^{(1)},\boldsymbol{\mu}^{(2)},\,\mathbf{\Omega}\in\mathbb{S}_{++}}{\operatorname{argmin}}\frac{n}{2}\ln|\mathbf{\Omega}|-\frac{1}{2}\sum_{g_i=1}(\mathbf{x}_i-\boldsymbol{\mu}^{(1)})^\top\mathbf{\Omega}(\mathbf{x}_i-\boldsymbol{\mu}^{(1)})-\frac{1}{2}\sum_{g_i=2}(\mathbf{x}_i-\boldsymbol{\mu}^{(2)})^\top\mathbf{\Omega}(\mathbf{x}_i-\boldsymbol{\mu}^{(2)})+\lambda||\mathbf{\Omega}||_1,$$

where $\mathbb{S}_{++}$ denotes the set of p-dimensional positive definite matrices and $||\mathbf{\Omega}||_1=\sum_{j\neq\ell}|\omega_{j\ell}|$. It results in $\hat{\boldsymbol{\mu}}^{(1)}=\bar{\mathbf{x}}^{(1)},\hat{\boldsymbol{\mu}}^{(2)}=\bar{\mathbf{x}}^{(2)}$, and

$$\hat{\mathbf{\Omega}}=\underset{\mathbf{\Omega}\in\mathbb{S}_{++}}{\operatorname{argmin}}\frac{n}{2}\ln|\mathbf{\Omega}|-\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i-\bar{\mathbf{x}}^{(g_i)})^\top\mathbf{\Omega}(\mathbf{x}_i-\bar{\mathbf{x}}^{(g_i)})+\lambda||\mathbf{\Omega}||_1. \tag{2.7}$$

This is equivalent to the graphical Lasso for the centered data $\mathbf{x}_1-\bar{\mathbf{x}}^{(g_1)},\ldots,\mathbf{x}_n-\bar{\mathbf{x}}^{(g_n)}$.

Instead of solving (2.7), we can also estimate the graph by neighborhood selection as proposed by Meinshausen and Bühlmann (2006). This method solves p node-wise regularized regressions, viz.

$$\underset{\gamma_0^{(1j)},\gamma_0^{(2j)},\boldsymbol{\gamma}^{(j)}}{\operatorname{argmin}}\frac{1}{2n}||\mathbf{X}_j^{(1)}-\gamma_0^{(1j)}-\mathbf{X}_{-j}^{(1)}\boldsymbol{\gamma}^{(j)}||_2^2+\frac{1}{2n}||\mathbf{X}_j^{(2)}-\gamma_0^{(2j)}-\mathbf{X}_{-j}^{(2)}\boldsymbol{\gamma}^{(j)}||_2^2+\lambda||\boldsymbol{\gamma}^{(j)}||_1,$$

where $\mathbf{X}_j^{(k)}$ denotes the jth feature of sample from group k and $\mathbf{X}_{-j}^{(k)}$ represents the other features. One can verify that

$$\hat{\boldsymbol{\gamma}}^{(j)} = \underset{\boldsymbol{\gamma}^{(j)}}{\operatorname{argmin}} \frac{1}{2n} ||\dot{\mathbf{X}}_j - \gamma_0 - \dot{\mathbf{X}}_{-j}\boldsymbol{\gamma}^{(j)}||_2^2 + \lambda||\boldsymbol{\gamma}^{(j)}||_1, \tag{2.8}$$

where $\dot{\mathbf{X}}$ denotes the data centered by subtracting corresponding group means. We can also use sequential Lasso (Luo and Chen, 2014b) for computational efficiency. The graph $\mathcal{G}$ is constructed by connecting nodes j and ℓ if $\hat{\gamma}_\ell^{(j)} \neq 0$ and/or $\hat{\gamma}_j^{(\ell)} \neq 0$.

Both approaches for estimating $\mathcal{G}$ have been justified theoretically (Meinshausen and Bühlmann, 2006; Yuan and Lin, 2007). We recommend to use neighborhood selection approaches for GSLDA. The main reason is that the former approaches, such as graphical Lasso, usually run through many iterations and can be slow for high-dimensional data ($p > 1000$). In contrast, neighborhood selection approaches only require p penalized regressions. Moreover, our direct interest is not $\boldsymbol{\Omega}$ but the graph $\mathcal{G}$ on which neighborhood selection focuses. We use the extended BIC (EBIC) (Chen and Chen, 2008) to select λ in (2.8). As suggested in Chen and Chen (2008), we choose $1 - 1/(2\log_n p)$ as the EBIC tuning parameter.

When we have an extra unlabeled dataset, denoted as $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}$, the likelihood becomes complicated because of the Gaussian mixture distribution of the unlabeled data. Thus it is difficult to estimate the parameters via likelihood. Moreover, the graph we need is directly related to $\tilde{\boldsymbol{\Omega}} = (X)^{-1}$ rather than $\boldsymbol{\Omega}$. Thus, a penalized likelihood approach is not suitable. Nevertheless, the neighborhood selection approaches are still valid by Lemma 1, because we are concerned with conditional correlation. In particular, we estimate the neighborhoods by

$$\hat{\tilde{\boldsymbol{\gamma}}}^{(j)} = \underset{\tilde{\boldsymbol{\gamma}}^{(j)}}{\operatorname{argmin}} \frac{1}{2(n+m)} ||\tilde{\mathbf{X}}_j - \tilde{\gamma}_0 - \tilde{\mathbf{X}}_{-j}\tilde{\boldsymbol{\gamma}}^{(j)}||_2^2 + \lambda||\tilde{\boldsymbol{\gamma}}^{(j)}||_1,$$

where $\tilde{\mathbf{X}} = (\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m})^\top$ denotes the combined feature matrix. Similarly, we use EBIC to select the tuning parameter λ.

2.3.2 Parameter estimation and tuning parameter selection

Given the graph $\mathcal{G}$, formulation (2.4) is a latent group Lasso problem (Obozinski et al., 2011). It can be transformed to an ordinary group Lasso problem as stated in Problem (2.3). There are many efficient algorithms to solve group Lasso problems, for example, groupwise majorization descent (Yang and Zou, 2015). For very high-dimensional data, we use an iterative proximal algorithm as in Yu and Liu (2016). For implementation, we use cross validation for tuning parameter selection.

2.3.3 Pre-screening

Suppose that there are some entries of $\boldsymbol{\delta}$ being zero. Then $\boldsymbol{\beta}^*$ can be a linear combination of only a few column vectors,

$$\boldsymbol{\beta}^* = \sum_{j \in J} \delta_j \mathbf{w}_j,$$

where $J = \{j : \delta_j \neq 0\}$. Using two-sample t tests for screening, we can specify $J' \subset \{1, \ldots, p\}$, which is a superset of J with a large probability. In particular, we have the following lemma.

Lemma 2. *Define the t-statistic* $T_j = \hat{\delta}_j / \{s_j^{(1)2}/n_1 + s_j^{(2)2}/n_2\}^{1/2}$, *where* $s_j^{(k)2}$ *is the sample variance of feature* $j \in \{1, \ldots, p\}$ *in group* $k \in \{1, 2\}$. *Assume* $\ln p = o(n^\gamma)$, $\ln |J| = o(n^{1/2-\gamma} B_n)$, *and* $\min_{j \in J} |\delta_j| / \sqrt{2\Sigma_{jj}} = B_n / n^\gamma$ *for some* $\gamma \in (0, 1/3)$ *and* $B_n \to \infty$. *Then there exists* $C > 0$ *such that*

$$\lim_{n \to \infty} \Pr \left\{ \min_{j \in J} |T_j| \geq C n^{\gamma/2}, \ \max_{j \notin J} |T_j| < C n^{\gamma/2} \right\} = 1.$$

The result in Lemma 2 was previously obtained by Fan and Fan (2008) and the corresponding proof is omitted. Lemma 2 guarantees the accuracy of our pre-screening procedure.

After feature screening, the proposed regularization can be simplified as follows:

$$||\boldsymbol{\beta}||_{\mathcal{G}_{J'}, \boldsymbol{\tau}} = \min_{\sum_{j \in J'} \mathbf{v}^{(j)} = \boldsymbol{\beta}, \, \mathrm{supp}(\mathbf{v}^{(j)}) \subseteq \mathcal{N}^{(j)}} \sum_{j \in J'} \tau_j ||\mathbf{v}^{(j)}||_2. \tag{2.9}$$

Compared with the original regularization (2.5), the new one in (2.9) is often simpler and enjoys computational advantages. Moreover, the new regularization (2.9) only requires part of the graph, i.e., the part corresponding to the support of $\{\boldsymbol{\omega}_j : j \in J'\}$. Graph estimation methods based on neighborhood selection fit into this idea naturally. When $\boldsymbol{\delta}$ is approximately sparse and $|J'| \ll p$,

the computational cost can be reduced substantially. Unlike the feature screening in Fan and Fan (2008), features outside J' are not necessarily excluded. Instead, they can be introduced into the model via connection with other features in J'.

2.4 Theoretical properties

In this section, we study the theoretical properties of GSLDA. In particular, the original GSLDA in (2.4) with a known graph $\mathcal{G}$ is considered. Since the semi-supervised GSLDA only differs from GSLDA in the graph used, we do not consider it separately. In Section 2.4.1, we show the selection consistency of GSLDA. In Section 2.4.2, we study the misclassification rate of the GSLDA and compare it with the Bayes error.

Before diving into the theoretical analysis, we first introduce some notations for our setting. We define, for an n-dimensional vector $\mathbf{a}$, $||\mathbf{a}||_\infty = \max(|a_1|, \ldots, |a_n|)$; for an $n \times m$ matrix $\mathbf{A}$, $||\mathbf{A}||_\infty = \max_i\{|A_{i1}| + \cdots + |A_{im}|\}$ and $|||\mathbf{A}|||_\infty = \max_{i,j} |A_{ij}|$. We consider the problem setting of standard LDA, in which both within-class populations are Gaussian, i.e., $\mathcal{N}(\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma})$ and $\mathcal{N}(\boldsymbol{\mu}^{(2)}, \boldsymbol{\Sigma})$. The discriminant vector of the Bayes rule, denoted as $\boldsymbol{\beta}^*$, is given in (2.1). Denote $A = \{j : \beta_j^* \neq 0\}$ the active set, and $s = |A|$. Define $\boldsymbol{\beta}^\dagger = \tilde{\boldsymbol{\Omega}}\boldsymbol{\delta}$, then $\boldsymbol{\beta}^\dagger$ is proportional to $\boldsymbol{\beta}^*$ (Proposition 1) and thus defines an equivalent classifier.

2.4.1 Selection consistency

Assume that the feature vectors are centralized, thus $\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2/n + \lambda||\boldsymbol{\beta}||_{\mathcal{G},\boldsymbol{\tau}}$. Denote $\tilde{\mathbf{S}} = \mathbf{X}^\top\mathbf{X}/n$, and $\kappa = ||\tilde{\boldsymbol{\Sigma}}_{A^\complement A}\tilde{\boldsymbol{\Sigma}}_{AA}^{-1}||_\infty$. Define

$$\tilde{\tau}_j = \min_\ell\{\tau_\ell : j \in \mathcal{N}^{(\ell)}\}, \quad \tau^* = \max_{j\in A} \tilde{\tau}_j, \quad \tau_* = \min_{j \in A^\complement} \tau_j |\mathcal{N}^{(j)}|^{-1/2}.$$

We present several assumptions to be used as follows.

(A1) $p = O\{\exp(n^\gamma)\}$, $s = o(n^a)$, for some $\gamma \in (0,1), a \in (0, (1-\gamma)/2)$.

(A2) For every $j \in \{1, \ldots, p\}$, either $\mathcal{N}^{(j)} \subseteq A$ or $\mathcal{N}^{(j)} \subseteq A^\complement$.

(A3) $||\tilde{\boldsymbol{\Sigma}}_{AA}^{-1}||_\infty$ is bounded by $\varphi < \infty$.

(A4) $||\tilde{\mathbf{\Sigma}}_{A^{\complement}A}\tilde{\mathbf{\Sigma}}_{AA}^{-1}||_{\infty} < \tau_*/\tau^*$.

(A5) $b = \min_{j\in A}|\beta_j^{\dagger}| \gg \sqrt{\ln p/n}$.

Here (A1) specifies the order of feature dimension as well as the number of discriminating features. By Assumption (A2), a discriminative feature can only be connected with other discriminative features. This is a reasonable condition in reality since a feature is often relevant for classification if it is related to another useful feature. Condition (A3) ensures that there is no extreme collinearity among discriminative features. Assumption (A4) is an irrepresentability condition that is often employed in showing the selection consistency of regularized estimators (Meinshausen and Bühlmann, 2006; Zhao and Yu, 2006).

It may not be immediately clear why we impose the irrepresentability condition (A4) on $\tilde{\mathbf{\Omega}}$ rather than $\mathbf{\Omega}$. Note that the more similarity between predictive and non-predictive features, the more difficult it is to achieve selection consistency. While $\mathbf{\Omega}$ encodes the within-class feature dependence, the relationship among features in the whole dataset is determined by the overall covariance. Thus we impose the condition on $\tilde{\mathbf{\Omega}}$. The main theoretical result on the selection consistency of the GSLDA is given in the following theorem.

Theorem 1 (Selection consistency)**.** *Under conditions (A1)–(A5), let* $\sqrt{\ln p/n} \le \lambda\tau^* \le O(b)$ *and* n *be sufficiently large, then the GSLDA recovers the active set* A *and* $||\hat{\boldsymbol{\beta}}_A - \boldsymbol{\beta}_A^{\dagger}||_{\infty} = O(\sqrt{\ln p/n})$ *with probability at least* $1 - O(p^{-C_1})$ *for some* $C_1 > 0$.

When we use an empty graph $\mathcal{G}$ and set $\tau_j = 1$ for all j, our GSLDA is equivalent to the DSDA method. In this special case, $\tau^* = \tau_* = 1$, and the selection consistency conditions are similar to those for DSDA (Mai et al., 2012).

2.4.2 Convergence rate

With respect to a classifier, the error rate is one of the most important performance measures. In this section, we investigate the misclassification rate of GSLDA. We first present some basic results on the classification problem. For a linear classifier $g_{\beta_0,\boldsymbol{\beta}}$, denote its classification error under our settings as $Q_{\beta_0,\boldsymbol{\beta}} = \Pr\{g_{\beta_0,\boldsymbol{\beta}}(X) \neq G\}$. Then we have the following results from Cai and Liu (2011).

Lemma 3 (Classification error rate in LDA setting)**.** *Under our setting,*

$$Q_{\beta_0,\boldsymbol{\beta}} = \frac{1}{2}\,\Phi\left(\frac{-\beta_0 - \boldsymbol{\beta}^\top \boldsymbol{\mu}^{(1)}}{\sqrt{\boldsymbol{\beta}^\top \boldsymbol{\Sigma}\boldsymbol{\beta}}}\right) + \frac{1}{2}\,\Phi\left(\frac{\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{\mu}^{(2)}}{\sqrt{\boldsymbol{\beta}^\top \boldsymbol{\Sigma}\boldsymbol{\beta}}}\right),$$

where Φ *denotes the cumulative distribution function of* $\mathcal{N}(0,1)$*. The misclassification rate of the Bayes classifier* $g_{\beta_0^*,\boldsymbol{\beta}^*}$ *is* $Q_{\beta_0^*,\boldsymbol{\beta}^*} = \Phi(-\Delta^{1/2}/2)$*, where* $\Delta = \boldsymbol{\delta}^\top \boldsymbol{\Omega}\boldsymbol{\delta}$*.*

Since Q is a continuous function of β_0 and $\boldsymbol{\beta}$, the misclassification rate of the GSLDA classifier is asymptotically the same as the Bayes error rate, i.e., $Q_{\hat{\beta}_0,\hat{\boldsymbol{\beta}}} \xrightarrow{p} Q_{\beta_0^*,\boldsymbol{\beta}^*}$, as long as $\hat{\boldsymbol{\beta}} \xrightarrow{p} \boldsymbol{\beta}^*$. A more interesting problem is the order of the misclassification rate of the GSLDA when $Q_{\beta_0^*,\boldsymbol{\beta}^*} \to 0$. To investigate this, we first introduce a new condition, under which we can construct an ℓ_2 error bound for the GSLDA estimator.

(A6) Denote $\mathcal{C}(A) = \{\boldsymbol{\Delta} \in \mathbb{R}^p : ||\boldsymbol{\Delta}_{A^\complement}||_{\mathcal{G},\boldsymbol{\tau}} \leq 3||\boldsymbol{\Delta}_A||_{\mathcal{G},\boldsymbol{\tau}}\}$, where $\boldsymbol{\Delta}_A = (\Delta_j \mathbf{1}(j \in A))_{p\times 1}$ and $\boldsymbol{\Delta}_{A^\complement} = (\Delta_j \mathbf{1}(j \notin A))_{p\times 1}$. For all $\boldsymbol{\Delta} \in \mathcal{C}(A)$, $\boldsymbol{\Delta}^\top \tilde{\boldsymbol{\Sigma}}\boldsymbol{\Delta}/\boldsymbol{\Delta}^\top\boldsymbol{\Delta} \geq \sigma > 0$.

This is actually a restricted eigenvalue condition, which is often used in showing the error bound for regularized estimators (Negahban et al., 2010). Compared to the irrepresentability condition (A4), this is much less stringent. With the new condition, we have the following ℓ_2 error bound for the GSLDA estimator.

Theorem 2 (ℓ_2-error bound)**.** *Under conditions (A1)–(A2) and (A6), let* $\lambda \geq 4C_2(1 + ||\boldsymbol{\beta}^\dagger||_1)\sqrt{\ln p/n}$ *for some* $C_2 > 0$ *and* n *be sufficiently large, then* $||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger||_2^2 \leq 9\lambda^2 s\tau^{*2}/\sigma^2$ *with probability at least* $1 - sp^{-C_3}$ *for some* $C_3 > 0$*.*

Based on Theorem 2 above, we can establish the asymptotic error rate of the GSLDA classifier as follows.

Theorem 3 (Convergence rate)**.** *Under conditions (A1)–(A2) and (A6), as* $n, p \to \infty$*, if* $\Delta \to \infty$*, we have*

$$Q_{\beta_0^*,\boldsymbol{\beta}^*} \to 0 \quad \textit{and} \quad Q_{\hat{\beta}_0,\hat{\boldsymbol{\beta}}}/Q_{\beta_0^*,\boldsymbol{\beta}^*} \xrightarrow{p} 1,$$

given $\lambda\tau^* = o[\min\{\lambda_{\max}(\boldsymbol{\Sigma})^{-1}\Delta^{-2}s^{-1/2}||\boldsymbol{\beta}^\dagger||_2^{-1}, \Delta^{-1}s^{-1/2}||\boldsymbol{\delta}||_2^{-1}\}]$ *and* $||\boldsymbol{\beta}^\dagger||_1 = o(n^{1-\gamma}\Delta^{-1})$*, where* Δ *is defined as in Lemma 3 and* $\lambda_{\max}(\boldsymbol{\Sigma})$ *denotes the largest eigenvalue of* $\boldsymbol{\Sigma}$*.*

That is, under mild conditions, the misclassification rate of the GSLDA classifier is of the same order as the Bayes error rate in this case.

2.5 Simulation study

To demonstrate the performance of the GSLDA methods, we compare them with several existing high-dimensional LDA extensions and other classification methods. The methods in comparison include the naive Bayes rule (NB), nearest shrunken centroids (NSC), sparse LDA (SLDA) (Shao et al., 2011), ℓ_1 penalized Logistic regression (PLR), penalized Fisher's discriminant analysis (PLDA) (Witten and Tibshirani, 2011), direct sparse discriminant analysis (DSDA) (Mai et al., 2012), linear programming discriminant (LPD) (Cai and Liu, 2011), and the ROAD (Fan et al., 2012). In particular, the methods NSC, PLR, PLDA and DSDA are implemented with `R` packages `pamr`, `glmnet`, `penalizedLDA` and `dsda`, respectively. We implement the LPD method via the parametric simplex algorithm (Vanderbei et al., 2015) as suggested in Pang et al. (2014).

Besides the above supervised methods, there are many semi-supervised clustering (or classification) methods; see, e.g., Pan and Shen (2007); Zhou et al. (2009); Liu et al. (2013). We have implemented the semi-supervised spectral clustering (SSSC) method proposed in Liu et al. (2013). Both the original and the semi-supervised GSLDA are implemented, and the latter is denoted as GSLDA-S. We also include the GSLDA methods with the true graph, denoted as GSLDA-O (with $\mathcal{G}$) and GSLDA-SO (with $\tilde{\mathcal{G}}$), in the comparison. To make a fair comparison, pre-screening is not employed in the numerical studies. The Bayes rule, denoted as Oracle, is used as a benchmark.

In the simulation, we fix the dimension $p = 200$ and the sample size $n = 200$. The labels $g_1, \ldots, g_n$ are generated with $\pi_1 = \pi_2 = 1/2$ and the features are sampled from $\mathcal{N}(\boldsymbol{\mu}^{(g_i)}, \boldsymbol{\Omega}^{-1})$ based on the labels. Moreover, we generate an independent dataset of sample size 2000 and remove the labels, for the semi-supervised methods. All tuning parameters are selected by 10-fold cross validation. We consider four different feature structures as follows.

Example 1. Blockwise sparse model. In this example, $\boldsymbol{\Sigma}^B$ is a 5×5 matrix with 1 for the diagonal and 0.7 for off-diagonal elements. We use 20 such blocks for the diagonal of the covariance matrix $\boldsymbol{\Sigma}$ and 0 for the rest, and let $\boldsymbol{\Omega} = \boldsymbol{\Sigma}$. The group means are generated such that $\mu_j^{(1)} = 0.5$ for $j \in \{5, 10, \ldots, 25\}$ and $\mu_j^{(1)} = 0$ otherwise; and $\boldsymbol{\mu}^{(2)} = -\boldsymbol{\mu}^{(1)}$.

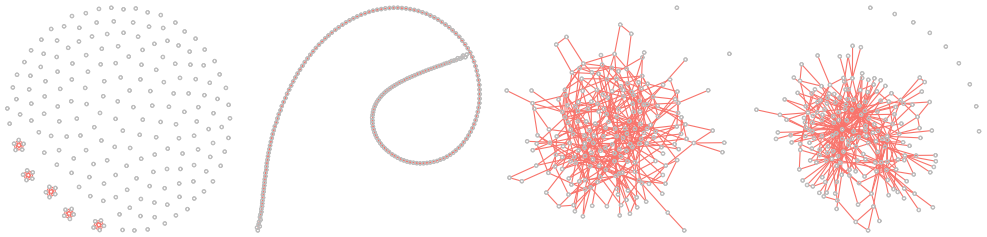

Figure 2.2: The graph structures used in the simulation study. From left to right: the blockwise sparse model, the AR(3) model, the random sparse model, the scale-free model. The last two plots use one realization for demonstration, and the graphs may vary among different realizations.

Example 2. AR(3) model. The precision matrix $\mathbf{\Omega}$ is generated such that $\omega_{jj} = 1$, and $\omega_{j\ell} = -2/3$ if $1 \leq |j-\ell| \leq 3$ and 0 otherwise. The group means are generated such that $\mu_j^{(1)} = 0.75$ for $j \in \{5, 10, \ldots, 25\}$ and $\mu_j^{(1)} = 0$ otherwise; and $\boldsymbol{\mu}^{(2)} = -\boldsymbol{\mu}^{(1)}$.

Example 3. Random sparse model. The graph $\mathcal{G}$ is generated in such a way that any two nodes are connected with probability 0.05. Based on $\mathcal{G}$, we generate the precision matrix $\mathbf{\Omega}$ by setting $\omega_{j\ell} = -0.5$ for all connected j and ℓ in the graph and 0 otherwise. We add $c\,\mathbf{I}_p$, where $c > 0$ and $\mathbf{I}_p$ is an identity matrix, to $\mathbf{\Omega}$ such that the eigenvalues are between 0 and 1. We standardize $\mathbf{\Omega}$ so that its diagonal elements are all 1. The group means are generated in such a way that $\mu_j^{(1)} = 0.75$ for all $j \in S$ and 0 otherwise; and $\boldsymbol{\mu}^{(2)} = -\boldsymbol{\mu}^{(1)}$.

Example 4. Scale-free random graph. The graph is generated in a way similar to the Barabasi–Albert (BA) model. Starting from an identity matrix $\mathbf{L} \in \mathbb{R}^{p\times p}$, at step i we randomly assign -0.5 to $\min\{\lfloor 0.05p \rfloor, i-1\}$ entries in row i with probability $\Pr(i,j) \propto \#\{L_{\ell j} \neq 0 : 1 \leq \ell \leq p\}, j < i$. Repeat the procedure until $i = p$. Then we get a lower triangular matrix. We construct $\mathbf{\Omega} = \mathbf{L}^\top \mathbf{L}$ and standardize it such that the eigenvalues are between 0 and 1. Denote the 6th to 10th most connected nodes as J. The group means are generated such that $\mu_j^{(1)} = 0.75$ for all $j \in J$ and 0 otherwise; and $\boldsymbol{\mu}^{(2)} = -\boldsymbol{\mu}^{(1)}$.

All four graph structures are displayed in Figure 2.2. The first two examples are fixed while the last two produce random graphs. Compared with the random sparse model, the scale-free random graphs are featured with hubs. For each graph structure, we repeat the simulation for 100 times

Table 2.1: Performance comparisons of different classification methods for Example 1.

	Error	FP	FN	Size
NB	27.01 (0.18)	—	—	—
NSC	14.17 (0.11)	0.71 (0.54)	20.27 (0.13)	5.44 (0.62)
SLDA	10.28 (0.16)	5.71 (1.29)	12.53 (0.31)	18.18 (1.61)
PLR	7.17 (0.13)	14.73 (0.56)	8.1 (0.24)	31.63 (0.69)
DSDA	6.76 (0.13)	23.26 (1.53)	6.79 (0.27)	41.47 (1.71)
LPD	7.80 (0.38)	37.20 (1.97)	5.73 (0.29)	56.47 (2.17)
ROAD	6.54 (0.12)	23.45 (1.24)	6.01 (0.24)	42.44 (1.37)
PLDA	14.16 (0.10)	3.62 (1.16)	19.53 (0.16)	9.09 (1.29)
SSSC	8.11 (0.10)	—	—	—
GSLDA	5.57 (0.07)	20.48 (2.17)	7.31 (0.25)	38.17 (2.33)
GSLDA-S	4.53 (0.06)	18.79 (2.26)	0.74 (0.11)	43.05 (2.29)
GSLDA-O	4.86 (0.08)	18.55 (2.63)	0 (0)	43.55 (2.63)
GSLDA-SO	4.52 (0.07)	16.43 (1.93)	0 (0)	41.43 (1.93)
Oracle	3.27 (0.01)	0 (0)	0 (0)	25 (0)

and evaluate the performance, both prediction and selection accuracy, of all classification methods. Table A.1 in the Appendix displays the graph estimation accuracy for all examples.

Tables 2.1–2.4 give a summary of the performance comparison of all methods in Examples 1 and 4. In particular, misclassification rates in percentage (Error), false positives (FP) and false negatives (FN) of $\boldsymbol{\beta}$ estimation are computed. The misclassification rate is evaluated based on an independent test dataset of size 20,000. All metrics are averaged over 100 simulations and the numbers within parentheses are the standard errors. Both the NB and the SSSC are not considered in the comparison of variable selection, since these methods do not perform variable selection.

From Tables 2.1–2.4, we observe that the two plug-in extensions of LDA, namely the naive Bayes and the NSC, perform worse than ℓ_1 penalized logistic regression and other direct LDA methods under these settings. This is expected because there is substantial correlation among the features while both the plug-in extensions of LDA use diagonal estimates of $\boldsymbol{\Sigma}$. In contrast, the performance of the direct LDA methods varies across the settings. For example, the DSDA has lower misclassification rates than the ROAD in most cases, while ROAD has better classification accuracy in Example 1. Utilizing the graph structures, high-dimensional LDA is further improved in GSLDA. As we can see from the results, GSLDA methods have the best performance among all methods in these four settings. In particular, the GSLDA method has lower misclassification rates than all other methods except its semi-supervised variant. Since the DSDA is the special case of

Table 2.2: Performance comparisons of different classification methods for Example 2.

	Error	FP	FN	Size
NB	36.59 (0.43)	—	—	—
NSC	17.46 (0.14)	42.75 (2.16)	25.96 (0.45)	55.79 (2.48)
SLDA	14.39 (0.12)	19.28 (1.72)	17.59 (0.43)	40.69 (2.17)
PLR	7.86 (0.11)	15.83 (0.42)	20.58 (0.29)	34.25 (0.54)
DSDA	6.96 (0.09)	25.13 (1.22)	17.21 (0.38)	46.92 (1.46)
LPD	8.84 (0.69)	34.48 (1.56)	17.98 (0.48)	55.50 (1.97)
ROAD	7.42 (0.12)	25.16 (0.98)	17.36 (0.35)	46.80 (1.17)
PLDA	16.48 (0.12)	2.26 (0.48)	32.69 (0.14)	8.57 (0.57)
SSSC	9.27 (0.17)	—	—	—
GSLDA	6.60 (0.10)	25.48 (1.83)	15.41 (0.43)	49.07 (2.19)
GSLDA-S	5.56 (0.07)	34.43 (2.52)	3.33 (0.41)	70.1 (2.77)
GSLDA-O	6.19 (0.09)	27.26 (1.72)	7.37 (0.47)	58.89 (2.08)
GSLDA-SO	5.79 (0.07)	30.78 (1.94)	2.16 (0.39)	67.62 (2.31)
Oracle	3.32 (0.01)	0 (0)	0 (0)	39 (0)

Table 2.3: Performance comparisons of different classification methods for Example 3.

	Error	FP	FN	Size
NB	36.86 (0.80)	—	—	—
NSC	24.16 (0.84)	29.15 (3.35)	44.78 (1.62)	50.37 (4.93)
SLDA	13.28 (0.72)	21.07 (2.29)	40.59 (1.57)	46.48 (3.87)
PLR	11.09 (0.12)	21.44 (0.56)	42.08 (0.39)	45.36 (0.75)
DSDA	10.94 (0.15)	30.32 (1.49)	38.12 (0.63)	58.20 (2.01)
LPD	13.19 (0.73)	41.67 (1.52)	39.84 (0.82)	67.83 (2.25)
ROAD	11.25 (0.15)	33.14 (1.46)	37.53 (0.55)	61.61 (1.92)
PLDA	26.31 (0.68)	22.34 (2.33)	50.89 (1.12)	37.45 (3.41)
SSSC	13.57 (0.91)	—	—	—
GSLDA	10.53 (0.10)	27.34 (1.91)	36.67 (0.85)	56.67 (2.67)
GSLDA-S	8.77 (0.08)	34.08 (2.77)	18.2 (0.72)	81.88 (3.37)
GSLDA-O	9.77 (0.08)	36.87 (2.54)	26.22 (0.78)	76.65 (3.27)
GSLDA-SO	8.91 (0.08)	35.17 (2.37)	16.31 (0.63)	84.86 (3.01)
Oracle	5.36 (0.02)	0 (0)	0 (0)	66 (0)

the GSLDA with an empty graph, it is a good benchmark to quantify the benefit of using graph structures. In most cases, the GSLDA provides better model selection than the DSDA. Therefore, utilizing the graph structure does help us to improve the LDA classifier in high dimensions.

With respect to the semi-supervised GSLDA, due to the large amount of unlabeled data, it often has better graph estimation and yields more accurate classifiers. In fact, the semi-supervised GSLDA has the lowest misclassification rates among all methods in all cases. Furthermore, the

Table 2.4: Performance comparisons of different classification methods for Example 4.

	Error	FP	FN	Size
NB	32.84 (0.28)	—	—	—
NSC	22.78 (0.12)	8.62 (1.76)	48.87 (0.76)	18.75 (2.49)
SLDA	17.53 (0.27)	19.83 (1.29)	38.23 (0.67)	40.60 (1.98)
PLR	14.60 (0.21)	16.51 (0.66)	35.5 (0.5)	40.01 (1.06)
DSDA	13.48 (0.17)	33.71 (1.9)	28.18 (0.65)	64.53 (2.47)
LPD	16.87 (0.36)	46.86 (1.66)	29.17 (0.71)	76.69 (2.31)
ROAD	13.95 (0.19)	36.9 (2.01)	27.71 (0.77)	68.19 (2.69)
PLDA	22.64 (0.12)	6.6 (1.06)	51.58 (0.45)	14.02 (1.48)
SSSC	12.08 (0.21)	—	—	—
GSLDA	10.46 (0.11)	21.53 (1.7)	15.69 (0.55)	64.84 (2.14)
GSLDA-S	9.15 (0.12)	12.87 (1.29)	5.03 (0.54)	66.84 (1.57)
GSLDA-O	10.39 (0.18)	28.29 (1.73)	19.44 (0.8)	67.85 (2.43)
GSLDA-SO	9.36 (0.17)	19.87 (1.69)	5.47 (0.71)	73.05 (2.31)
Oracle	4.62 (0.02)	0 (0)	0 (0)	59 (0)

semi-supervised GSLDA has superior model selection over the original GSLDA in most cases. This demonstrates the advantages of using unlabeled data.

We notice that models estimated by the semi-supervised GSLDA often have larger sizes, sometimes more false positives in coefficient vectors, than the original GSLDA classifiers. This is probably because the graph used in the semi-supervised GSLDA often has more edges. There are two possible reasons: (i) the true graph $\tilde{\mathcal{G}}$ corresponding to $\tilde{\mathbf{\Omega}}$ has more edges than $\mathcal{G}$, and (ii) graph estimation based on unlabeled data uses a much larger training dataset which often leads to denser graphs estimate. While a denser graph estimate may recover more connections among the features, it can also result in more false edges. This effect is enhanced by the difficulty of graph estimation with unlabeled data. As a consequence, the semi-supervised GSLDA may suffer from more false positives, as shown in Examples 2 and 3. To resolve this issue, we may consider to use more conservative graph estimation for the semi-supervised GSLDA.

2.6 Real data analysis

In this section, we implement our methods and several other existing classifiers on two real datasets. The first dataset is a genetic dataset with very high dimensions, and the second one consists of images of handwritten digits. We estimate the graphs from labeled training data and

unlabeled data. We find that GSLDA methods have a good performance in both datasets and utilizing the feature structure is beneficial.

2.6.1 Arcene cancer data

Nowadays, genetic diagnosis is an important tool in the clinical study and medical practice. By using the genetic information, we can estimate the potential risk of cancer for healthy people or determine cancer subtypes for patients. The Arcene dataset is a gene dataset of 88 cancer patients and 112 healthy individuals. The dataset contains 10,000 features and was originally used in the NIPS 2003 feature selection challenge (`https://archive.ics.uci.edu/ml/data\-sets/Arcene`). Out of the 10,000 features, 7000 are real genes while the other 3000 are noise features that have no predictive power and make the prediction harder. Besides the labeled data, there is an unlabeled dataset of 700 individuals, which is used to construct a graph for GSLDA-S. As in the previous simulation studies, we apply the GSLDA and other methods on the dataset.

The labeled data are randomly split into a training set and a test set, of sizes 150 and 50, respectively. All methods except the naive Bayes are tuned by 10-fold cross validation. The experiment is repeated 100 times and the results are summarized in Table 2.5.

