Graph Estimation and Cluster Analysis in High Dimensions

Kean Ming Tan

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2015

Reading Committee:

Daniela Witten, Chair

Noah Simon

Mathias Drton

Maitreya Dunham (GSR)

Program Authorized to Offer Degree:
Department of Biostatistics

University of Washington

Abstract

Graph Estimation and Cluster Analysis in High Dimensions

Kean Ming Tan

Chair of the Supervisory Committee:
Associate Professor Daniela Witten
Department of Biostatistics

In many applications, it is of interest to uncover patterns from a high-dimensional data set in which the number of features, p, is larger than the number of observations, n. We consider the areas of graph estimation and cluster analysis, which are often used to construct gene expression network and to partition the observations or features into subgroups, respectively. For graph estimation, we propose a framework to estimate graphical models with a few hub nodes that are densely-connected to many other nodes. We apply our framework to three widely used probabilistic graphical models: the Gaussian graphical model, the covariance graph model, and the binary Ising model. For cluster analysis, we propose a novel methodology for partitioning both observations and features into groups simultaneously, which we refer to as sparse biclustering. We also propose a framework to account for the correlation among the observations and features when we perform sparse biclustering. In addition, we study the statistical properties of convex clustering, a recent proposal for cluster analysis, which involves solving a convex optimization problem.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Daniela Witten, for her patience, encouragement, and active involvement in my experience throughout graduate school. Her guidance has allowed me to work on research areas that I am passionate about, and to expand my experience in a broad range of research topics. I would like to thank my committee members, Mathias Drton and Noah Simon, for their support and helpful feedback on my work. I am extremely grateful to Ali Shojaie, Maryam Fazel, Su-In Lee, and Maitreya Dunham for collaborating on research projects. I would like to thank Patrick Heagerty for providing me useful insights in healthcare data applications. I would also like to thank the faculty of the Departments of Biostatistics and Statistics for the education that I received over the course of my doctoral studies. I would like to thank my friends Shizhe, Ashley, Jenn, Xin Lu, Cesar, Yingying, and Xu for fun times without which the PhD would have been a monotonous experience. Finally, I thank my family and my grandparents Seow An and Siew-Hoey for their constant support, love, and encouragement.

DEDICATION

To my deceased father

Chapter 1

INTRODUCTION

In recent years, there has been considerable interest in uncovering patterns from high-dimensional data in which the number of features, p, is larger than the number of observations, n. We consider two popular tools for exploratory analysis in the high-dimensional setting:

1. Graphical models, which characterizes the relationships among variables. A graph consists of a set of p nodes, each representing a variable, and a set of edges between pairs of nodes. The presence of an edge between two nodes indicates that there is a relationship between the two variables [54].

2. Cluster analysis, which partitions the observations or features into groups based on some similarity measure. Some widely used clustering methodologies are hierarchical clustering, K-means clustering, and spectral clustering [44, 68].

As a motivating example, consider a gene expression data set with n subjects and p genes. Gene expression data is high-dimensional in the sense that the number of genes is much larger than the number of subjects. An interesting scientific problem is to explore the relationships among the genes. A typical way of visualizing the relationships among the genes is to estimate a gene regulatory network or an undirected graph from the gene expression data. On the other hand, one might be interested in exploring whether there are potential subgroups among subjects or genes that share similar gene expression levels. Clustering methods such as hierarchical clustering or K-means clustering are often performed for this purpose.

The problem of graph estimation is well-studied in the classical setting when $n > p$ (for a detailed review, we refer the reader to [54]). However, in the high-dimensional setting, the number of edges to be estimated is much larger than the number of observations. This poses significant challenges to existing methods for constructing graphical models. Many authors have studied the use of regularization or penalization on the parameter of interest. The ℓ_1 regularization has been particularly popular since it shrinks some of the parameters to exactly zero, resulting in a sparse and more interpretable graph [95]. However, the ℓ_1 penalty implicitly assumes that each edge is equally likely and independent of all edges. This is unrealistic in many real-world networks, in which we believe that certain nodes (which are not known *a priori*) have many more edges than other nodes. To address this issue, we propose a general framework to accommodate a network with very *densely-connected* hub nodes: nodes that are connected to a very substantial number of other nodes in the network. We present this work in Chapter 2.

Classical clustering methodologies such as hierarchical clustering and K-means clustering have been widely used in the scientific literature for the past century (see, for example, [44, 68]). Many of the classical methodologies involve clustering the n observations of the data matrix on the basis of the p features, or clustering the p features on the basis of the n observations. We will refer to such proposals as *one-way clustering*. In high dimensions, one might expect that the true underlying clusters differ only with respect to a small fraction of the features. To address this, a number of authors have developed sparse one-way clustering methodologies (among others, [76, 103, 106]).

However, in certain cases, we may be faced with *transposable* data, characterized by the fact that both the rows and columns are of scientific interest and may contain clusters or other structure [55]. In this setting, one-way clustering seems inappropriate since it does not reflect the fact that both the rows and the columns are of scientific interest. To address this shortcoming, a number of proposals have been made for *biclustering*, which involves simultaneously clustering the rows and columns of a data matrix (see, for instance, [15, 37, 55, 56]). However, most of the existing proposals do not have an underlying statistical

model. We propose a sparse biclustering procedure based on the assumption that the data is normally distributed with some bicluster-specific mean and common variance, and that some biclusters have approximately zero mean. We present this work in Chapter 3.

Most of the current one-way clustering methodologies such as hierarchical clustering, k-means clustering, and spectral clustering take a greedy approach. In recent years, a number of authors have proposed formulations for *convex clustering* [16, 47, 60, 77]. It involves solving a convex optimization problem with a squared error loss and a penalty that encourages the n observations to have the same estimated mean. Many authors have proposed efficient algorithms to solve the resulting convex optimization problem. However, the statistical properties of convex clustering are not well understood in the literature. We present the statistical properties of convex clustering in Chapter 4.

Chapter 2

LEARNING GRAPHICAL MODELS WITH HUBS

This work is published in *Journal of Machine Learning Research* [91].

Graphical models are used to model a wide variety of systems, such as gene regulatory networks and social interaction networks. A graph consists of a set of p nodes, each representing a variable, and a set of edges between pairs of nodes. The presence of an edge between two nodes indicates a relationship between the two variables. In this chapter, we consider two types of graphs: conditional independence graphs and marginal independence graphs. In a conditional independence graph, an edge connects a pair of variables if and only if they are conditionally dependent—dependent conditional upon the other variables. In a marginal independence graph, two nodes are joined by an edge if and only if they are marginally dependent—dependent without conditioning on the other variables.

In recent years, many authors have studied the problem of learning a graphical model in the high-dimensional setting, in which the number of variables p is larger than the number of observations n. Let $\mathbf{X}$ be a $n \times p$ matrix, with rows $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Throughout the rest of the text, we will focus on three specific types of graphical models:

1. A *Gaussian graphical model*, where $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$. In this setting, $(\mathbf{\Sigma}^{-1})_{jj'} = 0$ for some $j \neq j'$ if and only if the jth and j'th variables are conditionally independent [68]; therefore, the sparsity pattern of $\mathbf{\Sigma}^{-1}$ determines the conditional independence graph.

2. A *Gaussian covariance graph model*, where $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$. Then $\Sigma_{jj'} = 0$ for some $j \neq j'$ if and only if the jth and j'th variables are marginally independent. Therefore, the sparsity pattern of $\mathbf{\Sigma}$ determines the marginal independence graph.

3. A *binary Ising graphical model*, where $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are i.i.d. with density function

$$p(\mathbf{x}, \mathbf{\Theta}) = \frac{1}{Z(\mathbf{\Theta})} \exp\left[\sum_{j=1}^{p} \theta_{jj} x_j + \sum_{1 \le j < j' \le p} \theta_{jj'} x_j x_{j'}\right],$$

where $\mathbf{\Theta}$ is a $p \times p$ symmetric matrix, and $Z(\mathbf{\Theta})$ is the partition function, which ensures that the density sums to one. Here, $\mathbf{x}$ is a binary vector, and $\theta_{jj'} = 0$ if and only if the jth and j'th variables are conditionally independent. The sparsity pattern of $\mathbf{\Theta}$ determines the conditional independence graph.

Chapter outline In Chapter 2.1, we provide a brief summary on some recent proposals for graph estimation. We present the hub penalty function in Chapter 2.2. We then apply the hub penalty function to the Gaussian graphical model, the covariance graph model, and the binary Ising graphical model in Chapter 2.3, 2.4, and 2.5, respectively. In Chapter 2.6, we apply our approach to a webpage data set and a gene expression data set. We close with a discussion in Chapter 2.7. The proofs are in Appendix A.

2.1 Existing Work

To construct an interpretable graph when $p > n$, many authors have proposed applying an ℓ_1 penalty to the parameter encoding each edge, in order to encourage sparsity. For instance, such an approach is taken by [35], [86], [113], and [115] in the Gaussian graphical model; [7], [8], [12], [29], [87], and [110] in the covariance graph model; and [48], [57], and [83] in the binary Ising graphical model.

However, applying an ℓ_1 penalty to each edge can be interpreted as placing an independent double-exponential prior on each edge. Consequently, such an approach implicitly assumes that each edge is equally likely and independent of all other edges; this corresponds to an Erdős-Rényi graph in which most nodes have approximately the same number of edges [30]. This is unrealistic in many real-world networks, in which we believe that certain nodes (which, unfortunately, are not known *a priori*) have a lot more edges than other nodes. An example is the network of webpages in the World Wide Web, where a relatively small number

of webpages are connected to many other webpages [5]. A number of authors have shown that real-world networks are *scale-free*, in the sense that the number of edges for each node follows a power-law distribution; examples include gene-regulatory networks, social networks, and networks of collaborations among scientists (among others, [4, 5, 51, 58, 59, 74]). More recently, [42] have shown that certain genes, referred to as *super hubs*, regulate hundreds of downstream genes in a gene regulatory network, resulting in far denser connections than are typically seen in a scale-free network.

In this chapter, we refer to very densely-connected nodes, such as the "super hubs" considered in [42], as *hubs*. When we refer to hubs, we have in mind nodes that are connected to a very substantial number of other nodes in the network—and in particular, we are referring to nodes that are much more densely-connected than even the most highly-connected node in a scale-free network. An example of a network containing hub nodes is shown in Figure 2.1.

We propose a convex penalty function for estimating graphs containing hubs. Our formulation simultaneously identifies the hubs and estimates the entire graph. The penalty function yields a convex optimization problem when combined with a convex loss function. We consider the application of this hub penalty function in modeling Gaussian graphical models, covariance graph models, and binary Ising models. Our formulation does not require that we know *a priori* which nodes in the network are hubs.

In related work, several authors have proposed methods to estimate a scale-free Gaussian graphical model [22, 62]. However, those methods do not model hub nodes—the most highly-connected nodes that arise in a scale-free network are far less connected than the hubs that we consider in our formulation. Under a different framework, some authors proposed a screening-based procedure to identify hub nodes in the context of Gaussian graphical models [32, 45]. Our proposal outperforms such approaches when hub nodes are present (see discussion in Chapter 2.3.5.4).

In Figure 2.1, the performance of our proposed approach is shown in a toy example in the context of a Gaussian graphical model. We see that when the true network contains hub nodes (Figure 2.1(a)), our proposed approach (Figure 2.1(b)) is much better able to recover

the network than is the graphical lasso (Figure 2.1(c)), a well-studied approach that applies an ℓ_1 penalty to each edge in the graph [35].

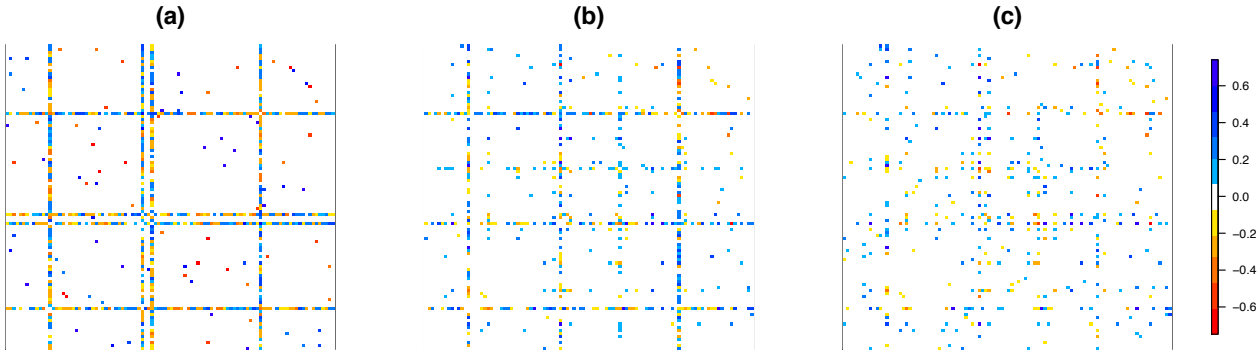

Figure 2.1: (a): Heatmap of the inverse covariance matrix in a toy example of a Gaussian graphical model with four hub nodes. White elements are zero and colored elements are non-zero in the inverse covariance matrix. Thus, colored elements correspond to edges in the graph. (b): Estimate from the *hub graphical lasso*. (c): Graphical lasso estimate.

2.2 The General Formulation

In this chapter, we present a general framework to accommodate network with hub nodes.

2.2.1 The Hub Penalty Function

Let $\mathbf{X}$ be a $n \times p$ data matrix, $\mathbf{\Theta}$ a $p \times p$ symmetric matrix containing the parameters of interest, and $\ell(\mathbf{X}, \mathbf{\Theta})$ a loss function (assumed to be convex in $\mathbf{\Theta}$). In order to obtain a sparse and interpretable graph estimate, many authors have considered the problem

$$\underset{\mathbf{\Theta} \in \mathcal{S}}{\text{minimize}} \quad \{\ell(\mathbf{X}, \mathbf{\Theta}) + \lambda \|\mathbf{\Theta} - \text{diag}(\mathbf{\Theta})\|_1\}, \tag{2.1}$$

where λ is a non-negative tuning parameter, $\mathcal{S}$ is some set depending on the loss function, and $\|\cdot\|_1$ is the sum of the absolute values of the matrix elements. For instance, in the case of a Gaussian graphical model, we could take $\ell(\mathbf{X}, \mathbf{\Theta}) = -\log\det\mathbf{\Theta} + \text{trace}(\mathbf{S\Theta})$, the negative log-likelihood of the data, where $\mathbf{S}$ is the empirical covariance matrix and $\mathcal{S}$ is the

set of $p \times p$ positive definite matrices. The solution to (2.1) can then be interpreted as an estimate of the inverse covariance matrix. The ℓ_1 penalty in (2.1) encourages zeros in the solution. But it typically does not yield an estimate that contains hubs.

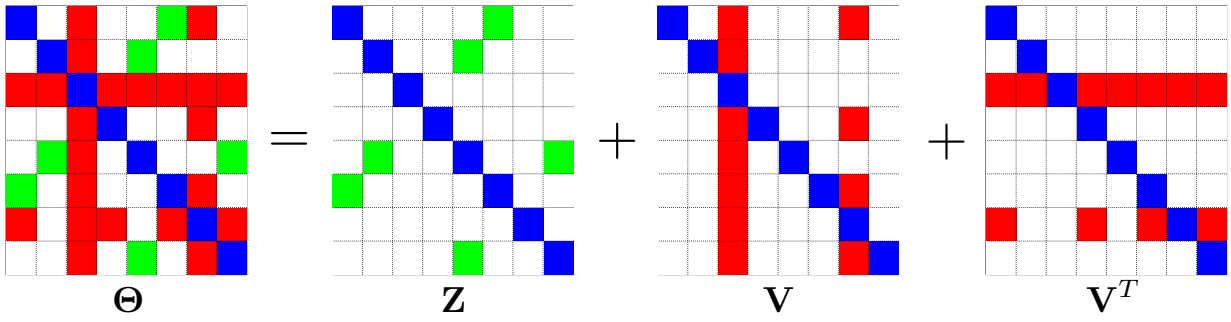

Figure 2.2: Decomposition of a symmetric matrix $\mathbf{\Theta}$ into $\mathbf{Z} + \mathbf{V} + \mathbf{V}^T$, where $\mathbf{Z}$ is sparse, and most columns of $\mathbf{V}$ are entirely zero. Blue, white, green, and red elements are diagonal, zero, non-zero in $\mathbf{Z}$, and non-zero due to two hubs in $\mathbf{V}$, respectively.

In order to explicitly model hub nodes in a graph, we wish to replace the ℓ_1 penalty in (2.1) with a convex penalty that encourages a solution that can be decomposed as $\mathbf{Z}+\mathbf{V}+\mathbf{V}^T$, where $\mathbf{Z}$ is a sparse symmetric matrix, and $\mathbf{V}$ is a matrix whose columns are either entirely zero or almost entirely non-zero (see Figure 2.2). The sparse elements of $\mathbf{Z}$ represent edges between non-hub nodes, and the non-zero columns of $\mathbf{V}$ correspond to hub nodes. We achieve this goal via the *hub penalty function*, which takes the form

$$\mathrm{P}(\mathbf{\Theta}) = \min_{\mathbf{V},\mathbf{Z}:\ \mathbf{\Theta}=\mathbf{V}+\mathbf{V}^T+\mathbf{Z}} \left\{ \lambda_1 \|\mathbf{Z} - \mathrm{diag}(\mathbf{Z})\|_1 + \lambda_2 \|\mathbf{V} - \mathrm{diag}(\mathbf{V})\|_1 + \lambda_3 \sum_{j=1}^{p} \|(\mathbf{V} - \mathrm{diag}(\mathbf{V}))_j\|_q \right\}. \quad (2.2)$$

Here λ_1, λ_2, and λ_3 are nonnegative tuning parameters. Sparsity in $\mathbf{Z}$ is encouraged via the ℓ_1 penalty on its off-diagonal elements, and is controlled by the value of λ_1. The ℓ_1 and ℓ_1/ℓ_q norms on the columns of $\mathbf{V}$ induce group sparsity when $q = 2$ [90, 114]; λ_3 controls the selection of hub nodes, and λ_2 controls the sparsity of each hub node's connections to other nodes. The convex penalty (2.2) can be combined with $\ell(\mathbf{X}, \mathbf{\Theta})$ to yield the convex

optimization problem

$$\underset{\mathbf{\Theta}\in\mathcal{S},\mathbf{V},\mathbf{Z}}{\text{minimize}} \quad \Big\{ \ell(\mathbf{X},\mathbf{\Theta}) + \lambda_1\|\mathbf{Z}-\text{diag}(\mathbf{Z})\|_1 + \lambda_2\|\mathbf{V}-\text{diag}(\mathbf{V})\|_1 + \lambda_3 \sum_{j=1}^{p} \|(\mathbf{V}-\text{diag}(\mathbf{V}))_j\|_q \Big\} \text{ subject to } \mathbf{\Theta} = \mathbf{V}+\mathbf{V}^T+\mathbf{Z}, \tag{2.3}$$

where the set $\mathcal{S}$ depends on the loss function $\ell(\mathbf{X},\mathbf{\Theta})$.

Note that when $\lambda_2 \to \infty$ or $\lambda_3 \to \infty$, then (2.3) reduces to (2.1). In this chapter, we take $q = 2$, which leads to estimation of a network containing dense hub nodes. Other values of q such as $q = \infty$ are also possible (see, for example, [73]). We note that the hub penalty function is closely related to recent work on overlapping group lasso penalties in the context of learning multiple sparse precision matrices [73].

2.2.2 Algorithm

In order to solve (2.3) with $q = 2$, we use an *alternating direction method of multipliers* (ADMM) algorithm (see, for example, [10, 25, 26]). ADMM is an attractive algorithm for this problem, as it allows us to decouple some of the terms in (2.3) that are difficult to optimize jointly. In order to develop an ADMM algorithm for (2.3) with guaranteed convergence, we reformulate it as a consensus problem, as in [65]. The convergence of the algorithm to the optimal solution follows from classical results (see, for instance, the review papers [10, 25]).

In greater detail, we let $\mathbf{B} = (\mathbf{\Theta}, \mathbf{V}, \mathbf{Z})$, $\tilde{\mathbf{B}} = (\tilde{\mathbf{\Theta}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Z}})$,

$$f(\mathbf{B}) = \ell(\mathbf{X},\mathbf{\Theta}) + \lambda_1\|\mathbf{Z}-\text{diag}(\mathbf{Z})\|_1 + \lambda_2\|\mathbf{V}-\text{diag}(\mathbf{V})\|_1 + \lambda_3 \sum_{j=1}^{p} \|(\mathbf{V}-\text{diag}(\mathbf{V}))\|_2,$$

and

$$g(\tilde{\mathbf{B}}) = \begin{cases} 0 & \text{if } \tilde{\mathbf{\Theta}} = \tilde{\mathbf{V}} + \tilde{\mathbf{V}}^T + \tilde{\mathbf{Z}} \\ \infty & \text{otherwise.} \end{cases}$$

Then, we can rewrite (2.3) as

$$\underset{\mathbf{B},\tilde{\mathbf{B}}}{\text{minimize}} \ \Big\{ f(\mathbf{B}) + g(\tilde{\mathbf{B}}) \Big\} \qquad \text{subject to } \mathbf{B} = \tilde{\mathbf{B}}. \tag{2.4}$$

Algorithm 1 ADMM Algorithm for Solving (2.3).