Table 2.5: Comparison of GSLDA and other methods on the Arcene dataset.

	Error	Size
NB	35.50 (0.62)	—
NSC	36.05 (0.61)	9934.46 (9.06)
SLDA	34.64 (0.73)	297 (4.17)
PLR	28.36 (0.65)	16.57 (0.90)
DSDA	28.29 (0.72)	30.96 (2.69)
LPD	31.59 (1.33)	10.95 (3.58)
ROAD	29.29 (0.64)	31.86 (3.43)
PLDA	34.36 (0.61)	9.39 (1.63)
SSSC	27.93 (0.83)	—
GSLDA	**22.57 (0.70)**	229.36 (6.39)
GSLDA-S	24.50 (0.68)	319.57 (8.37)

From Table 2.5, we can see that both GSLDA and semi-supervised GSLDA outperform other methods in prediction. Although semi-supervised GSLDA uses more data for graph estimation, its performance is inferior to GSLDA for this application, possibly due to the difficulty of graph estimation based on unlabeled data. In addition, the size of the unlabeled dataset is not substantially

Table 2.6: Comparison of GSLDA and other methods on the Semeion dataset.

	Error	Size
NB	13.81 (0.34)	—
NSC	15.21 (0.44)	84.74 (11.31)
SLDA	14.43 (0.67)	20.23 (2.80)
PLR	18.69 (0.88)	9.46 (0.40)
DSDA	13.76 (0.66)	16.76 (1.01)
LPD	17.15 (0.86)	15.32 (0.91)
ROAD	19.73 (0.98)	15.38 (1.25)
SSSC	13.97 (0.75)	—
GSLDA	12.65 (0.61)	28.46 (1.45)
GSLDA-S	**11.23 (0.56)**	33.28 (1.32)

larger than that of the labeled dataset. Compared with PLR, DSDA and ROAD, our methods have significantly larger model sizes. This may indicate that many genes are related to each other. It is likely that those genes contribute to cancer together, and including all of them in modeling can potentially make the classifier more robust. This characteristic may also contribute to the good performance of the proposed two GSLDA methods.

2.6.2 Semeion handwritten digits dataset

The Semeion dataset (`https://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+Digit`) consists of 1593 images of handwritten digits. Each digit is in the form of a 16×16 grayscale image and saved as a vector of 256 features. We take a subset of the dataset that only contains digits 1 and 7, which are generally difficult to distinguish. We randomly choose 40 images for training, and 80 for graph estimation of the semi-supervised GSLDA after removing labels. The remaining 200 images are used for testing. Other settings are the same as the cancer example. Table 6 gives a summary of the results.

As shown in Table 2.6, the semi-supervised GSLDA has excellent performance for this problem. It has the lowest misclassification rate among all methods in comparison. The original GSLDA method also has good classification accuracy for this problem. Moreover, we can see that both GSLDA methods have larger model sizes than other direct LDA methods, as in the previous analysis in Section 2.6.1.

2.7 Discussion

With many extensions in the literature, LDA can be readily applied to high-dimensional classification problems. In particular, the direct approaches of high-dimensional LDA are attractive due to their simplicity and good performance. Under the standard setting of LDA problems, we explore the relationship between the graph structure of features and the optimal discriminant vector $\boldsymbol{\beta}^*$. Our study shows that, by taking advantage of such structure, we can get better LDA classifiers in high dimensions. Based on this idea, we propose the GSLDA method. After investigating the overall graph structure of the Gaussian mixture population for unlabeled data, we further propose the semi-supervised GSLDA that can utilize unlabeled data. Both GSLDA methods have been evaluated on simulated and real data, which demonstrate the advantages of utilizing the graph structures. Moreover, we conclude that the performance of semi-supervised GSLDA depends on both the size of the unlabeled dataset and the graph complexity. When the graph structure is very complex, it is better to consider a conservative graph estimate for GSLDA. Finally, our focus in this chapter is on binary problems. It will be useful to extend the methods for multicategory problems.

CHAPTER 3

Joint Skeleton Estimation of Multiple Directed Acyclic Graphs for Heterogeneous Population

3.1 Introduction

Our method development is motivated by the problem of analyzing genome-wide gene expression data. In a typical gene expression study at human population, the variables of interest are the expression of $10,000 - 50,000$ genes, and the sample size is around a few hundreds. Co-expression of two genes implies a regulatory effect (one gene regulates the expression of the other gene) or that the two genes share some regulatory components. The co-expression of all the genes can be conveniently studied by a directed acyclic graph (DAG) in which a node represents a gene and directed edges specify regulatory effects. These graphs can be very useful for understanding the molecular basis of a disease or to prioritize drug target. For example, if disrupting a specific gene helps treating a certain type of cancer but no drug is available to target this gene, one may use a directed graph model to identify the immediate parent nodes of this gene as potential drug targets.

Despite the effectiveness of these aforementioned methods, they were originally developed for skeleton estimation of a homogeneous population. In practice, the samples may come from a heterogeneous population. For example, when we study gene expression data of patients with a certain type of cancer, the patients may belong to different subtypes. The co-expression pattern of genes, hence the DAG model, may vary across subtypes. Though we can estimate a DAG for each subtype separately, joint estimation can be more efficient by exploiting DAG similarities across subtypes. There have been extensive studies on the joint estimation of multiple Gaussian graphical models (Guo et al., 2011; Danaher et al., 2014). In contrast, not much work has been done for joint DAG or skeleton estimation (Oates et al., 2016). Furthermore, in practice, clustering and classification methods are commonly used to label samples into different classes with potential errors. However, these labels are often used as if they were true for multiple graph estimation. To

the best of our knowledge, no previous work has considered clustering or classification errors in graph estimation.

In this chapter, we propose a new method, MPenPC, to jointly estimate multiple skeletons for high dimensional variables measured in a set of heterogeneous samples. The MPenPC is a two-step method. It first jointly estimates the conditional independence graphs for multiple classes, and then applies the PC-stable algorithm to construct the skeletons. The MPenPC can accommodate both *hard labels* (i.e., discrete class assignment) or *soft labels* (i.e., posterior probability) for class assignments. Specifically, to use soft labels, we first estimate probabilistic class labels by clustering or classification, and then use them as weights in both steps of MPenPC. This approach benefits skeleton estimation by mitigating the impact of mistaken hard labels, as we will demonstrate in this chapter.

The rest of this chapter is organized as follows. In Section 3.2, we review some important properties of the DAG skeleton and existing estimation methods, then propose our method MPenPC. In Section 3.3, we give some implementation details of MPenPC. In Section 3.4, we discuss some theoretical properties of MPenPC. In Sections 3.5 and 3.6, we evaluate the performance of MPenPC by simulations and real data analysis, respectively. We conclude this chapter with some discussions on possible generalizations of MPenPC in Section 3.7.

3.2 Methodology

We first give an overview of DAG skeleton estimation methods under Gaussian settings in Section 3.2.1. Then in Sections 3.2.2 and 3.2.3, we introduce our MPenPC methods using hard and soft labels, respectively.

3.2.1 Review of DAG estimation

We first review some key concepts and properties of the DAG that will be used in this chapter. A node l is a *parent* of node j if there is an edge $l \to j$, and j is called a *child* of l. If two unconnected nodes are parents of a common child ($i \to j \leftarrow l$), then they form a *v-structure*. The *skeleton* of a DAG is the undirected graph formed by removing directions of all the edges in the DAG. For any DAG $\mathcal{D}$, we denote its skeleton by $\mathcal{D}^u$. There are often more than one DAGs that

can describe the conditional dependence embedded in a probability distribution. These DAGs are probabilistically equivalent, and they form a *Markov equivalence class*. It can be shown that two DAGs belong to a Markov equivalence class if and only if they share the skeleton and v-structures (Chickering, 2002).

We consider a DAG model, denoted by $\mathcal{D}$, for p random variables $X_1, \cdots, X_p$ under Gaussian settings. Given a sample $\{\mathbf{x}_i; i = 1, \cdots, n\}$, we assume $\mathbf{x}_i \overset{iid}{\sim} N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and denote the precision matrix by $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$. Given the DAG, the data generating process can be specified by a set of structure equations:

$$X_j = \nu_j + \sum_{l \in \mathbf{pa}_j} b_{j,l} X_l + Z_j, \tag{3.1}$$

for $j = 1, \cdots, p$, where $\mathbf{pa}_j$ represents the set of parent nodes of j, and $Z_j \sim N(0, \lambda_j^2)$ independently for all j. Let $X = (X_1, ..., X_p)^T$, $\boldsymbol{\nu} = (\nu_1, ..., \nu_p)^T$, and $\mathbf{Z} = (Z_1, ..., Z_p)^T$. The p structure equations can be rewritten in a concise form: $X = \boldsymbol{\nu} + \mathbf{B}X + \mathbf{Z}$, where $B_{j,l} = b_{j,l}$ for $l \in \mathbf{pa}_j$ and 0 otherwise. The direct results of the model equivalence are $\boldsymbol{\mu} = (\mathbf{I} - \mathbf{B})\boldsymbol{\nu}$, $\boldsymbol{\Sigma} = (\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\Lambda}(\mathbf{I} - \mathbf{B})^{-T}$, and

$$\boldsymbol{\Omega} = (\mathbf{I} - \mathbf{B})^T \boldsymbol{\Lambda}^{-1} (\mathbf{I} - \mathbf{B}), \tag{3.2}$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1^2, \cdots, \lambda_p^2)$. Let $\mathcal{G}$ be the conditional independence graph (CIG) for $X_1, \cdots, X_p$, then nodes j and l are not connected if and only if $\Omega_{j,l} = \Omega_{l,j} = 0$. Owing to equation (3.2), $\mathcal{G}$ is also a sparse graph because of the sparsity of $\mathbf{B}$. In fact, we can establish a relationship between the DAG $\mathcal{D}$ and $\mathcal{G}$ as follows.

Lemma 4. *For any two nodes $j, l \in \{1, \cdots, p\}$,*

1. *there is an edge $l \to j$ in $\mathcal{D}$ only when there is an edge $l - j$ in $\mathcal{G}$;*
2. *if there is an edge $j - l$ in $\mathcal{G}$, then j and l are either connected in $\mathcal{D}$, or they are parents of a v-structure in $\mathcal{D}$.*

We refer readers to Chapter 3.7 of Spirtes et al. (2000) for a proof of Lemma 4. Ha et al. (2016) took advantage of this relationship and proposed PenPC for skeleton estimation. It consists of two steps: i) estimate $\mathcal{G}$ by neighborhood selection, denoted by $\hat{\mathcal{G}}$, and ii) estimate the skeleton on the basis of $\hat{\mathcal{G}}$. The first step is a Gaussian graphical model estimation problem, which is well

studied in the literature (Meinshausen and Bühlmann, 2006; Yuan and Lin, 2007). Given a sparse $\hat{\mathcal{G}}$ estimated by neighborhood selection, the skeleton can be efficiently estimated by a modified PC-stable algorithm. In high dimensional cases, the PenPC method has shown significant advantages over the PC-stable algorithm, both in terms of accuracy and computational efficiency.

3.2.2 Joint estimation of multiple skeletons with hard labels

Denote the observed data by $\{(\mathbf{x}_i, g_i); i = 1, \cdots, n\}$, where $\mathbf{x}_i \in \mathbb{R}^p$ and $g_i \in \{1, \cdots, K\}$ are respectively the feature vector and the population label of the sample i. We assume the feature vector X is conditional Gaussian, i.e., $X|(G = k) \sim N(\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)})$ for $k = 1, \ldots, K$. For the k-th population, we denote the DAG and its skeleton by $\mathcal{D}^{(k)}$ and $\mathcal{D}_u^{(k)}$, the precision matrix by $\boldsymbol{\Omega}^{(k)}$, and the CIG by $\mathcal{G}^{(k)}$.

One can estimate a DAG skeleton for each population separately. However, a joint estimation approach can be more efficient when there is certain similarity across the DAGs. We propose a two-step method, the multi-PenPC (MPenPC), for multiple-skeleton estimation by extending the PenPC method. At the first step, we use joint neighborhood selection to estimate the CIGs (Meinshausen and Bühlmann, 2006). For any node j, the node-wise regression models of K populations are

$$X_j^{(k)} = \gamma_{j,0}^{(k)} + \sum_{l \neq j} \gamma_{j,l}^{(k)} X_l^{(k)} + \epsilon_j^{(k)}; \quad k = 1, \cdots, K, \tag{3.3}$$

where $X_j^{(k)}$ is the feature j of population k, and $\epsilon_j^{(k)}$ is an error term. For each class k, we define the coefficient vector $\boldsymbol{\gamma}_j^{(k)} = (\gamma_{j,1}^{(k)}, \cdots, \gamma_{j,j-1}^{(k)}, \gamma_{j,j+1}^{(k)}, \cdots, \gamma_{j,p}^{(k)})^T \in \mathbb{R}^{p-1}$. Denote $\boldsymbol{\gamma}_j^{*(k)} = -\boldsymbol{\Omega}_{-j,j}^{(k)}/\Omega_{j,j}^{(k)}$ the true coefficient vector, then $\gamma_{j,l}^{*(k)} \neq 0$ if and only if the edge $j - l$ exists in $\mathcal{G}^{(k)}$. Thus we can recover the graph $\mathcal{G}^{(k)}$ by estimating $\{\boldsymbol{\gamma}_j^{*(k)}; j = 1, \cdots, p\}$.

Denote $\mathbf{X}^{(k)}$ the feature matrix of the sample from population k, $\mathbf{X}_j^{(k)}$ the j-th column of $\mathbf{X}^{(k)}$, and $\mathbf{X}_{-j}^{(k)}$ the feature matrix without the j-th column. Then a joint neighborhood selection method can be formulated as

$$\underset{\boldsymbol{\gamma}_j}{\operatorname{argmin}} \frac{1}{2n} \sum_{k=1}^{K} ||\mathbf{X}_j^{(k)} - \gamma_{j,0}^{(k)} - \mathbf{X}_{-j}^{(k)} \boldsymbol{\gamma}_j^{(k)}||_2^2 + P(\boldsymbol{\gamma}_j), \tag{3.4}$$

where $\boldsymbol{\gamma}_j = (\gamma_{j,0}^{(1)}, \boldsymbol{\gamma}_j^{(1)T}, \cdots, \gamma_{j,0}^{(K)}, \boldsymbol{\gamma}_j^{(K)T})^T$. If the penalty function is defined as a summation of penalties across the K groups, i.e., $P(\boldsymbol{\gamma}_j) = \sum_{k=1}^{K} P^{(k)}(\boldsymbol{\gamma}_j^{(k)})$, then (3.4) is equivalent to estimating the neighborhood of node j separately for K populations. Thus $P(\boldsymbol{\gamma}_j)$ should possess a group selection effect to induce similar sparsity patterns for $\boldsymbol{\gamma}_j^{(k)}$'s.

There are various group penalties that encourage certain parameters to be zero or non-zero simultaneously. Some of them, including the group lasso (Yuan and Lin, 2006), are rigid in the sense that parameters in one group usually are either all zero or all non-zero. Whereas some others, such as group bridge and group exponential penalties (Huang et al., 2009; Breheny, 2015), can achieve a bi-level selection effect, i.e., the parameters in one group can contain both zero and non-zero values. In our case, the graphs are often only partially overlapped, so an edge may or may not be shared by all DAGs, and thus a group penalty with the bi-level selection effect is more appropriate. In this chapter, we employ the group exponential (GEL) penalty (Breheny, 2015) and the regularization in (3.4) becomes

$$P(\boldsymbol{\gamma}_j) = P_{\lambda,\tau}(\boldsymbol{\gamma}_j) = \lambda^2 \tau^{-1} \sum_{l \neq j} \{1 - \exp(-\lambda^{-1}\tau ||\boldsymbol{\gamma}_{j,l}||_1)\},$$

where $\boldsymbol{\gamma}_{j,l} = (\gamma_{j,l}^{(1)}, \cdots, \gamma_{j,l}^{(K)})^T$. The tuning parameters λ and τ can be chosen by the extended BIC (Chen and Chen, 2008). When there is only one class, the regularization becomes $P(\boldsymbol{\gamma}_j) = \lambda^2 \tau^{-1} \sum_{l \neq j} \{1 - \exp(-\lambda^{-1}\tau |\gamma_{j,l}|)\}$, which is non-convex and the resulting estimator has oracle properties (Fan and Li, 2001).

For the second step of our proposed MPenPC method, we apply the PC-stable algorithm (Colombo and Maathuis, 2014) for skeleton estimation while using $\{\hat{\mathcal{G}}^{(k)}; k = 1, \cdots, K\}$ as initial graphs. For completeness, we describe the implementation of the algorithm as follows. Assume we are estimating a DAG skeleton of p nodes using $\mathcal{G}$ as the initial graph, and a p-value cutoff α. Denote $\mathcal{N}(j)$ the neighborhood of node j in $\mathcal{G}$. Given an ordering of the p nodes, denoted by ORDER(p), the search of ordered node pairs will be based on ORDER(p). Starting with $s = 0$, we search over all ordered node pairs (j, l) such that $l \in \mathcal{N}(j)$ and $|\mathcal{N}(j) \setminus l| \geq s$. For each pair (j, l), we search over all size-s subsets of $\mathcal{N}(j) \setminus l$ for a d-separation set S, i.e. $(X_j \perp\!\!\!\perp X_l)|X_S$, by partial correlation tests. If we can find such a d-separation set $S(j, l)$, we record the separation set and stop searching. After searching over all the node pairs, we update $\mathcal{G}$ by deleting all edges between

node pairs with d-separation sets. Then we increase s by 1 and continue the procedure until each pair of adjacent nodes (j, l) satisfies $|\mathcal{N}(j) \setminus l| \leq s$. The resulting $\mathcal{G}$ is our estimated skeleton. With the d-separation sets $\{S(j, l)\}$, we can direct a subset of the edges by a set of deterministic rules.

3.2.3 Joint estimation of multiple skeletons with soft labels

In many real data analysis settings, the samples are labeled by experts or statistical methods (e.g., clustering or classification) with non-ignorable error rates. For example, breast cancer patients are typically classified into four major subtypes based on gene expression data, and some patients cannot be confidently classified into any subtype (Dai et al., 2015). To address this challenge, we propose to use *soft labeling*: instead of assigning a hard label to each observation, a probabilistic label vector $(w^{(1)}, \cdots, w^{(K)})^T$ is computed such that $w^{(k)} \approx \Pr(G = k|X)$. We refer to the method based on probabilistic labeling as *Soft* MPenPC.

Soft labels can be produced by either classification or clustering. When prior information or labels are available, we can compute probabilistic labels by soft classifiers, such as naive Bayes or quadratic discriminant analysis (QDA). Otherwise, we can estimate soft labels using probabilistic clustering methods, or apply soft classifiers on the clustered sample. The following toy example demonstrates that soft labels can provide more accurate estimates of class labels than hard labels. Let $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ represents n samples collected from a mixture of three Gaussian populations with equal weights. For any $k \in \{1, 2, 3\}$, the feature vector in class k has a Gaussian distribution $N(\boldsymbol{\mu}^{(k)}, \mathbf{I})$. Specifically, $\boldsymbol{\mu}^{(1)} = (2, 0)^T, \boldsymbol{\mu}^{(2)} = (0, 2)^T$, and $\boldsymbol{\mu}^{(3)} = (\sqrt{3} + 1, \sqrt{3} + 1)^T$. We create hard labels by k-means clustering with $k = 3$. Then we construct probabilistic labels using a naive Bayes classifier based on the clustering labels. We measure the accuracy of the estimated class labels by their average Manhattan distance from the true labels: $\sum_{i=1}^{n} \sum_{k=1}^{K} |\hat{w}_i^{(k)} - \mathbb{I}(g_i = k)|/n$, where $\hat{w}_i^{(k)} = \mathbb{I}(\hat{g}_i = k)$ for hard labels. The results show that soft labels often provide more accurate estimates of class labels (see Figure 1 in the Supplementary Materials).

Soft labels can be naturally incorporated into both steps of the MPenPC. Denote the full feature matrix $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)^T$, $n^{(k)} = \sum_{i=1}^{n} w_i^{(k)}$, $\mathbf{w}^{(k)} = (w_1^{(k)}, \cdots, w_n^{(k)})^T$, $\mathbf{W}^{(k)} = \text{diag}(\mathbf{w}^{(k)})$, and $\bar{\mathbf{x}}^{(k)} = \mathbf{X}^T \mathbf{w}^{(k)}/n^{(k)}$. In the neighborhood selection step, we use soft labels for weighted nodewise

regressions. That is, for node j we estimate its neighborhood by

$$\underset{\boldsymbol{\gamma}_j}{\operatorname{argmin}} \frac{1}{2n} \sum_{k=1}^{K} ||\mathbf{X}_j - \gamma_{j,0}^{(k)} - \mathbf{X}_{-j}\boldsymbol{\gamma}_j^{(k)})||^2_{\mathbf{W}^{(k)}} + P_{\lambda,\tau}(\boldsymbol{\gamma}_j), \tag{3.5}$$

where $\mathbf{X}_j$ denotes the j-th column of $\mathbf{X}$, and $||\mathbf{a}||_{\mathbf{W}} = (\mathbf{a}^T\mathbf{W}\mathbf{a})^{1/2}$. In the PC step, we can compute a weighted covariance matrix with soft labels: $\hat{\boldsymbol{\Sigma}}^{(k)} = (\mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^{(k)T})^T\mathbf{W}^{(k)}(\mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^{(k)T})/n^{(k)}$, for $k = 1, \cdots, K$. Based on the weighted covariance matrices, we can perform partial correlation tests and apply the PC-stable algorithm. In the partial correlation tests for population k, we use $n^{(k)}$ as the sample size.

3.3 Computation of soft labels and tuning parameter selection

In this section, we discuss some implementation details of the MPenPC methods. The neighborhood selection step is implemented based on the `grpreg` R package. The PC-stable algorithm in the skeleton estimation step is implemented using the `ParallelPC` R package.

3.3.1 Computation of soft labels

Given hard labels, which are often estimated by clustering, we can compute soft labels by probabilistic classification methods, for example, QDA and naive Bayes. Since we are interested in high dimensional problems, dimension reduction has to be done prior to applying methods like QDA. When hard labels are not available, we can either construct them by clustering, or estimate the soft labels directly by probabilistic clustering methods, such as Gaussian mixture models. From our experience, it is often better to perform dimension reduction before clustering, for example, by principal component analysis.

3.3.2 Parameter tuning

In the neighborhood selection step of MPenPC, we use extended BIC (EBIC) (Chen and Chen, 2008) for parameter tuning of λ and τ. For standard regression, EBIC is defined as

$$\text{EBIC}_\gamma = -2\log L_n(\hat{\boldsymbol{\beta}}) + s\log n + 2\psi\log\binom{p}{s}, 0 \le \psi \le 1,$$

where L_n denotes the likelihood, and $s = ||\hat{\boldsymbol{\beta}}||_0$. As suggested by Chen and Chen (2008), ψ can be taken as $1 - (2\log p/\log n)^{-1}$. For the nodewise regression in our proposed MPenPC, we consider the joint regression as a model with $(p-1)K$ parameters. Following the notations of (3.4), EBIC for the nodewise regression can be defined as

$$\text{EBIC}_{j,\gamma} = n\log\hat{\sigma}^2 + s\log n + 2\psi\log\binom{(p-1)K}{s},$$

where $s = \sum^{(k)} ||\hat{\boldsymbol{\gamma}}_j^{(k)}||_0$, and $\hat{\sigma}^2 = n^{-1}\sum_{k=1}^K ||\mathbf{X}_j^{(k)} - \hat{\gamma}_{j,0}^{(k)} - \mathbf{X}^{(k)}\hat{\boldsymbol{\gamma}}_j^{(k)}||_2^2$. For the soft MPenPC, we define $\hat{\sigma}^2 = n^{-1}\sum_{k=1}^K ||\mathbf{X}_j - \hat{\gamma}_{j,0}^{(k)} - \mathbf{X}\hat{\boldsymbol{\gamma}}_j^{(k)}||_{\mathbf{W}^{(k)}}^2$.

3.4 Theoretical properties

We first define some notations. For the k-th population, $\boldsymbol{\Omega}^{(k)}$ denotes the precision matrix, from which the CIG $\mathcal{G}^{(k)}$ can be deduced. We denote the neighborhood of node j in $\mathcal{G}^{(k)}$ by $A_j^{(k)} = \{l : \Omega_{j,l}^{(k)} \neq 0 \text{ and } l \neq j\}$, and its complement by $\overline{A}_j^{(k)} = \{l : \Omega_{j,l}^{(k)} = 0\}$. Moreover, denote $A_j = \cup_{k=1}^K A_j^{(k)}$ and $\overline{A}_j = \cap_{k=1}^K \overline{A}_j^{(k)}$. Then each node in A_j is connected with node j in *at least* one $\mathcal{G}^{(k)}$, and nodes in $\overline{A}_j$ are *not* connected with the node j in any $\mathcal{G}^{(k)}$.

We assume some regularity conditions (C1)-(C8) on the underlying model and tuning parameters. More details are provided in the Supplementary Materials. We have the following theorem regarding the neighborhood selection (Stage I) of the MPenPC with hard labels.

Theorem 4 (Stage I Consistency)**.** *(i) Under (C1) - (C5), with a probability of* $1 - O(1/p)$*, for all* $j \in \{1, \cdots, p\}$ *there is a local minimizer* $\hat{\boldsymbol{\gamma}}_j$ *to problem (3.4) such that* $j - l \in \hat{\mathcal{G}}^{(k)}$ *if* $l \in A_j^{(k)}$*, and* $j - l \notin \hat{\mathcal{G}}^{(k)}$ *if* $l \in \bar{A}_j$*;*

(ii) Under (C1)-(C6), with a probability of $1 - O(1/p)$*, for all* $j \in \{1, \cdots, p\}$ *there is a local minimizer* $\hat{\boldsymbol{\gamma}}_j$ *to problem (3.4) such that* $j - l \in \hat{\mathcal{G}}^{(k)}$ *if* $l \in A_j^{(k)}$*, and* $j - l \notin \hat{\mathcal{G}}^{(k)}$ *if* $l \in \bar{A}_j^{(k)}$*.*

Theorem 4 considers two different types of consistency for the neighborhood selection. In particular, with (C1)-(C5), the estimation is guaranteed to recover all edges that appear in at least one conditional independence graph $\mathcal{G}^{(k)}$. We call this group-union selection consistency. In this case, we can recover the *union* of the undirected graphs asymptotically. With condition (C6), we

can obtain stronger consistency that correctly identifies all edges for each graph. Similar results for the soft MPenPC, with a condition on the soft labels, are included in the Supplementary Materials.

To the extent of our knowledge, this is the first theoretical result regarding the selection consistency of GEL-penalized estimation. It can be extended to other bi-level group penalties. According to the irrepresentability condition (C6), estimating $\gamma_{j,l}^{(k)}$ correctly as 0 becomes more difficult as the edge $j-l$ being shared by more graphs. By replacing the GEL with another bi-level sparsity-inducing penalty composed of concave penalties for both levels (Breheny and Huang, 2009; Chen and Sun, 2017), condition (C6) can be greatly relaxed. We do not pursue that approach in this work, however, due to the additional computational burden for tuning parameter selection. Even if (C6) is not satisfied, the first part of Theorem 4 still guarantees consistent estimate of the union of edges from $\mathcal{G}^{(k)}$'s, and the next theorem guarantees that the PC step of the MPenPC can produce consistent skeleton estimates given the graph union.

Theorem 5 (Stage II Consistency)**.** *Assume we have perfect estimation of all the CIGs or their union from the Stage I, under (C1)-(C2) and (C8), there exists a p-value cutoff* $\alpha \to 0$ *such that the skeletons are recovered perfectly for all subpopulations with probability* $1 - O(\exp(-Cn^{1-2d_2})) \to 1$*, for some* $d_2 > 0$*.*

Therefore, by combining the results of Theorems 4 and 5, the MPenPC method produces a consistent skeleton estimation with probability going to 1.

3.5 Simulation studies

In this section, we use simulated examples to study the performance of our MPenPC methods. In particular, we justify the joint estimation and the usage of soft labeling by comparing the MPenPC with the original PenPC method. When applying the PenPC method, we can either estimate a single skeleton with all the samples, or estimate one skeleton for each class. We call the first approach as PenPC without grouping (PenPC - No Grouping) and the second as group-specific PenPC (Hard PenPC). For a comprehensive comparison, we also consider a group-specific PenPC with soft labels (Soft PenPC), which performs weighted regression at Stage I and uses weighted covariance for the PC-stable algorithm at Stage II. In this simulation, we implement the Hard and Soft MPenPC. The exponential penalty, i.e. $P(\boldsymbol{\theta}) = \lambda^2\tau^{-1}\sum_j\{1-\exp(-\lambda^{-1}\tau|\theta_j|)\}$, is used in both

PenPC approaches. We first introduce the simulation settings and the implementation of methods in Section 3.5.1. Then the methods are compared by stage in Sections 3.5.2 and 3.5.3.

3.5.1 Simulation settings

In the simulation, we consider Gaussian mixture settings under which the corresponding DAGs share some common edges. To this end, we first generate K DAGs with certain similarity, based on which we specify the Gaussian distributions. In particular, the k-th Gaussian component is specified through the structure equation model,

$$X = \boldsymbol{\nu}^{(k)} + \mathbf{B}^{(k)} X + Z, \tag{3.6}$$

where $Z \sim N(\mathbf{0}, \mathbf{I})$. For convenience, the variables are ordered such that $b_{j,l}^{(k)} = 0, \forall l \geq j$, where $b_{j,l}^{(k)}$ is the (j,l)-th entry of $\mathbf{B}^{(k)}$. The bias vector $\boldsymbol{\nu}^{(k)}$ is set as $\nu_j^{(k)} = \delta$ if $4(k-1) < j \leq 4k$ and 0 otherwise, so we can adjust the group difference via δ.

The DAGs are generated with two different models, the Erdos-Renyi (ER) and the Barabasi-Albert (BA), respectively. Denote p the dimension of $\mathbf{X}$, then $\{\mathbf{B}^{(k)}; k = 1, \cdots, K\}$ are $p \times p$ lower triangular matrices. With the two models, we generate $\mathbf{B}^{(k)}$'s as follows.

- (ER model) We generate $K+1$ random $p \times p$ matrices, denoted by $\mathbf{A}^{(0)}, \cdots, \mathbf{A}^{(K)}$, independently. Initialize each $\mathbf{A}^{(k)}$ with all 0's. Randomly select $\lceil \pi_E \cdot p(p-1)/2 \rceil$ entries in the lower triangular matrix (excluding the diagonal), and fill them with random values from Uniform($[-1, -0.5] \cup [0.5, 1]$). Using $\mathbf{A}^{(0)} = (a_{j,l}^{(0)})_{p \times p}$ as the basis matrix, we construct $\mathbf{B}^{(k)}$ by taking $b_{j,l}^{(k)} = a_{j,l}^{(0)}$ for $\lfloor \pi_0 p^2 \rfloor$ random entries and $b_{j,l}^{(k)} = a_{j,l}^{(k)}$ for the rest, where $\pi_0 \in [0,1]$, and $a_{j,l}^{(k)}$ is the (j,l)-the entry of $\mathbf{A}^{(k)}$.

- (BA Model) The procedure is basically the same as above, except $\mathbf{A}^{(k)}$'s. For each $\mathbf{A}^{(k)}$, we generate a random DAG with the BA model as follows. Starting from an empty DAG with the node 1, we add the node 2 and the edge $1 \to 2$ to the DAG. Then at each step, we add the node j and e random edges to the graph such that the probability of edge $l \to j$ is proportional to the neighborhood size of node l ($l < j$). Then we construct $\mathbf{A}^{(k)}$ by filling $a_{j,l}^{k}$ with random values from Uniform($[-1, -0.5] \cup [0.5, 1]$) if the edge $l \to j$ exists in the DAG.

Note that the graph sparsity is determined by π_E and e respectively in the two models, and π_0 tunes the similarity among the K DAGs. In particular, any two of the K DAGs have about $\pi_0^2 \times 100\%$ overlapping in expectation. Examples of ER and BA models with the same sparsity are shown in Figure 3.1. One may observe that the BA model has more variation on the degree of connections per node, with both hubs (i.e. heavily connected nodes) as well as nodes with few connections. This is due to the *scale free* property of the BA model. The different sparsity patterns of ER and BA models can result in different challenges in skeleton estimation.

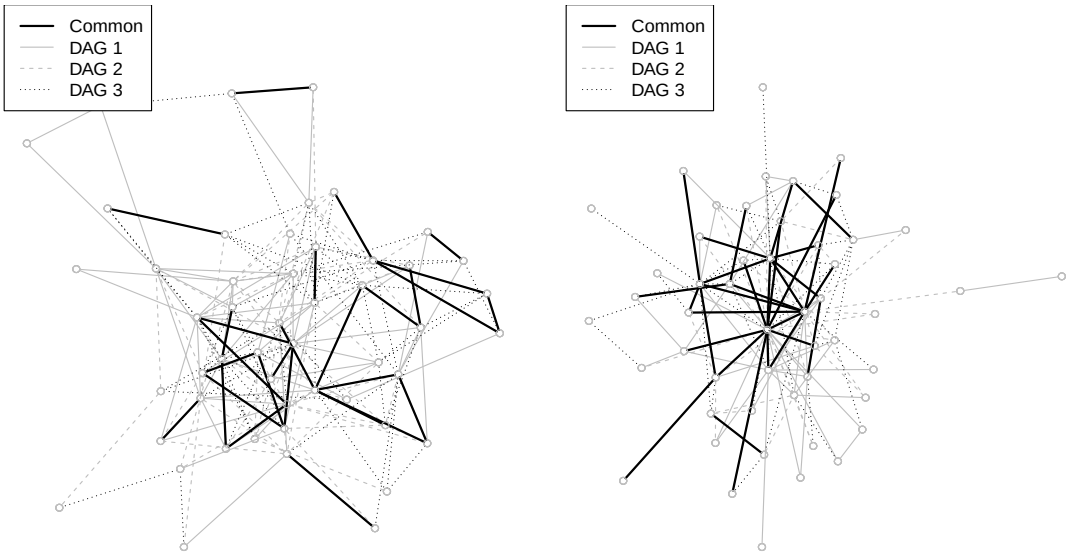

Figure 3.1: Examples of DAGs generated by the ER model (left) with $\pi_E = 0.02$ and the BA model (right) with $e = 2$. In both models, we set $K = 3$, $p = 50$, and $\pi_0 = 0.4$.

In the simulation, we assign equal weights to K classes, and generate n samples from the Gaussian mixture distribution, denoted by $\{\mathbf{x}_i; 1 \leq i \leq n\}$. We perform principal component analysis (PCA) on the whole data. Based on the first 20 principal components, the hard labels are constructed by the k-means clustering, denoted by $\{g_i; 1 \leq i \leq n\}$. Soft labels are then computed by QDA based on the hard labels, using the same 20 principal components. Let $K = 4$, $p = 500$, and $n = 400$. For the ER model, we set $\pi_E = 1/500$, and for the BA model, we set $e = 1$. For both models, we consider a *high overlapping* ($\pi_0 = 0.7$) setting as well as a *low-overlapping* one ($\pi_0 = 0.3$). In the latter case, any two graphs have only about 10% edges in common. For each simulation setting, we run 100 repetitions and evaluate the performance of all the methods by stage.

Evaluation criteria include true positive rate (TPR) and false positive rate (FPR) averaged over classes for both stages. In particular, denote $\mathcal{G}$ and $\hat{\mathcal{G}}$ as the true CIG or DAG skeleton and its estimate. Then

$$\text{TPR} = \frac{|\hat{\mathcal{G}} \cap \mathcal{G}|}{|\mathcal{G}|}, \quad \text{FPR} = \frac{|\hat{\mathcal{G}} \setminus \mathcal{G}|}{(p^2 - p)/2 - |\mathcal{G}|},$$

which are respectively 1 and 0 if $\hat{\mathcal{G}}$ recovers $\mathcal{G}$ perfectly. For Stage II, we also include the estimation accuracy results of the CPDAGs in terms of the average Structural Hamming Distance (SHD), which is computed as the number of flipping, addition, and deletion operations to turn each estimated CPDAG to the true CPDAG (the smaller the better).

Due to limited space, we only display the results for the BA example in the chapter. The results for the ER example and additional scenarios are included in the Supplementary Materials.

3.5.2 Stage I: neighborhood selection

In this subsection, we compare the CIG estimation (Stage I) of all methods under different settings. We select the tuning parameters of all methods (λ for PenPC, τ and λ for MPenPC) by EBIC. As we can see from Table 3.1 (upper panel), all four methods, except the PenPC without grouping, recover most of the edges effectively. Both the group-specific PenPC and the MPenPC benefit from using soft labels (Soft PenPC vs. Hard PenPC, Soft MPenPC vs. Hard MPenPC) – with slightly higher FPR, they have substantially higher TPR. We also notice that while the MPenPC methods can take advantage of the graph overlaps, they appear to be less effective than the group-specific PenPC under the low overlapping setting at this stage. However, the sparsity level of estimated graphs vary dramatically across methods, which poses challenges to our analysis. In particular, the estimates of the PenPC without grouping have far fewer edges than those of other methods, thus we do not include it in later comparisons.

For a more thorough comparison of the methods, we evaluate the estimation of all the methods at different sparsity levels by changing the tuning parameter λ in nodewise regressions. Figure 3.2 displays the results for BA model. In the high overlapping setting, the MPenPC methods always have higher TPRs than the PenPC methods at the same FPR level. Even in the low-overlapping setting, the Soft MPenPC has a better accuracy than the group-specific PenPC methods. Moreover,

both the group-specific PenPC and the MPenPC with soft labels outperform their counterparts with hard labels, which justifies the usage of soft labeling.

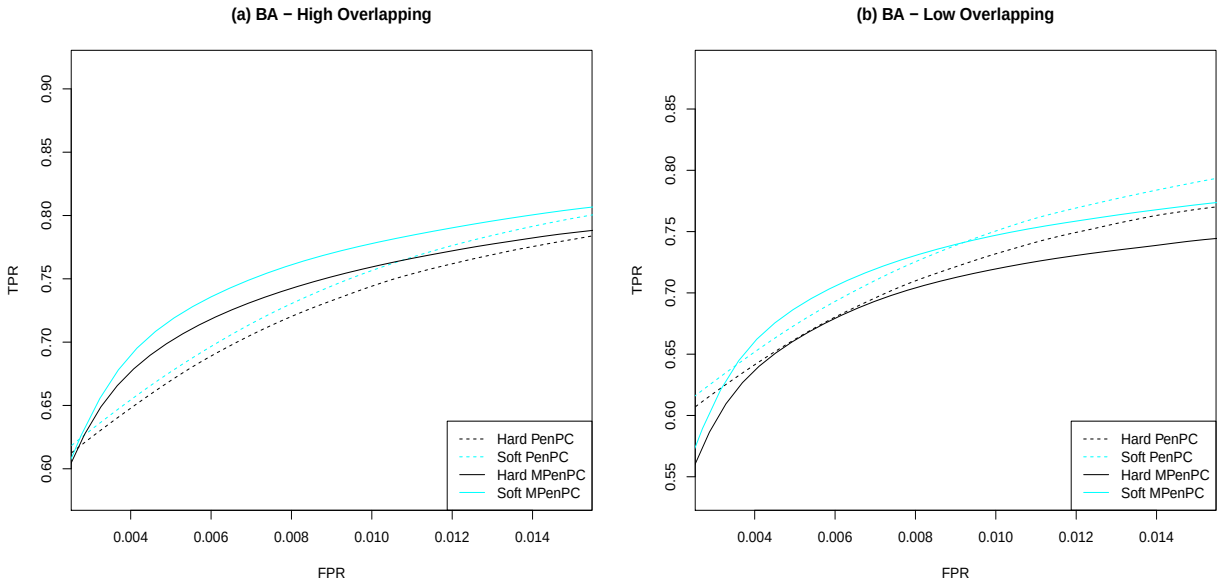

Figure 3.2: Neighborhood selection performance of different methods at Stage I in BA scenario: $K = 4$, $p = 500$, $e = 1$, $\delta^2 = 0.05$, $\pi_0 = 0.7$ for (a) and 0.3 for (b). The x-axes and y-axes represent FPR and TPR respectively. The graph sparsity at the neighborhood selection stage varies for different tuning parameter λ. The tuning parameter γ for MPenPC methods is preselected by EBIC.

3.5.3 Stage II: skeleton estimation

At Stage II, we apply the PC-stable algorithm to obtain skeleton estimation based on the CIGs estimated in Stage I (selected by EBIC). For each of the K classes, the input of the PC-stable algorithm includes an initial graph and a correlation matrix. The procedure is similar for all methods, except that the soft PenPC and the soft MPenPC use weighted correlation, and the PenPC without grouping uses the overall correlation of all samples. Table 3.1 (lower panel) summarizes the performance evaluation of all the methods at Stage II for the BA model with the p-value cutoff $\alpha = 0.02$.

From Table 3.1, we observe that all methods except the PenPC without grouping produce much sparser skeletons compared to the undirected graphs at Stage I. The sparsity of group-specific PenPC and MPenPC estimates becomes comparable. In the high overlapping setting, the MPenPC with soft labels has a clear edge over the group-specific PenPC with soft labels, with higher TPR

and lower FPR. The comparison is the same for their counterparts with hard labels. Under the low overlapping setting, the differences between the group-specific PenPC and the MPenPC become smaller compared with the results at Stage I. The results of CPDAG estimation also confirm the advantages of the proposed method in terms of lower SHD.

With a fixed initial graph, the sparsity of skeleton estimation can be tuned by α for each of the methods. For a complete comparison among the methods, we also present the results for a range of significance levels, namely $0.005, 0.01, \cdots, 0.3$, in Figure 3.3. As α increases, the skeleton estimation becomes less sparse and both FPR and TPR increase. With respect to the BA example, the relative performance of different methods is relatively stable at each significance level. The comparisons mostly conform to our earlier observations. In the high overlapping setting, both MPenPC variants have significant advantages over their group-specific PenPC counterpart. The PenPC and the MPenPC with soft labels are more accurate than those with hard labels. In the low overlapping setting, the performance of MPenPC methods and their group-specific PenPC counterpart are similar. The group-specific PenPC estimates have slightly higher TPRs than the MPenPC, but also with higher FPRs.

The simulation results for the ER model show similar patterns of relative performance of all the methods (Supplementary Table 1, Figures 2-3). We have also conducted simulations with non-overlapping DAGs. In this scenario, the group-specific PenPC methods have slightly better performance than the MPenPC methods, but soft-label-based methods still have better performance than hard-label-based methods (Supplementary Figures 4-7). In summary, our simulation study shows that our joint estimation methods, including both Hard and Soft MPenPC, often produce more accurate skeleton estimates than separate estimation, when the multiple DAGs have reasonable similarities. Moreover, for either the PenPC or the MPenPC method, the soft labels always benefit the estimation.

3.6 Cancer genomic applications

Breast cancer is the most commonly diagnosed cancer type in females and second leading cancer death in females (Siegel et al., 2016). Gene expression data collected from cancer samples are very informative to study the molecular characteristics of breast cancer. For example, based

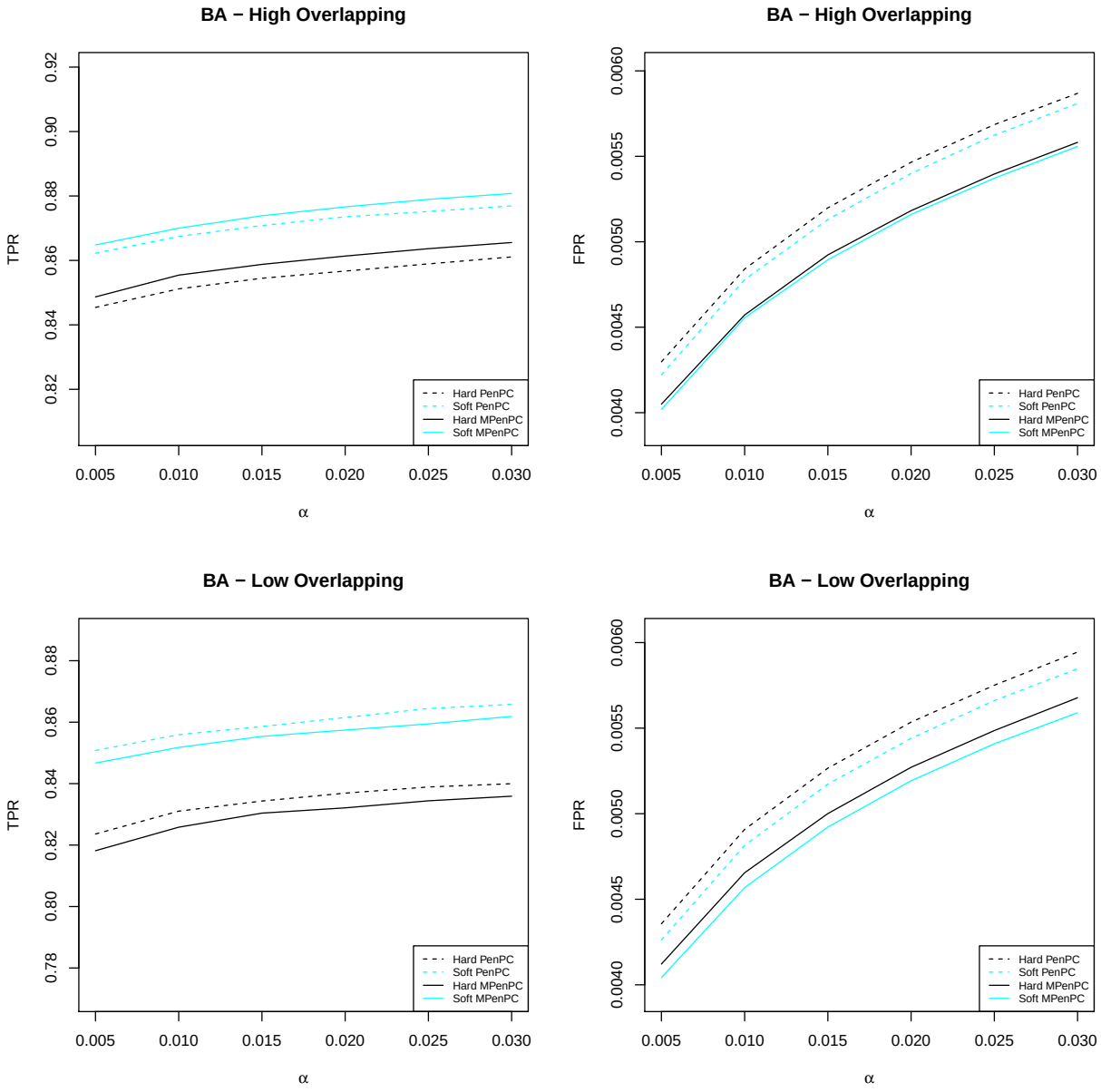

Figure 3.3: Skeleton estimation performance of different methods at Stage II in BA scenario. The high overlapping scenarios have $\pi_0 = 0.7$, and the low overlapping scenarios have $\pi_0 = 0.3$. The x-axes represent the significance level α of the PC-stable algorithm. The y-axes represent TPR (the left panel) and FPR (the right panel) respectively.

on the gene expression pattern, breast cancer can be divided into five subtypes: basal, HER2 over-expression (HER2), luminal A (lumA), luminal B (lumB), and normal-like (The Cancer Genome Atlas Network, 2012). We seek to use DAG skeleton to study gene co-expression patterns in breast cancer. To account for the similarity and differences across subtypes, we apply our MPenPC method to jointly estimate DAG skeletons of multiple subtypes.

We obtain level 3 gene expression data (UNC IlluminaHiSeq_RNASeqV2 pipline) of TCGA breast cancer patients (The Cancer Genome Atlas Network, 2012) from TCGA data portal. The gene expression dataset is in the format of raw counts of sequencing reads for more than 20,000 genes. We limit our analysis on 405 Caucasian patients and filter out genes with low expression in the majority of the patients. Specifically, we require the raw counts to be ≥ 20 for at least 25% of the individuals and 15,816 genes pass this filtering. The distribution of these 405 patients in the 5 breast cancer subtypes are: basal ($n = 64$), HER2 ($n = 21$), lumA ($n = 222$), lumB ($n = 91$), and normal-like ($n = 8$). In the following analysis, we remove the 8 individuals of normal-like subtype due to its small sample size.

To focus on genes that are more relevant to cancer biology, we select 3,466 genes that belong to at least one of 17 cancer-relevant gene sets (the Molecular Signatures Database C6 oncogenic signatures gene sets (Subramanian et al., 2005)). See the Supplementary Materials for a complete list of specific gene sets. For each of these 3,466 genes, we regress it against all other genes by penalized regression using the *log*-penalty (Sun et al., 2010) and select tuning parameters by EBIC. Two genes are connected if each of them is selected in the regression model for the other gene. Then we remove the genes that are not connected with any other gene of the same gene set, and use the remaining 1,528 genes in the following analysis.

As in the simulation study, we use different methods to estimate the skeletons for four subtypes. We use a PathwayCommons dataset (Cerami et al., 2010), which is based on multiple databases, as the benchmark. In this case, we have a single benchmark graph for all groups/subtypes. It is created by connecting any two genes that *interact with* or *are in complex with* each other. Since the subtypes have already been specified, we only need to create soft labels by classification. Naive Bayes is used to compute the soft labels. At the second stage of skeleton estimation (removing edges by conditional dependence testing), we use a significant level of $\alpha = 0.02$ for all methods.

We compare the performance of different methods at two stages. During Stage I, by varying tuning parameters, each method produces undirected graphs of varying sparsity level for each subtype. The MPenPC method with soft labels generally recover more edges defined in PathwayCommons than other methods, including the MPenPC with hard labels, at the same sparsity level (Figure 3.4). Since the EBIC-selected graph estimates of the four methods have very different sparsity levels, we compare the performances of these methods across sparsity levels of Stage I,

after applying the PC-stable algorithm with a fixed significance level $\alpha = 0.02$ during Stage II (Figure 3.5). We can see that the soft MPenPC again produces the best skeleton estimates among all methods. Therefore, joint estimation and soft labels can benefit the skeleton estimation in this application.

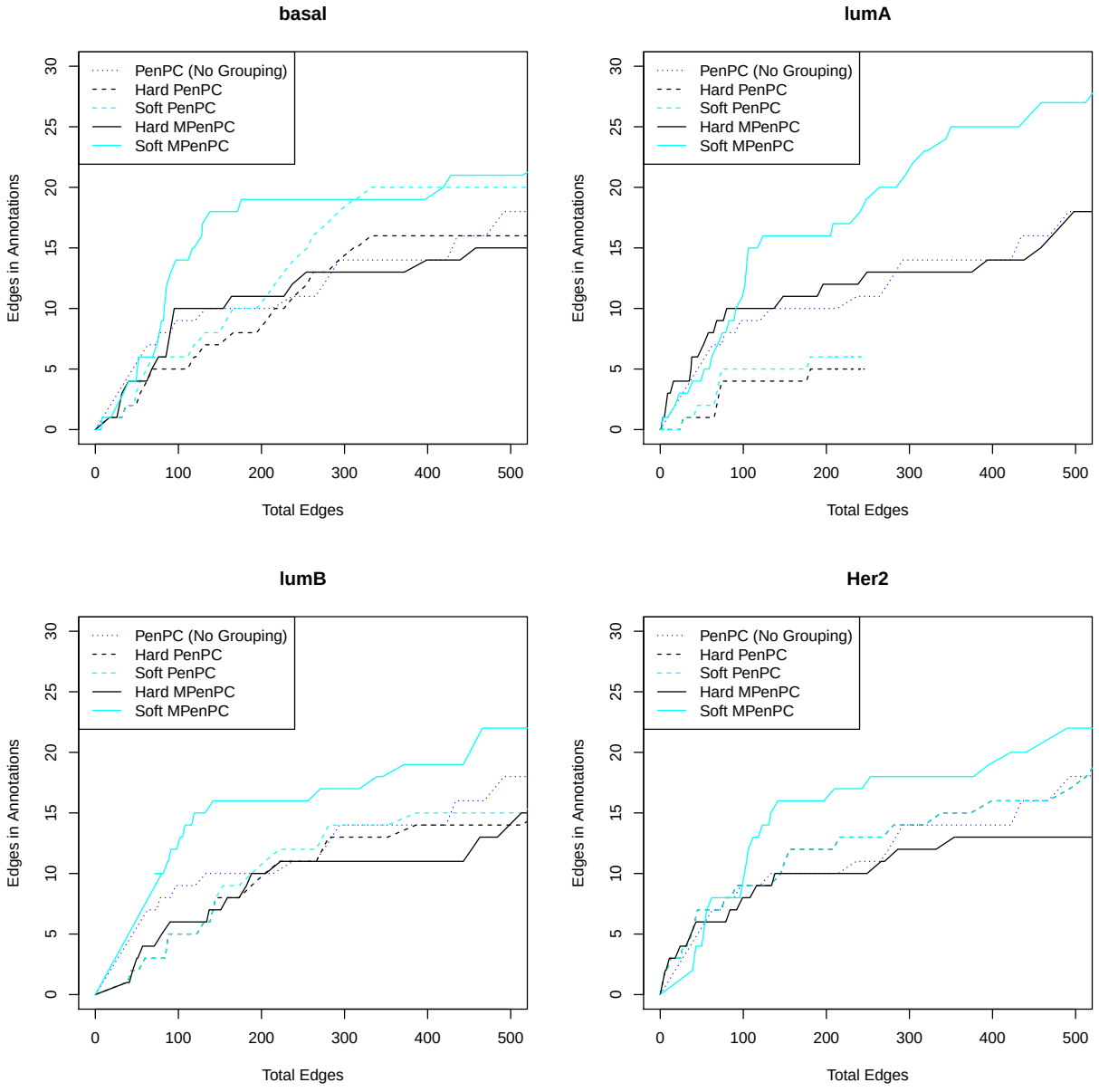

Figure 3.4: Performance comparison of all methods at Stage I by cancer subtype. The x-axes represent the total number of edges in estimated graphs corresponding to different λ values; the y-axes represent the number of overlapping edges in estimated graphs.

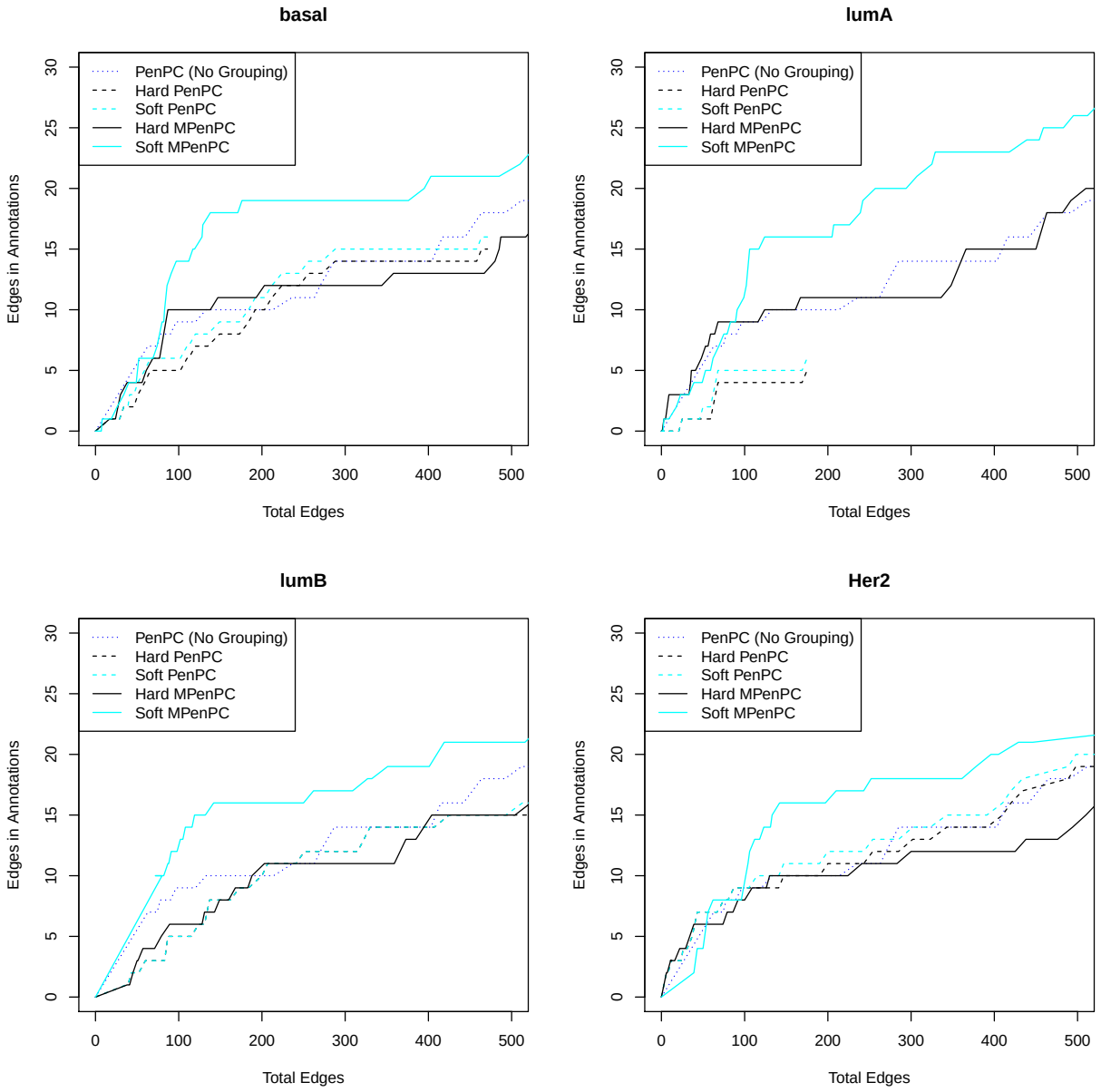

Figure 3.5: Performance comparison of all methods at Stage II by cancer subtype. The x-axes represent the significance level α of the PC-stable algorithm. The y-axes represent the number of overlapping edges (left panel) and total number of edges (right panel) in estimated skeletons.

To further explore the DAG skeleton within each gene set. We rerun our analysis using MPenPC for each gene set separately for three subtypes: basal, lumA or lumB. The HER2 subtype is not included due to its relatively small sample size. Since subtypes lumA and lumB are more similar, we expect more similarity between their skeleton estimates and less similarity between the skeleton

estimates for lumA/lumB and basal. This is indeed the case, as shown in Supplementary Figure 8 in the Supplementary Materials.

We further illustrate the skeleton estimates for these three cancer subtypes for a gene set related with TP53. TP53 is a tumor suppressor gene that induces cell cycle arrest and apoptosis in abnormal cells. TP53 pathway activity is among the major differences between basal and lumA/lumB cancers. TP53 mutation rate is 12%, 32%, and 84% for lumA, lumB, and basal, respectively. Integrative analysis of multiple types of -omic data from TCGA samples suggest TP53 pathway remains largely functional in luminal A samples, is often inactivated in a subset of luminal B samples, and is inactivated in most basal samples (The Cancer Genome Atlas Network, 2012). The gene set we analyzed related with TP53 is the union of two gene sets of MSigDB: P53_DN.V1_UP and P53_DN.V1_DN, which corresponds to genes that are up-regulated or down-regulated in NCI-60 panel of cell lines with mutated TP53. The DAG skeleton estimated by either soft MPenPC or soft PenPC show that genes involved in negative regulation of apoptosis are enriched among the genes with 4 or more connections in basal subtype, but less so for lumA/lumB, suggesting both methods identify biologically interesting subtype-specific features. In addition, soft MPenPC identify more edges shared between basal and lumA/lumB subtypes than soft PenPC (112 edges by MPenPC vs. 66 edges by PenPC, Chi-squared test p-value 1.4×10^{-6}), suggesting the advantage of joint analysis of multiple subtypes.

3.7 Conclusion

In this chapter, we propose the MPenPC method to estimate the DAG skeletons with heterogeneous samples. By taking advantage of the similarity among the DAGs, the MPenPC method can produce more accurate estimates than separate estimation. In particular, we take into account possible labeling errors in this scenario, and propose a remedy with soft labels. The effectiveness of our method is demonstrated with numerical examples.

The ideas of joint estimation and soft labels can also be combined with other DAG estimation methods. For example, Nandy et al. (2015) has demonstrated how to appropriately combine the CIG estimation and the greedy equivalence search for single-DAG estimation. Similar hybrid approaches may be used for heterogeneous populations.

Table 3.1: Performance of different methods at both stages for the BA-model examples ($K = 4$, $p = 500$, $e = 1$, $\pi_0 = 0.7$ (High Overlapping) or 0.3 (Low Overlapping), and $\delta^2 = 0.05$). TPR $= |\hat{\mathcal{G}} \cap \mathcal{G}|/|\mathcal{G}|$ and FPR $= |\hat{\mathcal{G}} \setminus \mathcal{G}|/[(p^2 - p)/2 - |\mathcal{G}|]$, where $\mathcal{G}$ and $\hat{\mathcal{G}}$ denote the true and the estimated graphs respectively. The numbers outside and inside parentheses are averages and standard errors respectively based on 100 repetitions.

Stage I. Neighborhood Selection		
High Overlapping	TPR	FPR
PenPC (No Grouping)	0.6469(0.0022)	0.0017(0.0000)
Hard PenPC	0.8176(0.0022)	0.0353(0.0000)
Soft PenPC	0.8474(0.0022)	0.0390(0.0002)
Hard MPenPC	0.8451(0.0019)	0.0446(0.0002)
Soft MPenPC	0.8675(0.0019)	0.0473(0.0002)
Low Overlapping	TPR	FPR
PenPC (No Grouping)	0.4515(0.0020)	0.0024(0.0000)
Hard PenPC	0.7991(0.0044)	0.0359(0.0000)
Soft PenPC	0.8338(0.0029)	0.0391(0.0000)
Hard MPenPC	0.7884(0.0048)	0.0452(0.0000)
Soft MPenPC	0.8178(0.0039)	0.0472(0.0000)
Stage II. Skeleton Estimation		
High Overlapping	TPR	FPR
PenPC (No Grouping)	0.6348(0.0025)	0.0016(0.0000)
Hard PenPC	0.8375(0.0030)	0.0055(0.0000)
Soft PenPC	0.8553(0.0030)	0.0054(0.0000)
Hard MPenPC	0.8629(0.0030)	0.0052(0.0000)
Soft MPenPC	0.8785(0.0030)	0.0052(0.0000)
Low Overlapping	TPR	FPR
PenPC (No Grouping)	0.4548(0.0017)	0.0022(0.0000)
Hard PenPC	0.8369(0.0060)	0.0055(0.0000)
Soft PenPC	0.8615(0.0046)	0.0054(0.0000)
Hard MPenPC	0.8321(0.0066)	0.0053(0.0000)
Soft MPenPC	0.8574(0.0051)	0.0052(0.0000)
Stage II. CPDAG Estimation		
High Overlapping	SHD	
PenPC (No Grouping)	612.99(3.57)	
Hard PenPC	565.46(2.94)	
Soft PenPC	550.22(2.27)	
Hard MPenPC	498.87(2.54)	
Soft MPenPC	492.61(2.28)	
Low Overlapping	SHD	
PenPC (No Grouping)	784.79(2.94)	
Hard PenPC	580.61(2.22)	
Soft PenPC	558.45(2.45)	
Hard MPenPC	511.22(2.55)	
Soft MPenPC	496.84(2.45)	

CHAPTER 4

Graphical Model Estimation for Single Cell RNA-seq Data

4.1 Introduction

In recent years, the single cell RNA sequencing (scRNA-seq) techniques are becoming increasingly popular in gene expression studies. While bulk RNA sequencing data measure the aggregated gene expression of a tissue sample that is composed of millions of cells, scRNA-seq data provide the gene expression quantification in each single cell. This helps researchers to study the biological heterogeneity at the cell level. For example, scRNA-seq revealed intratumoral heterogeneity in primary glioblastoma (Patel et al., 2014). Moreover, with scRNA-seq data, it is now feasible to study the unique gene interaction pattern of each person, which may provide valuable information for personalized treatment for diseases. In contrast, bulk RNA-seq data can only study gene interactions at a population level and require gene expression data from hundreds or even thousands of individuals.

While scRNA-seq techniques have shown significant advantages over traditional RNA-seq techniques in many aspects, particular features of scRNA-seq data pose great challenges for data analysis. For example, there are often many zero values in scRNA-seq data. They may be either due to the under-detection of mRNA below a certain level or simply because the gene is not expressed in those cells. Moreover, the gene expression of different cells are usually not independent. In particular, cells in the same cell lineage tend to be more alike than others. In this paper, we focus on the graphical model estimation based on scRNA-seq data and we design our method to accommodate such particular features of scRNA-seq data.

Traditionally, Gaussian graphical models are often employed to model the gene expression dependence across genes. Many methods have been proposed for the estimation of high-dimensional Gaussian graphical models (Meinshausen and Bühlmann, 2006; Yuan and Lin, 2007; Friedman et al., 2008; Cai et al., 2011). However, Gaussian graphical models assume that the features are Gaussian-

distributed. While this assumption is approximately true for traditional bulk RNA-seq data after certain transformations, it may not be appropriate for scRNA-seq data. Liu et al. (2009) and Liu et al. (2012) generalized Gaussian graphical models to non-Gaussian data by non-paranormal transformation, but it is valid only when the transformation is continuous.

There is a rich literature on count-data graphical models. For example, Yang et al. (2012a) and Chen et al. (2014) proposed a class of exponential family graphical models, of which the Poisson graphical model is a special case. But the model only allows negative dependence among the features. A number of approaches have been proposed to address this issue. Yang et al. (2013) proposed to remove such a restriction by modifying the base measure of the multivariate Poisson distribution. Allen et al. (2013) proposed a neighborhood selection approach that estimates the neighborhood of each node via penalized Poisson regression. See Inouye et al. (2017) for a comprehensive review. Recently, the Poisson-logNormal model has attracted a lot of attention for count-data graphical modeling. A variety of methods have been proposed for model estimation (Choi et al., 2017; Wu et al., 2018a; Sinclair and Hooker, 2017; Chiquet et al., 2018). However, none of them is feasible for high-dimensional modeling due to computational burdens.

In addition to the count feature of scRNA-seq data, the abundance of zero values and sample dependence pose additional challenges to the estimation of graphical models. While we can remove many rarely expressed genes, the modality at zero can still be an issue (McDavid et al., 2014). McDavid et al. (2016) proposed to model scRNA-seq data with a multivariate Hurdle distribution. Their Hurdle model consists of two parts: the first part determines whether a feature is zero or not and the second part determines the value of nonzero expression. Compared to a Gaussian graphical model, the number of parameters to be estimated is tripled, which makes the estimation difficult in high dimensions. For dependent continuous data, Zhou et al. (2014) proposed a matrix-variate Gaussian graphical model whose covariance matrix is the outer product of the feature covariance matrix and the sample covariance matrix. However, it is difficult to generalize this approach to non-Gaussian scenarios.

In this chapter, we propose two new neighborhood selection approaches for the graphical modeling of scRNA-seq data. The new methods account for the count nature and other characteristics of scRNA-seq data. The rest of this chapter is organized as follows. In Section 4.2, we briefly review existing graphical models in the literature and then propose our new graphical models. The

implementation and parameter tuning of the methods are detailed in Section 4.3. In Section 4.4, we examine the new methods with simulated examples and compare them with other methods. In Section 4.5, we investigate the characteristics of scRNA-seq data in depth with two real scRNA-seq datasets and evaluate the graph estimation of different methods. In Section 4.6, we conclude this chapter with a brief summary on our findings.

4.2 Graphical Models based on scRNA-seq Data

In this section, we first review existing count-data graphical models in Section 4.2.1. Then we introduce our Poisson-logNormal graphical model for scRNA-seq data and its estimation in Section 4.2.2. In Section 4.2.3, a Hurdle-logNormal graphical model is introduced to account for excessive zero values in scRNA-seq data.

Before diving into the model part, we first introduce some notations for later use. For any matrix $\mathbf{A}$, $\mathbf{A}_{i,\cdot}$ denotes its i-th row vector and $\mathbf{A}_j$ denotes its j-th column vector; $\mathbf{A}_{-j}$ denotes the matrix after removing the j-th column. For any vector $\mathbf{a}$, a_k denotes its k-th element and $\mathbf{a}_{-k}$ represents the vector after removing the k-th element. For any square matrix $\mathbf{A}$, $\text{diag}(\mathbf{A})$ denotes its diagonal vector; conversely, for any vector $\mathbf{a}$, $\text{diag}(\mathbf{a})$ represents a diagonal matrix with $\mathbf{a}$ as its diagonal. For vectors $\mathbf{a} = \{a_j\}$ and $\mathbf{b} = \{b_j\}$ of the same length, we denote their elementwise product by $\mathbf{a} \circ \mathbf{b} = \{a_j b_j\}$. The logit function is defined as $\text{logit}(\pi) = \log[\pi/(1-\pi)], \forall \pi \in (0,1)$, and its inverse function $\text{logit}^{-1}(x) = e^x/(1+e^x), \forall x \in \mathbb{R}$.

4.2.1 Existing count-data graphical models

Poisson distributions are often used to model count data. A straightforward generalization of Poisson to the multivariate scenario is

$$P(Y_1, \cdots, Y_p) = \exp\left\{ \sum_{j=1}^{p} (\theta_j Y_j - \log(Y_j!)) + \sum_{j,l} \theta_{jl} Y_j Y_l - A(\boldsymbol{\theta}) \right\}, \tag{4.1}$$

where $\theta_{jl} = \theta_{lj}, \forall j, l$ and $\boldsymbol{\theta} = \{\theta_j : 1 \le j \le p\} \cup \{\theta_{jl} : 1 \le j, l \le p\}$. Similar to the multivariate Gaussian distribution, Y_j and Y_l are conditionally independent given the rest if and only if $\theta_{jl} = \theta_{lj} = 0$. However, it can be shown that θ_{jl} must be non-positive to make the distribution valid

(Yang et al., 2012a; Chen et al., 2014). Yang et al. (2013) proposed to remove such a restriction via modification of the base measure $\log(Y_j!)$ in (4.1).