1. **Initialize** the parameters:

   (a) primal variables $\mathbf{\Theta}, \mathbf{V}, \mathbf{Z}, \tilde{\mathbf{\Theta}}, \tilde{\mathbf{V}}$, and $\tilde{\mathbf{Z}}$ to the $p \times p$ identity matrix.

   (b) dual variables $\mathbf{W}_1, \mathbf{W}_2$, and $\mathbf{W}_3$ to the $p \times p$ zero matrix.

   (c) constants $\rho > 0$ and $\tau > 0$.

2. **Iterate** until the stopping criterion $\frac{\|\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1}\|_F^2}{\|\mathbf{\Theta}_{t-1}\|_F^2} \leq \tau$ is met, where $\mathbf{\Theta}_t$ is the value of $\mathbf{\Theta}$ obtained at the tth iteration:

   (a) Update $\mathbf{\Theta}, \mathbf{V}, \mathbf{Z}$:

      i. $\mathbf{\Theta} = \underset{\mathbf{\Theta} \in \mathcal{S}}{\arg\min} \left\{ \ell(\mathbf{X}, \mathbf{\Theta}) + \frac{\rho}{2} \|\mathbf{\Theta} - \tilde{\mathbf{\Theta}} + \mathbf{W}_1\|_F^2 \right\}$.

      ii. $\mathbf{Z} = S(\tilde{\mathbf{Z}} - \mathbf{W}_3, \frac{\lambda_1}{\rho})$, $\mathrm{diag}(\mathbf{Z}) = \mathrm{diag}(\tilde{\mathbf{Z}} - \mathbf{W}_3)$. Here S denotes the soft-thresholding operator, applied element-wise to a matrix: $S(A_{ij}, b) = \mathrm{sign}(A_{ij}) \max(|A_{ij}| - b, 0)$.

      iii. $\mathbf{C} = \tilde{\mathbf{V}} - \mathbf{W}_2 - \mathrm{diag}(\tilde{\mathbf{V}} - \mathbf{W}_2)$.

      iv. $\mathbf{V}_j = \max\left(1 - \frac{\lambda_3}{\rho \|S(\mathbf{C}_j, \lambda_2/\rho)\|_2}, 0\right) \cdot S(\mathbf{C}_j, \lambda_2/\rho)$ for $j = 1, \ldots, p$.

      v. $\mathrm{diag}(\mathbf{V}) = \mathrm{diag}(\tilde{\mathbf{V}} - \mathbf{W}_2)$.

   (b) Update $\tilde{\mathbf{\Theta}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Z}}$:

      i. $\mathbf{\Gamma} = \frac{\rho}{6} \left[(\mathbf{\Theta} + \mathbf{W}_1) - (\mathbf{V} + \mathbf{W}_2) - (\mathbf{V} + \mathbf{W}_2)^T - (\mathbf{Z} + \mathbf{W}_3) \right]$.

      ii. $\tilde{\mathbf{\Theta}} = \mathbf{\Theta} + \mathbf{W}_1 - \frac{1}{\rho}\mathbf{\Gamma}$; iii. $\tilde{\mathbf{V}} = \frac{1}{\rho}(\mathbf{\Gamma} + \mathbf{\Gamma}^T) + \mathbf{V} + \mathbf{W}_2$; iv. $\tilde{\mathbf{Z}} = \frac{1}{\rho}\mathbf{\Gamma} + \mathbf{Z} + \mathbf{W}_3$.

   (c) Update $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$:

      i. $\mathbf{W}_1 = \mathbf{W}_1 + \mathbf{\Theta} - \tilde{\mathbf{\Theta}}$; ii. $\mathbf{W}_2 = \mathbf{W}_2 + \mathbf{V} - \tilde{\mathbf{V}}$; iii. $\mathbf{W}_3 = \mathbf{W}_3 + \mathbf{Z} - \tilde{\mathbf{Z}}$.

The scaled augmented Lagrangian for (2.4) takes the form

$$L(\mathbf{B}, \tilde{\mathbf{B}}, \mathbf{W}) = \ell(\mathbf{X}, \mathbf{\Theta}) + \lambda_1 \|\mathbf{Z} - \text{diag}(\mathbf{Z})\|_1 + \lambda_2 \|\mathbf{V} - \text{diag}(\mathbf{V})\|_1 + \lambda_3 \sum_{j=1}^{p} \|(\mathbf{V} - \text{diag}(\mathbf{V}))_j\|_2 + g(\tilde{\mathbf{B}}) + \frac{\rho}{2} \|\mathbf{B} - \tilde{\mathbf{B}} + \mathbf{W}\|_F^2,$$

where $\mathbf{B}$ and $\tilde{\mathbf{B}}$ are the primal variables, and $\mathbf{W} = (\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3)$ is the dual variable. Note that the scaled augmented Lagrangian can be derived from the usual Lagrangian by adding a quadratic term and completing the square [10].

A general algorithm for solving (2.3) is provided in Algorithm 1. The derivation is in Appendix A.1. Note that only the update for $\mathbf{\Theta}$ (Step 2(a)i) depends on the form of the convex loss function $\ell(\mathbf{X}, \mathbf{\Theta})$. In the following chapters, we consider special cases of (2.3) that lead to estimation of Gaussian graphical models, covariance graph models, and binary networks with hub nodes.

2.3 The Hub Graphical Lasso

Assume that $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$. The well-known *graphical lasso* problem (see, for example, [35]) takes the form of (2.1) with $\ell(\mathbf{X}, \mathbf{\Theta}) = -\log\det\mathbf{\Theta} + \text{trace}(\mathbf{S\Theta})$, and $\mathbf{S}$ the empirical covariance matrix of $\mathbf{X}$:

$$\underset{\mathbf{\Theta} \in \mathcal{S}}{\text{minimize}} \quad \left\{ -\log\det\mathbf{\Theta} + \text{trace}(\mathbf{S\Theta}) + \lambda \sum_{j \neq j'} |\Theta_{jj'}| \right\}, \tag{2.5}$$

where $\mathcal{S} = \{\mathbf{\Theta} : \mathbf{\Theta} \succ 0 \text{ and } \mathbf{\Theta} = \mathbf{\Theta}^T\}$. The solution to this optimization problem serves as an estimate for $\mathbf{\Sigma}^{-1}$. We now use the hub penalty function to extend the graphical lasso in order to accommodate hub nodes.

2.3.1 Formulation and Algorithm

We propose the *hub graphical lasso* (HGL) optimization problem, which takes the form

$$\underset{\mathbf{\Theta} \in \mathcal{S}}{\text{minimize}} \quad \left\{ -\log\det\mathbf{\Theta} + \text{trace}(\mathbf{S\Theta}) + \text{P}(\mathbf{\Theta}) \right\}. \tag{2.6}$$

Again, $\mathcal{S} = \{\mathbf{\Theta} : \mathbf{\Theta} \succ 0 \text{ and } \mathbf{\Theta} = \mathbf{\Theta}^T\}$. It encourages a solution that contains hub nodes, as well as edges that connect non-hubs (Figure 2.1). Problem (2.6) can be solved using Algorithm 1. The update for $\mathbf{\Theta}$ in Algorithm 1 (Step 2(a)i) can be derived by minimizing

$$-\log\det\mathbf{\Theta} + \text{trace}(\mathbf{S}\mathbf{\Theta}) + \frac{\rho}{2}\|\mathbf{\Theta} - \tilde{\mathbf{\Theta}} + \mathbf{W}_1\|_F^2 \tag{2.7}$$

with respect to $\mathbf{\Theta}$ (note that the constraint $\mathbf{\Theta} \in \mathcal{S}$ in (2.6) is treated as an implicit constraint, due to the domain of definition of the $\log\det$ function). This can be shown to have the solution

$$\mathbf{\Theta} = \frac{1}{2}\mathbf{U}\left(\mathbf{D} + \sqrt{\mathbf{D}^2 + \frac{4}{\rho}\mathbf{I}}\right)\mathbf{U}^T,$$

where $\mathbf{U}\mathbf{D}\mathbf{U}^T$ denotes the eigen-decomposition of $\tilde{\mathbf{\Theta}} - \mathbf{W}_1 - \frac{1}{\rho}\mathbf{S}$.

The complexity of the ADMM algorithm for HGL is $O(p^3)$ per iteration; this is the complexity of the eigen-decomposition for updating $\mathbf{\Theta}$. We now briefly compare the computational time for the ADMM algorithm for solving (2.6) to that of an interior point method (using the solver `Sedumi` called from `cvx`). On a 1.86 GHz Intel Core 2 Duo machine, the interior point method takes approximately 3 minutes, while ADMM takes only 1 second, on a data set with $p = 30$. We present a more extensive run time study for the ADMM algorithm for HGL in Appendix A.5.

2.3.2 Conditions for HGL Solution to be Block Diagonal

In order to reduce computations for solving the HGL problem, we now present a necessary condition and a sufficient condition for the HGL solution to be block diagonal, subject to some permutation of the rows and columns. The conditions depend only on the tuning parameters λ_1, λ_2, and λ_3. These conditions build upon similar results in the context of Gaussian graphical models from the recent literature (see, for example, [21, 69, 73, 104, 112]). Let $C_1, C_2, \ldots, C_K$ denote a partition of the p features.

Theorem 2.1. *A sufficient condition for the HGL solution to be block diagonal with blocks given by $C_1, C_2, \ldots, C_K$ is that $\min\left\{\lambda_1, \frac{\lambda_2}{2}\right\} > |S_{jj'}|$ for all $j \in C_k, j' \in C_{k'}, k \neq k'$.*

Theorem 2.2. *A necessary condition for the HGL solution to be block diagonal with blocks given by $C_1, C_2, \ldots, C_K$ is that* $\min\left\{\lambda_1, \frac{\lambda_2+\lambda_3}{2}\right\} > |S_{jj'}|$ *for all* $j \in C_k, j' \in C_{k'}, k \neq k'$.

The proofs of Theorems 2.1 and 2.2 are in Appendix A.2. Theorem 2.1 implies that one can screen the empirical covariance matrix $\mathbf{S}$ to check if the HGL solution is block diagonal (using standard algorithms for identifying the connected components of an undirected graph; see, for example, [94]). Suppose that the HGL solution is block diagonal with K blocks, containing $p_1, \ldots, p_K$ features, and $\sum_{k=1}^{K} p_k = p$. Then, one can simply solve the HGL problem on the features within each block separately. Recall that the bottleneck of the HGL algorithm is the eigen-decomposition for updating $\mathbf{\Theta}$. The block diagonal condition leads to massive computational speed-ups for implementing the HGL algorithm: instead of computing an eigen-decomposition for a $p \times p$ matrix in each iteration of the HGL algorithm, we compute the eigen-decomposition of K matrices of dimensions $p_1 \times p_1, \ldots, p_K \times p_K$. The computational complexity per-iteration is reduced from $O(p^3)$ to $\sum_{k=1}^{K} O(p_k^3)$.

We illustrate the reduction in computational time due to these results in an example with $p = 500$. Without exploiting Theorem 2.1, the ADMM algorithm for HGL (with a particular value of λ) takes 159 seconds; in contrast, it takes only 22 seconds when Theorem 2.1 is applied. The estimated precision matrix has 107 connected components, the largest of which contains 212 nodes.

2.3.3 Some Properties of HGL

We now present several properties of the HGL optimization problem (2.6), which can be used to provide guidance on the suitable range for the tuning parameters λ_1, λ_2, and λ_3. In what follows, $\mathbf{Z}^*$ and $\mathbf{V}^*$ denote the optimal solutions for $\mathbf{Z}$ and $\mathbf{V}$ in (2.6). Let $\frac{1}{s} + \frac{1}{q} = 1$ (recall that q appears in Equation 2.2).

Lemma 2.1. *A sufficient condition for $\mathbf{Z}^*$ to be a diagonal matrix is that* $\lambda_1 > \frac{\lambda_2+\lambda_3}{2}$.

Lemma 2.2. *A sufficient condition for $\mathbf{V}^*$ to be a diagonal matrix is that* $\lambda_1 < \frac{\lambda_2}{2} + \frac{\lambda_3}{2(p-1)^{1/s}}$.

Corollary 2.1. *A necessary condition for both* $\mathbf{V}^*$ *and* $\mathbf{Z}^*$ *to be non-diagonal matrices is that* $\frac{\lambda_2}{2} + \frac{\lambda_3}{2(p-1)^{1/s}} \leq \lambda_1 \leq \frac{\lambda_2+\lambda_3}{2}$.

Furthermore, (2.6) reduces to the graphical lasso problem (2.5) under a simple condition.

Lemma 2.3. *If* $q = 1$*, then (2.6) reduces to (2.5) with tuning parameter* $\min\left\{\lambda_1, \frac{\lambda_2+\lambda_3}{2}\right\}$.

Note also that when $\lambda_2 \to \infty$ or $\lambda_3 \to \infty$, (2.6) reduces to (2.5) with tuning parameter λ_1. However, throughout the rest of this chapter, we assume that $q = 2$, and λ_2 and λ_3 are finite.

The solution $\hat{\mathbf{\Theta}}$ of (2.6) is unique, since (2.6) is a strictly convex problem. We now consider the question of whether the decomposition $\hat{\mathbf{\Theta}} = \hat{\mathbf{V}} + \hat{\mathbf{V}}^T + \hat{\mathbf{Z}}$ is unique. We see that the decomposition is unique in a certain regime of the tuning parameters. For instance, according to Lemma 2.1, when $\lambda_1 > \frac{\lambda_2+\lambda_3}{2}$, $\hat{\mathbf{Z}}$ is a diagonal matrix and hence $\hat{\mathbf{V}}$ is unique. Similarly, according to Lemma 2.2, when $\lambda_1 < \frac{\lambda_2}{2} + \frac{\lambda_3}{2(p-1)^{1/s}}$, $\hat{\mathbf{V}}$ is a diagonal matrix and hence $\hat{\mathbf{Z}}$ is unique. Studying more general conditions on $\mathbf{S}$ and on λ_1, λ_2, and λ_3 such that the decomposition is guaranteed to be unique is a challenging problem and is outside of the scope of this chapter.

2.3.4 Tuning Parameter Selection

In this section, we propose a *Bayesian information criterion* (BIC)-type quantity for tuning parameter selection in (2.6). Recall from Chapter 2.2 that the hub penalty function (2.2) decomposes the parameter of interest into the sum of three matrices, $\mathbf{\Theta} = \mathbf{Z} + \mathbf{V} + \mathbf{V}^T$, and places an ℓ_1 penalty on $\mathbf{Z}$, and an ℓ_1/ℓ_2 penalty on $\mathbf{V}$.

For the graphical lasso problem in (2.5), many authors have proposed to select the tuning parameter λ such that $\hat{\mathbf{\Theta}}$ minimizes the following quantity:

$$-n \cdot \log\det(\hat{\mathbf{\Theta}}) + n \cdot \text{trace}(\mathbf{S}\hat{\mathbf{\Theta}}) + \log(n) \cdot |\hat{\mathbf{\Theta}}|,$$

where $|\hat{\mathbf{\Theta}}|$ is the cardinality of $\hat{\mathbf{\Theta}}$, that is, the number of unique non-zeros in $\hat{\mathbf{\Theta}}$ (see, for example, [115]).[1]

[1]The term $\log(n) \cdot |\hat{\mathbf{\Theta}}|$ is motivated by the fact that the degrees of freedom for an estimate involving the ℓ_1 penalty can be approximated by the cardinality of the estimated parameter [118].

Using a similar idea, we propose the following BIC-type quantity for selecting the set of tuning parameters $(\lambda_1, \lambda_2, \lambda_3)$ for (2.6):

$$\text{BIC}(\hat{\mathbf{\Theta}}, \hat{\mathbf{V}}, \hat{\mathbf{Z}}) = -n \cdot \log\det(\hat{\mathbf{\Theta}}) + n \cdot \text{trace}(\mathbf{S}\hat{\mathbf{\Theta}}) + \log(n) \cdot |\hat{\mathbf{Z}}| + \log(n) \cdot \Big(\nu + c \cdot [|\hat{\mathbf{V}}| - \nu]\Big),$$

where ν is the number of estimated hub nodes, that is, $\nu = \sum_{j=1}^{p} 1_{\{\|\hat{\mathbf{V}}_j\|_0 > 0\}}$, c is a constant between zero and one, and $|\hat{\mathbf{Z}}|$ and $|\hat{\mathbf{V}}|$ are the cardinalities (the number of unique non-zeros) of $\hat{\mathbf{Z}}$ and $\hat{\mathbf{V}}$, respectively.[2] We select the set of tuning parameters $(\lambda_1, \lambda_2, \lambda_3)$ for which the quantity $\text{BIC}(\hat{\mathbf{\Theta}}, \hat{\mathbf{V}}, \hat{\mathbf{Z}})$ is minimized. Note that when the constant c is small, $\text{BIC}(\hat{\mathbf{\Theta}}, \hat{\mathbf{V}}, \hat{\mathbf{Z}})$ will favor more hub nodes in $\hat{\mathbf{V}}$. We take $c = 0.2$ in this chapter.

2.3.5 Simulation Study

In this section, we compare HGL to two sets of proposals: proposals that learn an Erdős-Rényi Gaussian graphical model, and proposals that learn a Gaussian graphical model in which some nodes are highly-connected.

2.3.5.1 Notation and Measures of Performance

We start by defining some notation. Let $\hat{\mathbf{\Theta}}$ be the estimate of $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ from a given proposal, and let $\hat{\mathbf{\Theta}}_j$ be its jth column. Let $\mathcal{H}$ denote the set of indices of the hub nodes in $\mathbf{\Theta}$ (that is, this is the set of true hub nodes in the graph), and let $|\mathcal{H}|$ denote the cardinality of the set. In addition, let $\hat{\mathcal{H}}_r$ be the set of *estimated hub nodes*: the set of nodes in $\hat{\mathbf{\Theta}}$ that are among the $|\mathcal{H}|$ most highly-connected nodes, and that have at least r edges. The values chosen for $|\mathcal{H}|$ and r depend on the simulation set-up, and will be specified in each simulation study.

We now define several measures of performance that will be used to evaluate the various methods.

---

[2]The term $\log(n) \cdot |\hat{\mathbf{Z}}|$ is motivated by the degrees of freedom from the ℓ_1 penalty, and the term $\log(n) \cdot \Big(\nu + c \cdot [|\hat{\mathbf{V}}| - \nu]\Big)$ is motivated by an approximation of the degrees of freedom of the ℓ_2 penalty proposed in [114].

- Number of correctly estimated edges: $\sum_{j<j'} \left(1_{\{|\hat{\Theta}_{jj'}|>10^{-5} \text{ and } |\Theta_{jj'}|\neq 0\}}\right)$.

- Proportion of correctly estimated hub edges:

$$\frac{\sum_{j\in\mathcal{H},j'\neq j} \left(1_{\{|\hat{\Theta}_{jj'}|>10^{-5} \text{ and } |\Theta_{jj'}|\neq 0\}}\right)}{\sum_{j\in\mathcal{H},j'\neq j} \left(1_{\{|\Theta_{jj'}|\neq 0\}}\right)}.$$

- Proportion of correctly estimated hub nodes: $\frac{|\hat{\mathcal{H}}_r \cap \mathcal{H}|}{|\mathcal{H}|}$.

- Sum of squared errors: $\sum_{j<j'} \left(\hat{\Theta}_{jj'} - \Theta_{jj'}\right)^2$.

2.3.5.2 Data Generation

We consider three set-ups for generating a $p \times p$ adjacency matrix $\mathbf{A}$.

I - Network with hub nodes: for all $i < j$, we set $A_{ij} = 1$ with probability 0.02, and zero otherwise. We then set A_{ji} equal to A_{ij}. Next, we randomly select $|\mathcal{H}|$ hub nodes and set the elements of the corresponding rows and columns of $\mathbf{A}$ to equal one with probability 0.7 and zero otherwise.

II - Network with two connected components and hub nodes: the adjacency matrix is generated as $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_2 \end{pmatrix}$, with $\mathbf{A}_1$ and $\mathbf{A}_2$ as in Set-up I, each with $|\mathcal{H}|/2$ hub nodes.

III - Scale-free network:[3] the probability that a given node has k edges is proportional to $k^{-\alpha}$. [5] observed that many real-world networks have $\alpha \in [2.1, 4]$; we took $\alpha = 2.5$. Note that there is no natural notion of hub nodes in a scale-free network. While some nodes in a scale-free network have more edges than one would expect in an Erdős-Rényi graph, there is no clear distinction between "hub" and "non-hub" nodes, unlike

[3]Recall that our proposal is not intended for estimating a scale-free network.

in Set-ups I and II. In our simulation settings, we consider any node that is connected to more than 5% of all other nodes to be a hub node.[4]

We then use the adjacency matrix $\mathbf{A}$ to create a matrix $\mathbf{E}$, as

$$E_{ij} \overset{\text{i.i.d.}}{\sim} \begin{cases} 0 & \text{if } A_{ij} = 0 \\ \text{Unif}([-0.75, -0.25] \cup [0.25, 0.75]) & \text{otherwise,} \end{cases}$$

and set $\bar{\mathbf{E}} = \frac{1}{2}(\mathbf{E} + \mathbf{E}^T)$. Given the matrix $\bar{\mathbf{E}}$, we set $\boldsymbol{\Sigma}^{-1}$ equal to $\bar{\mathbf{E}} + (0.1 - \Lambda_{\min}(\bar{\mathbf{E}}))\mathbf{I}$, where $\Lambda_{\min}(\bar{\mathbf{E}})$ is the smallest eigenvalue of $\bar{\mathbf{E}}$. We generate the data matrix $\mathbf{X}$ according to $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma})$. Then, variables are standardized to have standard deviation one.

2.3.5.3 Comparison to Graphical Lasso and Neighborhood Selection

In this section, we compare the performance of HGL to two proposals that learn a sparse Gaussian graphical model.

- The graphical lasso (2.5), implemented using the `R` package `glasso`.
- The neighborhood selection approach of [70], implemented using the `R` package `glasso`. This approach involves performing p ℓ_1-penalized regression problems, each of which involves regressing one feature onto the others.

We consider the three simulation set-ups described in the previous section with $n = 1000$, $p = 1500$, and $|\mathcal{H}| = 30$ hub nodes in Set-ups I and II. Figure 2.3 displays the results, averaged over 100 simulated data sets. Note that the sum of squared errors is not computed for [70], since it does not directly yield an estimate of $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$.

HGL has three tuning parameters. To obtain the curves shown in Figure 2.3, we fixed $\lambda_1 = 0.4$, considered three values of λ_3 (each shown in a different color in Figure 2.3), and

[4]The cutoff threshold of 5% is chosen in order to capture the most highly-connected nodes in the scale-free network. In our simulation study, around three nodes are connected to at least $0.05 \times p$ other nodes in the network. The precise choice of cut-off threshold has little effect on the results obtained in the figures that follow.

used a fine grid of values of λ_2. The solid black circle in Figure 2.3 corresponds to the set of tuning parameters $(\lambda_1, \lambda_2, \lambda_3)$ for which the BIC as defined in Chapter 2.3.4 is minimized. The graphical lasso and [70] each involves one tuning parameter; we applied them using a fine grid of the tuning parameter to obtain the curves shown in Figure 2.3.

Results for Set-up I are displayed in Figures 2.3-I(a) through 2.3-I(d), where we calculate the proportion of correctly estimated hub nodes as defined in Chapter 2.3.5.1 with $r = 300$. Since this simulation set-up exactly matches the assumptions of HGL, it is not surprising that HGL outperforms the other methods. In particular, HGL is able to identify most of the hub nodes when the number of estimated edges is approximately equal to the true number of edges. We see similar results for Set-up II in Figures 2.3-II(a) through 2.3-II(d), where the proportion of correctly estimated hub nodes is as defined in Chapter 2.3.5.1 with $r = 150$.

In Set-up III, recall that we define a node that is connected to at least 5% of all nodes to be a hub. The proportion of correctly estimated hub nodes is then as defined in Chapter 2.3.5.1 with $r = 0.05 \times p$. The results are presented in Figures 2.3-III(a) through 2.3-III(d). In this set-up, only approximately three of the nodes (on average) have more than 50 edges, and the hub nodes are not as highly-connected as in Set-up I or Set-up II. Nonetheless, HGL outperforms the graphical lasso and [70].

Finally, we see from Figure 2.3 that the set of tuning parameters $(\lambda_1, \lambda_2, \lambda_3)$ selected using BIC performs reasonably well. In particular, the graphical lasso solution always has BIC larger than HGL, and hence, is not selected.

2.3.5.4 Comparison to Additional Proposals

In this section, we compare the performance of HGL to three additional proposals:

- The partial correlation screening procedure of [45]. The elements of the partial correlation matrix (computed using a pseudo-inverse when $p > n$) are thresholded based on their absolute value, and a hub node is declared if the number of nonzero elements in the corresponding column of the thresholded partial correlation matrix is sufficiently

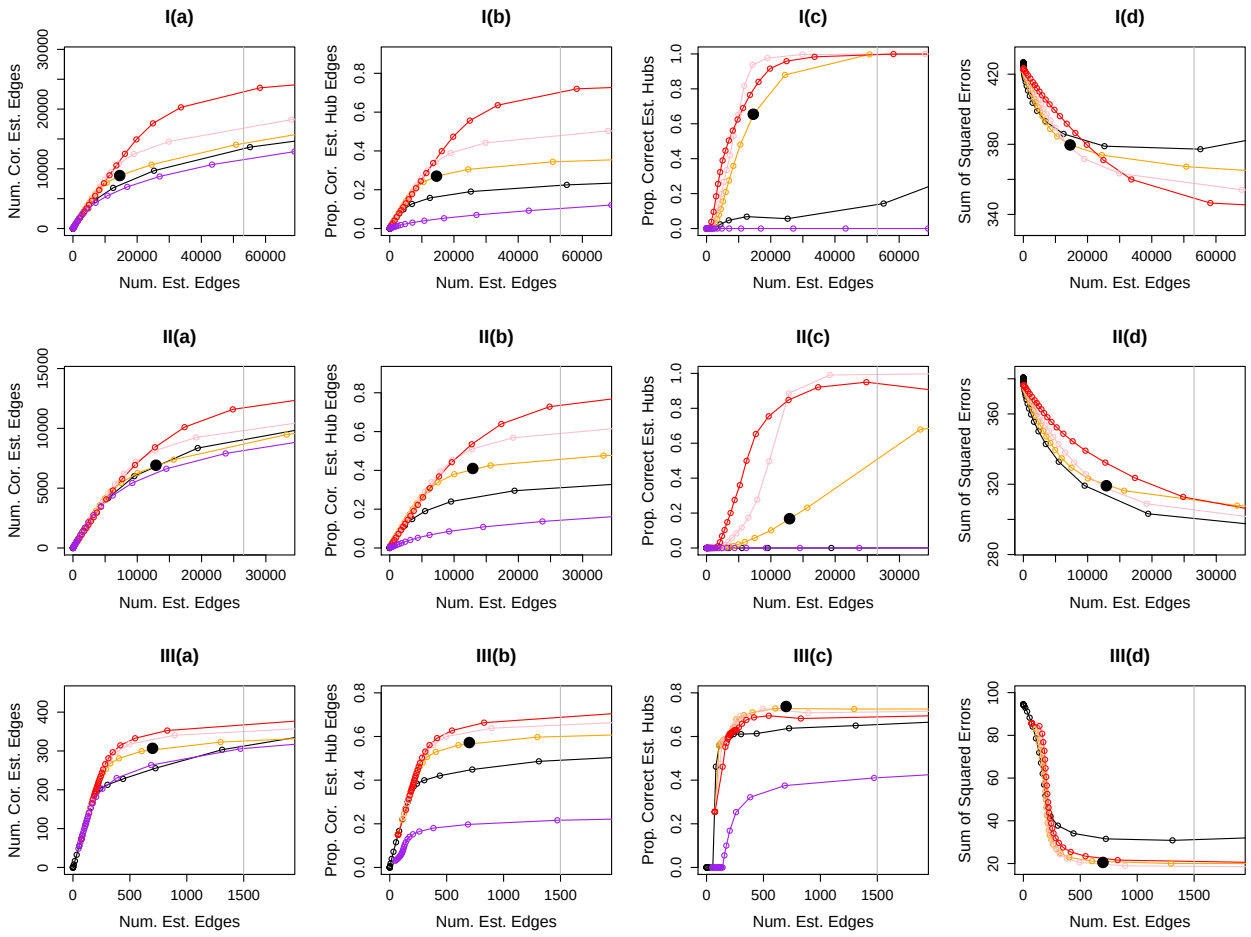

Figure 2.3: Simulation for Gaussian graphical model. Row I: Results for Set-up I. Row II: Results for Set-up II. Row III: Results for Set-up III. The results are for $n = 1000$ and $p = 1500$. In each panel, the x-axis displays the number of estimated edges, and the vertical gray line is the number of edges in the true network. The y-axes are as follows: Column (a): Number of correctly estimated edges; Column (b): Proportion of correctly estimated hub edges; Column (c): Proportion of correctly estimated hub nodes; Column (d): Sum of squared errors. The black solid circles are the results for HGL based on tuning parameters selected using the BIC-type criterion defined in Chapter 2.3.4. Colored lines correspond to the graphical lasso [35] (——); HGL with $\lambda_3 = 0.5$ (——), $\lambda_3 = 1$ (——), and $\lambda_3 = 2$ (——); neighborhood selection [70] (——).

large. Note that the purpose of [45] is to screen for hub nodes, rather than to estimate the individual edges in the network.

- The scale-free network estimation procedure of [62]. This is the solution to the non-convex optimization problem

$$\underset{\boldsymbol{\Theta}\in\mathcal{S}}{\text{minimize}} \quad \left\{ -\log\det\boldsymbol{\Theta} + \text{trace}(\mathbf{S}\boldsymbol{\Theta}) + \alpha\sum_{j=1}^{p}\log(\|\theta_{\setminus j}\|_1 + \epsilon_j) + \sum_{j=1}^{p}\beta_j|\theta_{jj}| \right\}, \tag{2.8}$$

  where $\theta_{\setminus j} = \{\theta_{jj'}|j' \neq j\}$, and ϵ_j, β_j, and α are tuning parameters. Here, $\mathcal{S} = \{\boldsymbol{\Theta} : \boldsymbol{\Theta} \succ 0 \text{ and } \boldsymbol{\Theta} = \boldsymbol{\Theta}^T\}$.

- Sparse partial correlation estimation procedure of [78], implemented using the `R` package `space`. This is an extension of the neighborhood selection approach of [70] that combines p ℓ_1-penalized regression problems in order to obtain a symmetric estimator. The authors claimed that the proposal performs well in estimating a scale-free network.

We generated data under Set-ups I and III (described in Chapter 2.3.5.2) with $n = 250$ and $p = 500$,[5] and with $|\mathcal{H}| = 10$ for Set-up I. The results, averaged over 100 data sets, are displayed in Figures 2.4 and 2.5.

To obtain Figures 2.4 and 2.5, we applied [62] using a fine grid of α values, and using the choices for β_j and ϵ_j specified by the authors: $\beta_j = 2\alpha/\epsilon_j$, where ϵ_j is a small constant specified in [62]. There are two tuning parameters in [45]: (1) ρ, the value used to threshold the partial correlation matrix, and (2) d, the number of non-zero elements required for a column of the thresholded matrix to be declared a hub node. We used $d = \{10, 20\}$ in Figures 2.4 and 2.5, and used a fine grid of values for ρ. Note that the value of d has no effect on the results for Figures 2.4(a)-(b) and Figures 2.5(a)-(b), and that larger values of d tend to yield worse results in Figures 2.4(c) and 2.5(c). For [78], we used a fine grid of tuning parameter values to obtain the curves shown in Figures 2.4 and 2.5. The sum of squared

[5] In this chapter, a small value of p was used due to the computations required to run the R package space, as well as computational demands of the [62] algorithm.

errors was not reported for [78] and [45] since they do not directly yield an estimate of the precision matrix. As a baseline reference, the graphical lasso is included in the comparison.

We see from Figure 2.4 that HGL outperforms the competitors when the underlying network contains hub nodes. It is not surprising that [62] yields better results than the graphical lasso, since the former approach is implemented via an iterative procedure: in each iteration, the graphical lasso is performed with an updated tuning parameter based on the estimate obtained in the previous iteration. [45] has the worst results in Figures 2.4(a)-(b); this is not surprising, since the purpose of [45] is to screen for hub nodes, rather than to estimate the individual edges in the network.

From Figure 2.5, we see that the performance of HGL is comparable to that of [62] and [78] under the assumption of a scale-free network; note that this is the precise setting for which [62]'s proposal is intended, and [78] reported that their proposal performs well in this setting. In contrast, HGL is not intended for the scale-free network setting (as mentioned in the Introduction, it is intended for a setting with hub nodes). Again, [62] and [78] outperform the graphical lasso, and [45] has the worst results in Figures 2.5(a)-(b). Finally, we see from Figures 2.4 and 2.5 that the BIC-type criterion for HGL proposed in Chapter 2.3.4 yields good results.

2.4 The Hub Covariance Graph

In this section, we consider estimation of a covariance matrix under the assumption that $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$; this is of interest because the sparsity pattern of $\mathbf{\Sigma}$ specifies the structure of the marginal independence graph (see, for example, [14, 23, 24]). We extend the covariance estimator of [110] to accommodate hub nodes.

2.4.1 Formulation and Algorithm

[110] proposed to estimate $\mathbf{\Sigma}$ using

$$\hat{\mathbf{\Sigma}} = \underset{\mathbf{\Sigma} \in \mathcal{S}}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{\Sigma} - \mathbf{S}\|_F^2 + \lambda \|\mathbf{\Sigma}\|_1 \right\}, \tag{2.9}$$

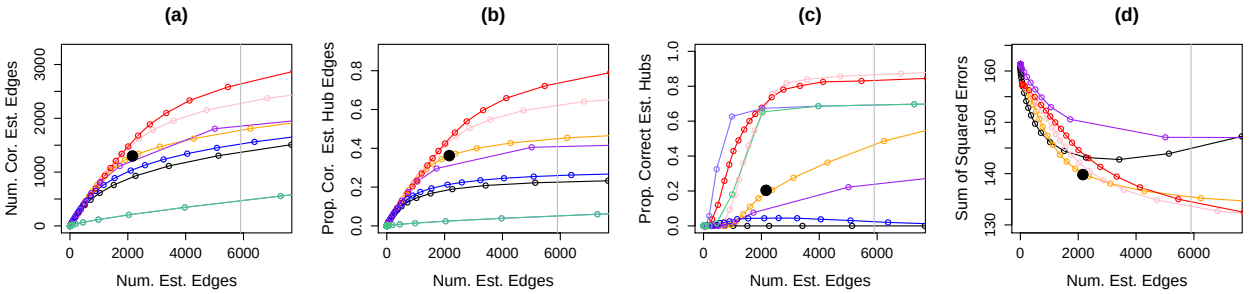

Figure 2.4: Simulation for the Gaussian graphical model. Set-up I was applied with $n = 250$ and $p = 500$. Details of the axis labels and the solid black circles are as in Figure 2.3. The colored lines correspond to the graphical lasso [35] (——); HGL with $\lambda_3 = 1$ (——), $\lambda_3 = 2$ (——), and $\lambda_3 = 3$ (——); the hub screening procedure [45] with $d = 10$ (——) and $d = 20$ (——); the scale-free network approach [62] (——); sparse partial correlation estimation [78] (——).

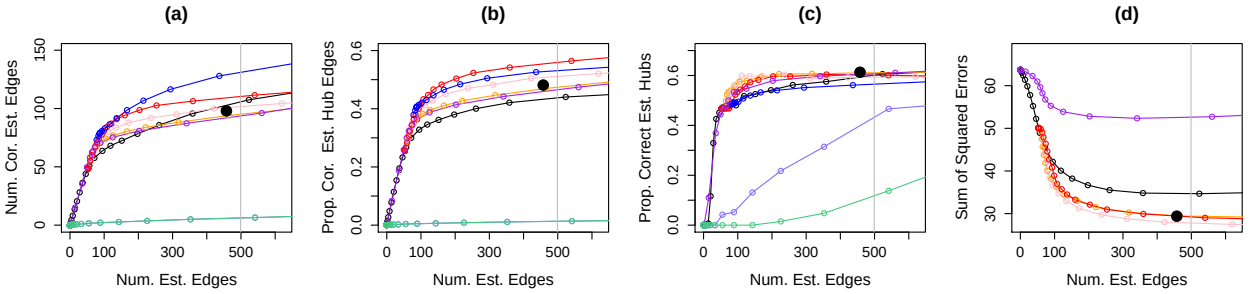

Figure 2.5: Simulation for the Gaussian graphical model. Set-up III was applied with $n = 250$ and $p = 500$. Details of the axis labels and the solid black circles are as in Figure 2.3. The colored lines correspond to the graphical lasso [35] (——); HGL with $\lambda_3 = 1$ (——), $\lambda_3 = 2$ (——), and $\lambda_3 = 3$ (——); the hub screening procedure [45] with $d = 10$ (——) and $d = 20$ (——); the scale-free network approach [62] (——); sparse partial correlation estimation [78] (——).

where $\mathbf{S}$ is the empirical covariance matrix, $\mathcal{S} = \{\mathbf{\Sigma} : \mathbf{\Sigma} \succeq \epsilon\mathbf{I} \text{ and } \mathbf{\Sigma} = \mathbf{\Sigma}^T\}$, and ϵ is a small positive constant; we take $\epsilon = 10^{-4}$. We extend (2.9) to accommodate hubs by imposing the hub penalty function (2.2) on $\mathbf{\Sigma}$. This results in the *hub covariance graph* (HCG) optimization problem,

$$\underset{\mathbf{\Sigma}\in\mathcal{S}}{\text{minimize}} \qquad \left\{\frac{1}{2}\|\mathbf{\Sigma} - \mathbf{S}\|_F^2 + \mathrm{P}(\mathbf{\Sigma})\right\},$$

which can be solved via Algorithm 1. To update $\mathbf{\Theta} = \mathbf{\Sigma}$ in Step 2(a)i, we note that

$$\underset{\mathbf{\Sigma}\in\mathcal{S}}{\arg\min}\left\{\frac{1}{2}\|\mathbf{\Sigma} - \mathbf{S}\|_F^2 + \frac{\rho}{2}\|\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}} + \mathbf{W}_1\|_F^2\right\} = \frac{1}{1+\rho}(\mathbf{S} + \rho\tilde{\mathbf{\Sigma}} - \rho\mathbf{W}_1)^+,$$

where $(\mathbf{A})^+$ is the projection of a matrix $\mathbf{A}$ onto the convex cone $\{\mathbf{\Sigma} \succeq \epsilon\mathbf{I}\}$. That is, if $\sum_{j=1}^p d_j\mathbf{u}_j\mathbf{u}_j^T$ denotes the eigen-decomposition of the matrix $\mathbf{A}$, then $(\mathbf{A})^+$ is defined as $\sum_{j=1}^p \max(d_j, \epsilon)\mathbf{u}_j\mathbf{u}_j^T$. The complexity of the ADMM algorithm is $O(p^3)$ per iteration, due to the complexity of the eigen-decomposition for updating $\mathbf{\Sigma}$.

2.4.2 Simulation Study

We compare HCG to two competitors for obtaining a sparse estimate of $\mathbf{\Sigma}$:

1. The non-convex ℓ_1-penalized log-likelihood approach of [8], using the `R` package `spcov`. This approach solves

$$\underset{\mathbf{\Sigma}\succ 0}{\text{minimize}}\left\{\log\det\mathbf{\Sigma} + \mathrm{trace}(\mathbf{\Sigma}^{-1}\mathbf{S}) + \lambda\|\mathbf{\Sigma}\|_1\right\}.$$

2. The convex ℓ_1-penalized approach of [110], given in (2.9).

We first generated an adjacency matrix $\mathbf{A}$ as in Set-up I in Chapter 2.3.5.2, modified to have $|\mathcal{H}| = 20$ hub nodes. Then $\bar{\mathbf{E}}$ was generated as described in Chapter 2.3.5.2, and we set $\mathbf{\Sigma}$ equal to $\bar{\mathbf{E}} + (0.1 - \Lambda_{\min}(\bar{\mathbf{E}}))\mathbf{I}$. Next, we generated $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$. Finally, we standardized the variables to have standard deviation one. In this simulation study, we set $n = 500$ and $p = 1000$.

Figure 2.6 displays the results, averaged over 100 simulated data sets. We calculated the proportion of correctly estimated hub nodes as defined in Chapter 2.3.5.1 with $r = 200$. We used a fine grid of tuning parameters for [110] in order to obtain the curves shown in each panel of Figure 2.6. HCG involves three tuning parameters, λ_1, λ_2, and λ_3. We fixed $\lambda_1 = 0.2$, considered three values of λ_3 (each shown in a different color), and varied λ_2 in order to obtain the curves shown in Figure 2.6.

Figure 2.6 does not display the results for the proposal of [8], due to computational constraints in the `spcov` `R` package. Instead, we compared our proposal to that of [8] using $n = 100$ and $p = 200$; those results are presented in Figure A.1 in Appendix A.4.

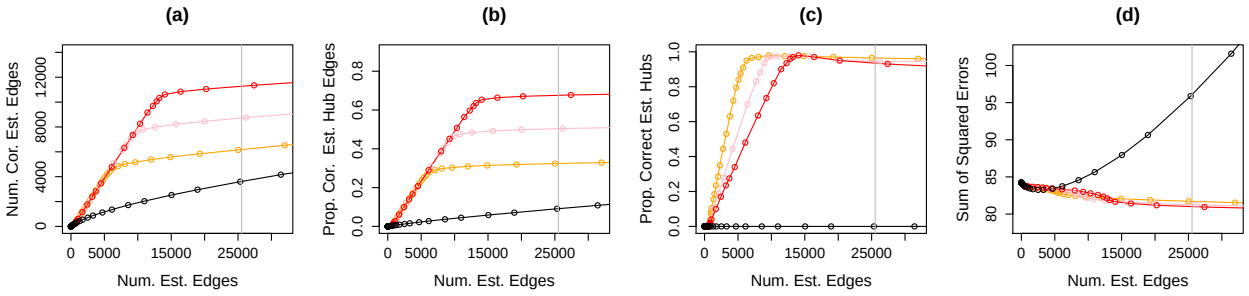

Figure 2.6: Covariance graph simulation with $n = 500$ and $p = 1000$. Details of the axis labels are as in Figure 2.3. The colored lines correspond to the proposal of [110] (——); HCG with $\lambda_3 = 1$ (——), $\lambda_3 = 1.5$ (——), and $\lambda_3 = 2$ (——).