Allen et al. (2013) proposed to only specify the conditional distributions, i.e.

$$Y_j|\mathbf{Y}_{-j} \sim \text{Poisson}(\exp(\beta_0^{(j)} + \tilde{\mathbf{Y}}_{-j}^\top \boldsymbol{\beta}^{(j)})), \tag{4.2}$$

where $\tilde{\mathbf{Y}}$ is the log-transformed $\mathbf{Y}$. In the corresponding graphical model, Y_j and Y_l are connected if $\beta_l^{(j)} \neq 0$ or $\beta_j^{(l)} \neq 0$. While (4.2) does not correspond to any valid joint distribution, it enjoys great flexibility.

The multivariate Poisson-logNormal model is another common choice in modeling count data. It assumes that

$$\begin{aligned} \mathbf{X} \sim & N(\boldsymbol{\mu}, \boldsymbol{\Omega}^{-1}), \\ Y_j|X_j \overset{ind}{\sim} & \text{Poisson}(e^{X_j}), j = 1, \cdots, p \end{aligned} \tag{4.3}$$

The corresponding graph is derived from the underlying Gaussian distribution, i.e. j and l are connected if $\Omega_{jl} \neq 0$. Although many different algorithms have been proposed (Choi et al., 2017; Wu et al., 2018a; Sinclair and Hooker, 2017; Chiquet et al., 2018), the estimation of Poisson-logNormal becomes infeasible for data with moderately high dimensions.

Recently, McDavid et al. (2016) proposed to model zero values in scRNA-seq datasets with a multivariate Hurdle model,

$$\log f(\mathbf{y}) = \mathbf{v}_\mathbf{y}^\top \mathbf{G} \mathbf{v}_\mathbf{y} + \mathbf{v}_\mathbf{y}^\top \mathbf{H} \mathbf{y} - \frac{1}{2} \mathbf{y}^\top \mathbf{K} \mathbf{y} - C(\mathbf{G}, \mathbf{H}, \mathbf{K}), \mathbf{y} \in \mathbb{R}^p, \tag{4.4}$$

where $\mathbf{v}_\mathbf{y} = (\mathbb{I}(y_1 \neq 0), \ldots, \mathbb{I}(y_n \neq 0))^\top$ and $\mathbf{G}$, $\mathbf{H}$, and $\mathbf{K}$ are interaction matrices. Assume $\mathbf{G}$, $\mathbf{H}$, and $\mathbf{K}$ are symmetric, then gene j and gene l are conditionally independent if and only if $G_{jl} = H_{jl} = K_{jl} = 0$.

4.2.2 Dependent Poisson graphical models

Let $\{Y_{ij} \in \mathbb{N} : i = 1, \cdots, n; j = 1, \cdots, p\}$ denote expression levels of p genes in n cells from a biological specimen. Then $\mathbf{y}_i = \mathbf{Y}_{i,\cdot}^T$ represents the expression levels of all genes in the cell i, and $\mathbf{Y}_j$ denotes the expression of gene j in all cells. To model the dependence among genes, we assume

that

$$Y_{ij}|\mathbf{Y}_{i,-j} \sim \text{Poisson}(\exp(\beta_0^{(j)} + \tilde{\mathbf{y}}_{i,-j}^\top \boldsymbol{\beta}^{(j)} + v_{ij})), \quad 1 \le i \le n, 1 \le j \le p, \tag{4.5}$$

where $\tilde{\mathbf{y}}_i = (\tilde{y}_{i1}, \cdots, \tilde{y}_{ip})^\top$ denotes the log-transformed gene expression and v_{ij} is a random effect term. The graph is defined as $\mathcal{G} = (V, E)$, where the vertex set $V = \{1, \cdots, p\}$ and the edge set $E = \{(j, l) : \beta_l^{(j)} \neq 0 \text{ or } \beta_j^{(l)} \neq 0\}$. We call Model (4.5) the dependent Poisson model.

With the random effect term v_{ij} in (4.5), we can model the dependence among samples in scRNA-seq data. In particular, we assume that $\mathbf{V}_j = (v_{1j}, \cdots, v_{nj})^\top \overset{ind}{\sim} N(\mathbf{0}, c_j\boldsymbol{\Sigma})$ for $j = 1, \cdots, p$, where $\boldsymbol{\Sigma}$ represents the sample dependence and c_j's are gene-specific factors. Both $\boldsymbol{\Sigma}$ and c_j's can be estimated from the data. Unlike existing methods for dependent-data graphical modeling, e.g. GEMENI (Zhou et al., 2014), by modeling the sample dependence with random effects, our dependent graphical model can be applied to non-Gaussian data.

While Model (4.5) does not model inflated zeros directly, the over-dispersion brought by the random effect can largely account for the issue of many zeros in the data. This matches our observations with many scRNA-seq data (see Figure 4.3, the bottom row). In practice, we find that it works well for different types of scRNA-seq data, even if some of them have excessive zeros that cannot be completely explained by over-dispersion. We will further discuss this in Sections 4.2.3 and 4.4.1.

Parameter estimation

We can estimate $\boldsymbol{\beta}^{(j)}$'s through penalized conditional log-likelihood. For simplicity of presentation, we use $\boldsymbol{\beta}$, c, $\mathbf{X}$, and $\mathbf{y}$ instead of $(\beta_0^{(j)}, \boldsymbol{\beta}^{(j)\top})^\top$, c_j, $(\mathbf{1}_n, \tilde{\mathbf{Y}}_{-j})$, and $\mathbf{Y}_j$ in the following when they would not cause misinterpretation. Denote $\mathbf{x}_i$ the i-th row vector of $\mathbf{X}$. Then the penalized conditional log-likelihood is

$$\underset{\boldsymbol{\beta}}{\text{argmin}} -\frac{1}{n} \log L_j(\boldsymbol{\beta}) + P_\lambda(\boldsymbol{\beta}), \tag{4.6}$$

where $P_\lambda(\boldsymbol{\beta})$ is a sparsity-inducing penalty. In particular, the conditional likelihood of $\mathbf{Y}_j$ given $\mathbf{Y}_{-j}$ is

$$L_j(\boldsymbol{\beta}) = \int_{\mathbb{R}^n} (2\pi)^{-n/2} |c^{-1}\boldsymbol{\Omega}|^{1/2} \exp[-S_j(\mathbf{v}; \boldsymbol{\beta})] \, \mathrm{d}\mathbf{v}, \tag{4.7}$$

where $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ and

$$S_j(\mathbf{v}; \boldsymbol{\beta}) = \frac{1}{2c}\mathbf{v}^T\boldsymbol{\Omega}\mathbf{v} - \mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta} + \mathbf{v}) + \sum_{i=1}^{n}\exp(\mathbf{x}_i^\top\boldsymbol{\beta} + v_i).$$

The optimization problem (4.6) can be very difficult to solve due to its integral part. Therefore, we propose to solve this problem via least square approximation (Wang and Leng, 2007). In particular, given a reasonable estimate of $\boldsymbol{\beta}$ and a representative value of $\mathbf{v}$, denoted by $\mathbf{b}$ and $\hat{\mathbf{v}}$, we approximate $S_j(\mathbf{v}; \boldsymbol{\beta})$ in the integrand of (4.7) by:

$$S(\mathbf{v}; \boldsymbol{\beta}) \approx \frac{1}{2}||\mathbf{v} - (c^{-1}\boldsymbol{\Omega} + \boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}(\mathbf{e} - \mathbf{X}\boldsymbol{\beta})||^2_{c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda}} + \frac{1}{2}||\mathbf{e} - \mathbf{X}\boldsymbol{\beta}||^2_{\boldsymbol{\Lambda}-\boldsymbol{\Lambda}(c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}},$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\hat{y}_1, \ldots, \hat{y}_n)$, $\mathbf{e} = ((y_1 - \hat{y}_1)/\hat{y}_1 + \mathbf{x}_1^\top\mathbf{b} + \hat{v}_1, \ldots, (y_n - \hat{y}_n)/\hat{y}_n + \mathbf{x}_n^\top\mathbf{b} + \hat{v}_n)^\top$, and $\hat{y}_i = e^{\mathbf{x}_i^\top\mathbf{b}+\hat{v}_i}$ for $1 \le i \le n$. Thus we have

$$L_j(\boldsymbol{\beta}) \approx |c^{-1}\boldsymbol{\Omega}|^{1/2}/|c^{-1}\boldsymbol{\Omega} + \boldsymbol{\Lambda}|^{1/2} \cdot \exp\left\{-\frac{1}{2}||\mathbf{e} - \mathbf{X}\boldsymbol{\beta}||^2_{\boldsymbol{\Lambda}-\boldsymbol{\Lambda}(c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}}\right\},$$

and (4.6) is approximated by a Lasso problem:

$$\underset{\boldsymbol{\beta}}{\mathrm{argmin}}\ \frac{1}{2n}||\mathbf{e} - \mathbf{X}\boldsymbol{\beta}||^2_{\boldsymbol{\Lambda}-\boldsymbol{\Lambda}(c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}} + P_\lambda(\boldsymbol{\beta}), \tag{4.8}$$

which can be solved easily. The details of the least square approximation are provided in the Appendix C.1.1.

Comparison with existing Poisson-logNormal graphical models

As we can see, our dependent Poisson graphical model (4.5) resembles the Poisson-logNormal graphical model (4.3) in that both of them are based on Poisson-logNormal distributions. Thus the two graphical models share some properties. For example, both of them can model the over-dispersion effect of scRNA-seq data and thus account for zero-inflation to certain extent.

Despite these similarities, the two graphical models have some key differences. A major distinction between them is the modeling of sample dependence. In particular, the Poisson-logNormal graphical model assumes that the samples are independent, while our model models the sample

dependence by taking advantage of the random effects. Moreover, the Poisson-logNormal graphical model specifies the joint distribution of all nodes through a latent Normal random vector, while our model assumes the conditional distribution of each node. The latent variables in the former model can cause great difficulties in the graph estimation and make it even impossible in high dimensions. In contrast, our model can be easily applied to scRNA-seq data of more than 1,000 dimensions as in the case studies of Section 4.5.

4.2.3 Dependent Hurdle graphical model

The dependent Poisson graphical model accounts for the cell dependence in scRNA-seq data and often works well in practice. However, there are still cases in which the data have excessive zeros that cannot be completely explained by the over-dispersion of the Poisson-logNormal distribution. To tackle with these cases, we propose a Hurdle-logNormal model as follows.

$$Y_{ij}|\mathbf{Y}_{i,-j} \sim \mathrm{B}(1, \mathrm{logit}^{-1}(\gamma_{0j} + \gamma_{1j}\eta_{ij})) \times \mathrm{SPoisson}(e^{\eta_{ij}}), \tag{4.9}$$

where γ_{0j} and γ_{1j} are unknown constants, $\eta_{ij} = \beta_0^{(j)} + \tilde{\mathbf{y}}_{i,-j}^\top \boldsymbol{\beta}^{(j)} + v_{ij}$, and v_{ij} is the random effect term. Here $\mathrm{B}(1,\pi)$ denotes a Bernoulli random variable with success probability π, and SPoisson denotes a shifted Poisson distribution, namely, for $X \sim \mathrm{SPoisson}(\mu)$

$$P(X = k) = \frac{\mu^{k-1}}{(k-1)!}e^{-\mu}, k = 1, 2, \cdots \tag{4.10}$$

The corresponding graph is defined as $\mathcal{G} = (V, E)$, where the vertex set $V = \{1, \cdots, p\}$ and the edge set $E = \{(j, l) : \beta_l^{(j)} \neq 0 \text{ or } \beta_j^{(l)} \neq 0\}$. We call Model (4.9) the dependent Hurdle model.

We choose the shifted Poisson distribution, instead of the more often used zero-truncated Poisson, for the positive part of the Hurdle model because the former is computationally much more efficient to evaluate. Moreover, the Bernoulli and the shifted Poisson models in (4.9) share the linear component η_{ij} instead of having two separate sets of parameters. This reduces the number of parameters and is also a reasonable assumption for scRNA-seq data with $\gamma_{1j} > 0$.

The graph can be estimated via neighborhood selection in a similar way as for dependent Poisson-logNormal models. For simplicity of presentation, we use $\boldsymbol{\beta}$, c, γ_0, γ_1, $\mathbf{X}$, and $\mathbf{y}$ instead

of $(\beta_0^{(j)}, \boldsymbol{\beta}^{(j)\top})^\top$, c_j, γ_{0j}, γ_{1j}, $(\mathbf{1}_n, \tilde{\mathbf{Y}}_{-j})$, and $\mathbf{Y}_j$ in the following when they would not cause misinterpretation. Given a reasonable estimate of $\boldsymbol{\beta}$, denoted as $\mathbf{b}$, we can first estimate γ_0 and γ_1. In particular, we can perform logistic regression of $\mathbf{z} = (\mathbb{I}(y_1 \neq 0), \cdots, \mathbb{I}(y_n \neq 0))^\top$ on $\boldsymbol{\xi} = \mathbf{X}\mathbf{b}$, then the coefficient estimates are our estimates of γ_0 and γ_1.

With respect to the integral form of likelihood, we can again take advantage of the least square approximation and estimate $\boldsymbol{\beta}$ by

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2n} ||\mathbf{e} - \mathbf{X}\boldsymbol{\beta}||_{\tilde{\boldsymbol{\Delta}}+\tilde{\boldsymbol{\Lambda}}-\mathbf{K}-\mathbf{K}^\top} + P_\lambda(\boldsymbol{\beta}), \tag{4.11}$$

where $\mathbf{e}$, $\tilde{\boldsymbol{\Delta}}$, $\tilde{\boldsymbol{\Lambda}}$, and $\mathbf{K}$ are defined in the Appendix C.1.2.

Both our dependent Hurdle-logNormal model and the multivariate Hurdle model (4.4) model the zero-inflation of scRNA-seq data. However, our model also considers the important dependence among cells in single cell profiling. In contrast, it is difficult to generalize Model (4.4) to account for the sample dependence. Moreover, we model the nonzero count data directly with the shifted Poisson distribution, while (4.4) treat them as logNormal random variables.

4.3 Implementation

In this section, we explain some implementation details of our methods. Specifically, we first discuss the estimation of sample dependence in Section 4.3.1, then illustrate how to get an initial estimates for the use of least squares approximation of the two dependent graphical models in Section 4.3.2.

4.3.1 Estimation of the sample dependence

For both dependent graphical models (4.5) and (4.9), it is necessary to pre-specify the sample dependence. This is equivalent to specifying the covariance matrix $\boldsymbol{\Sigma}$ and the variance factors c_j's. Under Model (4.5), we have approximately

$$\tilde{\mathbf{Y}}_j | \mathbf{Y}_{-j} \overset{ind}{\sim} N(\beta_0^{(j)}\mathbf{1} + \tilde{\mathbf{Y}}_{-j}\boldsymbol{\beta}^{(j)}, c_j\boldsymbol{\Sigma}), \tag{4.12}$$

for $j = 1, \cdots, p$. Thus we can estimate $\mathbf{\Sigma}$ and c_j's based on the regression residuals $\mathbf{e}_j = \tilde{\mathbf{Y}}_j - \beta_0^{(j)}\mathbf{1} - \tilde{\mathbf{Y}}_{-j}\boldsymbol{\beta}^{(j)}$. Specifically, we i) standardize $\mathbf{e}_j$ so that $\tilde{\mathbf{e}}_j^\top \tilde{\mathbf{e}}_j/n = 1$ and ii) estimate $\mathbf{\Sigma}$ by the sample covariance of $\{\tilde{\mathbf{e}}_j; j = 1, \cdots, p\}$ and c_j by the sample variance of $\mathbf{e}_j$.

Note that due to the large number of cells, the sample covariance matrix may not be a good estimate of $\mathbf{\Sigma}$, which may undermine the estimation of our dependent graphical models. Thus we use a regularized estimation of $\mathbf{\Sigma}$ instead. In particular, we estimate a regularized $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$ by graphical Lasso with $\{\tilde{\mathbf{e}}_1, \cdots, \tilde{\mathbf{e}}_p\}$ as input, then we estimate $\mathbf{\Sigma}$ by $\hat{\mathbf{\Omega}}^{-1}$. In the rare case of singular $\hat{\mathbf{\Omega}}$, we can estimate $\mathbf{\Sigma}$ by $(\hat{\mathbf{\Omega}} + \delta \mathbf{I}_n)^{-1}$ where δ is a small positive number.

4.3.2 Least squares approximation

For both the dependent Poisson and the dependent Hurdle graphical models, the model estimation is typically difficult due to the integral form of the likelihood function. We proposed to use least squares approximation to solve the problem in Section 4.2.2. To apply the least squares approximation, we first need a reasonable initial estimate of $\boldsymbol{\beta}$ (and c_{0j}, c_{1j} for the dependent Hurdle model).

Specifically, we can get an initial estimate of $\boldsymbol{\beta}^{(j)}$ by ℓ_1-penalized Poisson regression of $\mathbf{Y}_j$ on $\tilde{\mathbf{Y}}_{-j}$. While this may not be the best estimate of $\boldsymbol{\beta}^{(j)}$, it is a reasonable one under the dependent Poisson model. Under the dependent Hurdle model, it is also reliable as long as $c_{1j} \geq 0$, which is generally true for scRNA-seq data. We can select the tuning parameter of the penalized Poisson regression by AIC.

4.3.3 Tuning Parameter Selection

As discussed in Sections 4.2.2 and 4.2.3, both the dependent Poisson and the dependent Hurdle graphical models can be estimated via neighborhood selection. The tuning parameter λ in (4.8) and (4.11) determines the sparsity of the estimated graphs. In practice, we often need to select an appropriate λ and produce the graph estimation. We suggest to choose the tuning parameter by extended BIC (EBIC) (Chen and Chen, 2008). For the neighborhood selection of the node j, the

EBIC is defined as

$$\text{EBIC}_{\lambda}^{(j)} = -2\log L_j(\hat{\boldsymbol{\beta}}^{(j)}) + s\log n + 2\psi \log \binom{p}{s}, 0 \leq \psi \leq 1,$$

where the likelihood $L_j(\boldsymbol{\beta})$ can be computed via least square approximation, and $s = ||\hat{\boldsymbol{\beta}}^{(j)}||_0$. As suggested by Chen and Chen (2008), ψ can be taken as $1 - (2\log p/\log n)^{-1}$.

4.4 Simulation Studies

In this section, we use simulated examples to examine the performance of our new models in a variety of settings. Our methods, namely the dependent Poisson (`dep.poisson`) and the dependent Hurdle (`dep.hurdle`) models, are compared with other existing graph estimation methods, including the graphical Lasso (`glasso`), nonparanormal graphical Lasso (`glasso.npn`), the local Poisson graphical model (`poisson`), and the multivariate Hurdle graphical model (`hurdle`) (McDavid et al., 2016). In particular, with respect to the `glasso` and the `hurdle`, we estimate the graph based on log-transformed data, i.e. $\log(1+Y)$. For the `glasso.npn`, we use graphical Lasso to estimate the graph after performing Normal quantile transformation on the data. For `poisson`, the graph is estimated based on nodewise ℓ_1-penalized Poisson regression of $\mathbf{Y}_j$ on log-transformed $\mathbf{Y}_{-j}$.

In the following, we first introduce the simulations settings, including a non-zero-inflated setting and a zero-inflated one, in Section 4.4.1. Then we present the graph estimation results under different settings in Sections 4.4.2 and 4.4.3 respectively.

4.4.1 Simulation settings

For the simulations, we consider two different models, namely, the hierarchical Poisson-logNormal (HPLN) and the hierarchical Hurdle-logNormal (HHLN), for data generation. In particular, the HPLN model generates data as follows:

$$Y_{ij} \overset{ind}{\sim} \text{Poisson}(e^{Z_{ij}}); 1 \leq i \leq n, 1 \leq j \leq p, \tag{4.13}$$

where $Z_{ij} \sim N(\mu_j, \tau^2)$. The HHLN model generates data as follows:

$$Y_{ij} \sim \mathrm{B}(1, \mathrm{logit}(\gamma_0 + \gamma_1 Z_{ij})\}) \times \mathrm{SPoisson}(e^{Z_{ij}}). \tag{4.14}$$

where $Z_{ij} \sim N(\mu_j, \tau^2)$ and SPoisson denotes the shifted Poisson in (4.10). For both the HPLN and the HHLN models, we simulate $\mathbf{Z} = (Z_{ij})_{n\times p}$ by

$$\begin{aligned} \mathbf{Z}^{(1)} = (\mathbf{z}_1^{(1)\top}, \cdots, \mathbf{z}_n^{(1)\top})^\top, \text{ where } \mathbf{z}_i^{(1)\top} &\overset{iid}{\sim} N(\boldsymbol{\mu}, \boldsymbol{\Phi}), \\ \mathbf{Z}^{(2)} = (\mathbf{z}_1^{(2)}, \cdots, \mathbf{z}_p^{(2)}), \text{ where } \mathbf{z}_j^{(2)} &\overset{iid}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma}), \\ \mathbf{Z} &= \mathbf{Z}^{(1)} + \mathbf{Z}^{(2)}, \end{aligned} \tag{4.15}$$

where $\mathbf{z}_j^{(1)}$ is the j-th row vector of matrix $\mathbf{Z}^{(1)}$, $\mathbf{z}_j^{(2)}$ is the j-th column vector of matrix $\mathbf{Z}^{(2)}$, and the covariance matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Sigma}$ encode the gene interactions and the cell dependence respectively. Define the precision matrices $\boldsymbol{\Psi} = \boldsymbol{\Phi}^{-1}$ and $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$. The true graph $\mathcal{G} = (V, E)$ is defined as $V = \{1, \cdots, p\}$ and $E = \{(j, l); \Psi_{jl} \neq 0\}$.

It can be seen that the HPLN model (4.13) is closer to our dependent Poisson model (4.5) and produces moderate amount of zero values. The HHLN model (4.14) is closer to the dependent Hurdle model (4.9) and can produce excessive amount of zero values. With respect to the sample dependence, we set $\boldsymbol{\Sigma}$ as a blockwise sparse matrix, which resembles the cell dependence in real scRNA-seq data. Specifically, the diagonal of $\boldsymbol{\Sigma}$ consists of four square matrices $\boldsymbol{\Sigma}^{(1)}, \cdots, \boldsymbol{\Sigma}^{(4)}$ of dimensions $p/10$, $p/5$, $3p/10$, and $2p/5$. For $k = 1, \cdots, 4$,

$$\Sigma_{i_1,i_2}^{(k)} = \begin{cases} 1, & \text{if } i_1 = i_2; \\ 0.8, & \text{if } i_1 \neq i_2. \end{cases}$$

With respect to the underlying gene-gene interaction graph, we consider three different graph types and generate the corresponding precision matrix $\boldsymbol{\Psi}$ as follows.

- Banded Graph. The precision matrix $\boldsymbol{\Psi}$ is generated such that $\psi_{jj} = 1$, and $\psi_{jl} = -0.6$ if $|j - l| = 1$, -0.3 if $|j - l| = 2$, and 0 otherwise. Then we add $\delta \geq 0$ to its diagonal such that the minimal eigenvalue is greater than or equal to 0.05.

- Hub Graph. The nodes are divided into 10 equal-size groups: $\{(k-1)p/10+1, \cdots, kp/10\}$ for $k = 1, \ldots, 10$. Set $\psi_{jl} = -0.5$ for all $j = (k-1)p/10+1, (k-1)p/10+1 < l \leq kp/10$, and 0 otherwise. Then we add $\delta \geq 0$ to its diagonal such that the minimal eigenvalue is greater than or equal to 0.05.

- Random Graph. The graph is generated in a way similar to the Barabasi–Albert (BA) model. Starting from an p-dimensional null matrix $\mathbf{L} = \mathbf{O}$, at step i we randomly select $\min\{\lfloor 0.05p \rfloor, i-1\}$ entries in row i with probability $\Pr(i,j) \propto \#\{L_{\ell j} \neq 0 : 1 \leq \ell \leq p\}, j < i$ and assign their values by sampling from Uniform$(-0.8, -0.4)$. Repeat the procedure until $i = p$. Then we get a lower triangular matrix. We construct $\mathbf{\Psi} = \mathbf{L} + \mathbf{L}^\top$ and add $\delta \geq 0$ to its diagonal such that the minimal eigenvalue is greater than or equal to 0.05.

We compare the graph estimation methods under the above settings with $n = 100$ and $p = 80$. For each simulation setting, we run 50 experiments and take their average accuracy evaluation. The evaluation criteria we use include the false positive ratio (FPR), the true positive ratio (TPR), which are defined as follows. Denote the true graph $\mathcal{G}$ and an estimated graph $\hat{\mathcal{G}}$, then

$$\text{TPR} = \frac{|\hat{\mathcal{G}} \cap \mathcal{G}|}{|\mathcal{G}|}, \quad \text{FPR} = \frac{|\hat{\mathcal{G}} \setminus \mathcal{G}|}{(p^2-p)/2 - |\mathcal{G}|},$$

where $|\cdot|$ denotes the number of edges in graph.

4.4.2 Non-zero-inflated data

Under the simulation model (4.13), we estimate the graph with all aforementioned methods and evaluate their accuracy. The results for three different graph type settings are shown in Figure 4.1. We evaluate the whole solution paths of all methods, whose sparsity levels vary by their specific tuning parameters.

It can be seen that our dependent Poisson model has the best performance among all under this non-zero-inflated setting. Compared with the local Poisson model, our dependent Poisson model is significantly improved. This can be due to i) the consideration of sample dependence through the random effect v_{ij} in (4.5) and ii) the inclusion of the random effects that models the over-dispersion effect. The dependent Hurdle model has a similar performance as the Hurdle model (`hurdle`) in

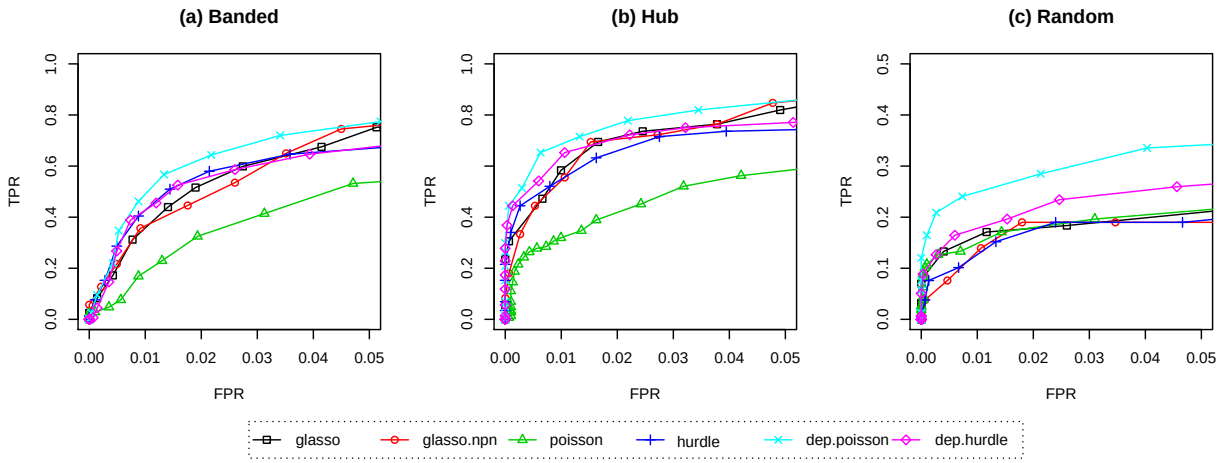

Figure 4.1: Performance of different graph estimation methods under the non-zero-inflated setting (4.13). (a) Banded graph: $\mu_1 = \cdots = \mu_p = 0$, $c = 1/3$; (b) Hub graph: $\mu_1 = \cdots = \mu_p = -1$, $c = 0.5$; (c) Random graph: $\mu_1 = \cdots = \mu_p = 1$, $c = 0.5$. The x-axes and y-axes represent FPR and TPR respectively. The sparsity of estimated graphs by each method varies by its specific tuning parameter.

estimating the banded graph while surpassing it in estimating the hub and the random graphs. This is probably because the latter has almost two times more parameters than the former. While there is a substantial proportion of zeros in the simulated data, it is completely caused by the over-dispersion of the Poisson-logNormal distribution (4.13). Therefore, it does not help to model the zero part specifically, either with `hurdle` or `dep.hurdle`.

4.4.3 Zero-inflated data

Compared to the HPLN model (4.13), the HHLN model (4.14) produces "actual" zero-inflation that cannot be explained by the over-dispersion. It may lead to non-ignorable bias if we fit a model without considering the zero-inflation. We estimate graphs using different methods under this setting and evaluate their performance. The results are shown in Figure 4.2.

According to the results in Figure 4.2, both our dependent Hurdle model and the multivariate Hurdle model capture the zero-inflation of the simulated data. While they have very close performance in estimating the random graphs, our model outperforms the multivariate Hurdle model in other cases. This again verifies the advantages of parameter sharing between the linear parts of the logistic and the Poisson models of our dependent Hurdle model.

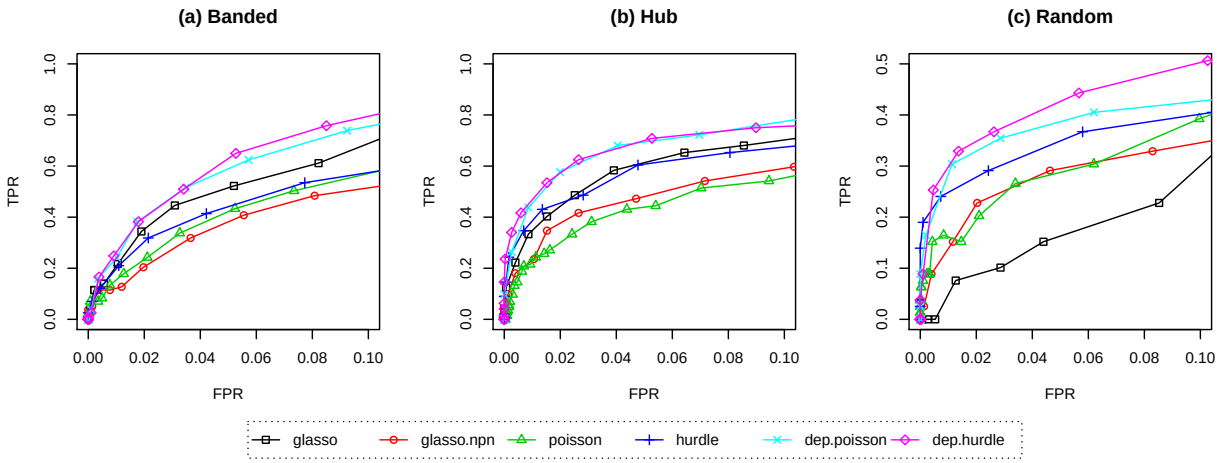

Figure 4.2: Performance of different graph estimation methods under the zero-inflated setting (4.14). (a) Banded graph: $\mu_1 = \cdots = \mu_p = 0$, $c = 1$, $\gamma_0 = -0.5$, $\gamma_1 = 0.5$; (b) Hub graph: $\mu_1 = \cdots = \mu_p = -1$, $c = 1$, $\gamma_0 = -0.5$, $\gamma_1 = 0.3$; (c) Random graph: $\mu_1 = \cdots = \mu_p = 2$, $c = 0.5$, $\gamma_0 = 0$, $\gamma_1 = 0.5$. The x-axes and y-axes represent FPR and TPR respectively. The sparsity of estimated graphs by each method varies by its specific tuning parameter.

We also find that the dependent Poisson model still have a very good performance, despite the data are actually from a model with zero-inflation. A possible reason is that under the Hurdle model (4.14) with $\gamma_1 > 0$, Y_{ij} is more likely to be 0 when Z_{ij} is small, in which case the shifted Poisson distribution also has a low center. Thus a Hurdle model with a positive γ_1 may not be too different from a Poisson model in some cases, and the random effect Poisson regression can still be a reasonable approximation to the data. This is also the case for many scRNA-seq data, so we can always try out the dependent Poisson model in practice.

4.5 Real Data Analysis

In this section, we illustrate the characteristics of scRNA-seq data with two real scRNA-seq datasets. Then we examine the performance of different graphical models on these data. There are several scRNA-seq techniques (Ziegenhain et al., 2017; Svensson et al., 2017) and they can be roughly divided into two groups based on characteristics of resulting scRNA-seq data. The first group, with Smart-seq2 as a typical example, sequence full length RNA and capture more genes per cell. The second group, with droplet based techniques as typical examples, quantify gene expression using unique molecular identifiers (UMIs) and capture less genes per cell but usually with higher

through put to process large number of cells. An UMI is a randomly generated barcode (4-10bp) to label each transcript molecule before amplification. Therefore, by counting UMIs instead of actual reads, one can remove most noise and bias due to amplification. However, UMI is only available for techniques that sequence 5' or 3' ends of the transcript molecule, and thus cannot sequence full length RNA. Since both groups of scRNA-seq techniques are popular in practice, we evaluate our method using two real datasets, one from each group of scRNA-seq techniques.

For the first dataset, Tirosh et al. (2016) disaggregated the melanoma tumors and profiled the single cells by Smart-seq2 technique. The expression of 23,682 genes were measured in 4,645 tumor cells. There are multiple cell types within the dataset, including T cell, B Bell, macrophages, endothelial cell, CAF (cancer associated fibroblast), and tumor (malignant) cells. The second dataset (Gierahn et al., 2017) measures the expression of 24,187 genes in 1,453 human macrophages cells of HEK293 cell line. The cells are profiled with the Seq-Well technique, which is similar to droplet technique. We call them Tirosh and Gierahn datasets respectively.

For the robustness of our analysis, we drop genes that are rarely expressed in the datasets. In particular, genes that are expressed in less than 30% cells of either dataset are removed from both datasets. After the screening, 1,960 genes remain in the two datasets.

4.5.1 Exploratory data analysis

The abundance of zero values is probably the most significant feature of scRNA-seq data. Figure 4.3 displays the expression proportion of 1,960 genes in the two datasets. Even after the screening, the proportions of zero values are still as high as around 50% in both datasets.

Since excessive zeros in the data can also be caused by over-dispersion, we fit the genes with the Poisson-logNormal (PLN) distribution. By comparing the expected zero proportions under the fitted PLN model with the actual zero proportions (Figure 4.3, right), we observe that it is approximately the case for the Gierahn data but very unlikely for the Tirosh one.