We see that HCG outperforms the proposals of [110] (Figures 2.6 and A.1) and [8] (Figure A.1). These results are not surprising, since those other methods do not explicitly model the hub nodes.

2.5 The Hub Binary Network

In this section, we focus on estimating a binary Ising Markov random field, which we refer to as a binary network. We refer the reader to [83] for an in-depth discussion of this type of graphical model and its applications.

In this set-up, each entry of the $n \times p$ data matrix $\mathbf{X}$ takes on a value of zero or one. We assume that the observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are i.i.d. with density

$$p(\mathbf{x}, \mathbf{\Theta}) = \frac{1}{Z(\mathbf{\Theta})} \exp \left[\sum_{j=1}^{p} \theta_{jj} x_j + \sum_{1 \leq j < j' \leq p} \theta_{jj'} x_j x_{j'} \right], \tag{2.10}$$

where $Z(\mathbf{\Theta})$ is the partition function, which ensures that the density sums to one. Here $\mathbf{\Theta}$ is a $p \times p$ symmetric matrix that specifies the network structure: $\theta_{jj'} = 0$ implies that the jth and j'th variables are conditionally independent.

In order to obtain a sparse graph, [57] considered maximizing an ℓ_1-penalized log-likelihood under this model. Due to the difficulty in computing the log-partition function, several authors have considered alternative approaches. For instance, [83] proposed a neighborhood selection approach. The proposal of [83] involves solving p logistic regression separately, and hence, the estimated parameter matrix is not symmetric. In contrast, several authors considered maximizing an ℓ_1-penalized pseudo-likelihood with a symmetric constraint on $\mathbf{\Theta}$ (see, for example, [39, 40, 48]).

2.5.1 Formulation and Algorithm

Under the model (2.10), the log-pseudo-likelihood for n observations takes the form

$$\sum_{j=1}^{p} \sum_{j'=1}^{p} \theta_{jj'} (\mathbf{X}^T \mathbf{X})_{jj'} - \sum_{i=1}^{n} \sum_{j=1}^{p} \log \left(1 + \exp \left[\theta_{jj} + \sum_{j' \neq j} \theta_{jj'} x_{ij'} \right] \right), \tag{2.11}$$

where $\mathbf{x}_i$ is the ith row of the $n \times p$ matrix $\mathbf{X}$. The proposal of [48] involves maximizing (2.11) subject to an ℓ_1 penalty on $\mathbf{\Theta}$. We propose to instead impose the hub penalty function (2.2) on $\mathbf{\Theta}$ in (2.11) in order to estimate a sparse binary network with hub nodes. This leads to the optimization problem

$$\underset{\mathbf{\Theta} \in \mathcal{S}}{\text{minimize}} \quad \left\{ -\sum_{j=1}^{p} \sum_{j'=1}^{p} \theta_{jj'} (\mathbf{X}^T \mathbf{X})_{jj'} + \sum_{i=1}^{n} \sum_{j=1}^{p} \log \left(1 + \exp \left[\theta_{jj} + \sum_{j' \neq j} \theta_{jj'} x_{ij'} \right] \right) + \mathrm{P}(\mathbf{\Theta}) \right\}, \tag{2.12}$$

where $\mathcal{S} = \{\mathbf{\Theta} : \mathbf{\Theta} = \mathbf{\Theta}^T\}$. We refer to the solution to (2.12) as the *hub binary network* (HBN). The ADMM algorithm for solving (2.12) is given in Algorithm 1. We solve the

update for $\mathbf{\Theta}$ in Step 2(a)i using the Barzilai-Borwein method [6]. The details are given in Appendix **??**.

2.5.2 Simulation Study

Here we compare the performance of HBN to the proposal of [48], implemented using the `R` package `BMN`.

We simulated a binary network with $p = 50$ and $|\mathcal{H}| = 5$ hub nodes. To generate the parameter matrix $\mathbf{\Theta}$, we created an adjacency matrix $\mathbf{A}$ as in Set-up I of Chapter 2.3.5.2 with five hub nodes. Then $\bar{\mathbf{E}}$ was generated as in Chapter 2.3.5.2, and we set $\mathbf{\Theta} = \bar{\mathbf{E}}$.

Each of $n = 100$ observations was generated using Gibbs sampling [39, 83]. Suppose that $x_1^{(t)}, \ldots, x_p^{(t)}$ is obtained at the tth iteration of the Gibbs sampler. Then, the $(t+1)$th iteration is obtained according to

$$x_j^{(t+1)} \sim \text{Bernoulli}\left(\frac{\exp(\theta_{jj} + \sum_{j \neq j'} \theta_{jj'} x_{j'}^{(t)})}{1 + \exp(\theta_{jj} + \sum_{j \neq j'} \theta_{jj'} x_{j'}^{(t)})}\right) \qquad \text{for } j = 1, \ldots, p.$$

We took the first 10^5 iterations as our burn-in period, and then collected an observation every 10^4 iterations, such that the observations were nearly independent [39].

The results, averaged over 100 data sets, are shown in Figure 2.7. We used a fine grid of values for the ℓ_1 tuning parameter for [48], resulting in curves shown in each panel of the figure. For HBN, we fixed $\lambda_1 = 5$, considered $\lambda_3 = \{15, 25, 30\}$, and used a fine grid of values of λ_2. The proportion of correctly estimated hub nodes was calculated using the definition in Chapter 2.3.5.1 with $r = 20$. Figure 2.7 indicates that HBN consistently outperforms the proposal of [48].

2.6 Real Data Application

We now apply HGL to a university webpage data set, and a brain cancer data set.

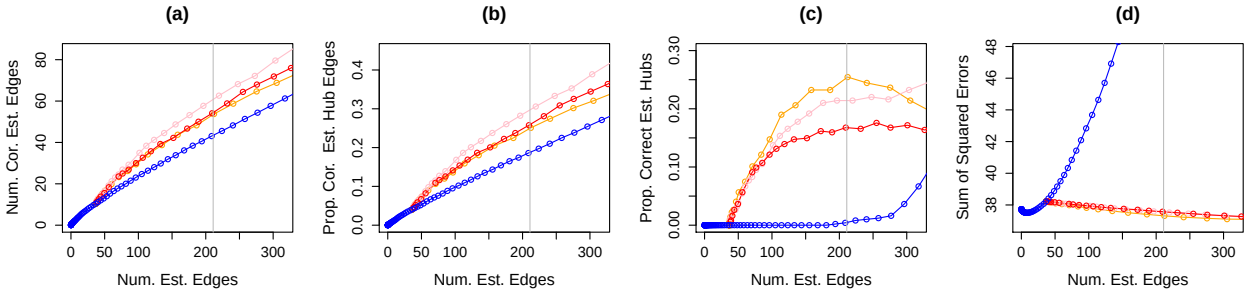

Figure 2.7: Binary network simulation with $n = 100$ and $p = 50$. Details of the axis labels are as in Figure 2.3. The colored lines correspond to the ℓ_1-penalized pseudo-likelihood proposal of [48] (——); and HBN with $\lambda_3 = 15$ (——), $\lambda_3 = 25$ (——), and $\lambda_3 = 30$ (——).

2.6.1 Application to University Webpage Data

We applied HGL to the university webpage data set from the "World Wide Knowledge Base" project at Carnegie Mellon University. This data set was pre-processed by [13]. The data set consists of the occurrences of various terms (words) on webpages from four computer science departments at Cornell, Texas, Washington and Wisconsin. We consider only the 544 student webpages, and select 100 terms with the largest entropy for our analysis. In what follows, we model these 100 terms as the nodes in a Gaussian graphical model.

The goal of the analysis is to understand the relationships among the terms that appear on the student webpages. In particular, we wish to identify terms that are hubs. We are not interested in identifying edges between non-hub nodes. For this reason, we fix the tuning parameter that controls the sparsity of $\mathbf{Z}$ at $\lambda_1 = 0.45$ such that the matrix $\mathbf{Z}$ is sparse. In the interest of a graph that is interpretable, we fix $\lambda_3 = 1.5$ to obtain only a few hub nodes, and then select a value of λ_2 ranging from 0.1 to 0.5 using the BIC-type criterion presented in Chapter 2.3.4. We performed HGL with the selected tuning parameters $\lambda_1 = 0.45$, $\lambda_2 = 0.25$, and $\lambda_3 = 1.5$.[6] The estimated matrices are shown in Figure 2.8.

Figure 2.8(a) indicates that six hub nodes are detected: *comput*, *research*, *scienc*, *software*,

[6]The results are qualitatively similar for different values of λ_1.

system, and *work*. For instance, the fact that *comput* is a hub indicates that many terms' occurrences are explained by the occurrence of the word *comput*. From Figure 2.8(b), we see that several pairs of terms take on non-zero values in the matrix $(\mathbf{Z} - \text{diag}(\mathbf{Z}))$. These include *(depart, univers)*; *(home, page)*; *(institut, technolog)*; *(graduat, student)*; *(univers, scienc)*, and *(languag,program)*. These results provide an intuitive explanation of the relationships among the terms in the webpages.

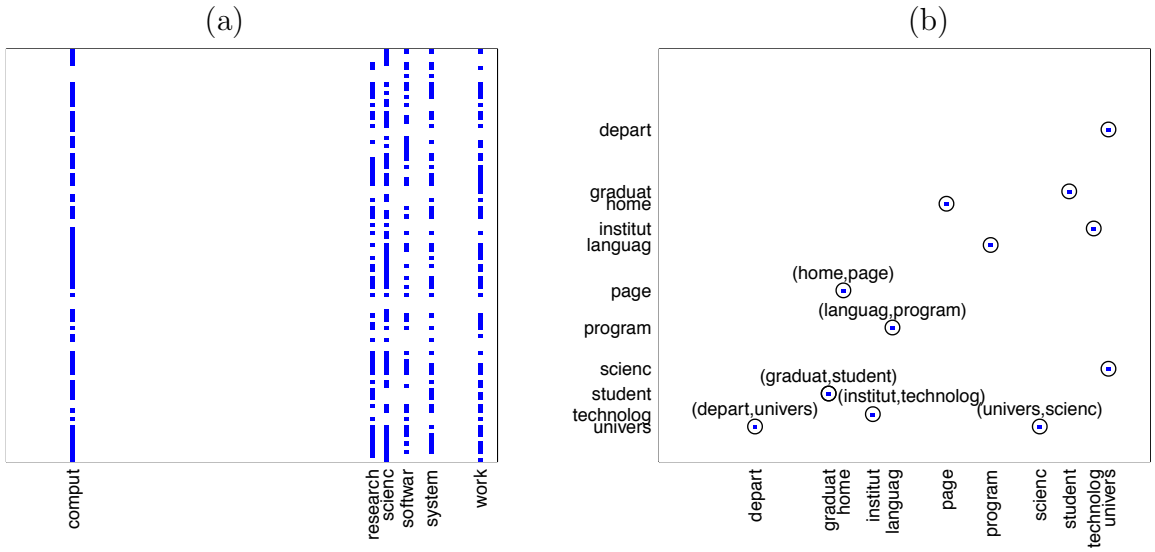

Figure 2.8: Results for HGL on the webpage data with tuning parameters selected using BIC: $\lambda_1 = 0.45$, $\lambda_2 = 0.25$, $\lambda_3 = 1.5$. Non-zero estimated values are shown, for *(a):* $(\mathbf{V} - \text{diag}(\mathbf{V}))$, and *(b):* $(\mathbf{Z} - \text{diag}(\mathbf{Z}))$.

2.6.2 Application to Gene Expression Data

We applied HGL to a publicly available cancer gene expression data set [102]. The data set consists of mRNA expression levels for 17,814 genes in 401 patients with glioblastoma multiforme (GBM), an extremely aggressive cancer with very poor patient prognosis. Among 7,462 genes known to be associated with cancer [82], we selected 500 genes with the highest variance.

We aim to reconstruct the gene regulatory network that represents the interactions among the genes, as well as to identify hub genes that tend to have many interactions with other genes. Such genes likely play an important role in regulating many other genes in the network. Identifying such regulatory genes will lead to a better understanding of brain cancer, and eventually may lead to new therapeutic targets. Since we are interested in identifying hub genes, and not as interested in identifying edges between non-hub nodes, we fix $\lambda_1 = 0.6$ such that the matrix $\mathbf{Z}$ is sparse. We fix $\lambda_3 = 6.5$ to obtain a few hub nodes, and we select λ_2 ranging from 0.1 to 0.7 using the BIC-type criterion presented in Chapter 2.3.4.

We applied HGL with this set of tuning parameters to the empirical covariance matrix corresponding to the 401×500 data matrix, after standardizing each gene to have variance one. In Figure 2.9, we plotted the resulting network (for simplicity, only the 438 genes with at least two neighbors are displayed). We found that five genes are identified as hubs. These genes are TRIM48, TBC1D2B, PTPN2, ACRC, and ZNF763, in decreasing order of estimated edges.

Interestingly, some of these genes have known regulatory roles. PTPN2 is known to be a signaling molecule that regulates a variety of cellular processes including cell growth, differentiation, mitotic cycle, and oncogenic transformation [67]. ZNF763 is a DNA-binding protein that regulates the transcription of other genes [67]. These genes do not appear to be highly-connected to many other genes in the estimate that results from applying the graphical lasso (2.5) to this same data set (results not shown). These results indicate that HGL can be used to recover known regulators, as well as to suggest other potential regulators that may be targets for follow-up analysis.

2.7 Discussion

We have proposed a general framework for estimating a network with hubs by way of a convex penalty function. The proposed framework has three tuning parameters, so that it can flexibly accommodate different numbers of hubs, sparsity levels within a hub, and connectivity levels among non-hubs. We have proposed a BIC-type quantity to select tuning

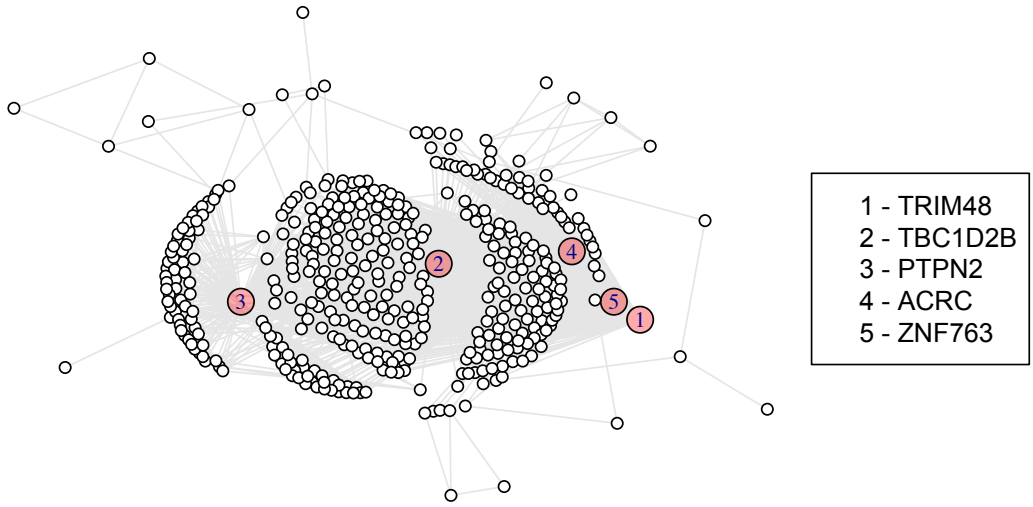

Figure 2.9: Results for HGL on the GBM data with tuning parameters selected using BIC: $\lambda_1 = 0.6$, $\lambda_2 = 0.4$, $\lambda_3 = 6.5$. Only nodes with at least two edges in the estimated network are displayed. Nodes displayed in pink were found to be hubs by the HGL algorithm.

parameters for our proposal. We note that tuning parameter selection in unsupervised settings remains a challenging open problem (see, for example, [33, 71]). In practice, tuning parameters could also be set based on domain knowledge or a desire for interpretability of the resulting estimates.

The framework proposed in this paper assumes an underlying model involving a set of edges between non-hub nodes, as well as a set of hub nodes. For instance, it is believed that such hub nodes arise in biology, in which "super hubs" in transcriptional regulatory networks may play important roles [42]. We note here that the underlying model of hub nodes assumed in this paper differs fundamentally from a scale-free network in which the degree of connectivity of the nodes follows a power law distribution—scale-free networks simply do not have such very highly-connected hub nodes. In fact, we have shown that existing techniques for estimating a scale-free network, such as [22] and [62], cannot accommodate the very dense hubs for which our proposal is intended.

As discussed in Chapter 2.2, the hub penalty function involves decomposing a parameter matrix $\mathbf{\Theta}$ into $\mathbf{Z} + \mathbf{V} + \mathbf{V}^T$, where $\mathbf{Z}$ is a sparse matrix, and $\mathbf{V}$ is a matrix whose columns are entirely zero or (almost) entirely non-zero. In this paper, we used an ℓ_1 penalty on $\mathbf{Z}$ in order to encourage it to be sparse. In effect, this amounts to assuming that the non-hub nodes obey an Erdős-Rényi network. But our formulation could be easily modified to accommodate a different network prior for the non-hub nodes. For instance, we could assume that the non-hub nodes obey a scale-free network, using the ideas developed in [22] and [62]. This would amount to modeling a scale-free network with hub nodes.

In this work, we applied the proposed framework to the tasks of estimating a Gaussian graphical model, a covariance graph model, and a binary network. The proposed framework can also be applied to other types of graphical models, such as the Poisson graphical model [1] or the exponential family graphical model [111].

An interesting question for future work is to study the theoretical statistical properties of the HGL formulation. For instance, in the context of the graphical lasso, it is known that the rate of statistical convergence depends upon the maximal degree of any node in the network [84]. It would be interesting to see whether HGL theoretically outperforms the graphical lasso in the setting in which the true underlying network contains hubs. Furthermore, it will be of interest to study HGL's hub recovery properties from a theoretical perspective.

Chapter 3

SPARSE BICLUSTERING OF TRANSPOSABLE DATA

This work is published in *Journal of Computational and Graphical Statistics* [92].

In recent years, much interest has centered around the unsupervised analysis of gene expression data and other types of high-dimensional biological data. Many proposals involve clustering the n observations on the basis of the p features, or clustering the p features on the basis of the n observations. We will refer to such proposals as *one-way clustering* in this chapter, since either the rows or columns of a data matrix are clustered, but not both. An overview of some popular one-way clustering procedures can be found in [44].

In certain cases, we may be faced with *transposable* data, characterized by the fact that both the rows and columns are of scientific interest and may contain clusters or other structure [55]. One such example is gene expression data, in which the rows represent tissue samples and the columns represent genes for which expression measurements were obtained. In this case, there may be subgroups among the rows (corresponding to distinct sets of patients, perhaps with different subtypes of a disease) or subgroups among the columns (corresponding to groups of genes with shared expression patterns, potentially revealing important biological pathways) [28]. In this setting, one-way clustering seems inappropriate since it does not reflect the fact that both the rows and the columns are of scientific interest. To address this shortcoming, a number of proposals have been made for *biclustering*, which involves simultaneously clustering the rows and columns of a data matrix (among others, [15, 19, 18, 37, 46, 55, 56, 66, 93]). We define a *bicluster* to be a subset of the data matrix, corresponding to a set of observations and a set of features, such that all elements within the subset are *similar* to each other; some authors refer to this as a *co-cluster*. The concept of similarity must be defined based on the data set and the scientific question.

(a)

2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0

(b)

4.0	5.0	7.0	3.0
5.0	6.0	8.0	4.0
3.0	4.0	6.0	2.0
1.0	2.0	4.0	0.0

(c)

0.5	1.0	2.0	1.5
2.0	4.0	8.0	6.0
1.5	3.0	6.0	4.5
1.0	2.0	4.0	3.0

Table 3.1: Biclusters with (a): constant values; (b): additive coherent values; and (c): multiplicative coherent values. Table adapted from [66].

In the literature, various authors have used the term *bicluster* in different ways. Three distinct types of biclusters are displayed in Table 3.1. The simplest type of bicluster is a *constant bicluster* (Table 3.1(a)), in which all elements take on approximately a constant value. Within an *additive coherent bicluster* (Table 3.1(b)), an additive model holds for each element; this is related to a two-way ANOVA model. Finally, a *multiplicative coherent bicluster* (Table 3.1(c)) stems from a multiplicative model. Biclustering proposals have taken a number of forms, and have been aimed at detecting all three types of biclusters.

Gene expression data is *high-dimensional*, in the sense that $p > n$. In this setting, it might be reasonable to assume that most genes do not contribute much to or differ between the biological conditions being studied, and so in a sense can be considered to be noise. A number of authors have recently suggested performing *sparse* one-way clustering of the observations in gene expression data, so that just a subset of the genes are used to cluster the observations [76, 103, 106, 109]. This can yield more accurate clusters, and also allows biologists to focus their research efforts on those selected genes.

In this chapter, we extend sparse one-way clustering to the biclustering problem. We propose sparse biclustering under the assumptions that (1) each matrix element is normally distributed with a bicluster-specific mean, and (2) the biclusters partition the rows and columns of the matrix. Our proposal can be thought of as a generalization of k-means

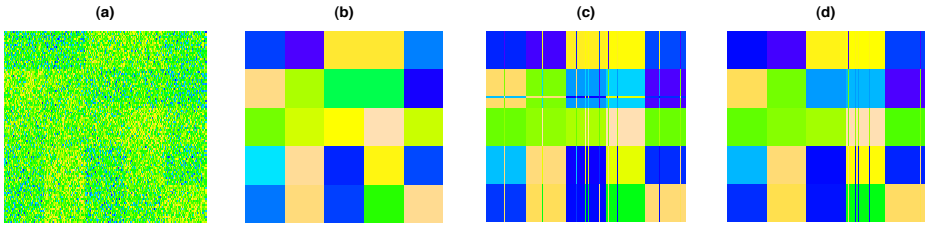

Figure 3.1: (a): A heatmap of a simulated 100×200 data set, with five row clusters and five column clusters. (b): True underlying mean signal within each cluster. (c): Mean signal estimated by independent 5-means clustering of the rows and 5-means clustering of the columns. (d): Mean signal estimated by biclustering, as described in Algorithm 2, with K=5, R=5, and λ=0. Biclustering results in more accurate clustering of both the rows and the columns than does independent 5-means clustering.

clustering to biclustering, and also a sparse and constrained version of the SVD. We can estimate the biclusters by maximizing the corresponding log likelihood. To achieve sparse biclustering, we maximize the ℓ_1-penalized log likelihood. The proposed approach is illustrated on a toy example in Figure 3.1, in which it is shown that biclustering can result in more accurate cluster discovery than independent one-way clustering of the rows and columns of a data matrix. Our approach identifies constant and contiguous biclusters, as in Table 3.1(a).

Chapter outline In Chapter 3.1, we provide a brief review of the biclustering literature. Chapter 3.2 contains our proposal for sparse biclustering, and in Chapter 3.3, we motivate our biclustering proposal further by exploring its connection with the singular value decomposition. In Chapter 3.4 we present an approach for selecting the tuning parameters associated with this proposal. In Chapter 3.5 we present the results of simulation studies, and Chapter 3.6 contains an application to a gene expression data set. We propose a more general formulation for biclustering using the matrix-variate normal distribution in Chapter 3.7. We close with a discussion in Chapter 3.8. The proofs are in Appendix **??**.

3.1 Existing Work

In the literature, biclustering proposals have taken a number of forms, and date back to at least [43]. For instance, some authors have independently clustered the rows and the columns of the data matrix, and others have suggested performing matrix factorization and examining the resulting singular vectors in order to identify biclusters. In addition, some biclustering proposals allow overlapping biclusters while some identify biclusters as contiguous block matrices. A detailed review of past proposals is outside of the scope of this chapter, but can be found in [66] and [79]. Here, we briefly review three proposals for biclustering that form the basis for comparisons in the later sections of this chapter. These three methods are included in comparisons because, like the proposal in this chapter, they assume that most elements of the data matrix take on a common mean value. If the data matrix is centered appropriately, then this leads to a sparse estimate of the mean matrix.

[55] introduced the *plaid model* for transposable data, in which $X_{ij} = \sum_{k=1}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk}$, where ρ_{ik} and κ_{jk} are binary values that equal one if the ith observation and jth variable belong to the kth bicluster. The plaid model identifies constant biclusters when $\theta_{ijk} = \mu_k$, and additive coherent biclusters result when $\theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}$. The parameters are estimated by minimizing the quantity $\sum_{i=1}^{n}\sum_{j=1}^{p}(X_{ij} - \sum_{k=1}^{K}\theta_{ijk}\rho_{ik}\kappa_{jk})^2$. [99] developed the *improved plaid* (IP) approach, an improved algorithm for this task, which is challenging due to the constraint that ρ_{ik} and κ_{jk} are binary.

More recently, [88] proposed an algorithm for finding constant biclusters, termed *large average submatrices* (LAS), using the model $X_{ij} = \sum_{k=1}^{K} \mu_k\, I_{(i,j)\in B_k} + \epsilon_{ij}$, where $I_{(i,j)\in B_k}$ is an indicator function for whether the ith row and jth column belong to the kth bicluster, μ_k is a mean term, and ϵ_{ij} is a noise term. The algorithm seeks to find a bicluster that maximizes a significance score on the residual matrix obtained by subtracting out the biclusters identified in previous iterations.

An entirely different approach based on the *singular value decomposition* (SVD) is taken by [46] and [56]. They proposed to identify multiplicative biclusters using a low-rank ap-

proximation: $\mathbf{X} \approx \sum_{k=1}^{K} s_k \mathbf{u}_k \mathbf{v}_k^T$, where s_k is a scalar and $\mathbf{u}_k$ and $\mathbf{v}_k$ are vectors of lengths n and p. [56] estimated the parameters subject to sparsity-inducing penalties on $\mathbf{u}_k$ and $\mathbf{v}_k$; we will refer to this as the *sparse SVD* (SSVD) approach. [46] imposed sparsity on the vectors $\mathbf{u}_k$ and $\mathbf{v}_k$ using a Bayesian approach. Both sets of authors declared the matrix elements corresponding to non-zero elements of $\mathbf{u}_k$ and $\mathbf{v}_k$ to make up the kth bicluster.

3.2 Sparse Biclustering

In what follows, $\mathbf{X}$ is a $n \times p$ matrix with n observations and p features. We assume that the n observations belong to K unknown and non-overlapping classes, $C_1, \ldots, C_K$, and the p features belong to R unknown and non-overlapping classes, $D_1, \ldots, D_R$.

3.2.1 An Approach for Biclustering

Assume that all matrix elements are independent, and that $X_{ij} \sim N(\mu_{kr}, \sigma^2)$ for $i \in C_k, j \in D_r$. We wish to estimate C_k, D_r, and μ_{kr} for $k = 1, \ldots, K$ and $r = 1, \ldots, R$. Maximizing the log likelihood of the data under this model is equivalent to

$$\underset{C_1,\ldots,C_K,D_1,\ldots,D_R,\boldsymbol{\mu}\in\mathbb{R}^{K\times R}}{\text{minimize}} \left\{ \sum_{k=1}^{K} \sum_{r=1}^{R} \sum_{i\in C_k} \sum_{j\in D_r} (X_{ij} - \mu_{kr})^2 \right\}, \tag{3.1}$$

which is easily seen to reduce to k-means clustering of the observations into K clusters if $R = p$, and k-means clustering of the features into R clusters if $K = n$. Note that solving (3.1) results in the discovery of KR biclusters, each of which consists of $|C_k||D_r|$ elements – namely, the observations in C_k and the features in D_r.

3.2.2 Sparse Biclustering

A shortcoming of (3.1) is that every row cluster C_k and column cluster D_r is assigned its own mean term μ_{kr}, where $\mu_{kr} \neq 0$ in general. If the data matrix $\mathbf{X}$ is centered so that its overall mean is zero, then we may suspect that some or many biclusters have a mean term that is approximately zero. In this setting, it may be worth incurring a little bit of additional bias by

estimating these mean terms to be exactly zero, in the interest of improved interpretability and reduced variance in the resulting biclusters. It is straightforward to induce sparsity on the mean elements by penalizing (3.1) using an ℓ_1 or *lasso* penalty [95]. We arrive at

$$\underset{C_1,\ldots,C_K,D_1,\ldots,D_R,\boldsymbol{\mu}\in\mathbb{R}^{K\times R}}{\text{minimize}} \left\{ \frac{1}{2}\sum_{k=1}^{K}\sum_{r=1}^{R}\sum_{i\in C_k}\sum_{j\in D_r}(X_{ij}-\mu_{kr})^2 + \lambda\sum_{k=1}^{K}\sum_{r=1}^{R}|\mu_{kr}| \right\}, \tag{3.2}$$

where λ is a nonnegative tuning parameter. As λ increases, (on average) an increasing number of μ_{kr}'s will be estimated to equal zero. If $\hat{\mu}_{kr} = 0$, then this indicates a bicluster (C_k, D_r) for which the overall mean is not substantially different from zero. We note that (3.2) can be viewed as an extension of some recent sparse one-way clustering proposals [76, 103, 109] to the biclustering setting, in the sense that if $R = p$ then we are performing sparse k-means clustering of the rows of the data matrix.

Algorithm 2 is a simple iterative approach for finding a local optimum of (3.2). It is a descent algorithm, and when $\lambda = 0$, it amounts to finding a local optimum of (3.1). We performed Algorithm 2 5,000 times on the same data matrix $\mathbf{X}$, generated as in Chapter 3.5.2, using random initializations of the row and column clusters. In 5,000 replications, the values of the objective function (3.2) were always within $\pm 0.5\%$ of the mean of the values.

We note that in the optimization problem (3.2), there is a complex interplay between the parameters K, R, and λ. For instance, when λ is extremely large, then $\mu_{kr} = 0$ for all $k = 1, \ldots, K$ and $r = 1, \ldots, R$, and so the values of $C_1, \ldots, C_K$ and $D_1, \ldots, D_R$ that minimize (3.2) are not unique. This problem can also manifest itself for more moderate values of λ. For instance, consider Step 2(a) of Algorithm 2, and suppose that $\mu_{kr} = \mu_{k'r} = 0$ for some $k \neq k'$ and for all $r = 1, \ldots, R$. Then in Step 2(b), $\sum_{r=1}^{R}\sum_{j\in D_r}(X_{ij} - \mu_{kr})^2 = \sum_{r=1}^{R}\sum_{j\in D_r}(X_{ij} - \mu_{k'r})^2$, and so C_k and $C_{k'}$ cannot be uniquely assigned. In our implementation of Algorithm 2, we address this problem when it occurs by simply merging the kth and k'th clusters, thereby reducing the total number of row clusters from K to $K - 1$. We take this approach in the interest of simplicity, though alternative procedures are possible and could lead to lower values of the objective (3.2).

Algorithm 2 Sparse Biclustering

1. Initialize $D_1, \ldots, D_R$ and $C_1, \ldots, C_K$ by performing one-way k-means clustering on the columns and on the rows of the mean-centered data matrix $\mathbf{X}$.

2. Iterate until convergence:

   (a) Holding $C_1, \ldots, C_K$ and $D_1, \ldots, D_R$ fixed, solve (3.2) with respect to $\boldsymbol{\mu}$. That is,

   $$\mu_{kr} = \frac{S(\sum_{i \in C_k} \sum_{j \in D_r} X_{ij}, \lambda)}{|C_k||D_r|}, \tag{3.3}$$

   where S is the soft-thresholding operator $S(a, b) = \text{sign}(a)(|a| - b)_+$, $|C_k|$ is the cardinality of C_k, and $|D_r|$ is the cardinality of D_r.

   (b) Holding $D_1, \ldots, D_R$ and $\boldsymbol{\mu}$ fixed, solve (3.2) with respect to $C_1, \ldots, C_K$, by assigning the ith observation to the row cluster for which $\sum_{r=1}^{R} \sum_{j \in D_r} (X_{ij} - \mu_{kr})^2$ is smallest.

   (c) Repeat Step 2(a).

   (d) Holding $C_1, \ldots, C_K$ and $\boldsymbol{\mu}$ fixed, solve (3.2) with respect to $D_1, \ldots, D_R$, by assigning the jth feature to the column cluster for which $\sum_{k=1}^{K} \sum_{i \in C_k} (X_{ij} - \mu_{kr})^2$ is smallest.

3.3 A Spectral Interpretation for Biclustering

[116] established that a relaxation of k-means clustering yields principal components analysis (PCA), or equivalently, that k-means can be interpreted as a constrained version of PCA in which the kth principal component must take on values in $\{0, \frac{1}{\sqrt{n_k}}\}$. We will now show that with $K = R$ (that is, the same number of row and column clusters), the biclustering optimization problem (3.1) can be relaxed in order to yield the SVD. We first present a lemma that provides an alternative characterization for the SVD.

Lemma 3.1. *Consider the optimization problem*

$$\underset{\mathbf{A}^T\mathbf{A}=\mathbf{I}_K, \mathbf{B}^T\mathbf{B}=\mathbf{I}_K}{\text{maximize}} ||\mathbf{A}^T\mathbf{X}\mathbf{B}||_F^2, \tag{3.4}$$

where $\mathbf{A}$ *and* $\mathbf{B}$ *are* $n \times K$ *and* $p \times K$ *orthogonal matrices and* $K \leq \min(n, p)$*. The solution is given by* $\mathbf{A} = \mathbf{U}_{1:K}\mathbf{Q}_1$ *and* $\mathbf{B} = \mathbf{V}_{1:K}\mathbf{Q}_2$*, where* $\mathbf{U}_{1:K}$ *and* $\mathbf{V}_{1:K}$ *are* $n \times K$ *and* $p \times K$ *matrices whose columns are the first* K *left and right singular vectors of* $\mathbf{X}$ *respectively, and* $\mathbf{Q}_1$ *and* $\mathbf{Q}_2$ *are any* $K \times K$ *orthogonal matrices.*

Finally, we present our theorem.

Theorem 3.1. *Consider the problem (3.4) with two additional constraints:*

1. *The elements of the* k*th column of* $\mathbf{A}$ *are 0 or* $\frac{1}{\sqrt{n_k}}$ *with* $n_k \in \mathbb{Z}^+$*,* $\sum_{k=1}^K n_k = n$*.*

2. *The elements of the* k*th column of* $\mathbf{B}$ *are 0 or* $\frac{1}{\sqrt{p_r}}$ *with* $p_r \in \mathbb{Z}^+$*,* $\sum_{r=1}^K p_r = p$*.*

This constrained version of (3.4) is equivalent to the biclustering optimization problem (3.1) with $K = R$*. Equivalently, a relaxed version of (3.1) yields the SVD.*

Theorem 3.1 elucidates the difference between performing independent k-means clustering on the rows and columns of a data matrix, and performing biclustering. For the relaxed problem, the two approaches are identical - that is, we know that performing PCA on the rows of a data matrix and PCA on the columns of a data matrix is equivalent to simply

computing the SVD of the data matrix. However, for the constrained problem, the two approaches are different, in the sense that k-means clustering and biclustering yield different solutions. Biclustering constitutes a more symmetric and systematic approach. A result closely-related to Theorem 3.1 can be found in [18].

3.4 Tuning Parameter Selection

The sparse biclustering proposal (3.2) involves three tuning parameters: the number of row clusters K, the number of column clusters R, and the sparsity parameter λ. Here we consider the problem of selecting these tuning parameters in an automated fashion.

3.4.1 Selection of K and R

In order to select K and R, we recast biclustering as a supervised learning problem, as follows. We leave out a random subset of elements from the data matrix $\mathbf{X}$, impute those left-out elements using the overall mean for the data matrix, and bicluster the resulting data matrix. We then assess the extent to which the estimated bicluster mean for the left-out elements differs from the true value of the left-out elements, using squared error loss. A related proposal appears in [107]. This approach, which assumes that λ is fixed, is described in greater detail in Algorithm 3.

In order to explore the performance of this approach for selecting K and R, we conducted a small simulation study with various values of n, p, K, and R. First, each row was randomly assigned into one of the row clusters with uniform probability, and each column was randomly assigned to one of the column clusters with uniform probability. Then, the elements of the matrix $\mathbf{X}$ were generated independently, $X_{ij} \overset{\text{i.i.d.}}{\sim} N(\mu_{kr}, 2^2)$ for $i \in C_k, j \in D_r$ where $\mu_{kr} \sim$ Unif(-3, 3). We quantified the extent to which Algorithm 3 correctly identified the values of K and R. Occasionally, Algorithm 3 may return multiple results – for instance, two results will be returned if both $(K = 3, R = 4)$ and $(K = 4, R = 3)$ satisfy the criterion in Step 3, and no pair of (K, R) for which $K + R < 7$ satisfies the criterion. In this case, we gave the algorithm "partial credit" according to the fraction of returned (K, R) pairs that are correct.

Algorithm 3 Selecting Number of Row Clusters K and Column Clusters R

1. Repeat the following procedure T times:

   (a) Let $\mathcal{M}$ denote a set containing np/T elements of the form (i, j), where (i, j) is drawn uniformly at random from $\{(1, 1), (1, 2), \ldots, (n, p)\}$.

   (b) Construct a new $n \times p$ matrix, $\mathbf{X}^*$, for which the elements in $\mathcal{M}$ are "missing" and are imputed using the mean of the non-missing values:

$$X^*_{ij} = \begin{cases} X_{ij} & \text{if} \quad (i,j) \in \mathcal{M}^c \\ \sum_{(i,j)\in\mathcal{M}^c} X_{ij}/|\mathcal{M}^c| & \text{if} \quad (i,j) \in \mathcal{M} \end{cases}. \tag{3.5}$$

   (c) For each pair of values (K, R) of interest:

      i. Perform sparse biclustering of $\mathbf{X}^*$ with K row and R column clusters.

      ii. Construct a $n \times p$ matrix $\mathbf{A}$ whose (i, j)th element equals the estimated value of μ_{kr}, where $i \in C_k$ and $j \in D_r$.

      iii. Calculate the mean squared error that results from estimating the "missing" elements using the corresponding bicluster means,

$$\sum_{(i,j)\in\mathcal{M}} (X_{ij} - A_{ij})^2/|\mathcal{M}|. \tag{3.6}$$

2. For each pair of values (K, R) that was considered in Step 1(c), compute $m_{K,R}$, the mean of the quantity (3.6) across all T iterations, as well as $s_{K,R}$, its standard error.

3. Identify the pairs (K, R) for which $m_{K,R} \leq m_{K+1,R+1} + s_{K+1,R+1}$.

4. Select the (K, R) from Step 3 for which $K + R$ is smallest.

Results are in Table 3.2.

Table 3.2: Simulation study to evaluate the performance of Algorithm 3 for tuning parameter selection. Results are reported over 50 simulated data sets. We report the overall accuracy, that is, the proportion of the data sets for which the correct values of both K and R were identified. We also report the mean (and standard errors) of the K and R values obtained.

True value of (K, R)	n	p	Overall Accuracy	Selected K	Selected R
$(K = 2, R = 4)$	100	100	56%	2 (0)	3.48 (0.0914)
		500	66%	2 (0)	3.60 (0.0857)
	500	100	70%	2 (0)	3.68 (0.0725)
		500	94%	2 (0)	3.94 (0.0339)
$(K = 6, R = 3)$	100	100	44%	5.26 (0.1100)	3 (0.0286)
		500	74%	5.7 (0.0769)	3 (0)
	500	100	68%	5.68 (0.0666)	3 (0)
		500	94%	5.92 (0.0481)	3 (0)

3.4.2 Selection of λ

We now assume that K and R are known, or else were already selected using Algorithm 3 with $\lambda = 0$. We select λ using an approach motivated by BIC. For a given value of λ, we perform sparse biclustering, and create a $(np) \times (q+1)$ design matrix, where q is equal to the number of non-zero $\hat{\mu}_{kr}$'s in the sparse biclustering output. The first column is a vector of 1's corresponding to an intercept, and the remaining columns contain 1's and 0's, indicating whether a given element of the matrix is part of the corresponding non-zero-mean bicluster in the sparse biclustering output. We fit a least squares regression model that uses this design matrix to predict the matrix elements, and compute BIC using the formula

$$\text{BIC} = np \times \log(\text{RSS}) + np \log(q)$$

where RSS is the usual residual sum of squares. We then select the value of λ that leads to the smallest value of BIC.

3.5 Simulation Studies

We compared the performance of our biclustering proposal to independent one-way k-means clustering of the rows and columns in a simulation setting with constant and contiguous non-zero biclusters (Simulation 1). In addition, we compared our biclustering proposal to a number of competitors under three simulation settings: in Simulation 2 there are constant and contiguous biclusters with some of the bicluster means exactly equal to zero, in Simulation 3 there are multiplicative biclusters, and in Simulation 4 there are overlapping biclusters.

3.5.1 Biclustering Methods Used in Our Comparisons

We compared the following biclustering methods, which were discussed in Chapters 3.1 and 3.2.

1. Independent one-way k-means clustering of the rows and of the columns.

2. Sparse biclustering using Algorithm 2, with several values of λ.

3. IP [99], which is a variant of the plaid model [55], using the `R` package `biclust` available on `CRAN` [52].

4. SSVD [56], using the `R` package `s4vd`, available on `CRAN` [89].

5. LAS [88], using Matlab code available at `https://genome.unc.edu/las/`.

3.5.2 Simulation 1: No Bicluster Means Exactly Equal Zero

We created $K = 4$ row clusters and $R = 5$ column clusters by randomly assigning each row to a row cluster and each column to a column cluster with uniform probability. We

generated a $n \times p$ data matrix $\mathbf{X}$, according to $X_{ij} \overset{\text{i.i.d.}}{\sim} N(\mu_{kr}, 4^2)$ for $i \in C_k, j \in D_r$, where $\mu_{kr} \sim \text{Unif}(-2, 2)$. Then, we mean-centered the matrix $\mathbf{X}$. We performed independent one-way k-means clustering on the rows and on the columns of the matrix, as well as sparse biclustering with various values of λ, as well as with λ selected automatically as described in Chapter 3.4.2.

The *clustering error rate* (CER; see, for example, [17, 106]) measures the disagreement between the true and estimated cluster labels. It is one minus the Rand index [81]. A high value of CER indicates disagreement between the true and estimated clusters, and a value of zero indicates perfect agreement. We used the CER to compare the estimated row and column clusters to the true row and column clusters. We defined the *sparsity rate* to be the fraction of the $\hat{\mu}_{kr}$'s that exactly equal zero, and we defined the *sparsity error rate* to be the proportion of $\hat{\mu}_{kr}$'s that were incorrectly set to zero or incorrectly set to be non-zero.

Results are reported in Table 3.3. We see that biclustering with $\lambda = 0$ leads to consistently better results than independent clustering of the rows and columns.