The dependence among samples is another characteristic of scRNA-seq data. With the two datasets, we first centralize the expression of each gene, then compute the correlation among cells. As we can see from Figure 4.4, there is substantial dependence among cells in the two datasets. In particular, 25.74% of the cell pairs in the Tirosh dataset have an absolute correlation coefficient

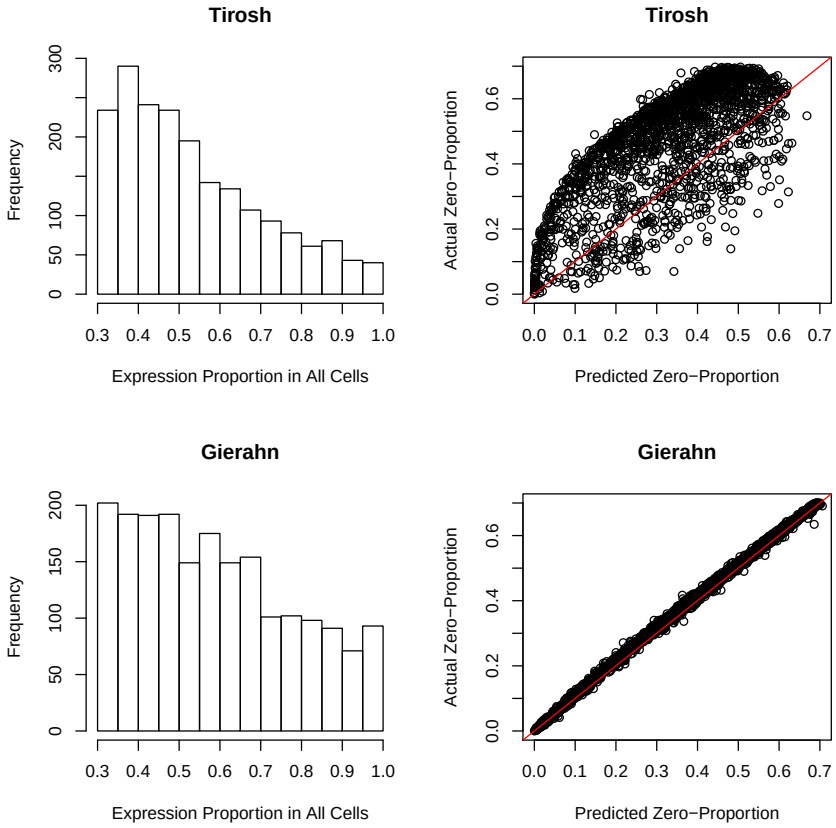

Figure 4.3: Left: Histogram of the expression proportions of 1,960 genes in the Tirosh and the Gierahn datasets. For example, more than 200 genes are expressed in only 30% – 35% of the cells in the Tirosh dataset. Right: The actual zero proportions versus the expected zero proportions for 1,960 genes under fitted PLN model.

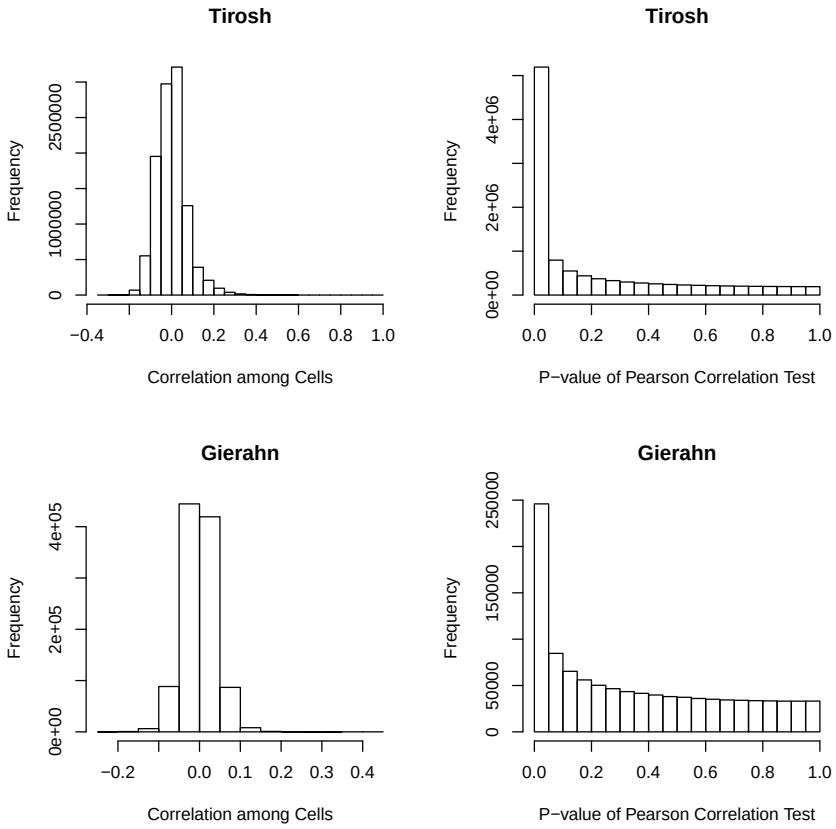

Figure 4.4: Left: Histogram of the sample correlation between cell pairs in the Tirosh and the Gierahn datasets. Right: Histogram of the P-values of Pearson correlation tests for all cell pairs.

greater than 0.1, and about 4% of the pairs have absolute correlation greater than 0.2. The Pearson test of correlation is rejected (significance level 1%) for 69.26% of the cell pairs in the Tirosh dataset and 20.60% of the cell pairs in the Gierahn dataset.

4.5.2 Graph estimation

To examine the performance of our models in practice, we estimate the gene interactions with the two datasets and evaluate their accuracy. As in Section 4.4, we estimate the graph with different models and compare them with the benchmark graph. Since the true gene interaction relationship is unknown, we construct a benchmark graph based on the PathwayCommons database (Cerami et al., 2010), which aggregates gene relationship findings from existing research literature. Specifically, we construct the benchmark graph by connecting all gene pairs that are annotated as "interacts-with" in the database.

The results are displayed in Figure 4.5. For the Tirosh data, the dependent Hurdle graphical model discovers significantly more true gene interactions than other methods at the same sparsity level. This matches our observation in Section 4.5.1 that the data are likely zero-inflated even taking into account of the over-dispersion effect. While the Poisson-logNormal distribution does not suit the Tirosh data well, the dependent Poisson model is greatly improved over the local Poisson graphical model and outperforms all methods except the dependent Hurdle model. At the same time, we observe that the multivariate Hurdle model also produces improved graph estimation compared to the graphical Lasso by modeling the zero-inflation.

For the Gierahn data, our dependent Poisson and dependent Hurdle graphical models have the best performance among all. As we mentioned in Section 4.5.1, the zero values in the data can be largely explained by the over-dispersion effect. While the dependent Hurdle model considers possible zero-inflation effect, it does not bring significant advantages over the dependent Poisson model. This coincides our observations in the simulation studies in Section 4.4.2. By comparing the local Poisson graphical model and our dependent Poisson model, the latter is improved by taking into account the sample dependence. In fact, there is not much difference among all other methods, including the multivariate Hurdle model.

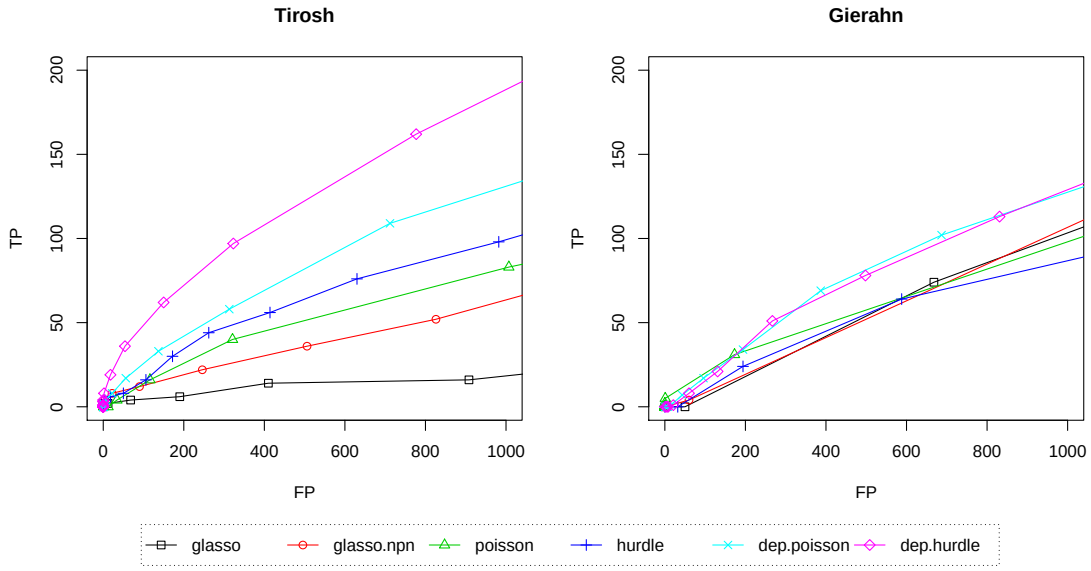

Figure 4.5: Accuracy evaluation of graph estimation with real scRNA-seq datasets (left: with dataset from Tirosh et al. (2016); right: with dataset from Gierahn et al. (2017))

.

4.6 Summary

As scRNA-seq data accumulate with the evolving sequencing techniques, statistical methods adapted to scRNA-seq data are in demand for accurate analysis. In this paper, we have explored the characteristics of scRNA-seq data, then proposed two graphical models, namely, the dependent Poisson and the dependent Hurdle models, that accommodates such characteristics. Particularly, we have also developed an efficient algorithm for model estimation.

With simulation and real case studies, we find that the dependent Poisson model works generally well for different types of data by taking into account the cell dependence and the over-dispersion effects of scRNA-seq data. For some scRNA-seq data with excessive zeros that cannot be accounted by the over-dispersion effect, the dependent Hurdle model can work better than the dependent Poisson one.

APPENDIX A

SUPPLEMENTARY MATERIALS FOR THE GSLDA METHOD

A.1 Some comments on the GSLDA method

A.1.1 A graphical display of the discriminant vector decomposition

$$\boldsymbol{\beta}^* = \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & 0 \\ \omega_{31} & 0 & \omega_{33} \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{pmatrix} = \begin{pmatrix} \delta_1\omega_{11} + \delta_2\omega_{21} + \delta_3\omega_{31} \\ \delta_1\omega_{12} + \delta_2\omega_{22} \\ \delta_1\omega_{13} + \delta_3\omega_{33} \end{pmatrix}$$

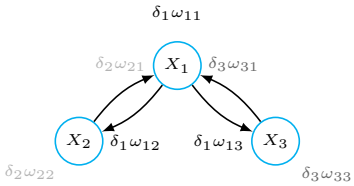

Figure A.1: A 3-dimensional LDA example demonstrating how marginal differences of the three features $(\delta_1, \delta_2, \delta_3)$ contribute to the predictive power of all features. Here $\omega_{23} = \omega_{32} = 0$. The terms around each node represent a decomposition of the corresponding coefficient. The gray scale of each term and the edge direction together indicate the source of the marginal differences.

A.1.2 Connection between GSLDA and existing methods

We first consider the case when $\mathcal{G}$ is a complete graph. Without loss of generality, we assume that there is a unique minimum weight, i.e., there exists an ℓ such $\tau_\ell < \tau_j$ for all $j \neq \ell$. In this case, for any $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\mathbf{v}^{(1)} + \cdots + \mathbf{v}^{(p)} = \boldsymbol{\beta}$, we have

$$\sum_{j=1}^{p} \tau_j \|\mathbf{v}^{(j)}\|_2 \geq \tau_\ell \sum_{j=1}^{p} \|\mathbf{v}^{(j)}\|_2 \geq \tau_\ell \|\boldsymbol{\beta}\|_2.$$

By taking $\mathbf{v}^{(l)} = \boldsymbol{\beta}$ and $\mathbf{v}^{(j)} = \mathbf{0}$ for all $j \neq \ell$, the regularization (2.5) becomes $||\boldsymbol{\beta}||_{\mathcal{G},\boldsymbol{\tau}} = \tau_\ell ||\boldsymbol{\beta}||_2$. Similarly, we can show the equivalence in the case where $\mathcal{G}$ consists of K disjoint complete subgraphs.

A.2 Numerical results

A.2.1 Graph estimation results

To better understand the performance of our proposed GSLDA methods, we also present the graph estimation results for the methods. In particular, we compare the graph estimation based on both labeled data (for supervised GSLDA) and unlabeled data (for semi-supervised GSLDA) with the true graphs, within-class graph $\mathcal{G}$ and overall graph $\tilde{\mathcal{G}}$. The accuracy metrics include false positives (TP) and false positives (FP).

Table A.1: Graph estimation accuracy for all examples in the simulations. The graphs are estimated with labeled data (L) after centering, or with unlabeled data (U). The former estimation is compared with $\mathcal{G}$, and the latter is compared with both $\mathcal{G}$ and $\tilde{\mathcal{G}}$. The results are averaged over 100 repetitions and the standard errors are provided in the parentheses.

Graph Type	Data	TP	FP	Size	True Size
Block Sparse	L	51.54 (0.25)	6.86 (0.48)	58.4 (0.55)	$\mathcal{G}$: 100 (0)
	U	100 (0)	82.96 (0.66)	182.96 (0.66)	$\mathcal{G}$: 100 (0)
	U	176.48 (0.46)	6.48 (0.38)	182.96 (0.66)	$\tilde{\mathcal{G}}$: 600 (0)
AR(3)	L	468.02 (1.31)	69.64 (1.24)	537.66 (1.34)	$\mathcal{G}$: 1188 (0)
	U	1178.04 (0.38)	76.72 (0.87)	1254.76 (1.01)	$\mathcal{G}$: 1188 (0)
	U	1235.76 (0.75)	19 (0.61)	1254.76 (1.01)	$\tilde{\mathcal{G}}$: 2508 (0)
Random Sparse	L	353.14 (2.02)	69.34 (1.25)	422.48 (1.73)	$\mathcal{G}$: 818 (0)
	U	814.52 (0.18)	72.74 (1.04)	887.26 (1.12)	$\mathcal{G}$: 818 (0)
	U	866.14 (0.87)	21.12 (0.70)	887.26 (1.12)	$\tilde{\mathcal{G}}$: 2426 (0)
Scale-Free	L	374.92 (1.37)	32.44 (0.92)	407.36 (1.48)	$\mathcal{G}$: 776 (0)
	U	709.88 (0.73)	103.5 (1.00)	813.38 (1.18)	$\mathcal{G}$: 776 (0)
	U	799.08 (1.05)	14.3 (0.53)	813.38 (1.18)	$\tilde{\mathcal{G}}$: 3564 (0)

A.2.2 Additional simulation results

The misclassification rates may not reflect the comprehensive performance of classification models, especially when the classes are unbalanced. Thus we present the receiver operating characteristic (ROC) curve for the classification models. Besides the balanced class setting as in the main text, we also consider an unbalanced class setting in which Class-0 accounts for 80% of the whole dataset. As we can see from Figures A.2 and A.3, our methods still outperforms other methods in terms of higher sensitivities at each specificity level.

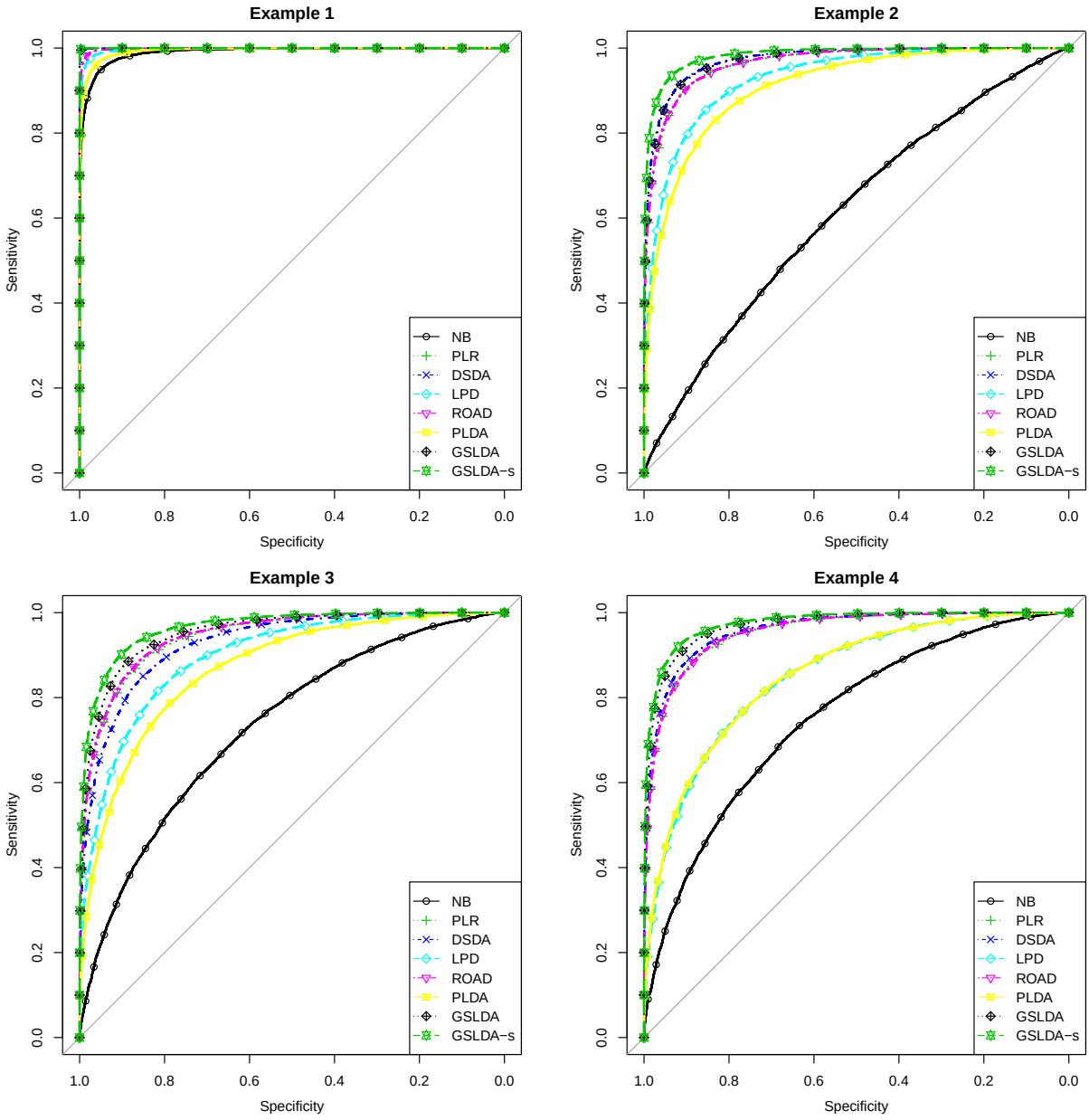

Figure A.2: ROC Curve under the balanced setting for the four examples. The proportion of Class-0 sample is 50%. The ROC curve is computed based on 100 repetitions.

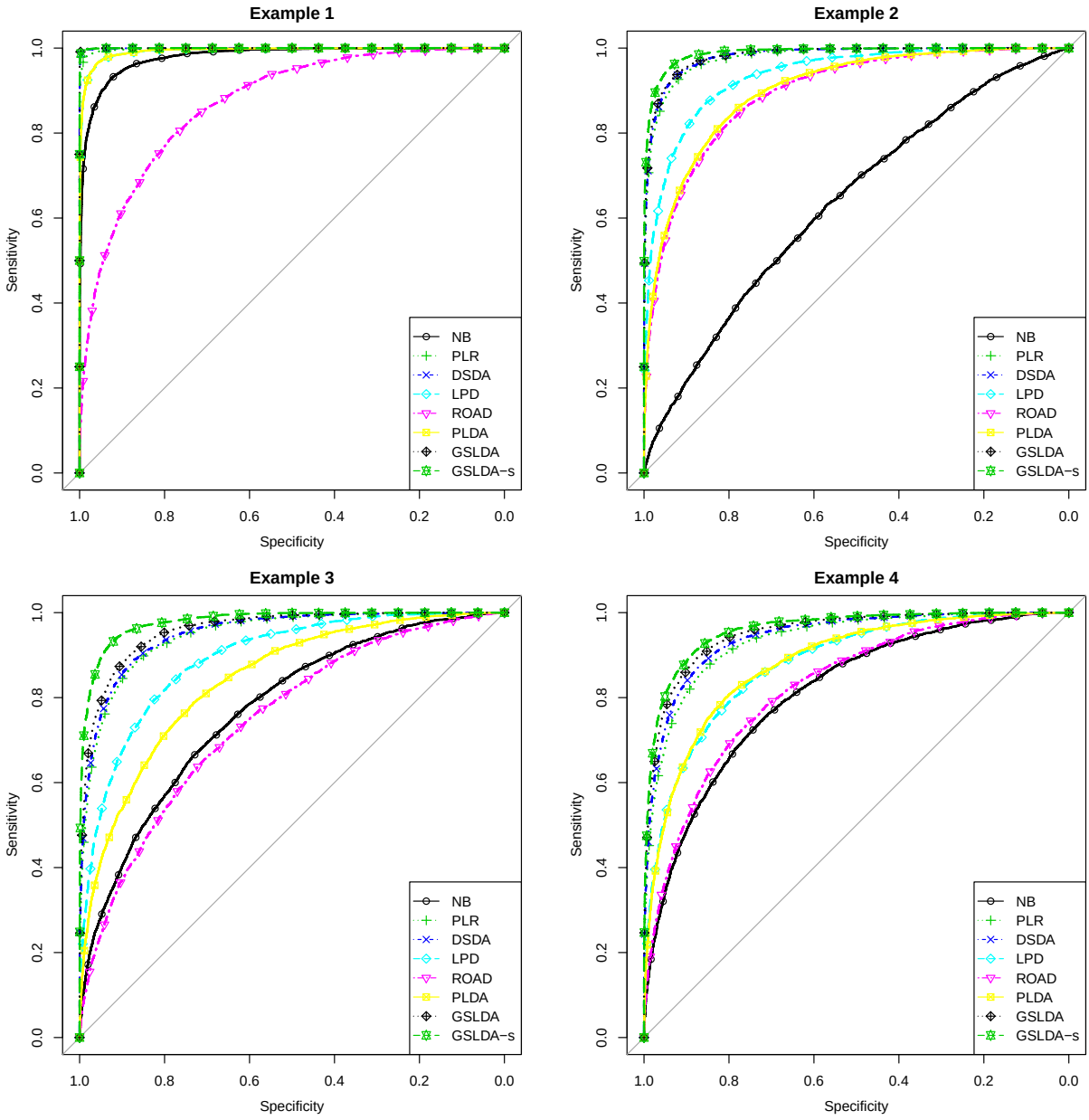

Figure A.3: ROC Curve under the unbalanced setting for the four examples. In particular, the proportion of Class-0 sample is 80%. The ROC curve is computed based on 100 repetitions.

A.3 Proofs to the theoretical results

A.3.1 Proof of Proposition 1

The random variable $\mathbf{X}$ can be represented as $\mathbf{X} = \xi\mathbf{Z}_1 + (1-\xi)\mathbf{Z}_2$, where (i) $\xi \sim \text{Bin}(1, \pi_1)$ is a Bernoulli random variable and (ii) $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are from the two population components, respectively. Moreover, ξ, $\mathbf{Z}_1$, and $\mathbf{Z}_2$ are mutually independent. We have $(\mathbf{Z}_1) = (\mathbf{Z}_2) = \boldsymbol{\Sigma}$, $\text{E}\mathbf{Z}_1 = \boldsymbol{\mu}^{(1)}$, and $\text{E}\mathbf{Z}_2 = \boldsymbol{\mu}^{(2)}$. Then $\text{E}(\mathbf{X}) = \pi_1\boldsymbol{\mu}^{(1)} + \pi_2\boldsymbol{\mu}^{(2)}$ and

$$\begin{aligned}\text{E}(\mathbf{X}\mathbf{X}^\top) =& \text{E}\{\xi^2\mathbf{Z}_1\mathbf{Z}_1^\top + (1-\xi)^2\mathbf{Z}_2\mathbf{Z}_2^\top + \xi(1-\xi)\mathbf{Z}_1\mathbf{Z}_2^\top + \xi(1-\xi)\mathbf{Z}_2\mathbf{Z}_1^\top\} \\ =& \pi_1\text{E}(\mathbf{Z}_1\mathbf{Z}_1^\top) + \pi_2\text{E}(\mathbf{Z}_2\mathbf{Z}_2^\top).\end{aligned}$$

Thus the overall covariance matrix is $(\mathbf{X}) = \boldsymbol{\Sigma} + \pi_1\pi_2\boldsymbol{\delta}\boldsymbol{\delta}^\top$, where $\boldsymbol{\delta} = \boldsymbol{\mu}^{(1)} - \boldsymbol{\mu}^{(2)}$.

Now we verify the inverse matrix of $(\mathbf{X})$, i.e., the overall precision matrix of the mixture distribution. By setting $c = \pi_1\pi_2/(1 + \pi_1\pi_2\boldsymbol{\delta}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta})$, we have

$$(\boldsymbol{\Sigma}^{-1} - c\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta}\boldsymbol{\delta}^\top\boldsymbol{\Sigma}^{-1})(\boldsymbol{\Sigma} + \pi_1\pi_2\boldsymbol{\delta}\boldsymbol{\delta}^\top) = \mathbf{I} + \pi_1\pi_2\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta}\boldsymbol{\delta}^\top - c\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta}\boldsymbol{\delta}^\top - \pi_1\pi_2 c\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta}\boldsymbol{\delta}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta}\boldsymbol{\delta}^\top = \mathbf{I}.$$

Denote $\boldsymbol{\beta}^* = \boldsymbol{\Sigma}^{-1}\boldsymbol{\delta}$. Then we have $(\mathbf{X})^{-1} = \boldsymbol{\Sigma}^{-1} - c\boldsymbol{\beta}^*\boldsymbol{\beta}^{*\top}$. □

A.3.2 Proof of Theorem 1

Before the proof, we introduce a lemma from Cai et al. (2011). The proof is omitted.

Lemma 5. *Let $\xi_1, \ldots, \xi_n$ be independent random variables with mean zero. Suppose that there exists some $t > 0$ and $\bar{B}_n$ such that $\sum_{k=1}^n \text{E}(\xi_k^2 e^{t|\xi_k|}) \le \bar{B}_n^2$. Set $C_t = t + t^{-1}$. Then uniformly for $x \in (0, \bar{B}_n]$,*

$$\Pr\left(\sum_{k=1}^n \xi_k \ge C_t\bar{B}_n x\right) \le \exp(-x^2).$$

Denote $\xi_1 = ||\tilde{\mathbf{S}}_{AA} - \tilde{\boldsymbol{\Sigma}}_{AA}||_\infty$, $\xi_2 = ||\tilde{\mathbf{S}}_{AA}^{-1} - \tilde{\boldsymbol{\Sigma}}_{AA}^{-1}||_\infty$, and $\xi = ||\tilde{\mathbf{S}}_{A^\complement A}\tilde{\mathbf{S}}_{AA}^{-1} - \tilde{\boldsymbol{\Sigma}}_{A^\complement A}\tilde{\boldsymbol{\Sigma}}_{AA}^{-1}||_\infty$. With simple calculations, one can show that for all $\boldsymbol{\beta} \in \mathbb{R}^p$,

$$\min_{\beta_0\in\mathbb{R}} \sum_{i=1}^n (y_i - \beta_0 - \mathbf{x}_i^\top\boldsymbol{\beta})^2 = \sum_{i=1}^n (y_i - \dot{\mathbf{x}}_i^\top\boldsymbol{\beta})^2,$$

where $\dot{\mathbf{x}}_i = \mathbf{x}_i - (\mathbf{x}_1 + \cdots + \mathbf{x}_n)/n$ is the centralized feature vector. Thus the loss function of GSLDA in (2.4) is equivalent to $\sum_{i=1}^{n}(y_i - \dot{\mathbf{x}}_i{}^\top \boldsymbol{\beta})^2/n + \lambda||\boldsymbol{\beta}||_{\mathcal{G},\boldsymbol{\tau}}$. In the rest of our proof, we assume the sample $\mathbf{X}$ has been centralized. Then the GSLDA formulation (2.4) becomes

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}} ||Y - \mathbf{X}\boldsymbol{\beta}||_2^2/n + \lambda||\boldsymbol{\beta}||_{\mathcal{G},\boldsymbol{\tau}}. \tag{A.1}$$

Under the assumption (A1), we can define

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}\in\mathbb{R}^s}{\operatorname{argmin}} ||Y - \mathbf{X}_A\boldsymbol{\gamma}||_2^2/n + \lambda||\boldsymbol{\gamma}||_{\mathcal{G}_A,\boldsymbol{\tau}_A}, \tag{A.2}$$

where $\mathcal{G}_A$ denotes the subgraph of $\mathcal{G}$ corresponding to A. If we can show that (i) all elements of $\hat{\boldsymbol{\gamma}}$ are non-zero; and (ii) $\hat{\boldsymbol{\beta}}$ with $\hat{\boldsymbol{\beta}}_A = \hat{\boldsymbol{\gamma}}$ and $\hat{\boldsymbol{\beta}}_{A^\complement} = \mathbf{0}$ solves (A.1); then, GSLDA estimation recovers all significant features accurately.

We first show statement (i). By Section 4.6 of Obozinski et al. (2011), the formulation (A.2) is equivalent to $\hat{\boldsymbol{\gamma}} = \sum_{j\in A}\hat{\mathbf{u}}^{(j)}$ where

$$\{\hat{\mathbf{u}}^{(j)} : j \in A\} = \underset{\mathbf{u}^{(j)}\in\mathbb{R}^s:\operatorname{supp}(\mathbf{u}^{(j)})\subseteq\mathcal{N}^{(j)}\cap A, j\in A}{\operatorname{argmin}} ||Y - \sum_{j\in A}\mathbf{X}_A\mathbf{u}^{(j)}||_2^2/n + \lambda\sum_{j\in A}\tau_j||\mathbf{u}^{(j)}||_2.$$

Since this is a convex optimization problem, any solution $\{\mathbf{u}^{(j)} : j \in A\}$ satisfies the KKT conditions (Boyd and Vandenberghe, 2004), which are for all $j \in A$, either

$$\mathbf{u}^{(j)} \neq \mathbf{0} \quad \text{and} \quad 2\mathbf{X}_{\mathcal{N}^{(j)}}^\top(Y - \mathbf{X}_A\boldsymbol{\gamma})/n = \lambda\tau_j\mathbf{u}_{\mathcal{N}^{(j)}}^{(j)}/||\mathbf{u}^{(j)}||_2,$$

or

$$\mathbf{u}^{(j)} = \mathbf{0} \quad \text{and} \quad 2||\mathbf{X}_{\mathcal{N}^{(j)}}^\top(Y - \mathbf{X}_A\boldsymbol{\gamma})||_2/n \leq \lambda\tau_j,$$

where $\boldsymbol{\gamma} = \sum_{j\in A} \mathbf{u}^{(j)}$. Thus we have $||\mathbf{X}_A^\top(Y - \mathbf{X}_A\hat{\boldsymbol{\gamma}})||_\infty/n \le \lambda\tau^*/2$, and we can write $\hat{\boldsymbol{\gamma}}$ as $\hat{\boldsymbol{\gamma}} = \tilde{\mathbf{S}}_{AA}^{-1}(\hat{\boldsymbol{\delta}}_A + \lambda\tau^*\mathbf{t}_A/2)$, where $\mathbf{t}_A \in \mathbb{R}^s$ satisfies $||\mathbf{t}_A||_\infty \le 1$. We have

$$\begin{aligned}
||\hat{\boldsymbol{\gamma}} - \boldsymbol{\beta}_A^\dagger||_\infty &= ||(\tilde{\mathbf{S}}_{AA}^{-1} - \tilde{\boldsymbol{\Sigma}}_{AA}^{-1})\boldsymbol{\delta}_A - \tilde{\mathbf{S}}_{AA}^{-1}(\boldsymbol{\delta}_A - \hat{\boldsymbol{\delta}}_A) + \tilde{\mathbf{S}}_{AA}^{-1}\lambda\tau^*\mathbf{t}_A/2||_\infty \\
&\le ||\boldsymbol{\delta}_A||_\infty\xi_2 + (\varphi + \xi_2)||\hat{\boldsymbol{\delta}}_A - \boldsymbol{\delta}_A||_\infty + \lambda\tau^*(\varphi + \xi_2)/2 \\
&\le \xi_2(||\boldsymbol{\delta}_A||_\infty + ||\hat{\boldsymbol{\delta}}_A - \boldsymbol{\delta}_A||_\infty + \lambda\tau^*/2) + \varphi(||\hat{\boldsymbol{\delta}}_A - \boldsymbol{\delta}_A||_\infty + \lambda\tau^*/2) \\
&\le \frac{\varphi^2\xi_1}{1-\varphi\xi_1}(||\boldsymbol{\delta}_A||_\infty + ||\hat{\boldsymbol{\delta}}_A - \boldsymbol{\delta}_A||_\infty + \lambda\tau^*/2) + \varphi(||\hat{\boldsymbol{\delta}}_A - \boldsymbol{\delta}_A||_\infty + \lambda\tau^*/2) \equiv L_1,
\end{aligned}$$

in which the second inequality holds for sufficiently large n because $\varphi\xi_1 \le 1$ and $\xi_2 \le (1 - \varphi\xi_1)^{-1}\varphi^2\xi_1$. If $\xi_1 \le \epsilon$ and $||\hat{\boldsymbol{\delta}}_A - \boldsymbol{\delta}_A||_\infty \le \epsilon$, then $L_1 = O(\epsilon) + \lambda\tau^*\varphi/2 > 0$, which proves (i). By Lemma 5, the statement (i) is true with probability at least $1 - 2s^2\exp(-a_1 n\epsilon^2/s^2) - 2s\exp(-a_2 n\epsilon^2)$, for some positive a_1 and a_2.

Now we prove statement (ii). The formulation (A.1) is equivalent to $\hat{\boldsymbol{\beta}} = \sum_{j=1}^p \hat{\mathbf{v}}^{(j)}$, where

$$\{\hat{\mathbf{v}}^{(1)}, \ldots, \hat{\mathbf{v}}^{(p)}\} = \underset{\mathbf{v}^{(j)}:\mathrm{supp}(\mathbf{v}^{(j)})\subseteq\mathcal{N}^{(j)},\ 1\le j\le p}{\operatorname{argmin}} \frac{1}{n}\left\|Y - \sum_{j=1}^p \mathbf{X}\mathbf{v}^{(j)}\right\|_2^2 + \lambda\sum_{j=1}^p ||\mathbf{v}^{(j)}||_2. \tag{A.3}$$

This is also a convex optimization problem and the KKT conditions of formulation (A.3) are for all $j \in \{1, \ldots, p\}$, either

$$\mathbf{v}^{(j)} \ne \mathbf{0} \quad \text{and} \quad 2\mathbf{X}_{\mathcal{N}^{(j)}}^\top(Y - \mathbf{X}\boldsymbol{\beta})/n = \lambda\tau_j\mathbf{v}_{\mathcal{N}^{(j)}}^{(j)}/||\mathbf{v}^{(j)}||_2, \tag{A.4}$$

or

$$\mathbf{v}^{(j)} = \mathbf{0} \quad \text{and} \quad 2||\mathbf{X}_{\mathcal{N}^{(j)}}^\top(Y - \mathbf{X}\boldsymbol{\beta})||_2/n \le \lambda\tau_j, \tag{A.5}$$

where $\boldsymbol{\beta} = \mathbf{v}^{(1)} + \cdots + \mathbf{v}^{(p)}$. Let $\mathbf{v}^{(j)} = 0$ for all $j \in A^\complement$, and $\mathbf{v}_A^{(j)} = \mathbf{u}^{(j)}$, $\mathbf{v}_{A^\complement}^{(j)} = \mathbf{0}$ for all $j \in A$. Then $\boldsymbol{\beta}_A = \hat{\boldsymbol{\gamma}}$ and $\boldsymbol{\beta}_{A^\complement} = \mathbf{0}$.