3.5.3 Simulation 2: Some Bicluster Means Exactly Equal Zero

We modified Simulation 1 so that $\mu_{kr} \sim \text{Unif}[(-2.5, -1.5) \cup (1.5, 2.5)]$ or $\mu_{kr} = 0$ with equal probability. We compared sparse biclustering with several competitors as described in Chapter 3.5.1:

- For IP, we used the `R` package `biclust` to identify constant biclusters, with a background layer, and with *row* and *column release* parameters set to 0.5 as in [99].

- For LAS, we used the default settings in the Matlab code. We discarded biclusters with a significance-based score below one, as those tend to contain the entire matrix.

- For SSVD, we obtained a rank-1 through rank-4 approximation using the `R` package `sv4d`; note that in our simulation set-up, the rank of the true underlying mean matrix

Table 3.3: Results from one-way k-means clustering and sparse biclustering for Simulation 1 with $n = 200$, over 50 simulated data sets. We report the mean (and standard error) of the CER of the rows and columns, and the mean (and standard error) of the sparsity rate. Note that $\bar{\lambda}$ is the mean of λ selected across 50 simulations using the approach of Chapter 3.4.2. The correct values of K and R were used, since CER is not comparable across different numbers of clusters.

p	Method	Row CER	Column CER	Sparsity Rate
200	k-means	0.0873 (0.0079)	0.1055 (0.0078)	-
	Bicluster λ=0	0.0547 (0.0066)	0.0559 (0.0056)	-
	Bicluster λ=200	0.0520 (0.0053)	0.0575 (0.0057)	0.0779 (0.0071)
	Bicluster λ=400	0.0589 (0.0063)	0.0699 (0.0065)	0.1665 (0.0111)
	Bicluster λ=800	0.0865 (0.0091)	0.0971 (0.0078)	0.2588 (0.0127)
	Bicluster $\bar{\lambda} = 320$	0.0534 (0.0057)	0.0644 (0.0063)	0.1338 (0.0110)
500	k-means	0.0254 (0.0048)	0.0755 (0.0061)	-
	Bicluster λ=0	0.0108 (0.0034)	0.0474 (0.0043)	-
	Bicluster λ=200	0.0109 (0.0032)	0.0475 (0.0044)	0.0237 (0.0052)
	Bicluster λ=400	0.0095 (0.0031)	0.0478 (0.0042)	0.0560 (0.0061)
	Bicluster λ=800	0.0122 (0.0034)	0.0557 (0.0051)	0.1158 (0.0089)
	Bicluster $\bar{\lambda} = 442$	0.0100 (0.0032)	0.0480 (0.0043)	0.0891 (0.009)

is four. Sparsity parameters were selected using BIC. The adaptive weight parameters were set to two as in [56]. Only the best results obtained are reported.

We quantify the success of the approaches via the proportion of zero elements in the underlying mean matrix that are correctly identified (correct zeros), and the proportion of non-zero elements in the underlying mean matrix that are correctly identified (correct non-zeros). We also report sparsity rate and sparsity error rate as defined in Chapter 3.5.2. Finally, for one-way k-means clustering and for our sparse biclustering proposal, we report row and column CER; we do not report this for the other competitors, since they do not provide a partition of the rows and columns, and instead simply identify (possibly overlapping) hotspots in the matrix.

The results are presented in Table 3.4. We see that a substantial benefit is obtained by performing sparse biclustering rather than one-way k-means clustering, in terms of CER. Now, we discuss the performance of various biclustering methods in terms of proportion of correctly identified zeros and non-zeros, and also the sparsity error rate. We see from Table 3.4 that IP fails to identify any biclusters in this simulation set-up. This is due to the fact that the signal-to-noise ratio in this setting is too low; in related simulation set-ups with a higher signal-to-noise ratio, IP's performance is improved. SSVD and LAS perform comparably in this setting, and by far the best overall performance is achieved by our sparse biclustering proposal with a large value of λ. For instance, when $\lambda = 1000$ and $p = 200$, the sparsity error rate is only 14.2%.

Table 3.4: Results of various competitors in Simulation 2 with $n = 200$. We report the mean (and standard error) over 50 simulated data sets of the CER of the rows and columns, proportion of correctly identified zeros and non-zeros, sparsity rate, and sparsity error rate. Note that $\bar{\lambda}$ is the mean of λ selected across 50 simulations using the approach of Section 3.4.2.

p	Method	Row CER	Column CER	C. Zeros	C. Non-zeros	Sparsity Rate	Sparsity Error Rate
200	k-means	0.0460 (0.009)	0.0725 (0.008)	-	-	-	-
	Bicluster λ=0	0.0306 (0.008)	0.0434 (0.007)	-	-	-	-
	Bicluster λ=200	0.0289 (0.007)	0.0425 (0.007)	0.264 (0.035)	0.994 (0.002)	0.135 (0.018)	0.372 (0.021)
	Bicluster λ=500	0.0313 (0.008)	0.0482 (0.007)	0.574 (0.053)	0.985 (0.004)	0.295 (0.028)	0.217 (0.025)
	Bicluster λ=1000	0.0552 (0.010)	0.0723 (0.009)	0.749 (0.042)	0.962 (0.007)	0.392 (0.238)	0.142 (0.022)
	Bicluster $\bar{\lambda}$=475	0.0292 (0.007)	0.0456 (0.007)	0.684 (0.053)	0.987 (0.002)	0.345 (0.028)	0.166 (0.026)
	IP	-	-	1.000 (0.000)	0.000 (0.000)	1.000 (0.000)	0.498 (0.020)
	SSVD rank-2	-	-	0.683 (0.047)	0.489 (0.052)	0.609 (0.048)	0.388 (0.017)
	LAS	-	-	0.366 (0.008)	0.932 (0.004)	0.217 (0.007)	0.353 (0.012)
500	k-means	0.0168 (0.005)	0.0494 (0.007)	-	-	-	-
	Bicluster λ=0	0.0100 (0.004)	0.0375 (0.006)	-	-	-	-
	Bicluster λ=200	0.0097 (0.004)	0.0374 (0.006)	0.127 (0.028)	0.998 (0.001)	0.063 (0.013)	0.440 (0.021)
	Bicluster λ=500	0.0103 (0.004)	0.0379 (0.006)	0.287 (0.045)	0.995 (0.001)	0.151 (0.025)	0.354 (0.024)
	Bicluster λ=1000	0.0112 (0.004)	0.0401 (0.007)	0.511 (0.058)	0.994 (0.001)	0.261 (0.032)	0.244 (0.028)
	Bicluster $\bar{\lambda}$=663	0.0098 (0.004)	0.0383 (0.006)	0.530 (0.059)	0.994 (0.0013)	0.264 (0.029)	0.242 (0.029)
	IP	-	-	1.000 (0.000)	0.000 (0.000)	1.000 (0.000)	0.498 (0.020)
	SSVD rank-2	-	-	0.594 (0.045)	0.623 (0.043)	0.503 (0.044)	0.373 (0.016)
	LAS	-	-	0.443 (0.011)	0.953 (0.004)	0.244 (0.008)	0.305 (0.013)

3.5.4 Simulation 3: Multiplicative Biclusters

This simulation study is adapted from [56]. Let $\mathbf{M} = d\mathbf{u}_1\mathbf{v}_1^T$ be a 100×50 matrix with $d = 50$,

$$\tilde{\mathbf{u}}_1 = [10, 9, 8, 7, 6, 5, 4, 3, r(2, 17), r(0, 75)]^T,$$

$$\tilde{\mathbf{v}}_1 = [10, -10, 8, -8, 5, -5, r(3, 5), r(-3, 5), r(0, 34)]^T,$$

$\mathbf{u}_1 = \tilde{\mathbf{u}}_1/\|\tilde{\mathbf{u}}_1\|_2$, and $\mathbf{v}_1 = \tilde{\mathbf{v}}_1/\|\tilde{\mathbf{v}}_1\|_2$, where $r(a, b)$ denotes a vector of length b with all entries equal a. Then, let $\mathbf{X} = \mathbf{M} + \epsilon$ where $\epsilon_{ij} \overset{\text{i.i.d.}}{\sim} N(0, 1)$. Figures 3.2(a)-(b) display the data matrix $\mathbf{X}$ and the underlying mean matrix $\mathbf{M}$. As mentioned in [56], this is a challenging biclustering problem since some non-zero entries in $\mathbf{M}$ are small relative to the noise. In particular, this setting is challenging for our sparse biclustering proposal, due to the presence of multiplicative biclusters, as opposed to the contiguous constant bicluster setting for which our proposal is intended.

We performed sparse biclustering with K, R automatically selected using Algorithm 3, and with various values of λ. For IP, LAS, and SSVD, the tuning parameters used are as in Chapter 3.5.3 unless specified otherwise. For IP, we set the `R` package `biclust` to identify the most flexible model discussed in [55], and ran the algorithm without the background layer. For SSVD, we set the parameters in the `R` package `s4vd` such that one bicluster is identified.

The results (averaged over 100 simulations) are summarized in Table 3.5. It is not surprising that SSVD has the best results in this simulation set-up, as in this set-up there are multiplicative biclusters. Though they have low sparsity error rates, both IP and LAS fail to correctly identify most of the non-zero elements in the underlying mean matrix. It is not surprising that LAS performs poorly in this simulation set-up, as LAS was developed to identify constant biclusters.

How does sparse biclustering perform in this setting, which clearly violates the constant and contiguous bicluster model? Sparse biclustering with $\lambda = 0$ has a sparsity error rate of 0.92, due to the fact that when $\lambda = 0$, all elements in the estimated mean matrix are non-

zero. However, for a moderate value of λ, sparse biclustering performs well, even though it is designed to identify contiguous constant biclusters. This is because the multiplicative bicluster in Figure 3.2(b) can be approximated as the *union of a number of constant biclusters*. Therefore, sparse biclustering leads to Figure 3.2(c), which is a very accurate approximation of Figure 3.2(b). In particular, Figure 3.2(c) resulted from our sparse biclustering proposal with $K = 3$ and $R = 5$; note that these values were selected automatically by Algorithm 3.

Table 3.5: Results for Simulation 3, averaged over 100 simulated data sets. For sparse biclustering, K and R were automatically chosen using Algorithm 3. Note that $\bar{\lambda}$ is the mean of λ selected across 100 simulations using the approach of Chapter 3.4.2. Standard errors are in parentheses.

Method	Sparsity Rate	C. Zeros	C. Non-zeros	Sparsity Error Rate
Bicluster λ=0	0.000 (0.000)	0.000 (0.000)	1.000 (0.000)	0.920 (0.000)
Bicluster λ=80	0.829 (0.012)	0.895 (0.013)	0.940 (0.005)	0.101 (0.012)
Bicluster λ=90	0.872 (0.009)	0.944 (0.010)	0.951 (0.005)	0.056 (0.009)
Bicluster λ=100	0.878 (0.014)	0.950 (0.015)	0.955 (0.005)	0.050 (0.013)
Bicluster λ=110	0.804 (0.024)	0.871 (0.025)	0.963 (0.004)	0.122 (0.023)
Bicluster $\bar{\lambda} = 11.6$	0.310 (0.029)	0.336 (0.032)	0.986 (0.004)	0.612 (0.029)
SSVD	0.886 (0.002)	0.963 (0.002)	0.997 (0.001)	0.034 (0.002)
IP	0.972 (0.001)	0.997 (0.001)	0.307 (0.008)	0.059 (0.001)
LAS	0.920 (0.002)	0.963 (0.002)	0.575 (0.009)	0.068 (0.002)

3.5.5 Simulation 4: Overlapping Multiplicative Biclusters

In this section, we investigate an example with overlapping multiplicative biclusters. Let $\mathbf{M} = d\mathbf{u}_1\mathbf{v}_1^T + d\mathbf{u}_2\mathbf{v}_2^T$ be a 100×50 matrix with $d = 50$, $\mathbf{u}_1, \mathbf{v}_1$ as defined in Chapter 3.5.4,

$$\tilde{\mathbf{u}}_2 = [r(0,13), 10, 9, 8, 7, 6, 5, 4, 3, r(2,17), r(0,62)]^T,$$

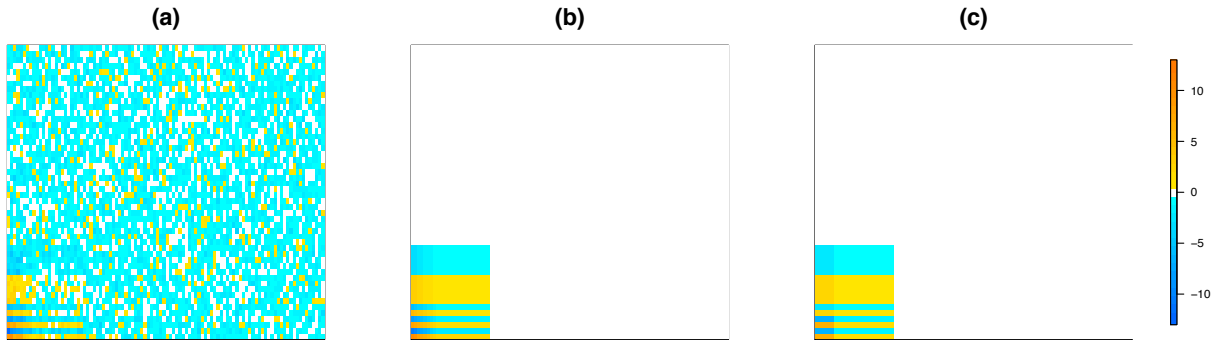

Figure 3.2: Heatmaps of (a): data matrix, generated according to Simulation 3. (b) Underlying means used to generate data. (c) Mean matrix estimated by sparse biclustering, with K and R automatically chosen ($K = 3$, $R = 5$) and $\lambda = 10$; 84% of the elements are estimated to equal zero.

$$\tilde{\mathbf{v}}_2 = [r(0,9), 10, -9, 8, -7, 6, -5, r(4,5), r(-3,5), r(0,25)]^T,$$

$\mathbf{u}_2 = \tilde{\mathbf{u}}_2/\|\tilde{\mathbf{u}}_2\|$, and $\mathbf{v}_2 = \tilde{\mathbf{v}}_2/\|\tilde{\mathbf{v}}_2\|$. Then, let $\mathbf{X} = \mathbf{M} + \epsilon$ where $\epsilon_{ij} \overset{\text{i.i.d.}}{\sim} N(0,1)$. Heatmaps of $\mathbf{X}$ and $\mathbf{M}$ are shown in Figures 3.3(a)-(b).

We performed the biclustering methods described in the previous section, with the SSVD parameters set to identify two biclusters. We expect SSVD to perform well in this set-up, since there are multiplicative overlapping biclusters. In contrast, sparse biclustering's assumption of constant and non-overlapping biclusters is clearly violated. Nonetheless, sparse biclustering performs competitively (Table 3.6), since the multiplicative and overlapping biclusters can be very accurately approximated using sparse biclustering using a sufficiently large value of K and R (Figure 3.3(c)). A similar fact was noted in [38].

3.6 Application to a Gene Expression Data Set

In this section, we consider a lung cancer gene expression data set previously analyzed by [56] and [63], consisting of measurements for 56 samples and 12,625 genes. 17 samples correspond to normal subjects, 20 correspond to subjects with pulmonary carcinoid tumors, 13 correspond to colon metastases, and six correspond to small cell carcinomas. We selected

Table 3.6: Results for Simulation 4. Details are as in Table 3.5.

Method	Sparsity Rate	C. Zeros	C. Non-zeros	Sparsity Error Rate
Bicluster λ=40	0.648 (0.020)	0.718 (0.023)	0.775 (0.007)	0.274 (0.019)
Bicluster λ=60	0.770 (0.018)	0.849 (0.021)	0.706 (0.007)	0.171 (0.017)
Bicluster λ=80	0.813 (0.016)	0.895 (0.017)	0.679 (0.007)	0.136 (0.015)
Bicluster λ=100	0.859 (0.012)	0.950 (0.014)	0.687 (0.004)	0.088 (0.011)
Bicluster λ=120	0.823 (0.009)	0.915 (0.010)	0.727 (0.006)	0.112 (0.009)
Bicluster $\bar{\lambda} = 12.2$	0.262 (0.021)	0.294 (0.024)	0.928 (0.006)	0.616 (0.020)
SSVD	0.792 (0.008)	0.897 (0.006)	0.834 (0.028)	0.112 (0.004)
IP	0.944 (0.012)	0.995 (0.001)	0.358 (0.007)	0.097 (0.001)
LAS	0.877 (0.002)	0.963 (0.002)	0.634 (0.005)	0.084 (0.002)

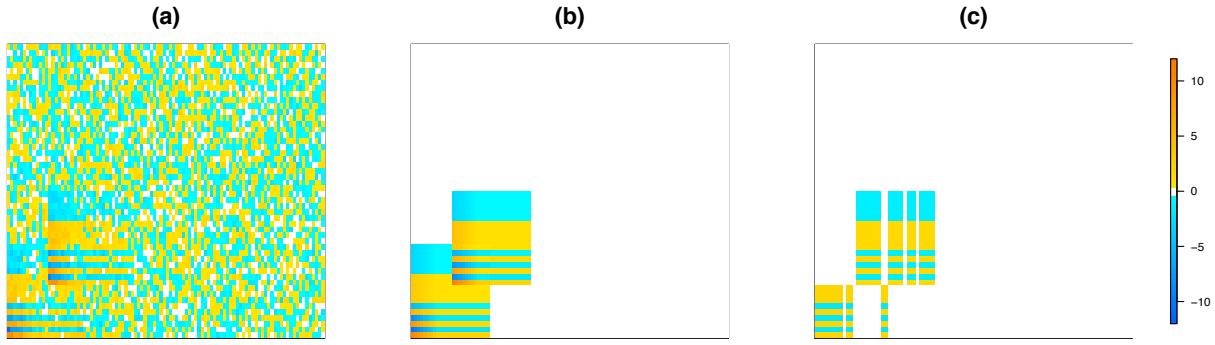

Figure 3.3: Heatmaps of (a): data matrix, generated according to Simulation 4. (b) Underlying means used to generate data. (c) Mean matrix estimated by sparse biclustering, with K and R automatically chosen ($K = 3$, $R = 6$) and $\lambda = 70$; 88% of the elements are exactly equal to zero.

5,000 genes with largest variance, and we mean-centered the 56×5000 data matrix. The goal is to discover sets of genes whose expression differs from the baseline in a subset of the patients.

We performed sparse biclustering using $K = 4$ (which we know to be the true number of row clusters), $R = 10$, and $\lambda = 1500$. A heatmap of the resulting estimated mean matrix is shown in Figure 3.4. For visualization purposes, we reordered the genes based on the estimated clusters to which they belong. From Figure 3.4, we see that one subject with small cell carcinoma is assigned to a cluster of pulmonary carcinoid tumors via sparse biclustering. Imposing sparsity in estimating the bicluster means provides substantial benefits in interpretation of the image plot, as $\hat{\mu}_{kr} = 0$ for many values of k and r. Furthermore, we see from Figure 3.4 that there is substantial variation among the estimated bicluster means. For instance, the genes in the second column cluster have a very large mean value in normal patients and a very small mean value in carcinoid patients.

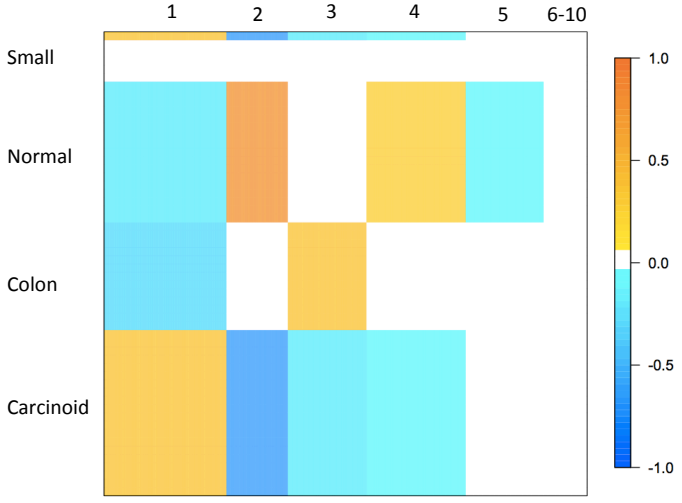

Figure 3.4: Heatmap of the estimated mean matrix from sparse biclustering using $K = 4$, $R = 10$, and $\lambda = 1500$ on a subset of the lung cancer data set consisting of the 5,000 genes with highest variance. The rows are ordered by true cancer subtype. The genes are reordered based on the estimated clusters for visualization purposes. The column labels are the gene clusters. Note that all elements in column clusters 6-10 are estimated to equal zero.

The estimated mean matrix shown in Figure 3.4 is similar to the three image plots obtained using SSVD in [56]. This is not surprising, since our biclustering proposal can be interpreted as a constrained version of the SVD (see Chapter 3.3). However, SSVD has a major interpretational disadvantage relative to our proposal: whereas sparse biclustering explicitly returns cluster labels for both the rows and columns of the data matrix, the SSVD instead returns a series of sparse singular vectors. The analyst must then take a *post hoc* approach to interpret these singular vectors in order to determine the row and column clusters. In other words, SSVD does not directly output a single interpretable figure as in Figure 3.4.

We note that Algorithm 3 led to selection of $K = 5$ and $R = 25$ on this example. One of these row clusters contains just a single subject, and the others correspond perfectly to the subjects' cancer types. Here we reported results using $R = 10$ instead of $R = 25$ for simplicity; however, using $R = 25$, a figure that is qualitatively very similar to Figure 3.4 emerges.

3.7 Matrix-variate Normal Biclustering

Recently, proposals have emerged to use the matrix-variate normal distribution to model high-dimensional transposable data [2, 41]. To indicate that a $n \times p$ data matrix $\mathbf{X}$ has a matrix-variate normal distribution, we write

$$\mathbf{X} \sim MVN(\mathbf{A}, \mathbf{\Sigma}, \mathbf{\Delta}), \tag{3.7}$$

where $\mathbf{A}$ is a $n \times p$ matrix containing the mean for each element of $\mathbf{X}$, $\mathbf{\Sigma}$ is a $n \times n$ covariance matrix for the rows of $\mathbf{X}$, and $\mathbf{\Delta}$ is a $p \times p$ covariance matrix for the columns of $\mathbf{X}$.

A consequence of the matrix-variate normal model (3.7) is that the rows and columns of $\mathbf{X}$ are marginally multivariate normal. For instance, letting $\mathbf{X}_i$ and $\mathbf{A}_i$ be the ith rows of $\mathbf{X}$ and $\mathbf{A}$, respectively, then

$$\mathbf{X}_i \sim N(\mathbf{A}_i, \Sigma_{ii}\mathbf{\Delta}). \tag{3.8}$$

We note that in the case $\mathbf{\Sigma} = \mathbf{\Delta} = \mathbf{I}$, this model reduces to $X_{ij} \overset{\text{i.i.d.}}{\sim} N(0, 1)$.

3.7.1 General Formulation of Matrix-variate Normal Biclustering

Now assume that the $n \times p$ data matrix $\mathbf{X}$ is drawn from a matrix-variate normal distribution of the form (3.7) and that $\mathbf{A}$ has *constant biclusters*: that is, for all $i \in C_k$ and $j \in D_r$, $A_{ij} = \mu_{kr}$. Without loss of generality, suppose that the rows and columns are ordered such that $k < k'$, $i \in C_k$, and $i' \in C_{k'}$ implies that $i < i'$, and similarly $r < r'$, $j \in D_r$, and $j' \in D_{r'}$ implies that $j < j'$. In other words, we use the model

$$\mathbf{X} \sim \text{MVN}\left(\begin{pmatrix} (\mu_{11}) & \dots & (\mu_{1R}) \\ \vdots & \ddots & \vdots \\ (\mu_{K1}) & \dots & (\mu_{KR}) \end{pmatrix}, \mathbf{\Sigma}, \mathbf{\Delta}\right), \tag{3.9}$$

where (μ_{kr}) is a $|C_k| \times |D_r|$ matrix, all of whose elements equal μ_{kr}. This is a natural formulation for biclustering since it easily accommodates constant biclusters as well as arbitrary row and column covariances. Fitting the model (3.9) requires estimating the $n \times n$ matrix $\mathbf{\Sigma}$ and the $p \times p$ matrix $\mathbf{\Delta}$ using the $n \times p$ matrix $\mathbf{X}$; a proposal to do this using ℓ_1 or ℓ_2 penalties is presented in [2].

A further simplification to the model (3.9) is natural. Though we might expect correlation between observations within a row cluster, or between features within a column cluster, correlations between observations in two different row clusters or between features in two different column clusters are less easily interpreted. This leads to the model

$$\mathbf{X} \sim \text{MVN}\left(\begin{pmatrix} (\mu_{11}) & \dots & (\mu_{1R}) \\ \vdots & \ddots & \vdots \\ (\mu_{K1}) & \dots & (\mu_{KR}) \end{pmatrix}, \begin{pmatrix} \mathbf{\Sigma}_1 & & \\ & \ddots & \\ & & \mathbf{\Sigma}_K \end{pmatrix}, \begin{pmatrix} \mathbf{\Delta}_1 & & \\ & \ddots & \\ & & \mathbf{\Delta}_R \end{pmatrix}\right), \tag{3.10}$$

where $\mathbf{\Sigma}$ and $\mathbf{\Delta}$ are now block diagonal with blocks of dimension $|C_1| \times |C_1|, \dots, |C_K| \times |C_K|$ and $|D_1| \times |D_1|, \dots, |D_R| \times |D_R|$, respectively. The formulation (3.10) is attractive not only because it provides a natural model for biclustering, but also because it has as special cases some well-known formulations for one-way clustering. In particular, consider (3.10) with $R = p$ and $\mathbf{\Sigma}_k = \mathbf{I}$ for $k = 1, \dots, K$. Then (3.10) amounts to a simple and well-studied model in which all observations come from a multivariate normal distribution with a common

diagonal covariance matrix and a cluster-specific mean vector [34]. If furthermore $\boldsymbol{\Delta} = \sigma^2\mathbf{I}$, then this amounts to the usual formulation for one-way k-means clustering. By symmetry of the matrix normal distribution, (3.10) also reduces to model-based clustering or k-means clustering of the columns. Note that if we assume that $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$ and $\boldsymbol{\Delta} = \mathbf{I}$, then this corresponds to our proposal in Chapter 3.2.

3.7.2 Sparse Matrix-variate Normal Biclustering

The log likelihood corresponding to (3.10) takes the form

$$l(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Delta}) = \frac{p}{2}\sum_{k=1}^{K}\log|\boldsymbol{\Sigma}_k^{-1}| + \frac{n}{2}\sum_{r=1}^{R}\log|\boldsymbol{\Delta}_r^{-1}| - \frac{1}{2}\sum_{k=1}^{K}\sum_{r=1}^{R}\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}(\mathbf{X}_{k,r}-\mu_{kr})\boldsymbol{\Delta}_r^{-1}(\mathbf{X}_{k,r}-\mu_{kr})^T), \tag{3.11}$$

where $\mathbf{X}_{k,r}$ is a $|C_k| \times |D_r|$ submatrix of $\mathbf{X}$ that consists of the elements X_{ij} for $i \in C_k$ and $j \in D_r$. We would like to fit the model (3.10) by maximizing (3.11). However, two problems arise. First, the maximum likelihood estimates of $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Delta}_r$ may be singular. Second, we may want to encourage sparsity in μ_{kr}. To address these two points, we propose to maximize the penalized log likelihood

$$\begin{aligned} l_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Delta}) &= -\frac{1}{2}\sum_{k=1}^{K}\sum_{r=1}^{R}\mathrm{tr}\left(\boldsymbol{\Sigma}_k^{-1}(\mathbf{X}_{k,r}-\mu_{kr})\boldsymbol{\Delta}_r^{-1}(\mathbf{X}_{k,r}-\mu_{kr})^T\right) + \frac{p}{2}\sum_{k=1}^{K}\log|\boldsymbol{\Sigma}_k^{-1}| \\ &+ \frac{n}{2}\sum_{r=1}^{R}\log|\boldsymbol{\Delta}_r^{-1}| - \lambda\sum_{k=1}^{K}\sum_{r=1}^{R}|\mu_{kr}| - \alpha\sum_{k=1}^{K}||\boldsymbol{\Sigma}_k^{-1}||^d - \beta\sum_{r=1}^{R}||\boldsymbol{\Delta}_r^{-1}||^d. \end{aligned} \tag{3.12}$$

Here, α, β, and λ are nonnegative parameters that determine the extent of penalization. We take $d = 1$ or $d = 2$. The last two terms in (3.12) can be understood as $||\mathbf{W}||^d = \sum_{i,j}|W_{ij}|^d$.

To maximize (3.12), we take an iterative approach in which we update the parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Delta}$, $C_1, \ldots, C_K$, $D_1, \ldots, D_R$ sequentially, holding all other parameters fixed as we update the current set of parameters. We begin with two simple lemmas.

Lemma 3.2. *With $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$, $\boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_R^{-1}$, $C_1, \ldots, C_K$, and $D_1, \ldots, D_R$ held fixed, then maximizing (3.12) with respect to $\boldsymbol{\mu}$ results in the update*

$$\mu_{kr} = S\left(\frac{\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{X}_{k,r}^T)}{\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{1}^T)}, \frac{\lambda}{\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{1}^T)}\right), \tag{3.13}$$

where $\mathbf{1}$ *is a* $|C_k| \times |D_r|$ *matrix comprised solely of 1's, and* S *is the soft-thresholding operator.*

Lemma 3.3. *With* $\boldsymbol{\mu}$, $\boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_R^{-1}$, $C_1, \ldots, C_K$, *and* $D_1, \ldots, D_R$ *held fixed, maximizing (3.12) with respect to* $\boldsymbol{\Sigma}_k^{-1}$ *reduces to*

$$\underset{\boldsymbol{\Sigma}_k^{-1}}{\text{maximize}} \left\{ \log |\boldsymbol{\Sigma}_k^{-1}| - \text{tr}(\boldsymbol{\Sigma}_k^{-1} \mathbf{S}_k) - (2\alpha/p)||\boldsymbol{\Sigma}_k^{-1}||^d \right\} \tag{3.14}$$

where $\mathbf{S}_k = \frac{1}{p} \sum_{r=1}^{R} (\mathbf{X}_{k,r} - \mu_{kr}) \boldsymbol{\Delta}_r^{-1} (\mathbf{X}_{k,r} - \mu_{kr})^T$.

Note that if $d = 1$, the graphical lasso algorithm [35] can be used to solve (3.14), and the estimate for $\boldsymbol{\Sigma}_k^{-1}$ will be sparse if the tuning parameter α is sufficiently large. When $d = 2$, then a simple analytical solution in terms of the eigenvectors and eigenvalues of $\mathbf{S}_k$ is available [105]. A similar approach can be used to solve (3.12) with respect to $\boldsymbol{\Delta}_r^{-1}$, with $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$ held fixed.

In order to update $C_1, \ldots, C_K$ with $D_1, \ldots, D_R$, $\boldsymbol{\Delta}^{-1}$, $\boldsymbol{\Sigma}^{-1}$, and $\boldsymbol{\mu}$ held fixed, we note that by (3.8), the ith row of $\mathbf{X}$ has a multivariate normal distribution given by

$$\mathbf{X}_i \sim N(\boldsymbol{\mu}_k, \Sigma_{ii} \boldsymbol{\Delta}) \tag{3.15}$$

if that observation belongs to the kth cluster. In (3.15), $\boldsymbol{\mu}_k$ is a p-vector whose jth element equals μ_{kr} if $j \in D_r$. So we update the row cluster of the ith observation by assigning that observation to the class for which the log likelihood resulting from (3.15) is largest. We note that this approach for updating the row clusters is not completely rigorous, since we are assigning each observation to a new row cluster without regard to the covariance structure among the rows. In particular, this approach is not guaranteed to increase the log likelihood, but performs well empirically. A similar approach is taken to update the column clusters.

The steps just described for maximizing (3.12) are summarized in Algorithm 4. Although Steps 2(b) and 2(d) in Algorithm 4 could potentially lead to a decrease in (3.12), in our experience, the algorithm tends to converge within 35 iterations in the simulation set-up of Chapter 3.7.3.

Algorithm 4 Matrix-variate Normal Biclustering

1. Initialize $C_1, \ldots, C_K$, $D_1, \ldots, D_R$, $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$, $\boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_R^{-1}$, $\boldsymbol{\mu}$.

2. Iterate until convergence or until a fixed number of iterations is reached:

   (a) Holding $C_1, \ldots, C_K$ and $D_1, \ldots, D_R$ fixed, perform the following updates:

      i. Holding $\boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\Delta}^{-1}$ fixed, update $\boldsymbol{\mu}$ using (3.13).

      ii. Holding $\boldsymbol{\mu}$ and $\boldsymbol{\Delta}^{-1}$ fixed, update $\boldsymbol{\Sigma}_k^{-1}$ as in Lemma 3.3 for $k = 1, \ldots, K$.

      iii. Holding $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}^{-1}$ fixed, update $\boldsymbol{\Delta}_r^{-1}$ as in Lemma 3.3 for $r = 1, \ldots, R$.

   (b) Holding $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$, $\boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_R^{-1}$, $\boldsymbol{\mu}$, and $D_1, \ldots, D_R$ fixed, update the row clustering. To do this, iterate through the rows and assign each row to the row cluster for which the log likelihood resulting from (3.15) is largest.

   (c) Repeat Step 2(a).

   (d) Holding $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$, $\boldsymbol{\Delta}_1^{-1}, \ldots, \boldsymbol{\Delta}_R^{-1}$, $\boldsymbol{\mu}$, and $C_1, \ldots, C_K$ fixed, update the column clustering, as in Step 2(b), with the roles of the rows and columns reversed.

3.7.3 A Simulation Study

We created $K = 4$ row and $R = 5$ column clusters by randomly assigning each row to a row cluster with uniform probability, and each column to a column cluster with uniform probability. We generated a $n \times p$ mean matrix $\mathbf{A}$ as follows: for each $i \in C_k$ and $j \in D_r$, $A_{ij} = \mu_{kr}$, where $\mu_{kr} \sim \text{Unif}[(-2.5, -1.5) \cup (1.5, 2.5)]$ or $\mu_{kr} = 0$ with equal probability. Then, the $n \times p$ matrix $\mathbf{X}$ is generated according to $\mathbf{X} \sim \text{MVN}(\mathbf{A}, \mathbf{\Sigma}, \mathbf{\Delta})$, where $\mathbf{\Sigma}$ and $\mathbf{\Delta}$ are block diagonal covariance matrices with blocks corresponding to the row and column cluster memberships, respectively.

We performed one-way k-means clustering on the rows and on the columns, sparse biclustering, and matrix-variate normal biclustering with $d = 1$. We considered the cases when $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}^{-1}$ are known and unknown. We set the tuning parameters α and β in (3.12) to equal 0.05. In addition, we considered IP, LAS, and SSVD, where the tuning parameters were chosen as described in Chapter 3.5.3. The same evaluation criteria as in Chapter 3.5.3 were used to evaluate the performance of various biclustering methods. Results are reported in Table 3.7.

We see that matrix-variate normal biclustering leads to consistently better results than sparse biclustering and one-way clustering of the rows and columns via k-means. When both $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}^{-1}$ are known, matrix-variate normal biclustering results in the lowest CER.

Table 3.7: Results for simulation study with $n = p = 200$ as described in Chapter 3.7.3. Sparse biclustering and MVN biclustering were performed, with various values of λ, and with λ chosen automatically ($\bar{\lambda}$). MVN biclustering was performed with $\mathbf{\Sigma}^{-1}$ and $\mathbf{\Delta}^{-1}$ known (MVN bicluster known) and unknown (MVN bicluster).

Method	Row CER	Column CER	C. Zeros	C. Non-zeros	Sparsity Rate	Sparsity Error Rate
k-means	0.124 (0.013)	0.145 (0.008)	-	-	-	-
Bicluster $\lambda = 0$	0.075 (0.013)	0.081 (0.010)	-	-	-	-
Bicluster $\lambda = 200$	0.068 (0.012)	0.078 (0.009)	0.556 (0.031)	0.978 (0.003)	0.272 (0.014)	0.248 (0.023)
Bicluster $\lambda = 400$	0.065 (0.012)	0.079 (0.009)	0.782 (0.029)	0.960 (0.006)	0.394 (0.015)	0.139 (0.020)
Bicluster $\bar{\lambda} = 430$	0.066 (0.012)	0.078 (0.009)	0.791 (0.033)	0.962 (0.007)	0.398 (0.019)	0.137 (0.023)
MVN bicluster $\lambda = 0$	0.071 (0.013)	0.081 (0.010)	-	-	-	-
MVN bicluster $\lambda = 15$	0.060 (0.012)	0.073 (0.009)	0.649 (0.028)	0.975 (0.005)	0.323 (0.013)	0.199 (0.020)
MVN bicluster $\lambda = 30$	0.087 (0.014)	0.095 (0.011)	0.809 (0.025)	0.922 (0.013)	0.432 (0.015)	0.141 (0.018)
MVN bicluster $\bar{\lambda} = 18.8$	0.060 (0.012)	0.073 (0.010)	0.716 (0.039)	0.969 (0.009)	0.354 (0.019)	0.169 (0.025)
MVN bicluster known, $\lambda = 0$	0.027 (0.008)	0.044 (0.007)	-	-	-	-
MVN bicluster known, $\lambda = 100$	0.025 (0.008)	0.041 (0.007)	0.475 (0.027)	0.997 (0.001)	0.245 (0.018)	0.258 (0.016)
MVN bicluster known, $\lambda = 250$	0.034 (0.008)	0.053 (0.009)	0.693 (0.027)	0.987 (0.006)	0.358 (0.020)	0.155 (0.014)
MVN bicluster known, $\bar{\lambda} = 257.5$	0.057 (0.017)	0.048 (0.009)	0.712 (0.039)	0.993 (0.002)	0.344 (0.020)	0.163 (0.026)
IP	-	-	1.000 (0.000)	0.000 (0.000)	1.000 (0.000)	0.500 (0.020)
SSVD rank-2	-	-	0.716 (0.040)	0.449 (0.051)	0.640 (0.044)	0.387 (0.014)
LAS	-	-	0.334 (0.006)	0.917 (0.004)	0.208 (0.005)	0.376 (0.012)

3.7.4 Application to real data

We again consider the lung cancer data set described in Chapter 3.6. Once again, we selected 5,000 genes with largest variance, and mean-centered the data matrix. We performed MVN biclustering with $K = 4$, $R = 10$, $\lambda = 1500$, $\alpha = 0.35$, $\beta = 0.35$, and $d = 1$, where α, β, and d are given in (3.12). A heatmap of the estimated mean matrix resulting from MVN biclustering is shown in Figure 3.5.

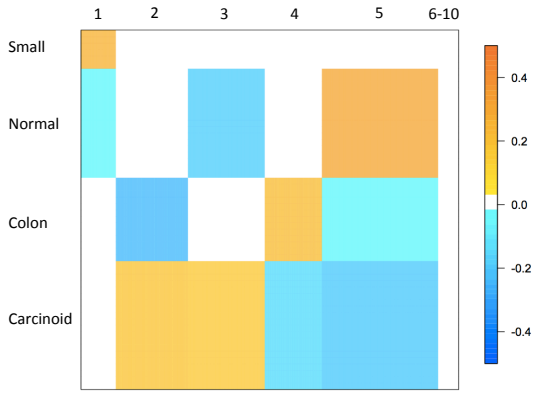

Figure 3.5: Heatmap of the estimated mean matrix from MVN biclustering using $K = 4$, $R = 10$, $\lambda = 1500$, $\alpha = 0.35$, and $\beta = 0.35$ on a subset of the lung cancer data set consisting of the 5,000 genes with highest variance. Details are as in Figure 3.4.

We see from Figure 3.5 that MVN biclustering perfectly identifies the four types of subjects. On this data set, since α is large and n is small, the estimate for $\mathbf{\Sigma}^{-1}$ obtained is diagonal – in other words, here our MVN biclustering does not model conditional dependencies among the samples. In contrast, the estimate obtained for $\mathbf{\Delta}^{-1}$ has many non-zero elements within each of the blocks. In particular, 13.45% of the partial correlations in cluster 1, 73% of the partial correlations in cluster 2, 58.23% of the partial correlations in cluster 3, 40.96% of the partial correlations in cluster 4, 73.22% of the partial correlations in cluster 5, and 0.057% of the partial correlations in clusters 6-10 are non-zero. By inspection of Figure 3.5, we see that the gene clusters with expression levels that differ substantially among

cancer subtypes tend to contain genes that are conditionally dependent. This is scientifically plausible, since we believe that genes that participate in the same pathways tend to be conditionally dependent, and may have similar expression levels in each biological condition.

3.8 Discussion

In this chapter, we have proposed a novel approach for biclustering. Sparsity in the bicluster means is achieved using an ℓ_1 penalty, and our biclustering proposal is extended to a more general setting using the matrix-variate normal distribution. We have shown that k-means clustering can be seen as a special case of our biclustering proposal. Just as a relaxation of k-means clustering yields PCA, a relaxation of our biclustering approach yields the SVD.

A possible drawback of our sparse biclustering proposal is that it does not allow for overlapping biclusters — that is, it assigns each element of the data matrix to exactly one bicluster. While allowing for overlapping biclusters can be beneficial in certain contexts [66], we argue that it results in too much complexity as well as challenges in interpretation. Furthermore, we demonstrate in Chapters 3.5.4 and 3.5.5 that even though our sparse biclustering proposal assumes constant and contiguous biclusters, it performs competitively when there are multiplicative biclusters and overlapping biclusters.

Chapter 4

STATISTICAL PROPERTIES OF CONVEX CLUSTERING

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ be a data matrix with n observations and p features. We assume for convenience that the rows of $\mathbf{X}$ are unique. The goal of clustering is to partition the n observations into subgroups based on some similarity measure. Traditional clustering methods such as hierarchical clustering, k-means clustering, and spectral clustering take a greedy approach (see, e.g., [44]).

In recent years, several authors have proposed formulations for *convex clustering* [16, 47, 60, 77]. Convex clustering of the rows, $\mathbf{X}_{1.}, \ldots, \mathbf{X}_{n.}$, of a data matrix $\mathbf{X}$ involves solving the convex optimization problem

$$\underset{\mathbf{U} \in \mathbb{R}^{n \times p}}{\text{minimize}} \frac{1}{2} \sum_{i=1}^{n} \|\mathbf{X}_{i.} - \mathbf{U}_{i.}\|_2^2 + \lambda \mathrm{Q}_q(\mathbf{U}), \tag{4.1}$$

where $\mathrm{Q}_q(\mathbf{U}) = \sum_{i<i'} \|\mathbf{U}_{i.} - \mathbf{U}_{i'.}\|_q$ for $q = \{1, 2, \infty\}$. The penalty $\mathrm{Q}_q(\mathbf{U})$ generalizes the fused lasso penalty proposed in [96], and encourages the rows of $\hat{\mathbf{U}}$, the solution to (4.1), to take on a small number of unique values. On the basis of $\hat{\mathbf{U}}$, we define the estimated clusters as follows.