For $j \in A$, (A.4) holds owing to the definition of $\hat{\boldsymbol{\gamma}}$. For $j \in A^\complement$, $2||\mathbf{X}_{\mathcal{N}^{(j)}}^\top(Y - \mathbf{X}\boldsymbol{\beta})||_2/n \le 2\sqrt{|\mathcal{N}^{(j)}|}||\mathbf{X}_{\mathcal{N}^{(j)}}^\top(Y - \mathbf{X}\boldsymbol{\beta})||_\infty/n$. Denote $\eta = ||\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}||_\infty$. If $||\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}||_\infty \le \epsilon$ and $||\tilde{\mathbf{S}}_{A^\complement A} - \tilde{\boldsymbol{\Sigma}}_{A^\complement A}||_\infty \le \epsilon$,

then

$$\begin{aligned}||\mathbf{X}_{A^{\complement}}^{\top}(Y-\mathbf{X}_A\hat{\boldsymbol{\gamma}})||_\infty/n &= ||\tilde{\mathbf{S}}_{A^{\complement}A}\tilde{\mathbf{S}}_{AA}^{-1}(\hat{\boldsymbol{\delta}}_A+\lambda\tau^*\mathbf{t}_A/2)-\hat{\boldsymbol{\delta}}_{A^{\complement}}||_\infty \\ &\leq (||\boldsymbol{\delta}_A||_\infty+1+\epsilon+\kappa+\lambda\tau^*/2)\epsilon+\lambda\tau^*\kappa/2 \\ &\leq O(\epsilon)+\lambda\tau_*/2.\end{aligned}$$

By Lemma 5, the statement (ii) is true with probability at least $1-2ps\exp(-a_1n\epsilon^2/s^2)-2p\exp(-a_2n\epsilon^2)$. By taking $\epsilon=\sqrt{\ln p/n}$, the active set is recovered and $||\hat{\boldsymbol{\beta}}_A-\boldsymbol{\beta}_A^{\dagger}||_\infty\leq O(\sqrt{\ln p/n})$ with probability at least $1-2s^2\exp(-a_1n\epsilon^2/s^2)-2ps\exp(-a_1n\epsilon^2/s^2)-2p\exp(-a_2n\epsilon^2)=1-O(p^{-C_1})$ for some $C_1>0$. □

A.3.3 Proof of Theorem 2

The proof uses the following lemma from Negahban et al. (2010).

Lemma 6. *Denote $\mathcal{M}$ a subspace of $\mathbb{R}^p$ and $\mathcal{M}^{\perp}$ its orthogonal complement. For a regularized estimation problem*

$$\hat{\boldsymbol{\theta}}=\arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p}L(\boldsymbol{\theta})+\lambda R(\boldsymbol{\theta}),$$

where

(i) R is a norm and is decomposable with respect to $(\mathcal{M},\mathcal{M}^{\perp})$, i.e., $R(\boldsymbol{\theta}+\boldsymbol{\eta})=R(\boldsymbol{\theta})+R(\boldsymbol{\eta})$ for all $\boldsymbol{\theta}\in\mathcal{M},\boldsymbol{\eta}\in\mathcal{M}^{\perp}$;

(ii) L is convex and differentiable, and satisfies restricted strong convex condition with curvature κ_L, i.e., $\delta L(\boldsymbol{\theta}^*,\boldsymbol{\Delta})=L(\boldsymbol{\theta}^*+\boldsymbol{\Delta})-L(\boldsymbol{\theta}^*)-\nabla L(\boldsymbol{\theta}^*)^{\top}\boldsymbol{\Delta}\geq\kappa_L||\boldsymbol{\Delta}||_2^2$ for some $\boldsymbol{\theta}^*$, for all $\boldsymbol{\Delta}$ such that $R(\boldsymbol{\Delta}_{\mathcal{M}^{\perp}})\leq 3R(\boldsymbol{\Delta}_{\mathcal{M}})+4R(\boldsymbol{\theta}^*_{\mathcal{M}^{\perp}})$.

Let $\lambda\geq 2R^*\{\nabla L(\boldsymbol{\theta}^*)\}$, where R^* denote the dual norm of R, then any solution $\hat{\boldsymbol{\theta}}$ to the problem satisfies

$$||\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*||_2\leq 9\lambda^2\psi(\mathcal{M})/\kappa_L^2+4\lambda R^*(\boldsymbol{\theta}^*_{\mathcal{M}^{\perp}})/\kappa_L,$$

where $\psi(\mathcal{M})=\sup_{\mathbf{u}\in\mathcal{M}/\{0\}}R(\mathbf{u})/||\mathbf{u}||_2$.

Proof. In the GSLDA formulation, the loss function is $L(\beta_0,\boldsymbol{\beta})=||\mathbf{y}-\beta_0\mathbf{1}-\mathbf{X}\boldsymbol{\beta}||_2^2/n$, and the regularization is $R(\boldsymbol{\beta})=||\boldsymbol{\beta}||_{\mathcal{G},\boldsymbol{\tau}}$. It has been shown in Obozinski et al. (2011) that R is a norm

and its dual norm is $R^*(\mathbf{u}) = \max_{1\le j\le p} \tau_j^{-1}||\mathbf{u}_{\mathcal{N}^{(j)}}||_2$. When we take $\tau_j = \sqrt{|\mathcal{N}^{(j)}|}$, $R^*(\mathbf{u}) \le \max_j ||\mathbf{u}_{\mathcal{N}^{(j)}}||_\infty = ||\mathbf{u}||_\infty$.

For some $\epsilon > 0$, denote the event $\chi = \{||\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}||_\infty \le \epsilon, |||\tilde{\mathbf{S}}_{\cdot,A} - \tilde{\boldsymbol{\Sigma}}_{\cdot,A}|||_\infty \le \epsilon\}$. Then by Lemma 4, $\Pr(\chi) \ge 1 - 2p\exp(-a_2 n\epsilon^2) - 2ps\exp(-a_1 n\epsilon^2)$. Under the event χ,

$$
\begin{aligned}
||\nabla L(\boldsymbol{\beta}^\dagger)||_\infty &= ||2n^{-1}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\dagger)||_\infty \\
&= ||2(\hat{\boldsymbol{\delta}} - \tilde{\mathbf{S}}\boldsymbol{\beta}^\dagger)||_\infty = ||2(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) - 2(\tilde{\mathbf{S}} - \tilde{\boldsymbol{\Sigma}})\boldsymbol{\beta}^\dagger||_\infty \le 2||\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}||_\infty + 2||(\tilde{\mathbf{S}} - \tilde{\boldsymbol{\Sigma}})_{\cdot,A}\boldsymbol{\beta}^\dagger_A||_\infty \\
&\le 2||\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}||_\infty + 2||(\tilde{\mathbf{S}} - \tilde{\boldsymbol{\Sigma}})_{\cdot,A}||_\infty||\boldsymbol{\beta}^\dagger_A||_1 \le 2\epsilon + 2||\boldsymbol{\beta}^\dagger_A||_1\epsilon.
\end{aligned}
$$

We take $\epsilon = C_2\sqrt{\ln p/n}$ where $C_2 > (a_1 \wedge a_2)^{-1}$. Then $\lambda \ge 2R^*(\nabla L(\boldsymbol{\beta}^\dagger))$. Under the event χ with $\epsilon \le c\sigma$ for some $c > 0$, for sufficiently large n, we have $\boldsymbol{\Delta}^\top\tilde{\mathbf{S}}\boldsymbol{\Delta} \ge \boldsymbol{\Delta}^\top\tilde{\boldsymbol{\Sigma}}\boldsymbol{\Delta} - |\boldsymbol{\Delta}^\top(\tilde{\mathbf{S}} - \tilde{\boldsymbol{\Sigma}})\boldsymbol{\Delta}| \ge \sigma||\boldsymbol{\Delta}||_2^2/2$ for $\boldsymbol{\Delta} \in \mathcal{C}(A)$. Thus $\delta L(\boldsymbol{\beta}^*, \boldsymbol{\Delta}) \ge 2\boldsymbol{\Delta}^\top\tilde{\mathbf{S}}\boldsymbol{\Delta} \ge \sigma||\boldsymbol{\Delta}||_2^2$ for all $\boldsymbol{\Delta} \in \mathcal{C}(A)$.

We take $\mathcal{M} = \{\boldsymbol{\beta} \in \mathbb{R}^p : \boldsymbol{\beta}_{A^\complement} = \mathbf{0}\}$. Then $\boldsymbol{\beta}^\dagger \in \mathcal{M}$ and $\boldsymbol{\beta}^\dagger_{\mathcal{M}^\perp} = \mathbf{0}$. Moreover,

$$
\psi(\mathcal{M}) = \sup_{\boldsymbol{\beta}\in\mathcal{M}} \frac{R(\boldsymbol{\beta})}{||\boldsymbol{\beta}||_2} = \sup_{\boldsymbol{\beta}_{A^\complement}=\mathbf{0}} \frac{\min_{\sum \mathbf{v}^{(j)}=\boldsymbol{\beta}} \sum \tau_j||\mathbf{v}^{(j)}||_2}{||\boldsymbol{\beta}||_2} \le \sup_{\boldsymbol{\beta}_{A^\complement}=\mathbf{0}} \frac{\sum_{j\in A}\tilde{\tau}_j|\beta_j|}{||\boldsymbol{\beta}_A||_2} \le \tau^*\sqrt{s}\,.
$$

Therefore, by Lemma 5, we have

$$
||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger||_2^2 \le 9\lambda^2 s\,\tau^{*2}/\sigma^2,
$$

with probability at least $1 - 2ps\exp(-a_1 n\epsilon^2) - 2p\exp(-a_2 n\epsilon^2) \ge 1 - sp^{-C_3}$ where $C_3 = C_2(a_1 \vee a_2) - 1 > 0$. □

A.3.4 Proof of Theorem 3

We use the same notations as in the proof above. Without loss of generality, we assume $\boldsymbol{\mu}^{(1)} + \boldsymbol{\mu}^{(2)} = \mathbf{0}$, then $\boldsymbol{\mu}^{(1)} = \boldsymbol{\delta}/2, \boldsymbol{\mu}^{(2)} = -\boldsymbol{\delta}/2$, and $\beta_0^\dagger = 0$. According to Proposition 2, we have

$$
\begin{aligned}
& Q(\hat{\beta}_0, \hat{\boldsymbol{\beta}})/Q(\beta_0^\dagger, \boldsymbol{\beta}^\dagger) - 1 \\
&= \left\{ \Phi\left(\frac{-\hat{\beta}_0 - \hat{\boldsymbol{\beta}}^\top \boldsymbol{\mu}^{(1)}}{\sqrt{\hat{\boldsymbol{\beta}}^\top \boldsymbol{\Sigma} \hat{\boldsymbol{\beta}}}} \right) - \Phi\left(\frac{-\boldsymbol{\beta}^{\dagger T} \boldsymbol{\mu}^{(1)}}{\sqrt{\boldsymbol{\beta}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{\beta}^\dagger}} \right) + \Phi\left(\frac{\hat{\beta}_0 + \hat{\boldsymbol{\beta}}^\top \boldsymbol{\mu}^{(2)}}{\sqrt{\hat{\boldsymbol{\beta}}^\top \boldsymbol{\Sigma} \hat{\boldsymbol{\beta}}}} \right) - \Phi\left(\frac{\boldsymbol{\beta}^{\dagger T} \boldsymbol{\mu}^{(2)}}{\sqrt{\boldsymbol{\beta}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{\beta}^\dagger}} \right) \right\} \\
&\qquad \times \left\{ \Phi\left(\frac{-\boldsymbol{\beta}^{\dagger T} \boldsymbol{\mu}^{(1)}}{\sqrt{\boldsymbol{\beta}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{\beta}^\dagger}} \right) + \Phi\left(\frac{\boldsymbol{\beta}^{\dagger T} \boldsymbol{\mu}^{(2)}}{\sqrt{\boldsymbol{\beta}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{\beta}^\dagger}} \right) \right\}^{-1} \\
&\leq \max \left\{ \Phi\left(\frac{-\hat{\beta}_0 - \hat{\boldsymbol{\beta}}^\top \boldsymbol{\delta}/2}{\sqrt{\hat{\boldsymbol{\beta}}^\top \boldsymbol{\Sigma} \hat{\boldsymbol{\beta}}}} \right), \Phi\left(\frac{\hat{\beta}_0 - \hat{\boldsymbol{\beta}}^\top \boldsymbol{\delta}/2}{\sqrt{\hat{\boldsymbol{\beta}}^\top \boldsymbol{\Sigma} \hat{\boldsymbol{\beta}}}} \right) \right\} \Big/ \Phi\left(\frac{-\boldsymbol{\beta}^{\dagger T} \boldsymbol{\delta}/2}{\sqrt{\boldsymbol{\beta}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{\beta}^\dagger}} \right) - 1 \\
&\equiv \max\{R^{(1)}, R^{(2)}\}.
\end{aligned}
$$

We will use the following property of standard Gaussian distribution function (Cai and Liu, 2011):

$$
|\Phi(x_0 + r)/\Phi(x_0) - 1| \leq c_1 |r| (|x_0| + 1) \exp(c_2 |x_0 r|). \tag{A.6}
$$

For $k \in \{1, 2\}$, let

$$
r^{(k)} = \left| (\hat{\beta}_0 + \hat{\boldsymbol{\beta}}^\top \boldsymbol{\mu}^{(k)}) / \sqrt{\hat{\boldsymbol{\beta}}^\top \boldsymbol{\Sigma} \hat{\boldsymbol{\beta}}} - \boldsymbol{\beta}^{\dagger T} \boldsymbol{\mu}^{(k)} / \sqrt{\boldsymbol{\beta}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{\beta}^\dagger} \right|.
$$

Since $\boldsymbol{\beta}^{\dagger T} \boldsymbol{\delta} / \sqrt{\boldsymbol{\beta}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{\beta}^\dagger} = \Delta^{1/2}$, it suffices to verify the orders of $r^{(k)}$ and Δ.

According to Theorem 2, $||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger||_2 \leq 3\lambda\tau^* \sqrt{s}/\sigma$ with probability going to 1. Moreover, by the definition of $\boldsymbol{\beta}^\dagger$, we have $\boldsymbol{\beta}^\dagger = 4/(4 + \Delta)\boldsymbol{\beta}^*$, and $\boldsymbol{\beta}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{\beta}^\dagger = 16\Delta/(4 + \Delta)^2$. Since

$$
\begin{aligned}
|\hat{\boldsymbol{\beta}}^\top \boldsymbol{\Sigma} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{\beta}^\dagger| &\leq |(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger)^\top \boldsymbol{\Sigma} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger)| + 2|(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger)^\top \boldsymbol{\Sigma} \boldsymbol{\beta}^\dagger| \\
&\leq \lambda_{\max}(\boldsymbol{\Sigma}) ||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger||_2^2 + 2\lambda_{\max}(\boldsymbol{\Sigma}) ||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger||_2 ||\boldsymbol{\beta}^\dagger||_2 \\
&\leq 3\lambda_{\max}(\boldsymbol{\Sigma}) ||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger||_2 ||\boldsymbol{\beta}^\dagger||_2 \leq 9\lambda_{\max}(\boldsymbol{\Sigma}) \lambda\tau^* \sqrt{s}\, ||\boldsymbol{\beta}^\dagger||_2/\sigma,
\end{aligned}
$$

for sufficiently large n, we have

$$\begin{aligned}|(\hat{\boldsymbol{\beta}}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}})^{-1/2}-(\boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger)^{-1/2}| &= \left|\frac{(\hat{\boldsymbol{\beta}}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}})^{1/2}-(\boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger)^{1/2}}{(\hat{\boldsymbol{\beta}}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}})^{1/2}(\boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger)^{1/2}}\right| \\ &= \left|\frac{\hat{\boldsymbol{\beta}}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger}{(\hat{\boldsymbol{\beta}}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}})^{1/2}\cdot(\boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger)^{1/2}[(\hat{\boldsymbol{\beta}}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}})^{1/2}+(\boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger)^{1/2}]}\right| \\ &\le \left|\frac{\hat{\boldsymbol{\beta}}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger}{3/4(\boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger)^{3/2}}\right| \le \frac{(4+\Delta)^3}{4\Delta^{3/2}}\lambda_{\max}(\boldsymbol{\Sigma})\lambda\tau^*\sqrt{s}\,||\boldsymbol{\beta}^\dagger||_2/\sigma,\end{aligned}$$

in which the first inequality holds because $\hat{\boldsymbol{\beta}}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}} \ge \boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger/2$ for sufficiently large n. Moreover, under χ we have,

$$\begin{aligned}|(\hat{\beta}_0+\hat{\boldsymbol{\beta}}^\top\boldsymbol{\delta}/2)-(\boldsymbol{\beta}^{\dagger T}\boldsymbol{\delta}/2)| &\le |(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^\dagger)^\top\boldsymbol{\delta}|/2+|-\bar{\mathbf{x}}^\top\hat{\boldsymbol{\beta}}| \\ &\le ||\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^\dagger||_2||\boldsymbol{\delta}||_2/2+|\bar{\mathbf{x}}^\top\boldsymbol{\beta}^\dagger|+|\bar{\mathbf{x}}^\top(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^\dagger)| \\ &\le 3\lambda\tau^*\sqrt{s}\,||\boldsymbol{\delta}||_2/(2\sigma)+||\boldsymbol{\beta}^\dagger||_1\epsilon.\end{aligned}$$

Therefore,

$$\begin{aligned}r^{(1)} &\le |(\hat{\beta}_0+\hat{\boldsymbol{\beta}}^\top\boldsymbol{\delta}/2)\{(\hat{\boldsymbol{\beta}}^\top\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}})^{-1/2}-(\boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger)^{-1/2}\}| \\ &\quad+|(\hat{\beta}_0+\hat{\boldsymbol{\beta}}^\top\boldsymbol{\delta}/2)-\boldsymbol{\beta}^{\dagger T}\boldsymbol{\delta}/2|/(\boldsymbol{\beta}^{\dagger T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\dagger)^{1/2} \\ &\le \frac{(4+\Delta)^2}{2\Delta^{1/2}}\lambda\tau^*\lambda_{\max}(\boldsymbol{\Sigma})\sqrt{s}\,||\boldsymbol{\beta}^\dagger||_2/\sigma+\frac{4+\Delta}{4\Delta^{1/2}}\{2\lambda\tau^*\sqrt{s}\,||\boldsymbol{\delta}||_2/\sigma+||\boldsymbol{\beta}^\dagger||_1\epsilon\}.\end{aligned}$$

Using the property (A.6), then we have

$$\begin{aligned}R^{(1)} = \left|\frac{\Phi(-\Delta^{1/2}/2+r^{(1)})}{\Phi(-\Delta^{1/2}/2)}-1\right| &\le c_1|r^{(1)}|(\Delta^{1/2}/2+1)\exp(c_2r^{(1)}\Delta^{1/2}/2) \\ &= O\{r^{(1)}\Delta^{1/2}\exp(c_2r^{(1)}\Delta^{1/2}/2)\}.\end{aligned}$$

Since $\Delta^2\lambda\tau^*\lambda_{\max}(\boldsymbol{\Sigma})\sqrt{s}\,||\boldsymbol{\beta}^\dagger||_2/\sigma \to 0$, $\Delta\lambda\tau^*\sqrt{s}\,||\boldsymbol{\delta}||_2/\sigma \to 0$, and $\Delta n^{\gamma-1}||\boldsymbol{\beta}^\dagger||_1 \to 0$, we have $r^{(1)}\Delta^{1/2}\to 0$ and thus $R^{(1)} \xrightarrow{p} 0$. Similarly, we can show $R^{(2)} \xrightarrow{p} 0$, which proves the theorem.

APPENDIX B

SUPPLEMENTARY MATERIALS FOR THE MPENPC METHOD

B.1 Soft Label Demonstration

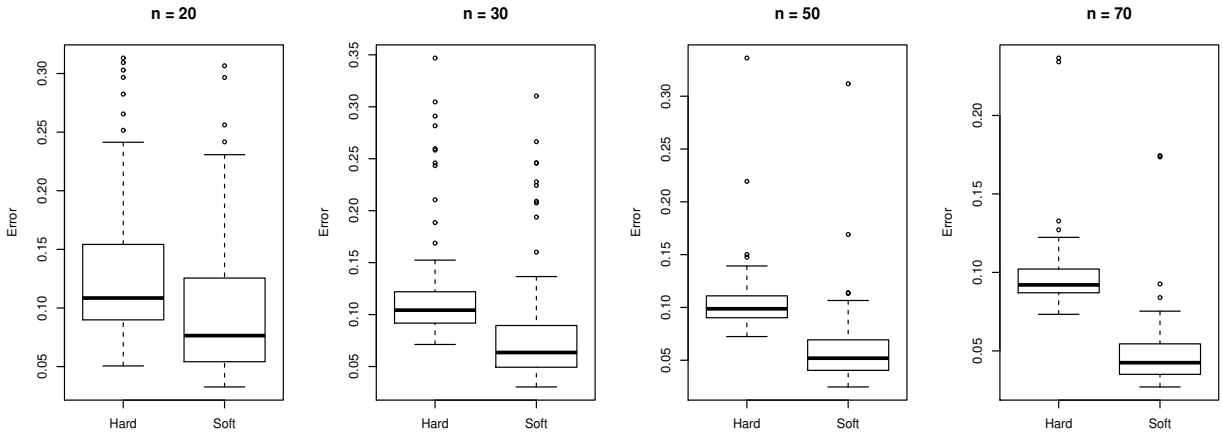

Figure B.1: Accuracy of the estimated hard and soft labels for the toy example with varying sample sizes. The y-axis denotes the Manhattan distance between the estimated labels and true labels. Each boxplot is produced based on 100 repetitions.

B.2 Simulation Results

B.2.1 Results of the ER Model Scenario

B.2.2 Additional Settings

In addition to the simulations in the main text, we also perform simulation studies for i) a scenario with non-overlapping skeletons and ii) a scenario with common group means. The goal of these simulation settings is to study the robustness of the MPenPC. We present the estimation results of all methods as follows.

Non-Overlapping DAGs

Figure B.4 shows the graph estimation accuracy of all methods in Stage I with varying λ, and Figure B.5 displays the skeleton estimation accuracy with different significance levels. Table B.2

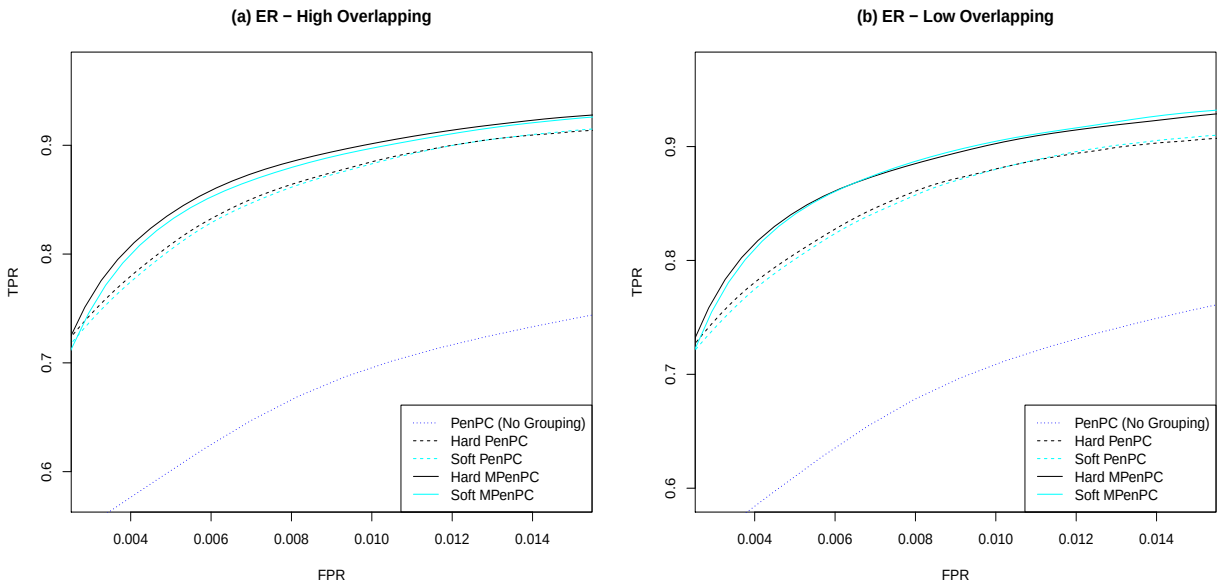

Figure B.2: Neighborhood selection performance of different methods at Stage I in the ER-model example: $K = 4$, $p = 500$, $\pi_E = 1/500$, $\delta^2 = 0.05$, $\pi_0 = 0.7$ for (a) and $\pi_0 = 0.3$ for (b). The x-axes and y-axes represent FPR and TPR respectively. The graph sparsity at the neighborhood selection stage varies for different choices of the tuning parameter λ. The tuning parameter γ for MPenPC methods is preselected by EBIC.

summarizes the performance of all methods with EBIC-selected tuning parameters by stage. We can see from the results that, even when the DAGs are not overlapping, our methods still have reasonable performance. The Soft MPenPC is only slightly worse than the Soft PenPC in this case.

B.2.3 Common Group Mean

In this scenario, we use the BA model with $K = 4$, $p = 500$, $e = 1$, $\pi_0 = 0.7$, and $\delta^2 = 0$. The soft labels are estimated with QDA based on the first 20 principal components. The simulation results are presented in Figures B.6 and B.7. We can see from the results that the methods based on soft labels (Soft MPenPC, Soft PenPC) still outperform those based on hard labels (Hard MPenPC, Hard PenPC). Based on the simulation results, we can see that as long as the classification method is chosen appropriately, it is typically beneficial to use the soft labels.

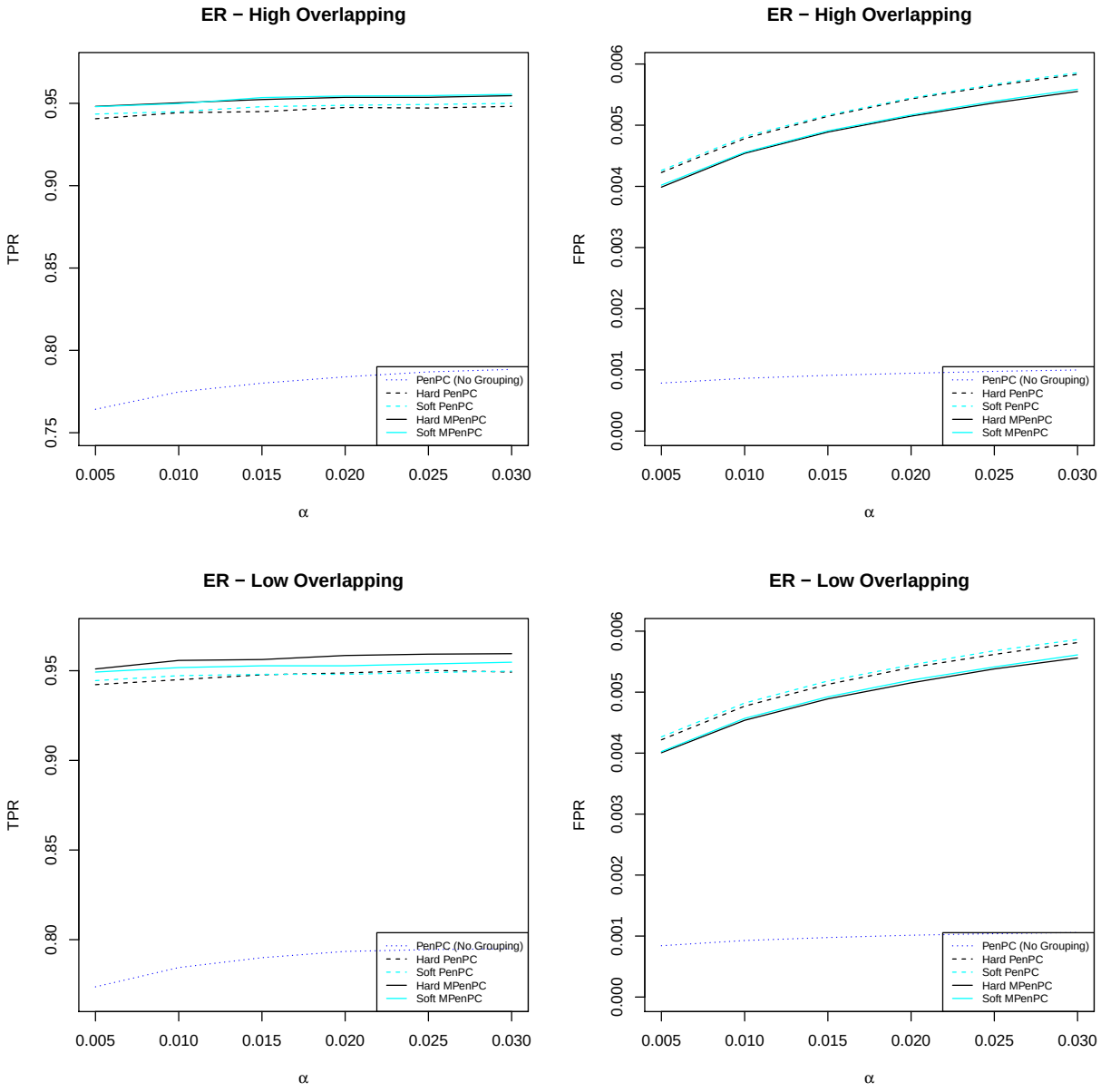

Figure B.3: Skeleton estimation performance of different methods at Stage II in the ER-model example. The high overlapping scenarios have $\pi_0 = 0.7$, and the low overlapping scenarios have $\pi_0 = 0.3$. The x-axes represent the significance level α of the PC-stable algorithm. The y-axes represent TPR (the left panel) and FPR (the right panel) respectively.

B.3 Assumptions and Proofs of the Theoretical Results

B.3.1 Regularity Conditions

Given a random sample X from the heterogeneous population, denote $\pi^{(k)} = \Pr(G = k|X)$, $\xi_{j,l}^{(k)} = w^{(k)} X_j X_l$ and $\zeta_{j,l}^{(k)} = w^{(k)} X_j \eta_l^{(k)}$, where $\eta_l^{(k)} = X_l + X_{-l}^T \mathbf{\Omega}_{-l,l}^{(k)} / \Omega_{l,l}^{(k)}$. Denote $\rho_{j,l|S}^{(k)}$ the partial

Table B.1: Performance of different methods at both stages for the ER-model example ($K = 4$, $p = 500$, $\pi_E = 1/500$, $e = 1$, $\delta^2 = 0.05$, $\pi_0 = 0.7$ for high overlapping and $\pi_0 = 0.3$ for low overlapping). The numbers outside and inside parentheses are averages and standard errors respectively based on 100 repetitions.

Stage I. Neighborhood Selection		
High Overlapping	TPR	FPR
PenPC (No Grouping)	0.6232(0.0021)	0.0019(0.0000)
Hard PenPC	0.8076(0.0024)	0.0352(0.0002)
Soft PenPC	0.8176(0.0021)	0.0384(0.0001)
Hard MPenPC	0.8368(0.0020)	0.0642(0.0002)
Soft MPenPC	0.8450(0.0019)	0.0666(0.0002)
Low Overlapping	TPR	FPR
PenPC (No Grouping)	0.4234(0.0018)	0.0024(0.0000)
Hard PenPC	0.8007(0.0030)	0.0355(0.0000)
Soft PenPC	0.8129(0.0022)	0.0380(0.0000)
Hard MPenPC	0.7921(0.0025)	0.0641(0.0000)
Soft MPenPC	0.8033(0.0019)	0.0656(0.0000)
Stage II. Skeleton Estimation		
High Overlapping	TPR	FPR
PenPC (No Grouping)	0.464(0.0027)	0.0058(0)
Hard PenPC	0.6595(0.0046)	0.0075(0)
Soft PenPC	0.6994(0.0045)	0.007(0)
Hard MPenPC	0.6668(0.0045)	0.007(0)
Soft MPenPC	0.7034(0.0045)	0.0065(0)
Low Overlapping	TPR	FPR
PenPC (No Grouping)	0.3076(0.0026)	0.0077(0)
Hard PenPC	0.6365(0.0034)	0.0079(0)
Soft PenPC	0.6952(0.0023)	0.007(0)
Hard MPenPC	0.6329(0.0035)	0.0074(0)
Soft MPenPC	0.6923(0.0023)	0.0065(0)
Stage II. CPDAG Estimation		
High Overlapping	SHD	
PenPC (No Grouping)	1141.27(4.97)	
Hard PenPC	1297.33(5.21)	
Soft PenPC	1232.73(4.62)	
Hard MPenPC	1236.6(7.22)	
Soft MPenPC	1164.74(4.19)	
Low Overlapping	SHD	
PenPC (No Grouping)	1164.74(4.19)	
Hard PenPC	1344.86(5.27)	
Soft PenPC	1226.09(5.03)	
Hard MPenPC	1291.52(5.19)	
Soft MPenPC	1173.37(4.79)	

correlation between the X_j and X_l given $\{X_i : i \in S\}$ in class k. For each class k, let $A_{jl}^{(k)}$ denote

Table B.2: Performance of different methods in the non-overlapping example (BA model, $K = 4$, $p = 500$, $e = 1$, $\delta^2 = 0.05$, $\pi_0 = 0$). The numbers outside and inside parentheses are averages and standard errors respectively based on 100 repetitions.

Stage I. Neighborhood Selection		
Non-Overlapping	TPR	FPR
PenPC (No Grouping)	0.4002(0.0018)	0.0027(0.0000)
Hard PenPC	0.7946(0.0023)	0.0352(0.0001)
Soft PenPC	0.8247(0.0018)	0.0378(0.0001)
Hard MPenPC	0.771(0.0021)	0.0641(0.0002)
Soft MPenPC	0.801(0.0015)	0.0653(0.0003)
Stage II. Skeleton Estimation		
Non-Overlapping	TPR	FPR
PenPC (No Grouping)	0.4072(0.0018)	0.0026(0.0000)
Hard PenPC	0.8354(0.0026)	0.0054(0.0000)
Soft PenPC	0.8588(0.0019)	0.0053(0.0000)
Hard MPenPC	0.8288(0.0026)	0.0052(0.0000)
Soft MPenPC	0.8523(0.002)	0.0051(0.0000)

Table B.3: Performance of different methods in the common group mean example (BA model, $K = 4$, $p = 500$, $e = 1$, $\pi_0 = 0.7$, $\delta^2 = 0$). The numbers outside and inside parentheses are averages and standard errors respectively based on 100 repetitions.

Stage I. Neighborhood Selection		
Non-Overlapping	TPR	FPR
PenPC (No Grouping)	0.6404(0.0039)	0.0019(1e-04)
Hard PenPC	0.8167(0.0035)	0.0352(1e-04)
Soft PenPC	0.8288(0.0036)	0.0387(3e-04)
Hard MPenPC	0.8421(0.0037)	0.0641(3e-04)
Soft MPenPC	0.8552(0.0028)	0.0671(5e-04)
Stage II. Skeleton Estimation		
Non-Overlapping	TPR	FPR
PenPC (No Grouping)	0.6317(0.0073)	0.0017(0)
Hard PenPC	0.8442(0.0047)	0.0055(0)
Soft PenPC	0.8562(0.0048)	0.0054(0)
Hard MPenPC	0.8486(0.0054)	0.0052(0)
Soft MPenPC	0.8606(0.0056)	0.0052(0)

the Markov blanket of j and l, after excluding their common children and descendants, and $C_{j,l}^{(k)}$ denote the set of nodes that can be common children or descendants.

Moreover, for a matrix $\mathbf{A}$, let $\lambda_{min}(\mathbf{A})$ denote its minimum eigenvalue. Denote $\|\mathbf{A}\|_2$ the spectral norm, $\|\mathbf{A}\|_1$ the max column absolute summation, $\|\mathbf{A}\|_\infty$ the max row absolute summation, $|\mathbf{A}|_\infty$ the max absolute value of its elements.

The following assumptions will be used in the theoretical analysis of our MPenPC method:

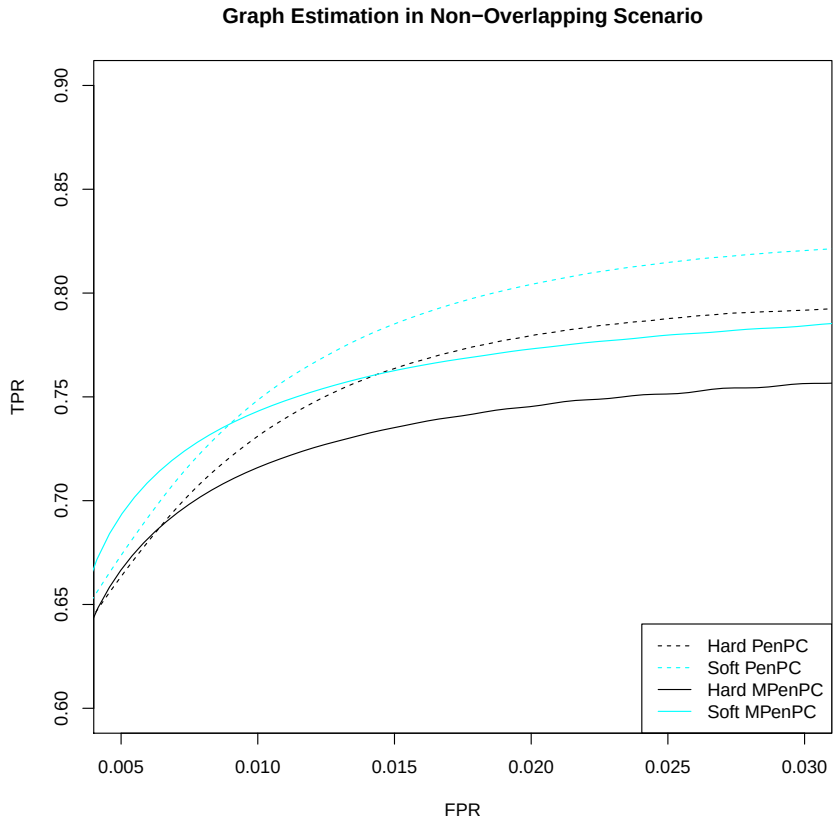

Figure B.4: Neighborhood selection performance of different methods at Stage I in the non-overlapping example (BA model, $K = 4$, $p = 500$, $e = 1$, $\delta^2 = 0.05$, $\pi_0 = 0$). The x-axis and y-axis represent FPR and TPR respectively. The graph sparsity at the neighborhood selection stage varies with the tuning parameter λ. The other tuning parameter γ for MPenPC methods is preselected by EBIC.

(C1) $\forall k, n^{(k)} = O(n)$;

(C2) $p = O(\exp(n^a))$, $q := \max_j |A_j| = O(n^b)$, where $a \in [0, 1)$ and $b \in [0, (1-a)/2)$;

(C3) $\delta \geq O(n^{-d_1})$, where $\delta = \frac{1}{2} \min_{k,j,l:\Omega^{(k)}_{j,l} \neq 0} \left| \Omega^{(k)}_{j,l} / \Omega^{(k)}_{j,j} \right|$ and $d_1 \in (0, (1-a-b)/2)$;

(C4) For any A with $|A| \leq q$, $\lambda_{min}(\mathbf{\Sigma}^{(k)}_{A,A}) \geq C_1 > 0$, $\forall k \in \{1, \cdots, K\}$. Moreover, $\max_{j,k} \Sigma^{(k)}_{j,j} < C_2$ for some $C_2 > 0$;

(C5) $\lambda = o(\tau\delta)$, $\lambda \gg n^{-(1-a-b)}$, $\lambda \exp(-\tau\delta/\lambda) = o(q^{-1/2}\delta)$, and $\tau \exp(-\tau\delta/(2\lambda)) < C_1/2$;

(C6) For any $j \in \{1, \cdots, p\}$, if $\boldsymbol{\gamma}_{j,l}$ is partially zero for some l, then

$$\left\| \left(\mathbf{\Sigma}^{(k)}_{A^{(k)}_j, A^{(k)}_j} \right)^{-1} \mathbf{\Sigma}^{(k)}_{A^{(k)}_j, l} \right\|_1 \leq \exp\left\{ -\lambda^{-1}\tau \left[\|\boldsymbol{\gamma}_{j,l}\|_1 + (\|\boldsymbol{\gamma}_{j,l}\|_0 - 1)\delta \right] \right\},$$

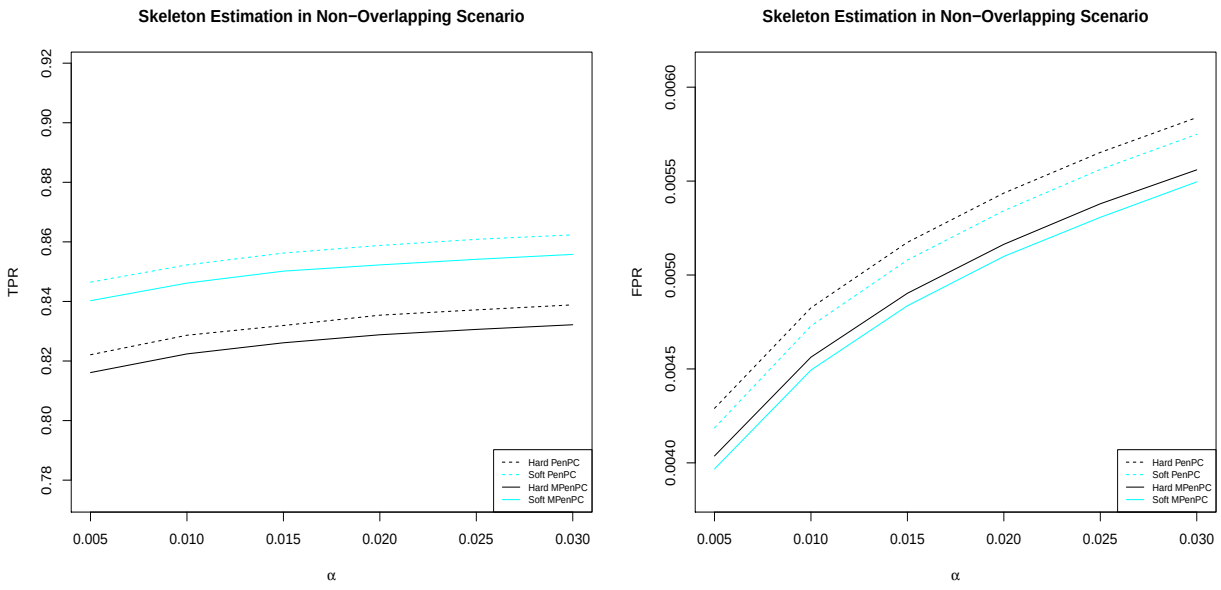

Figure B.5: Skeleton estimation performance of different methods at Stage II in the non-overlapping example (BA model, $K = 4$, $p = 500$, $e = 1$, $\delta^2 = 0.05$, $\pi_0 = 0$). The x-axes represent the significance level α of the PC-stable algorithm. The y-axes represent TPR (the left panel) and FPR (the right panel) respectively.

for all $k \in \{1, \cdots, K\}$ s.t. $\gamma_{j,l}^{(k)} = 0$.

(C7) For each k, $w_i^{(k)}$'s are independent for $i = 1, \cdots, n$ and $\mathbb{E}w_i^{(k)} = \pi_i^{(k)} := \Pr(G = k|X_i)$ with bounded $\mathrm{Var}(w_i^{(i)})$. In addition, for all $i \in \{1, \cdots, n\}$ and $j, l \in \{1, \cdots, p\}$, $\mathbb{E}(\xi_{i,j,l}^{(k)}) = \pi_i^{(k)}\Sigma_{j,l}^{(k)}$, $\mathbb{E}(\zeta_{i,j,l}^{(k)}) = 0$, $\mathbb{E}[\xi_{i,j,l}^{(k)2}\exp(t|\xi_{i,j,l}^{(k)}|)] \le C_5 < \infty$, and $\mathbb{E}[\zeta_{i,j,l}^{(k)2}\exp(t|\zeta_{i,j,l}^{(k)}|)] \le C_6 < \infty$.

(C8) $\forall j, l, k, S \in \Pi_{jl}^{(k)}$ such that $\rho_{j,l|S}^{(k)} \neq 0$, we have

$$c \le |\rho_{j,l|S}^{(k)}| \le M < 1,$$

where $c = O(n^{-d_2})$ for some $d_2 \in (0, (1 - a \vee b)/2)$. Here $\Pi_{j,l}^{(k)} = \{A_{jl}^{(k)} \backslash D_{jl}^{(k)} : D_{jl}^{(k)} \subseteq C_{jl}^{(k)}\}$.

Condition (C1) requires each class to have a non-vanishing weight, and one of its direct implications is $K = O(1)$. (C2)-(C4) are common conditions for selection consistency of penalized estimation. In particular, (C2) specifies the sparsity level of the conditional independence graph $\mathcal{G}^{(k)}$. (C3) gives a lower bound on the minimal size of nonzero regression coefficients. (C4) ensures the correlations among the neighbor nodes are not too strong. (C5) specifies the orders of tuning parameters, λ and τ. (C6) is an *irrepresentability condition* for the GEL penalty and it guarantees the bi-level

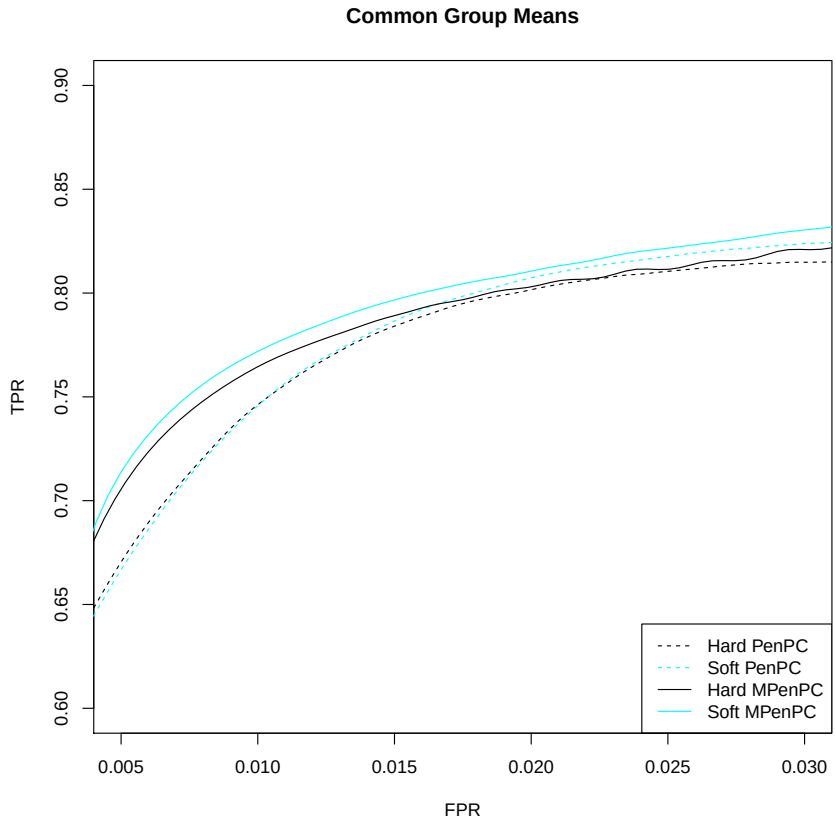

Figure B.6: Neighborhood selection performance of different methods at Stage I in the common group mean example (BA model, $K = 4$, $p = 500$, $e = 1$, $\pi_0 = 0.7$, $\delta^2 = 0$). The x-axis and y-axis represent FPR and TPR respectively. The graph sparsity at the neighborhood selection stage varies with the tuning parameter λ. The other tuning parameter γ for MPenPC methods is preselected by EBIC.

selection consistency. Through (C7) we make certain assumptions on the accuracy of the soft labels. Condition (C8) is commonly used in proving the consistency of the PC algorithm and other skeleton estimation methods (Nandy et al., 2015).

B.3.2 Theorem 4

Before we prove the theorem, we introduce the following lemma.

Lemma 2. Define the events

$$\chi = \{\mathbf{X} : |\hat{\mathbf{\Sigma}}^{(k)} - \mathbf{\Sigma}^{(k)}|_\infty \leq C_3\sqrt{\log p/n}, k = 1, \cdots, K\},$$

$$\mathcal{E} = \{\max_{j,l} |\mathbf{X}_l^{(k)T}\boldsymbol{\epsilon}_j|/n^{(k)} \leq C_4\sqrt{n^{-1}\log p}, k = 1, \cdots, K\}.$$

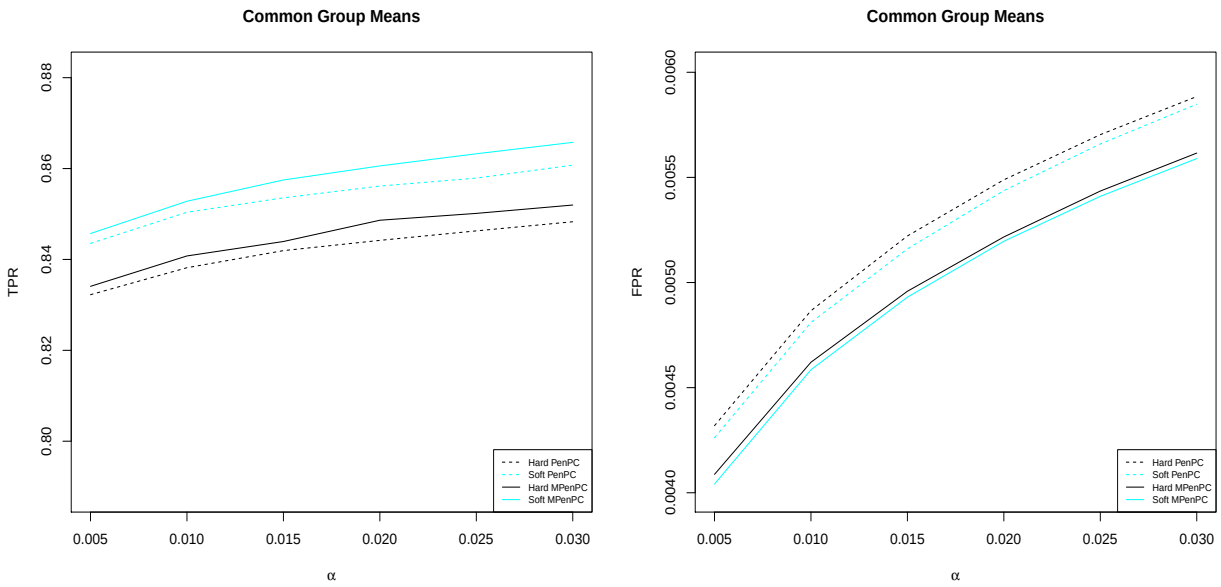

Figure B.7: Skeleton estimation performance of different methods at Stage II in the common group mean example (BA model, $K = 4$, $p = 500$, $e = 1$, $\pi_0 = 0.7$, $\delta^2 = 0$). The x-axes represent the significance level α of the PC-stable algorithm. The y-axes represent TPR (the left panel) and FPR (the right panel) respectively.

Then

(a) Under (C1), (C2), (C4), and (C5), $P(\chi) = 1 - O(p^{-1})$ and $P(\mathcal{E}) = 1 - O(p^{-1})$;

(b) Under χ, we have

$$||(\mathbf{X}_{A_j}^{(k)^T}\mathbf{X}_{A_j}^{(k)})^{-1}||_2 \leq 2/(nC_1),$$
$$||(\mathbf{X}_{A_j}^{(k)^T}\mathbf{X}_{A_j}^{(k)})^{-1}||_\infty = O(q^{1/2}/n^{(k)}),$$
$$||\mathbf{X}_{A_j^c}^{(k)^T}\mathbf{X}_{A_j}^{(k)}(\mathbf{X}_{A_j}^{(k)T}\mathbf{X}_{A_j}^{(k)})^{-1}||_\infty = O(q^{1/2}).$$

This is the Lemma 7 in Ha et al. (2016) and thus we skip the proof. We will use this lemma in proving the selection consistency.

Proof. We prove the two parts of Theorem 1 respectively. Without loss of generality, we assume the dataset has been centered by group.

(i) According to the KKT conditions, $\boldsymbol{\gamma}_j$ with $\boldsymbol{\gamma}_{j,\overline{A}_j}^{(k)} = \mathbf{0}, \forall k$ is a strict local minimizer of (4) if

$$\mathbf{X}_{A_j}^{(k)T}(\mathbf{X}_j^{(k)} - \mathbf{X}_{-j}^{(k)T}\boldsymbol{\gamma}_j^{(k)}) = nP'_{\lambda,\tau}(\boldsymbol{\gamma}_{j,A_j}^{(k)}), \tag{B.1}$$

$$||\mathbf{X}_{\overline{A}_j}^{(k)T}(\mathbf{X}_j^{(k)} - \mathbf{X}_{-j}^{(k)T}\boldsymbol{\gamma}_j^{(k)})||_\infty \leq n\lambda, \tag{B.2}$$

$$||(\mathbf{X}_{A_j}^{(k)T}\mathbf{X}_{A_j}^{(k)})^{-1}||_2 < 1/(n\kappa_j^{(k)}), \tag{B.3}$$

for all k, where $\kappa_j^{(k)} = \max_{l\in A_j} -\partial^2 P_{\lambda,\tau}(\boldsymbol{\gamma}_j)/\partial\gamma_{jl}^{(k)2}$.

Define $\mathcal{N}_j = \left\{\boldsymbol{\gamma}_j \in \mathbb{R}^{pK} : \left\|\boldsymbol{\gamma}_{j,A_j}^{(k)} - \boldsymbol{\gamma}_{j,A_j}^{*(k)}\right\|_\infty \leq Cn^{-d_1} \leq \delta/2, \boldsymbol{\gamma}_{j,\overline{A}_j}^{(k)} = \mathbf{0}, \forall k\right\}$. We will show that under $\chi \cap \mathcal{E}$ there exists a $\hat{\boldsymbol{\gamma}}_j \in \mathcal{N}_j$ that satisfies the KKT conditions above.

Define a function $\phi^{(k)}(\boldsymbol{\gamma}_{j,A_j}^{(k)}) = \boldsymbol{\gamma}_{j,A_j}^{(k)} - \boldsymbol{\gamma}_{j,A_j}^{*(k)} - \mathbf{u}_j^{(k)}$, where $\mathbf{u}_j^{(k)} = (\mathbf{X}_{A_j}^{(k)T}\mathbf{X}_{A_j}^{(k)})^{-1}\{\mathbf{X}_{A_j}^{(k)T}\boldsymbol{\epsilon}_j^{(k)} - nP'(\boldsymbol{\gamma}_{j,A_j}^{(k)})\}$. For any $\boldsymbol{\gamma}_j \in \mathcal{N}_j$,

$$\begin{aligned}
\left\|\mathbf{u}^{(k)}\right\|_\infty \leq& \left\|(\mathbf{X}_{A_j}^{(k)T}\mathbf{X}_{A_j}^{(k)})^{-1}\right\|_\infty \left\|\mathbf{X}_{A_j}^{(k)T}\boldsymbol{\epsilon}_j^{(k)} - nP'(\boldsymbol{\gamma}_{j,A_j}^{(k)})\right\|_\infty \\
\leq& \left\|(\mathbf{X}_{A_j}^{(k)T}\mathbf{X}_{A_j}^{(k)})^{-1}\right\|_\infty \left\{\left\|\mathbf{X}_{A_j}^{(k)T}\boldsymbol{\epsilon}_j^{(k)}\right\|_\infty + n\left\|P'(\boldsymbol{\gamma}_{j,A_j}^{(k)})\right\|_\infty\right\} \\
=& O(\sqrt{q}/n^{(k)})\{C_4 n^{(k)}\sqrt{\log p/n} + n\lambda\exp(-\lambda^{-1}\tau\delta)\} \\
\leq& O(\sqrt{q\log p/n}) + O(\sqrt{q}\lambda\exp(-\lambda^{-1}\tau\delta)).
\end{aligned}$$

Thus, by (C1)-(C2) and (C5), $\sqrt{q\log p/n} = o(n^{-d_1})$ and $\sqrt{q}\lambda\exp(-\lambda^{-1}\tau\delta) = o(n^{-d_1})$. Thus $\left\|\mathbf{u}^{(k)}\right\|_\infty = o(n^{-d_1})$ for $\boldsymbol{\gamma}_j \in \mathcal{N}_j$. By the continuity of $\phi^{(k)}$ and Miranda's Existence Theorem, there exists a $\hat{\boldsymbol{\gamma}}_j \in \mathcal{N}_j$ such that $\phi^{(k)}(\hat{\boldsymbol{\gamma}}_{j,A_j}^{(k)}) = 0, \forall k$. Therefore, the condition (B.1) is satisfied.

Given the $\hat{\boldsymbol{\gamma}}_j$ defined as above, for any k and j,

$$\begin{aligned}
&\left\|\mathbf{X}_{\overline{A}_j}^{(k)T}\left(\mathbf{X}_j^{(k)} - \mathbf{X}_{A_j}^{(k)}\hat{\boldsymbol{\gamma}}_{j,A_j}^{(k)}\right)\right\|_\infty \\
=& \left\|\mathbf{X}_{\overline{A}_j}^{(k)T}\boldsymbol{\epsilon}_j^{(k)} - \mathbf{X}_{\overline{A}_j}^{(k)T}\mathbf{X}_{A_j}^{(k)}\left(\hat{\boldsymbol{\gamma}}_{j,A_j}^{(k)} - \boldsymbol{\gamma}_{j,A_j}^{*(k)}\right)\right\|_\infty \\
=& \left\|\mathbf{X}_{\overline{A}_j}^{(k)T}\boldsymbol{\epsilon}_j^{(k)} - \mathbf{X}_{\overline{A}_j}^{(k)T}\mathbf{X}_{A_j}^{(k)}\left(\mathbf{X}_{A_j}^{(k)T}\mathbf{X}_{A_j}^{(k)}\right)^{-1}\left(\mathbf{X}_{A_j}^{(k)}\boldsymbol{\epsilon}_j^{(k)} - nP'(\hat{\boldsymbol{\gamma}}_{j,A_j}^{(k)})\right)\right\|_\infty \\
\leq& \left\|\mathbf{X}_{\overline{A}_j}^{(k)T}\boldsymbol{\epsilon}_j^{(k)}\right\|_\infty + \left\|\mathbf{X}_{\overline{A}_j}^{(k)T}\mathbf{X}_{A_j}^{(k)}\left(\mathbf{X}_{A_j}^{(k)T}\mathbf{X}_{A_j}^{(k)}\right)^{-1}\right\|_\infty \times \left\{\left\|\mathbf{X}_{A_j}^{(k)}\boldsymbol{\epsilon}_j^{(k)}\right\|_\infty + n\left\|P'(\hat{\boldsymbol{\gamma}}_{j,A_j}^{(k)})\right\|_\infty\right\} \\
\leq& O(C_4 n^{(k)}\sqrt{\log p/n}) + O(\sqrt{q})\{C_4 n^{(k)}\sqrt{\log p/n} + n\lambda\exp(-\lambda^{-1}\tau\delta)\}.
\end{aligned}$$

By (C1)-(C2) and (C5), $\left\|\mathbf{X}_{\overline{A}_j}^{(k)T}\left(\mathbf{X}_j^{(k)}-\mathbf{X}_{A_j}^{(k)}\hat{\boldsymbol{\gamma}}_{j,A_j}^{(k)}\right)\right\|_\infty = o(\lambda)$, for all k and j. That is, $\hat{\boldsymbol{\gamma}}_j$ satisfies (B.2).

Because $\hat{\boldsymbol{\gamma}}_j \in \mathcal{N}_j$,

$$\begin{aligned}\kappa_j^{(k)} =& \max_{l\in A_j}\left\{-\left.\frac{\partial^2 P_{\lambda,\tau}(\boldsymbol{\gamma}_j)}{\partial\gamma_{jl}^{(k)2}}\right|_{\boldsymbol{\gamma}_j=\hat{\boldsymbol{\gamma}}_j}\right\}\\ =&\tau\exp\left\{-\lambda^{-1}\tau\min_{l\in A_j}\|\hat{\boldsymbol{\gamma}}_{j,l}\|_1\right\}\\ \leq&\tau\exp\left\{-\lambda^{-1}\tau\delta/2\right\} < C_1/2.\end{aligned}$$

By combining (C5) and Lemma 2, $\hat{\boldsymbol{\gamma}}_j$ satisfies (B.3), which completes the proof.

(ii) The proof is quite similar to that of part (i), except that the KKT conditions become

$$\mathbf{X}_{A_j^{(k)}}^{(k)T}(\mathbf{X}_j^{(k)}-\mathbf{X}_{-j}^{(k)T}\boldsymbol{\gamma}_j^{(k)}) = nP'_{\lambda,\tau}(\boldsymbol{\gamma}_{j,A_j^{(k)}}^{(k)}), \tag{B.4}$$

$$\left\|\mathbf{X}_{\overline{A}_j^{(k)}}^{(k)T}(\mathbf{X}_j^{(k)}-\mathbf{X}_{-j}^{(k)T}\boldsymbol{\gamma}_j^{(k)})\right\|_\infty \leq n\lambda, \tag{B.5}$$

$$\left\|\left(\mathbf{X}_{A_j^{(k)}}^{(k)T}\mathbf{X}_{A_j^{(k)}}^{(k)}\right)^{-1}\right\|_2 < 1/(n\kappa_j^{(k)}), \tag{B.6}$$

for all k, where $\kappa_j^{(k)} = \max_{l\in A_j^{(k)}} -\partial^2 P_{\lambda,\tau}(\boldsymbol{\gamma}_j)/\partial\gamma_{jl}^{(k)2}$.

Define $\mathcal{N}_j^* = \left\{\boldsymbol{\gamma}_j\in\mathbb{R}^{pK} : \left\|\boldsymbol{\gamma}_{j,A_j^{(k)}}^{(k)} - \boldsymbol{\gamma}_{j,A_j^{(k)}}^{*(k)}\right\|_\infty \leq Cn^{-d_1} \leq \delta/2, \boldsymbol{\gamma}_{j,\overline{A}_j^{(k)}}^{(k)} = \mathbf{0}, \forall k\right\}$. Then it suffices to show that there exists a $\hat{\boldsymbol{\gamma}}_j \in \mathcal{N}_j^*$ that satisfies the KKT conditions above. We will show that these conditions are always satisfied under $\chi\cap\mathcal{E}$.

Define a function $\phi^{(k)}(\boldsymbol{\gamma}_{j,A_j^{(k)}}^{(k)}) = \boldsymbol{\gamma}_{j,A_j^{(k)}}^{(k)} - \boldsymbol{\gamma}_{j,A_j^{(k)}}^{*(k)} - \mathbf{u}_j^{(k)}$, where $\mathbf{u}_j^{(k)} = \left(\mathbf{X}_{A_j^{(k)}}^{(k)T}\mathbf{X}_{A_j^{(k)}}^{(k)}\right)^{-1}$ $\times\left\{\mathbf{X}_{A_j^{(k)}}^{(k)T}\boldsymbol{\epsilon}_j^{(k)} - nP'(\boldsymbol{\gamma}_{j,A_j^{(k)}}^{(k)})\right\}$. Similar to part (i), we can show that $\left\|\mathbf{u}^{(k)}\right\|_\infty = o(n^{-d_1})$. By the continuity of $\phi^{(k)}$ and Miranda's Existence Theorem, $\phi^{(k)}(\boldsymbol{\gamma}_{j,A_j^{(k)}}^{(k)}) = 0, \forall k$, has a root within $\mathcal{N}_j^*$, which is denoted as $\hat{\boldsymbol{\gamma}}_j$ and satisfies (B.4).

Given the $\hat{\boldsymbol{\gamma}}_j$ defined as above, under $\chi\cap\mathcal{E}$, we have that for any $l\in\overline{A}_j^{(k)}$, $\mathbf{X}_l^{(k)T}\left(\mathbf{X}_j^{(k)}-\mathbf{X}_{A_j^{(k)}}^{(k)}\hat{\boldsymbol{\gamma}}_{j,A_j^{(k)}}^{(k)}\right) = \boldsymbol{\eta}_l^{(k)T}\Pi_j^{(k)}(\boldsymbol{\epsilon}_j^{(k)}) + n\boldsymbol{\Sigma}_{A_j^{(k)},l}^{(k)T}\left(\boldsymbol{\Sigma}_{A_j^{(k)},A_j^{(k)}}^{(k)}\right)^{-1}P'(\hat{\boldsymbol{\gamma}}_{j,A_j^{(k)}}^{(k)}) + n\boldsymbol{\eta}_l^{(k)T}\mathbf{X}_{A_j^{(k)}}^{(k)}\left(\mathbf{X}_{A_j^{(k)}}^{(k)T}\mathbf{X}_{A_j^{(k)}}^{(k)}\right)^{-1}P'(\hat{\boldsymbol{\gamma}}_{j,A_j^{(k)}}^{(k)})$, where $\boldsymbol{\eta}_l^{(k)}$ is the regression error of $\mathbf{X}_l^{(k)}$ on $\mathbf{X}_{A_j^{(k)}}^{(k)}$

(thus $\boldsymbol{\eta}_l^{(k)}$ and $\mathbf{X}^{(k)}_{A_j^{(k)}}$ are independent) and $\Pi_j^{(k)}(\mathbf{v}) = \left\{\mathbf{I} - \mathbf{X}^{(k)}_{A_j^{(k)}}\left(\mathbf{X}^{(k)T}_{A_j^{(k)}}\mathbf{X}^{(k)}_{A_j^{(k)}}\right)^{-1}\mathbf{X}^{(k)T}_{A_j^{(k)}}\right\}\mathbf{v}$ computes the difference between $\mathbf{v}$ and its projection on $\mathbf{X}^{(k)}_{A_j^{(k)}}$. By (C6), we have

$$\begin{aligned}
&\left|n\boldsymbol{\Sigma}^{(k)T}_{A_j^{(k)},l}\left(\boldsymbol{\Sigma}^{(k)}_{A_j^{(k)},A_j^{(k)}}\right)^{-1}P'(\hat{\boldsymbol{\gamma}}^{(k)}_{j,A_j^{(k)}})\right| \\
\leq & n\left\|\left(\boldsymbol{\Sigma}^{(k)}_{A_j^{(k)},A_j^{(k)}}\right)^{-1}\boldsymbol{\Sigma}^{(k)}_{A_j^{(k)},l}\right\|_1\left\|P'(\hat{\boldsymbol{\gamma}}^{(k)}_{j,A_j^{(k)}})\right\|_\infty \\
\leq & n\lambda\exp(-\lambda^{-1}\tau\delta) \\
= & o(\lambda).
\end{aligned}$$

By (C1)-(C2) and (C6), we have

$$\left|\boldsymbol{\eta}_l^{(k)T}\Pi_j^{(k)}(\boldsymbol{\epsilon}_j^{(k)}) + n\boldsymbol{\eta}_l^{(k)T}\mathbf{X}^{(k)}_{A_j^{(k)}}\left(\mathbf{X}^{(k)T}_{A_j^{(k)}}\mathbf{X}^{(k)}_{A_j^{(k)}}\right)^{-1}P'(\hat{\boldsymbol{\gamma}}^{(k)}_{j,A_j^{(k)}})\right| = o(\lambda).$$

Therefore, $\left\|\mathbf{X}^{(k)T}_{\bar{A}_j^{(k)}}\left(\mathbf{X}_j^{(k)} - \mathbf{X}^{(k)}_{A_j^{(k)}}\hat{\boldsymbol{\gamma}}^{(k)}_{j,A_j^{(k)}}\right)\right\|_\infty = o(\lambda)$, for all k and j. Thus, (B.5) is satisfied. Lastly, (B.6) is always satisfied for any $\hat{\boldsymbol{\gamma}}_j \in \mathcal{N}_j$, which completes the proof.

□

B.3.3 Theorem 1' (Soft MPEN)

We can prove the same consistency results as in Theorem 1 for the Soft MPenPC.

Theorem 1'. (i) Under (C1) - (C5) and (C7), with a probability of $1 - O(1/p)$, for all $j \in \{1, \cdots, p\}$ there is a local minimizer $\hat{\boldsymbol{\gamma}}_j$ to problem (5) such that $j - l \in \hat{\mathcal{G}}^{(k)}$ if $l \in A_j^{(k)}$, and $j - l \notin \hat{\mathcal{G}}^{(k)}$ if $l \in \bar{A}_j$;

(ii) Under (C1)-(C7), with a probability of $1 - O(1/p)$, for all $j \in \{1, \cdots, p\}$ there is a local minimizer $\hat{\boldsymbol{\gamma}}_j$ to problem (5) such that $j - l \in \hat{\mathcal{G}}^{(k)}$ if $l \in A_j^{(k)}$, and $j - l \notin \hat{\mathcal{G}}^{(k)}$ if $l \in \bar{A}_j^{(k)}$.

Before the proof, we introduce a lemma from Cai et al. (2011).

Lemma 3. Let $\xi_1, \cdots, \xi_n$ be independent random variables with mean zero. Suppose that there exists some $t > 0$ and $\bar{B}_n$ such that

$$\sum_{k=1}^{n} \mathbb{E}(\xi_k^2 e^{t|\xi_k|}) \leq \bar{B}_n^2.$$

Then uniformly for $x \in (0, \bar{B}_n]$,

$$\Pr\left(\sum_{k=1}^{n} \xi_k \geq C_t \bar{B}_n x\right) \leq \exp(-x^2),$$

where $C_t = t + t^{-1}$.

This is the Lemma 1 of Cai et al. (2011) and we omit its proof here.

Proof. The proof of Theorem 1' is similar to that of Theorem 1. It suffices to show Lemma 2 for weighted estimation of $\mathbf{\Sigma}^{(k)}$ and $\mathbf{X}_j^{(k)T}\boldsymbol{\epsilon}_l$. Particularly, χ and $\mathcal{E}$ are now defined as

$$\chi = \{\mathbf{X} : |\hat{\mathbf{\Sigma}}^{(k)} - \mathbf{\Sigma}^{(k)}|_\infty \leq C_3\sqrt{\log p/n}, \forall k\},$$

$$\mathcal{E} = \{\max_{j,l} |\mathbf{X}_l^{(k)T}\mathbf{W}^{(k)}\boldsymbol{\eta}_j^{(k)}|/n^{(k)} \leq C_4\sqrt{n^{-1}\log p}, \forall k\},$$

where $\hat{\mathbf{\Sigma}}^{(k)} = (\mathbf{X}^T\mathbf{W}^{(k)}\mathbf{X})/\mathbf{1}^T\mathbf{w}^{(k)}$ and $n^{(k)} = \mathbf{1}^T\mathbf{w}^{(k)}$.

Since $\hat{\Sigma}_{j,l}^{(k)} = \sum_{i=1}^n \xi_{i,j,l}^{(k)}/n^{(k)}$ and $\mathbf{X}_l^{(k)T}\mathbf{W}^{(k)}\boldsymbol{\eta}_j^{(k)} = \sum_{i=1}^n \zeta_{i,j,l}^{(k)}$, by Lemma 3, we have

$$\Pr\left(\left|\sum_{i=1}^{n}(\xi_{i,j,l}^{(k)} - \pi_i^{(k)}\Sigma_{j,l}^{(k)})\right| \geq \sqrt{nC_5}C_t\Delta\right) \leq \exp(-\Delta^2),$$

and

$$\Pr\left(\left|\sum_{i=1}^{n}\zeta_{i,j,l}^{(k)}\right| \geq \sqrt{nC_6}C_t\Delta\right) \leq \exp(-\Delta^2).$$

Moreover, by the Strong Law of Large Number, $n^{-1}\sum_{i=1}^n w_i^{(k)} \xrightarrow{a.s.} n^{-1}\sum_{i=1}^n \pi_i^{(k)}$. By taking $\Delta = \{C_3/(C_tC_5)(\log p)^{1/2}\}$ and $\{C_4/(C_tC_6)(\log p)^{1/2}\}$, we have $\Pr(\chi) = 1 - O(p^{-1})$ and $\Pr(\mathcal{E}) = 1 - O(p^{-1})$, which is the counterpart of Lemma 2 for Soft MPenPC. □

B.3.4 Theorem 5

We use a similar technique as in Ha et al. (2016) to show Theorem 4.

Proof. Given the CIG estimates from Stage I, the PC algorithm starts with a sparse initial graph $\hat{\mathcal{G}}^{(k)}$ for each class k. Based on our assumption, $\hat{\mathcal{G}}^{(k)}$ contains all edges in the skeleton $\mathcal{D}^{u(k)}$ plus at most $q(K-1)$ false edges for each node. Denote the union graph of all CIGs as $\mathcal{G}$, then the edges of $\hat{\mathcal{G}}^{(k)}$ form a subset of edges of $\mathcal{G}$.

For each edge $i-j$ in $\hat{\mathcal{G}}^{(k)}$, define $\mathcal{K}$ to be any set in Π_{jl} with $|\mathcal{K}| < n^{(k)} - 3$. Let $\nu_j = |A_j|$ for all j. By Lemma 3 of Kalisch and Bühlmann (2007),

$$\sup_{j,l,\mathcal{K}} \Pr(|\hat{z}^{(k)}_{j,l|\mathcal{K}} - z^{(k)}_{j,l|\mathcal{K}}| > \gamma) \leq O(n^{(k)} - \nu_j - \nu_l) \exp\{-C_7(n^{(k)} - \nu_j - \nu_j)\gamma^2\},$$

where $z^{(k)}_{j,l|\mathcal{K}} = \frac{1}{2}\log\{(1+\rho^{(k)}_{j,l|\mathcal{K}})/(1-\rho^{(k)}_{j,l|\mathcal{K}})\}$ and $\hat{z}^{(k)}_{j,l|\mathcal{K}} = \frac{1}{2}\log\{(1+\hat{\rho}^{(k)}_{j,l|\mathcal{K}})/(1-\hat{\rho}^{(k)}_{j,l|\mathcal{K}})\}$, for some $\gamma \in (0,2)$ and $C_7 > 0$.

Denote $E^I_{j,l|\mathcal{K}} = \{\text{Falsely reject } z^{(k)}_{j,l|\mathcal{K}} = 0\}$ and $E^{II}_{j,l|\mathcal{K}} = \{\text{Falsely accept } z_{j,l|\mathcal{K}} = 0\}$. Choose $\alpha = 2\{1-\Phi(\sqrt{n^{(k)}}c/2)\}$, then

$$\begin{aligned}\sup_{j,l,\mathcal{K}} \Pr(E^I_{j,l|\mathcal{K}}) =& \sup_{j,l,\mathcal{K}} \Pr\left(|\hat{z}^{(k)}_{j,l|\mathcal{K}} - z^{(k)}_{j,l|\mathcal{K}}| > \sqrt{n^{(k)}/(n^{(k)} - |\mathcal{K}| - 3)}c/2\right) \\ \leq& O(n^{(k)} - \nu_j - \nu_l)\exp\{-C_7'(n^{(k)} - \nu_j - \nu_l)c^2\},\end{aligned}$$

and

$$\begin{aligned}\sup_{j,l,\mathcal{K}} \Pr(E^{II}_{j,l|\mathcal{K}}) =& \sup_{j,l,\mathcal{K}} \Pr\left(|\hat{z}^{(k)}_{j,l|\mathcal{K}}| \leq \sqrt{n^{(k)}/(n^{(k)} - |\mathcal{K}| - 3)}c/2\right) \\ \leq& O(n^{(k)} - \nu_j - \nu_l)\exp\{-C_7''(n^{(k)} - \nu_j - \nu_l)c^2\},\end{aligned}$$

for some $C_7', C_7'' > 0$.

Therefore,

$$
\begin{aligned}
&\Pr(\text{An error occurs in the PC step of MPenPC}) \\
\leq &\sum_{j=1}^{p} \sum_{l \in \mathcal{N}_j} 2^{\nu_j + \nu_l} O(n^{(k)} - \nu_j - \nu_l) \exp\{-C_8 (n^{(k)} - \nu_j - \nu_l) c^2\} \\
\leq &\sum_{j=1}^{p} \sum_{l \in \mathcal{N}_j} O(n) 2^{2q} \exp\{-C_8 (n^{(k)} - 2q) c^2\} \\
\leq &O(n) pq \exp\{2q - C_8 (n^{(k)} - 2q) c^2\} = O(\exp(-C n^{1-2d_2})),
\end{aligned}
$$

for some C and $d_2 > 0$. The last equation holds because of (C1) and (C2). □

B.4 Datasets for the Real Data Analysis

B.4.1 Cancer-Relevant Gene Sets

We use 17 cancer-relevant gene sets in the Molecular Signatures Database (MSigDB) C6 oncogenic signatures gene sets (`http://software.broadinstitute.org/gsea/msigdb`) in Section 5. These gene sets are: RB_DN.V1, RB_P107_DN.V1, RB_P130_DN.V1, P53_DN.V1, P53_DN.V2, PTEN_DN.V1, PTEN_DN.V2, JNK_DN.V1, MYC_UP.V1, AKT_UP.V1, AKT_UP_MTOR_DN.V1, EGFR_UP.V1, ERB2_UP.V1, CYCLIN_D1_KE_.V1, CYCLIN_D1_UP.V1, BRCA1_DN.V1, and KRAS.BREAST_UP.V1. Particularly, each of them is created by combining two corresponding gene sets (UP/DN). For instance, the RB_DN.V1 gene set we use is the union of two gene sets of MSigDB: RB_DN.V1_UP and RB_DN.V1_DN,

B.4.2 PathwayCommons Dataset for Benchmark Graph

We use the PathwayCommons datasets (`http://www.pathwaycommons.org/archives/PC2/v9/`) that are annotated by "interact with each other" and "are in complex with each other".

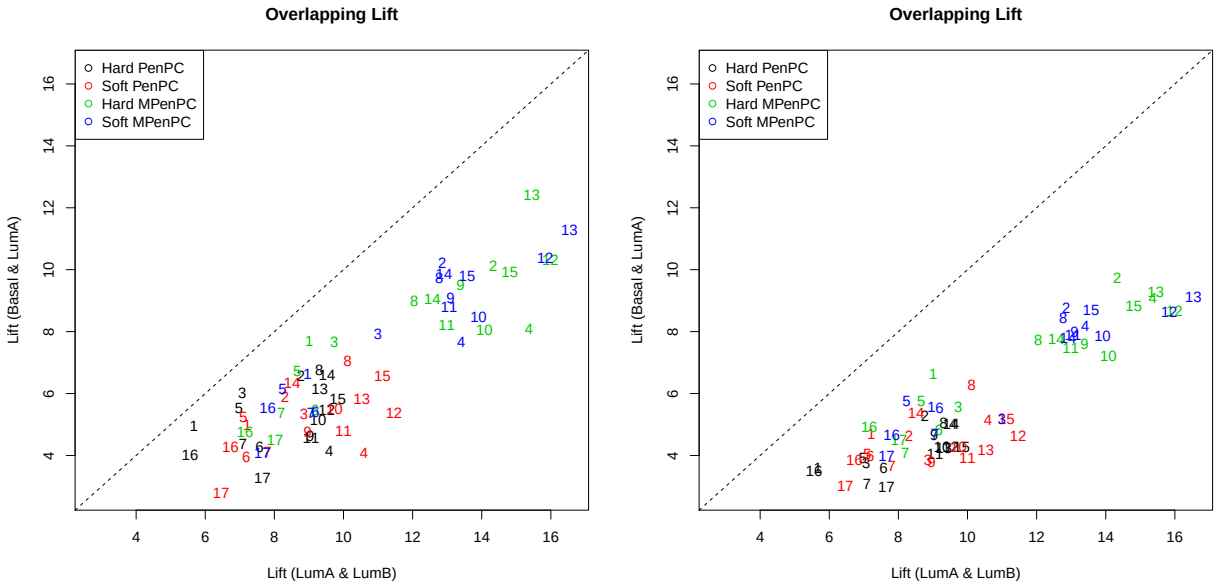

Figure B.8: The similarity, measured by $\text{lift}(\mathcal{G}_1, \mathcal{G}_2) = (p^2 - p)/2 \cdot |\mathcal{G}_1 \cap \mathcal{G}_2|/(|\mathcal{G}_1| \cdot |\mathcal{G}_2|)$, between the LumA/LumB skeleton estimates and that between the Basal/Lum skeleton estimates. Both the x-axes represent lift values of the LumA/LumB skeleton comparison. The y-axes represent lift values of the Basal/LumA (left panel) and the Basal/LumB (right panel) skeleton comparisons. The similarities are computed based on skeleton estimation for each gene set by all methods. Different methods are in different colors, and the numbers annotate the 17 cancer-relevant gene sets.

APPENDIX C

SUPPLEMENTARY MATERIALS FOR DEPENDENT GRAPHICAL MODELS

C.1 Least Square Approximation

C.1.1 Dependent Poisson Model

For the dependent Poisson-logNormal model, the conditional likelihood of the j-th node is

$$L_j(\boldsymbol{\beta}) = \int_{\mathbb{R}^n} (2\pi)^{-n/2} |c^{-1}\boldsymbol{\Omega}|^{1/2} \exp[-S_j(\mathbf{v};\boldsymbol{\beta})]\, \mathrm{d}\mathbf{v}, \tag{C.1}$$

where

$$S_j(\mathbf{v};\boldsymbol{\beta}) = \frac{1}{2c}\mathbf{v}^T\boldsymbol{\Omega}\mathbf{v} - \mathbf{y}^\top(\mathbf{X}\boldsymbol{\beta}+\mathbf{v}) + \sum_{i=1}^{n}\exp(\mathbf{X}_{i,\cdot}^\top\boldsymbol{\beta}+v_i).$$

With Taylor's expansion about $(\boldsymbol{\beta},\mathbf{v}) = (\mathbf{b},\mathbf{0})$, where $\mathbf{b}$ is a reasonable estimation of $\boldsymbol{\beta}$, we have

$$\begin{aligned} S_j(\mathbf{v};\boldsymbol{\beta}) &\approx \frac{1}{2c}\mathbf{v}^T\boldsymbol{\Omega}\mathbf{v} + \frac{1}{2}\sum_{i=1}^{n} e^{\mathbf{x}_i^\top\mathbf{b}}(e_i - \mathbf{x}_i^\top\boldsymbol{\beta} - v_i)^2 \\ &= \frac{1}{2}\mathbf{v}^\top(c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda})\mathbf{v} - (\mathbf{e}-\mathbf{X}\boldsymbol{\beta})^\top\boldsymbol{\Lambda}\mathbf{v} + \frac{1}{2}(\mathbf{e}-\mathbf{X}\boldsymbol{\beta})^\top\boldsymbol{\Lambda}(\mathbf{e}-\mathbf{X}\boldsymbol{\beta}) \\ &= \frac{1}{2}||\mathbf{v} - (c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}(\mathbf{e}-\mathbf{X}\boldsymbol{\beta})||^2_{c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda}} + \frac{1}{2}||\mathbf{e}-\mathbf{X}\boldsymbol{\beta}||^2_{\boldsymbol{\Lambda}-\boldsymbol{\Lambda}(c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}}, \end{aligned}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(e^{\mathbf{x}_1^\top\mathbf{b}},\ldots,e^{\mathbf{x}_n^\top\mathbf{b}})$ and $\mathbf{e} = ((y_1 - e^{\mathbf{x}_1^\top\mathbf{b}})/e^{\mathbf{x}_1^\top\mathbf{b}} + \mathbf{x}_1^\top\mathbf{b},\ldots,(y_1 - e^{\mathbf{x}_n^\top\mathbf{b}})/e^{\mathbf{x}_n^\top\mathbf{b}} + \mathbf{x}_n^\top\mathbf{b})^\top$. Thus we have,

$$\begin{aligned} L_j(\boldsymbol{\beta}) \approx& \int_{\mathbb{R}^n} (2\pi)^{-n/2}|c^{-1}\boldsymbol{\Omega}|^{1/2}\exp\left[-\frac{1}{2}||\mathbf{v}-(c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}(\mathbf{e}-\mathbf{X}\boldsymbol{\beta})||^2_{c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda}}\right]\mathrm{d}\mathbf{v} \\ &\times \exp\left(-\frac{1}{2}||\mathbf{e}-\mathbf{X}\boldsymbol{\beta}||^2_{\boldsymbol{\Lambda}-\boldsymbol{\Lambda}(c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}}\right) \\ =& |c^{-1}\boldsymbol{\Omega}|^{1/2}|c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda}|^{-1/2}\exp\left(-\frac{1}{2}||\mathbf{e}-\mathbf{X}\boldsymbol{\beta}||^2_{\boldsymbol{\Lambda}-\boldsymbol{\Lambda}(c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}}\right), \end{aligned} \tag{C.2}$$

and we can perform neighborhood selection by solving (4.2.2).

C.1.2 Dependent Hurdle Model

For the dependent Hurdle-logNormal model, the conditional likelihood of the j-th node is

$$L_j(\boldsymbol{\beta}) = \int_{\mathbb{R}^n} (2\pi)^{-n/2} |c^{-1}\boldsymbol{\Omega}|^{1/2} \exp[-S_j(\mathbf{v};\boldsymbol{\beta})]\, \mathrm{d}\mathbf{v}, \tag{C.3}$$

where

$$S_j(\mathbf{v};\boldsymbol{\beta}) = \frac{1}{2c}\mathbf{v}^T\boldsymbol{\Omega}\mathbf{v} + \sum_{i=1}^{n} \log(1 + e^{C_0 + C_1(\mathbf{x}_i^\top\boldsymbol{\beta} + v_i)}) - \sum_{y_i>0} [C_0 + (y_i - 1 + C_1)(\mathbf{x}_i^\top\boldsymbol{\beta} + v_i) - e^{\mathbf{x}_i^\top\boldsymbol{\beta} + v_i}].$$

Let $z_i = \mathbb{I}(y_i > 0)$, $\hat{z}_i = e^{C_0 + C_1\mathbf{x}_i^\top\mathbf{b}}/(1 + e^{C_0 + C_1\mathbf{x}_i^\top\mathbf{b}})$, $\hat{y}_i = e^{\mathbf{x}_i^\top\mathbf{b}}$ for $i = 1, \cdots, n$. With Taylor's expansion about $(\boldsymbol{\beta}, \mathbf{v}) = (\mathbf{b}, \mathbf{0})$, we have the following approximation up to a constant:

$$\begin{aligned}
S_j(\mathbf{v};\boldsymbol{\beta}) \approx & \frac{1}{2c}\mathbf{v}^T\boldsymbol{\Omega}\mathbf{v} + \frac{C_1^2}{2}\sum_{i=1}^{n} \hat{z}_i(1-\hat{z}_i)(e_i^{(0)} - \mathbf{x}_i^\top\boldsymbol{\beta} - v_i)^2 + \frac{1}{2}\sum_{y_i>0} \hat{y}_i(e_i^{(1)} - \mathbf{x}_i^\top\boldsymbol{\beta} - v_i)^2 \\
= & \frac{1}{2c}\mathbf{v}^T\boldsymbol{\Omega}\mathbf{v} + \frac{1}{2}||\mathbf{e}^{(0)} - \mathbf{X}\boldsymbol{\beta} - \mathbf{v}||_{\boldsymbol{\Delta}}^2 + \frac{1}{2}||\mathbf{e}^{(1)} - \mathbf{X}\boldsymbol{\beta} - \mathbf{v}||_{\boldsymbol{\Lambda}}^2 \\
= & \frac{1}{2}\mathbf{v}^T(c^{-1}\boldsymbol{\Omega} + \boldsymbol{\Delta} + \boldsymbol{\Lambda})\mathbf{v} - [(\mathbf{e}^{(0)} - \mathbf{X}\boldsymbol{\beta})^\top\boldsymbol{\Delta} + (\mathbf{e}^{(1)} - \mathbf{X}\boldsymbol{\beta})^\top\boldsymbol{\Lambda}]\mathbf{v} + \frac{1}{2}||\mathbf{e}^{(0)} - \mathbf{X}\boldsymbol{\beta}||_{\boldsymbol{\Delta}}^2 \\
& + \frac{1}{2}||\mathbf{e}^{(1)} - \mathbf{X}\boldsymbol{\beta}||_{\boldsymbol{\Lambda}}^2 \\
= & \frac{1}{2}||\mathbf{v} - (c^{-1}\boldsymbol{\Omega} + \boldsymbol{\Delta} + \boldsymbol{\Lambda})^{-1}[\boldsymbol{\Delta}(\mathbf{e}^{(0)} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\Lambda}(\mathbf{e}^{(1)} - \mathbf{X}\boldsymbol{\beta})]||_{c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Delta}+\boldsymbol{\Lambda}}^2 \\
& - \frac{1}{2}||\boldsymbol{\Delta}(\mathbf{e}^{(0)} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\Lambda}(\mathbf{e}^{(1)} - \mathbf{X}\boldsymbol{\beta})||_{(c^{-1}\boldsymbol{\Omega}+\boldsymbol{\Delta}+\boldsymbol{\Lambda})^{-1}}^2 + \frac{1}{2}||\mathbf{e}^{(0)} - \mathbf{X}\boldsymbol{\beta}||_{\boldsymbol{\Delta}}^2 \\
& + \frac{1}{2}||\mathbf{e}^{(1)} - \mathbf{X}\boldsymbol{\beta}||_{\boldsymbol{\Lambda}}^2,
\end{aligned}$$

where $\boldsymbol{\Delta} = \mathrm{diag}(C_1^2\hat{z}_1(1-\hat{z}_1), \ldots, C_1^2\hat{z}_n(1-\hat{z}_n))$, $\boldsymbol{\Lambda} = \mathrm{diag}(\hat{y}_1\mathbb{I}(y_1 > 0), \ldots, \hat{y}_n\mathbb{I}(y_n > 0))$, $\mathbf{e}^{(0)} = (\mathbf{x}_1^\top\mathbf{b} - (\hat{z}_1 - z_1)/[C_1\hat{z}_1(1-\hat{z}_1)], \ldots, \mathbf{x}_n^\top\mathbf{b} - (\hat{z}_n - z_n)/[C_1\hat{z}_n(1-\hat{z}_n)])$, and $\mathbf{e}^{(1)} = (\mathbf{x}_1^\top\mathbf{b} - (\hat{y}_1 - y_1 + 1)/\hat{y}_1, \ldots, \mathbf{x}_n^\top\mathbf{b} - (\hat{y}_n - y_n + 1)/\hat{y}_n)$. Thus we have $L_j(\boldsymbol{\beta}) \propto \exp\left(-\frac{1}{2}||(\tilde{\boldsymbol{\Delta}} + \tilde{\boldsymbol{\Lambda}} - \mathbf{K} - \mathbf{K}^\top)^{-1}(\tilde{\boldsymbol{\Delta}}\mathbf{e}^{(0)} + \tilde{\boldsymbol{\Lambda}}\mathbf{e}^{(1)} - \mathbf{K}\mathbf{e}^{(0)} - \mathbf{K}^\top\mathbf{e}^{(1)}) - \mathbf{X}\boldsymbol{\beta}||_{\tilde{\boldsymbol{\Delta}}+\tilde{\boldsymbol{\Lambda}}-\mathbf{K}-\mathbf{K}^\top}^2\right)$, where $\tilde{\boldsymbol{\Delta}} = \boldsymbol{\Delta} - \boldsymbol{\Delta}(c^{-1}\boldsymbol{\Omega} + \boldsymbol{\Delta} + \boldsymbol{\Lambda})^{-1}\boldsymbol{\Delta}$, $\tilde{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda} - \boldsymbol{\Lambda}(c^{-1}\boldsymbol{\Omega} + \boldsymbol{\Delta} + \boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}$, $\mathbf{K} = \boldsymbol{\Delta}(c^{-1}\boldsymbol{\Omega} + \boldsymbol{\Delta} + \boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}$. Define $\mathbf{e} = (\tilde{\boldsymbol{\Delta}} + \tilde{\boldsymbol{\Lambda}} - \mathbf{K} - \mathbf{K}^\top)^{-1}(\tilde{\boldsymbol{\Delta}}\mathbf{e}^{(0)} + \tilde{\boldsymbol{\Lambda}}\mathbf{e}^{(1)} - \mathbf{K}\mathbf{e}^{(0)} - \mathbf{K}^\top\mathbf{e}^{(1)})$, then we can perform neighborhood selection by solving (4.11).

BIBLIOGRAPHY

Allen, G. I., Liu, Z., et al. (2013). A local poisson graphical model for inferring networks from sequencing data. *IEEE Trans NanoBiosci*, 12(3):189–98.

Bickel, P. J. and Levina, E. (2004). Some theory for fisher's linear discriminant function, 'naive bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, 10(6):989–1010.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Bondell, H. D. and Reich, B. J. (2008). Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Breheny, P. (2015). The group exponential lasso for bi-level variable selection. *Biometrics*, 71(3):731–740.

Breheny, P. and Huang, J. (2009). Penalized methods for bi-level variable selection. *Statistics and its interface*, 2(3):369.

Cai, D., He, X., and Han, J. (2007). Semi-supervised discriminant analysis. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–7. IEEE.

Cai, T. and Liu, W. (2011). A direct estimation approach to sparse linear discriminant analysis. *Journal of the American Statistical Association*, 106(496):1566–1577.

Cai, T., Liu, W., and Luo, X. (2011). A constrained $\ell 1$ minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607.

Cerami, E. G., Gross, B. E., Demir, E., Rodchenkov, I., Babur, Ö., Anwar, N., Schultz, N., Bader, G. D., and Sander, C. (2010). Pathway commons, a web resource for biological pathway data. *Nucleic acids research*, 39(suppl_1):D685–D690.

Chen, J. and Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.

Chen, S., Witten, D. M., and Shojaie, A. (2014). Selection and estimation for mixed graphical models. *Biometrika*, 102(1):47–64.

Chen, T.-H. and Sun, W. (2017). Prediction of cancer drug sensitivity using high-dimensional omic features. *Biostatistics*, 18(1):1–14.

Chickering, D. M. (2002). Learning equivalence classes of bayesian-network structures. *Journal of Machine Learning Research*, 2(Feb):445–498.

Chiquet, J., Mariadassou, M., and Robin, S. (2018). Variational inference for sparse network reconstruction from count data. *arXiv preprint arXiv:1806.03120*.

Choi, Y., Coram, M., Peng, J., and Tang, H. (2017). A poisson log-normal model for constructing gene covariation network using RNA-seq data. *Journal of Computational Biology*, 24(7):721–731.

Clemmensen, L., Hastie, T., Witten, D., and Ersbøll, B. (2011). Sparse discriminant analysis. *Technometrics*, 53(4):406–413.

Colombo, D. and Maathuis, M. H. (2014). Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(1):3741–3782.

Dai, X., Li, T., Bai, Z., Yang, Y., Liu, X., Zhan, J., and Shi, B. (2015). Breast cancer intrinsic subtype classification, clinical use and future trends. *American Journal of Cancer Research*, 5(10):29292943.

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.

Fan, J. and Fan, Y. (2008). High dimensional classification using features annealed independence rules. *Annals of statistics*, 36(6):2605.

Fan, J., Feng, Y., and Tong, X. (2012). A road to classification in high dimensional space: the regularized optimal affine discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(4):745–771.

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.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

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

Friedman, N. and Koller, D. (2003). Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine Learning*, 50(1-2):95–125.

Gierahn, T. M., Wadsworth II, M. H., Hughes, T. K., Bryson, B. D., Butler, A., Satija, R., Fortune, S., Love, J. C., and Shalek, A. K. (2017). Seq-well: portable, low-cost RNA sequencing of single cells at high throughput. *Nature methods*, 14(4):395.

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

Ha, M. J., Sun, W., and Xie, J. (2016). Penpc : A two-step approach to estimate the skeletons of high-dimensional directed acyclic graphs. *Biometrics*, 72(1):146–155.

Han, S. W., Chen, G., Cheon, M.-S., and Zhong, H. (2016). Estimation of directed acyclic graphs through two-stage adaptive lasso for gene network inference. *Journal of the American Statistical Association*, 111(515):1004–1019.

Hand, D. J. et al. (2006). Classifier technology and the illusion of progress. *Statistical science*, 21(1):1–14.

Hastie, T., Tibshirani, R., and Buja, A. (1994). Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89(428):1255–1270.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer New York, New York, NY, second edition.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243.

Huang, J., Ma, S., Xie, H., and Zhang, C.-H. (2009). A group bridge approach for variable selection. *Biometrika*, 96(2):339–355.

Inouye, D. I., Yang, E., Allen, G. I., and Ravikumar, P. (2017). A review of multivariate distributions for count data derived from the poisson distribution. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(3):e1398.

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(Mar):613–636.

Kim, S., Pan, W., and Shen, X. (2013). Network-based penalized regression with application to genomic data. *Biometrics*, 69(3):582–593.

Li, C. and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182.

Liu, B., Shen, X., and Pan, W. (2013). Semi-supervised spectral clustering with application to detect population stratification. *Frontiers in genetics*, 4:215.

Liu, H., Han, F., Yuan, M., Lafferty, J., Wasserman, L., et al. (2012). High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326.

Liu, H., Lafferty, J., and Wasserman, L. (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10(Oct):2295–2328.

Liu, Y. and Yuan, M. (2011). Reinforced multicategory support vector machines. *Journal of Computational and Graphical Statistics*, 20(4):901–919.

Luo, S. and Chen, Z. (2014a). Edge detection in sparse gaussian graphical models. *Computational Statistics & Data Analysis*, 70:138–152.

Luo, S. and Chen, Z. (2014b). Sequential lasso cum ebic for feature selection with ultra-high dimensional feature space. *Journal of the American Statistical Association*, 109(507):1229–1240.

Mai, Q., Yang, Y., and Zou, H. (2015). Multiclass sparse discriminant analysis. *arXiv preprint arXiv:1504.05845*.

Mai, Q. and Zou, H. (2013). A note on the connection and equivalence of three sparse linear discriminant analysis methods. *Technometrics*, 55(2):243–246.

Mai, Q., Zou, H., and Yuan, M. (2012). A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika*, 99(1):29–42.

McDavid, A., Dennis, L., Danaher, P., Finak, G., Krouse, M., Wang, A., Webster, P., Beechem, J., and Gottardo, R. (2014). Modeling bi-modality improves characterization of cell cycle on gene expression in single cells. *PLoS computational biology*, 10(7):e1003696.

McDavid, A., Gottardo, R., Simon, N., and Drton, M. (2016). Graphical models for zero-inflated single cell gene expression. *arXiv preprint arXiv:1610.05857*.

Meier, L., Van De Geer, S., and Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71.

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

Min, W., Liu, J., and Zhang, S. (2018). Network-regularized sparse logistic regression models for clinical risk prediction and biomarker discovery. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 15(3):944–953.

Nandy, P., Hauser, A., and Maathuis, M. H. (2015). High-dimensional consistency in score-based and hybrid structure learning. *arXiv preprint arXiv:1507.02608*.

Negahban, S. N., Ravikumar, P., Wainwright, M. J., and Yu, B. (2012;2010;). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557.

Oates, C. J., Smith, J. Q., Mukherjee, S., and Cussens, J. (2016). Exact estimation of multiple directed acyclic graphs. *Statistics and Computing*, 26(4):797–811.

Obozinski, G., Jacob, L., and Vert, J.-P. (2011). Group lasso with overlaps: the latent group lasso approach. *arXiv preprint arXiv:1110.0413*.

Pan, W. and Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8:1145–1164.

Pan, W., Xie, B., and Shen, X. (2010). Incorporating predictor network in penalized regression with application to microarray data. *Biometrics*, 66(2):474–484.

Pang, H., Liu, H., and Vanderbei, R. (2014). The fastclime package for linear programming and large-scale precision matrix estimation in r. *The Journal of Machine Learning Research*, 15(1):489–493.

Patel, A. P., Tirosh, I., Trombetta, J. J., Shalek, A. K., Gillespie, S. M., Wakimoto, H., Cahill, D. P., Nahed, B. V., Curry, W. T., Martuza, R. L., et al. (2014). Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401.

Pearl, J. (2009). *Causality*. Cambridge university press.

Ravikumar, P., Wainwright, M. J., Lafferty, J. D., et al. (2010). High-dimensional ising model selection using 611-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319.

Schmidt, M., Niculescu-Mizil, A., and Murphy, K. (2007). Learning graphical model structure using l1-regularization paths. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, volume 2, pages 1278–1283. AAAI Press.

Shao, J., Wang, Y., Deng, X., and Wang, S. (2011). Sparse linear discriminant analysis by thresholding for high dimensional data. *The Annals of statistics*, 39(2):1241–1265.

Siegel, R. L., Miller, K. D., and Jemal, A. (2016). Cancer statistics, 2016. *CA: A Cancer Journal for Clinicians*, 66(1):7–30.

Sinclair, D. and Hooker, G. (2017). Sparse inverse covariance estimation for high-throughput microRNA sequencing data in the poisson log-normal graphical model. *arXiv preprint arXiv:1708.04490*.

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, Prediction, and Search*. Adaptive computation and machine learning. MIT Press.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550.

Sun, W., Ibrahim, J. G., and Zou, F. (2010). Genomewide multiple-loci mapping in experimental crosses by iterative adaptive penalized regression. *Genetics*, 185(1):349–359.

Svensson, V., Natarajan, K. N., Ly, L.-H., Miragaia, R. J., Labalette, C., Macaulay, I. C., Cvejic, A., and Teichmann, S. A. (2017). Power analysis of single-cell rna-sequencing experiments. *Nature methods*, 14(4):381.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572.

Tirosh, I., Izar, B., Prakadan, S. M., Wadsworth, M. H., Treacy, D., Trombetta, J. J., Rotem, A., Rodman, C., Lian, C., Murphy, G., et al. (2016). Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science*, 352(6282):189–196.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78.

Vanderbei, R. J. et al. (2015). *Linear programming*. Springer.

Voorman, A., Shojaie, A., and Witten, D. (2013). Graph estimation with joint additive models. *Biometrika*, 101(1):85–101.

Wang, H. and Leng, C. (2007). Unified lasso estimation by least squares approximation. *Journal of the American Statistical Association*, 102(479):1039–1048.

Witten, D. M. and Tibshirani, R. (2011). Penalized classification using fisher's linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):753–772.

Wu, H., Deng, X., and Ramakrishnan, N. (2018a). Sparse estimation of multivariate poisson log-normal models from count data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 11(2):66–77.

Wu, M., Zhu, L., Feng, X., et al. (2018b). Network-based feature screening with applications to genome data. *The Annals of Applied Statistics*, 12(2):1250–1270.

Wu, M. C., Zhang, L., Wang, Z., Christiani, D. C., and Lin, X. (2009). Sparse linear discriminant analysis for simultaneous testing for the significance of a gene set/pathway and gene selection. *Bioinformatics*, 25(9):1145–1151.

Yang, E., Allen, G., Liu, Z., and Ravikumar, P. K. (2012a). Graphical models via generalized linear models. In *Advances in Neural Information Processing Systems*, pages 1358–1366.

Yang, E., Ravikumar, P. K., Allen, G. I., and Liu, Z. (2013). On poisson graphical models. In *Advances in Neural Information Processing Systems*, pages 1718–1726.

Yang, S., Yuan, L., Lai, Y.-C., Shen, X., Wonka, P., and Ye, J. (2012b). Feature grouping and selection over an undirected graph. pages 922–930. ACM.

Yang, Y. and Zou, H. (2015). A fast unified algorithm for solving group-lasso penalize learning problems. *Statistics and Computing*, 25(6):1129–1141.

Yu, G. and Liu, Y. (2016). Sparse regression incorporating graphical structure among predictors. *Journal of the American Statistical Association*, 111(514):707–720.

Yuan, M. (2010). High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11(Aug):2261–2286.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

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

Zhang, C. and Liu, Y. (2013). Multicategory large-margin unified machines. *The Journal of Machine Learning Research*, 14(1):1349–1386.

Zhang, C., Liu, Y., Wang, J., and Zhu, H. (2016). Reinforced angle-based multicategory support vector machines. *Journal of Computational and Graphical Statistics*, 25(3):806–825.

Zhang, W., Wan, Y.-w., Allen, G. I., Pang, K., Anderson, M. L., and Liu, Z. (2013). Molecular pathway identification using biological network-regularized logistic models. *BMC genomics*, 14(8):S7.

Zhao, P. and Yu, B. (2006). On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov):2541–2563.

Zhao, S. and Shojaie, A. (2016). A significance test for graph-constrained estimation. *Biometrics*, 72(2):484–493.

Zhou, H., Pan, W., and Shen, X. (2009). Penalized model-based clustering with unconstrained covariance matrices. *Electronic journal of statistics*, 3:1473.

Zhou, S. et al. (2014). Gemini: Graph estimation with matrix variate normal instances. *The Annals of Statistics*, 42(2):532–562.

Zhu, Y., Shen, X., and Pan, W. (2013). Simultaneous grouping pursuit and feature selection over an undirected graph. *Journal of the American Statistical Association*, 108(502):713–725.

Ziegenhain, C., Vieth, B., Parekh, S., Reinius, B., Guillaumet-Adkins, A., Smets, M., Leonhardt, H., Heyn, H., Hellmann, I., and Enard, W. (2017). Comparative analysis of single-cell rna sequencing methods. *Molecular cell*, 65(4):631–643.