Definition 4.1. The ith and i'th observations are estimated by convex clustering to belong to the same cluster if and only if $\hat{\mathbf{U}}_{i.} = \hat{\mathbf{U}}_{i'.}$.

The tuning parameter λ controls the number of unique rows of $\hat{\mathbf{U}}$, i.e., the number of estimated clusters. When $\lambda = 0$, $\hat{\mathbf{U}} = \mathbf{X}$, and so each observation belongs to its own cluster. As λ increases, the number of unique rows of $\hat{\mathbf{U}}$ will decrease. For sufficiently large λ, all rows of $\hat{\mathbf{U}}$ will be identical, and so all observations will be estimated to belong to a single cluster. Note that (4.1) is strictly convex, and therefore the solution $\hat{\mathbf{U}}$ is unique.

To simplify our analysis of convex clustering, we rewrite (4.1). Let $\mathbf{x} = \text{vec}(\mathbf{X}) \in \mathbb{R}^{np}$ and let $\mathbf{u} = \text{vec}(\mathbf{U}) \in \mathbb{R}^{np}$, where the vec$(\cdot)$ operator is such that $x_{(i-1)p+j} = X_{ij}$ and $u_{(i-1)p+j} = U_{ij}$. Construct $\mathbf{D} \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right] \times np}$, and define the index set $\mathcal{C}(i, i')$ such that the $p \times np$ submatrix $\mathbf{D}_{\mathcal{C}(i,i')}$ satisfies $\mathbf{D}_{\mathcal{C}(i,i')}\mathbf{u} = \mathbf{U}_{i.} - \mathbf{U}_{i'.}$. Furthermore, define $\text{P}_q(\mathbf{Du}) = \sum_{i<i'} \|\mathbf{D}_{\mathcal{C}(i,i')}\mathbf{u}\|_q = \sum_{i<i'} \|\mathbf{U}_{i.} - \mathbf{U}_{i'.}\|_q = \text{Q}_q(\mathbf{U})$. Problem (4.1) can be rewritten as

$$\underset{\mathbf{u} \in \mathbb{R}^{np}}{\text{minimize}} \; \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|_2^2 + \lambda \text{P}_q(\mathbf{Du}). \tag{4.2}$$

When $q = 1$, (4.2) is an instance of the generalized lasso problem studied in [97]. Let $\hat{\mathbf{u}}$ be the solution to (4.2). By Definition 4.1, the ith and i'th observations belong to the same cluster if and only if $\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}} = 0$. In what follows, we work with (4.2) instead of (4.1) for convenience.

Let $\mathbf{D}^\dagger \in \mathbb{R}^{np \times \left[p \cdot \binom{n}{2}\right]}$ be the Moore-Penrose pseudo-inverse of $\mathbf{D}$. We state some properties of $\mathbf{D}$ and $\mathbf{D}^\dagger$ that will prove useful in later chapters.

Lemma 4.1. *The matrices* $\mathbf{D}$ *and* $\mathbf{D}^\dagger$ *have the following properties.*

1. $\text{rank}(\mathbf{D}) = p(n-1)$.

2. $\mathbf{D}^\dagger = \frac{1}{n}\mathbf{D}^T$.

3. $(\mathbf{D}^T\mathbf{D})^\dagger\mathbf{D}^T = \mathbf{D}^\dagger$ *and* $(\mathbf{D}\mathbf{D}^T)^\dagger\mathbf{D} = (\mathbf{D}^T)^\dagger$.

4. $\mathbf{D}(\mathbf{D}^T\mathbf{D})^\dagger\mathbf{D}^T = \frac{1}{n}\mathbf{D}\mathbf{D}^T$ *is a projection matrix onto the column space of* $\mathbf{D}$.

5. *Define* $\Lambda_{\min}(\mathbf{D})$ *and* $\Lambda_{\max}(\mathbf{D})$ *as the minimum non-zero singular value and maximum singular value of the matrix* $\mathbf{D}$*, respectively. Then,* $\Lambda_{\min}(\mathbf{D}) = \Lambda_{\max}(\mathbf{D}) = \sqrt{n}$.

Recently, [80] studied the statistical properties of a closely related problem to convex clustering with $q = 1$. On the other hand, [117] studied the condition needed for convex clustering with $q = 2$ to recover the correct clusters. The authors assume that the observations are within some fixed constant of the mean and that the n observations are partitioned

into two non-overlapping clusters, D_1 and D_2. They showed that if the cluster sizes are approximately equal, a sufficient condition for convex clustering to recover the correct clusters is

$$\min_{i \in D_1, i' \in D_2} \|\mathbf{X}_{i\cdot} - \mathbf{X}_{i'\cdot}\|_2 = \Omega(\sqrt{p}),$$

where we use the Landau symbol $f(n) = \Omega(g(n))$ to indicate that there exists a constant $c > 0$ such that $f(n) \geq c \cdot g(n)$ for two sequences $f(n)$ and $g(n)$.

Chapter outline In Chapters 4.1 and 4.2, we study the dual problem of (4.2), and use it to establish that convex clustering is closely related to *single linkage clustering*. In Chapter 4.3, we present some properties of convex clustering. More specifically, we characterize the range of the tuning parameter λ in (4.2) such that convex clustering yields a non-trivial solution. We also provide a finite sample bound for the prediction error, and an unbiased estimator of the degrees of freedom for convex clustering. In Chapter 4.4, we conduct a simulation study to evaluate the empirical performance of convex clustering relative to some existing proposals. We close with a discussion in Chapter 4.5. The proofs are in Appendix C.

4.1 Dual Problem of Convex Clustering

In this chapter, we analyze convex clustering (4.2) by studying its dual problem. Let $\mathrm{P}_q^*(\cdot)$ denote the dual norm of $\mathrm{P}_q(\cdot)$. We refer the reader to [11] for an overview of the concept of duality.

Lemma 4.2. *The dual problem of convex clustering (4.2) is*

$$\underset{\boldsymbol{\nu} \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}}{\text{minimize}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}^T\boldsymbol{\nu}\|_2^2 \qquad \text{subject to } \mathrm{P}_q^*(\boldsymbol{\nu}) \leq \lambda, \tag{4.3}$$

where $\boldsymbol{\nu} \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}$ *is the dual variable. Furthermore, let* $\hat{\mathbf{u}}$ *and* $\hat{\boldsymbol{\nu}}$ *be the solutions to (4.2) and (4.3), respectively. Then,*

$$\mathbf{D}\hat{\mathbf{u}} = \mathbf{D}\mathbf{x} - \mathbf{D}\mathbf{D}^T\hat{\boldsymbol{\nu}}. \tag{4.4}$$

While (4.2) is strictly convex, its dual problem (4.3) is not strictly convex, since $\mathbf{D}$ is not of full rank by Lemma 4.1.1. Therefore, the solution $\hat{\boldsymbol{\nu}}$ to (4.3) is not unique. Lemma 4.1.4 indicates that $\frac{1}{n}\mathbf{D}\mathbf{D}^T$ is a projection matrix onto the column space of $\mathbf{D}$. Thus, the solution $\mathbf{D}\hat{\mathbf{u}}$ in (4.4) can be interpreted as the difference between $\mathbf{D}\mathbf{x}$, the pairwise difference between rows of $\mathbf{X}$, and the projection of a dual variable onto the column space of $\mathbf{D}$.

We now consider a modification to the convex clustering problem (4.2). Recall from Definition 4.1 that the ith and i'th observations are in the same estimated cluster if $\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}} = \mathbf{0}$. This motivates us to estimate $\mathbf{u}' = \mathbf{D}\mathbf{u}$ directly by solving

$$\underset{\mathbf{u}' \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}}{\text{minimize}} \frac{1}{2}\|\mathbf{D}\mathbf{x} - \mathbf{u}'\|_2^2 + \lambda \mathrm{P}_q(\mathbf{u}'). \tag{4.5}$$

We establish a connection between (4.2) and (4.5) by studying the dual problem of (4.5).

Lemma 4.3. *The dual problem of (4.5) is*

$$\underset{\boldsymbol{\nu}' \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}}{\text{minimize}} \frac{1}{2}\|\mathbf{D}\mathbf{x} - \boldsymbol{\nu}'\|_2^2 \qquad \text{subject to } \mathrm{P}_q^*(\boldsymbol{\nu}') \leq \lambda, \tag{4.6}$$

where $\boldsymbol{\nu}' \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}$ is the dual variable. Furthermore, let $\hat{\mathbf{u}}'$ and $\hat{\boldsymbol{\nu}}'$ be the solutions to (4.5) and (4.6), respectively. Then,

$$\hat{\mathbf{u}}' = \mathbf{D}\mathbf{x} - \hat{\boldsymbol{\nu}}'. \tag{4.7}$$

We see that the solution for convex clustering in (4.4) and the solution to our modified problem in (4.7) are closely related. In particular, both solutions involve taking the difference between $\mathbf{D}\mathbf{x}$ and some function of a dual variable that has $\mathrm{P}_q^*(\cdot)$ norm less than or equal to λ. The main difference is that for convex clustering (4.4), we project the dual variable into the column space of $\mathbf{D}$.

Problem (4.5) is quite simple, and in fact it amounts to a thresholding operation on $\mathbf{D}\mathbf{x}$. For instance, when $q = 1$ or $q = 2$, the solution $\hat{\mathbf{u}}'$ is obtained by performing soft thresholding on $\mathbf{D}\mathbf{x}$, or group soft thresholding on $\mathbf{D}_{\mathcal{C}(i,i')}\mathbf{x}$ for all $i < i'$, respectively [3].

4.2 Convex Clustering and Single Linkage Clustering

4.2.1 Connection to Single Linkage Clustering

We now establish a connection between convex clustering and single linkage clustering by showing that the estimated clusters of (4.5) with $q = 2$ are equivalent to those of single linkage clustering.

Let $\hat{\mathbf{u}}'$ be the solution to (4.5). It can be verified that $\hat{\mathbf{u}}'_{\mathcal{C}(i,i')} = \mathbf{0}$ if and only if $\|\mathbf{D}_{\mathcal{C}(i,i')}\mathbf{x}\|_2 = \|\mathbf{X}_{i\cdot} - \mathbf{X}_{i'\cdot}\|_2 \leq \lambda$. For convenience, we define the set

$$\hat{\mathcal{S}}(\lambda) = \{(i, i') : \|\mathbf{X}_{i\cdot} - \mathbf{X}_{i'\cdot}\|_2 < \lambda, i < i'\}. \tag{4.8}$$

It might be tempting to conclude that a pair of observations (i, i') belong to the same cluster if $\hat{\mathbf{u}}'_{\mathcal{C}(i,i')} = \mathbf{0}$, or equivalently, $(i, i') \in \hat{\mathcal{S}}(\lambda)$. However, by inspection of (4.8), it could happen that $(i, i') \in \hat{\mathcal{S}}(\lambda)$ and $(i', i'') \in \hat{\mathcal{S}}(\lambda)$, but $(i, i'') \notin \hat{\mathcal{S}}(\lambda)$. To overcome this problem, we define the $n \times n$ adjacency matrix $\mathbf{A}(\lambda)$ as

$$A_{ii'}(\lambda) = \begin{cases} 1 & \text{if } i = i', \\ 1 & \text{if } (i, i') \in \hat{\mathcal{S}}(\lambda) \text{ or } (i', i) \in \hat{\mathcal{S}}(\lambda), \\ 0 & \text{if } (i, i') \notin \hat{\mathcal{S}}(\lambda). \end{cases} \tag{4.9}$$

Subject to a rearrangement of the rows and columns, $\mathbf{A}(\lambda)$ is a block-diagonal matrix with R blocks. On the basis of $\mathbf{A}(\lambda)$, we define R estimated clusters: the indices of the observations in the rth cluster are the same as the indices of the observations in the rth block.

We now present a lemma on the equivalence between single linkage clustering and the clusters identified by (4.5) using (4.9).

Lemma 4.4. *[50, 72]. Let $\hat{D}_1, \ldots, \hat{D}_K$ denote the clusters that result from performing single linkage clustering on the dissimilarity matrix defined by the Euclidean distance between the observations, and cutting the dendrogram at the height of $\lambda > 0$. Let $\hat{E}_1, \ldots, \hat{E}_R$ index the blocks within the adjacency matrix $\mathbf{A}(\lambda)$. Then $K = R$, and there exists a permutation π such that $D_k = E_{\pi(k)}$ for $k = 1, \ldots, K$.*

In other words, Lemma 4.4 implies that single linkage clustering and (4.5) yield the same estimated clusters. Recalling the connection between (4.2) and (4.5) established in Chapter 4.1, this implies a close connection between convex clustering with $q = 2$ and single linkage clustering.

4.2.2 Sufficient Condition for Consistent Clustering

We have shown that convex clustering is closely related to single linkage clustering. We now study the properties of single linkage clustering, in order to establish that the conditions needed for single linkage clustering to successfully recover the true clusters are similar to those of convex clustering. Suppose that the n observations are partitioned into K clusters $D_1, \ldots, D_K$, i.e., D_k is the index set for the observations in the kth cluster. Let $\mathcal{S} = \{(i, i') : i, i' \in D_k, i < i'\}$ be a set containing pairs of indices for observations that belong to the same cluster. Let $\boldsymbol{\mu}_k \in \mathbb{R}^p$ be the mean vector for the kth cluster. We will now establish a sufficient condition on the minimum distance between $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_{k'}$ such that the event $\left\{\hat{\mathcal{S}}(\lambda) = \mathcal{S}\right\}$ holds with high probability.

Assumption 4.1. The minimum distance between the means of any two clusters is

$$\delta = \min_{k \neq k'} \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'}\|_2 \geq 2\sqrt{2}\sigma\sqrt{p}\sqrt{1 + \sqrt{\frac{12 \log n}{p}} + 6\frac{\log n}{p}}.$$

Theorem 4.1. *Assume that* $\mathbf{X}_{i\cdot} \sim \text{MVN}(\boldsymbol{\mu}_k, \sigma^2\mathbf{I})$ *for* $i \in D_k$. *Then, given Assumption 4.1, we have that* $\Pr\left(\hat{\mathcal{S}}(\delta/2) = \mathcal{S}\right) \geq 1 - \frac{2}{n}$.

Theorem 4.1 guarantees that with high probability, (4.5) identifies the correct cluster memberships for the observations, provided that the minimum signal δ is sufficiently large. Assuming $\log n = o(p)$, we obtain $\delta = \Omega(\sqrt{p})$. We see that the signal requirement for (4.5) with $q = 2$ is on the same order as that of convex clustering (4.1) with $q = 2$ [117]. This suggests that (4.5), or equivalently, single linkage clustering, may do as well as convex clustering. We will explore this in Chapter 4.4 in a simulation study.

4.3 Properties of Convex Clustering

We now study the statistical properties of convex clustering (4.2) with $q = 1$ and $q = 2$. In Chapter 4.3.1, we establish the range of the tuning parameter λ in (4.2) such that convex clustering yields a non-trivial solution with more than one cluster. We provide finite sample bounds for the prediction error of convex clustering in Chapter 4.3.2. Finally, we provide unbiased estimates of the degrees of freedom for convex clustering in Chapter 4.3.3.

4.3.1 Range of λ that Yields Non-trivial Solution

In this chapter, we establish the range of the tuning parameter λ such that convex clustering (4.2) yields a non-trivial solution.

Lemma 4.5. *Let*

$$\lambda_{\text{upper}} := \begin{cases} \min\limits_{\boldsymbol{\omega}} \left\| \frac{1}{n}\mathbf{D}\mathbf{x} + \left(\mathbf{I} - \frac{1}{n}\mathbf{D}\mathbf{D}^T\right)\boldsymbol{\omega} \right\|_\infty & \text{for } q = 1, \\ \min\limits_{\boldsymbol{\omega}} \left\{ \max\limits_{i<i'} \left\{ \left\| \left(\frac{1}{n}\mathbf{D}\mathbf{x} + \left(\mathbf{I} - \frac{1}{n}\mathbf{D}\mathbf{D}^T\right)\boldsymbol{\omega}\right)_{\mathcal{C}(i,i')} \right\|_2 \right\} \right\} & \text{for } q = 2. \end{cases} \tag{4.10}$$

Convex clustering (4.2) with $q = 1$ or $q = 2$ yields a non-trivial solution of more than one cluster if and only if $\lambda < \lambda_{\text{upper}}$.

By Lemma 4.5, we see that calculating λ_{upper} boils down to solving a convex optimization problem. This can be solved using a standard solver such as `CVX` in `MATLAB`. In the absence of such a solver, a loose upper bound on λ can be obtained by taking λ_{upper} to be $\|\frac{1}{n}\mathbf{D}\mathbf{x}\|_\infty$ for $q = 1$, or $\max\limits_{i<i'} \|\frac{1}{n}\mathbf{D}_{\mathcal{C}(i,i')}\mathbf{x}\|_2$ for $q = 2$.

Therefore, to obtain the entire solution path of convex clustering, we need only consider values of λ that satisfy $\lambda \leq \lambda_{\text{upper}}$.

4.3.2 Bounds on Prediction Error

We provide finite sample bounds for the prediction error of convex clustering (4.2). Let λ be the tuning parameter in (4.2) and let $\lambda' = \frac{\lambda}{np}$.

Lemma 4.6. *Assume that* $\mathbf{x} \sim \text{MVN}(\mathbf{u}, \sigma^2\mathbf{I})$. *Let* $\hat{\mathbf{u}}$ *be the estimate obtained from (4.2) with* $q = 1$. *If* $\lambda' \geq 4\sigma\sqrt{\frac{\log\left(p\cdot\binom{n}{2}\right)}{n^3p^2}}$, *then*

$$\frac{1}{2np}\|\hat{\mathbf{u}} - \mathbf{u}\|_2^2 \leq \frac{3\lambda'}{2}\|\mathbf{D}\mathbf{u}\|_1 + \sigma^2\left[\frac{1}{n} + 2\sqrt{\frac{\log(np)}{n^2p}} + 2\frac{\log(np)}{np}\right]$$

holds with probability at least $1 - \frac{2}{p\cdot\binom{n}{2}} - \frac{1}{np}$.

We see from Lemma 4.6 that the average prediction error is bounded by the oracle quantity $\|\mathbf{D}\mathbf{u}\|_1$ and a second term that decays to zero as $n, p \to \infty$. Convex clustering with $q = 1$ is prediction consistent only if $\lambda'\|\mathbf{D}\mathbf{u}\|_1 = o(1)$. We now provide a scenario for which $\lambda'\|\mathbf{D}\mathbf{u}\|_1 = o(1)$ holds.

Suppose that we are in the high-dimensional setting in which $p > n$ and the true underlying clusters differ only with respect to a fixed number of features [106]. Also, suppose that each element of $\mathbf{D}\mathbf{u}$ — that is, $U_{ij} - U_{i'j}$ — is of order $O(1)$. Therefore, $\|\mathbf{D}\mathbf{u}\|_1 = O(n^2)$, since by assumption only a fixed number of features have different means across clusters. Assume that $\sqrt{\frac{n\log\left(p\cdot\binom{n}{2}\right)}{p^2}} = o(1)$. Under these assumptions, convex clustering with $q = 1$ is prediction consistent. Next, we present a finite sample bound on the prediction error for convex clustering with $q = 2$.

Lemma 4.7. *Assume* $\mathbf{x} \sim \text{MVN}(\mathbf{u}, \sigma^2\mathbf{I})$. *Let* $\hat{\mathbf{u}}$ *be the estimate obtained from (4.2) with* $q = 2$. *If* $\lambda' \geq 4\sigma\sqrt{\frac{\log\left(p\cdot\binom{n}{2}\right)}{n^3p}}$, *then*

$$\frac{1}{2np}\|\hat{\mathbf{u}} - \mathbf{u}\|_2^2 \leq \frac{3\lambda'}{2}\sum_{i<i'}\|\mathbf{D}_{\mathcal{C}(i,i')}\mathbf{u}\|_2 + \sigma^2\left[\frac{1}{n} + 2\sqrt{\frac{\log(np)}{n^2p}} + 2\frac{\log(np)}{np}\right]$$

holds with probability at least $1 - \frac{2}{p\cdot\binom{n}{2}} - \frac{1}{np}$.

Under the scenario described above, $\|\mathbf{D}_{\mathcal{C}(i,i')}\mathbf{u}\|_2 = O(1)$, and therefore $\sum_{i<i'}\|\mathbf{D}_{\mathcal{C}(i,i')}\mathbf{u}\|_2 = O(n^2)$. Convex clustering with $q = 2$ is prediction consistent if $\sqrt{\frac{n\log\left(p\cdot\binom{n}{2}\right)}{p}} = o(1)$.

4.3.3 Degrees of Freedom

Convex clustering recasts the clustering problem as a penalized regression problem, for which the notion of degrees of freedom is established [27]. Under this framework, we provide an unbiased estimator of the degrees of freedom for clustering. Recall that $\hat{\mathbf{u}}$ is the solution to convex clustering (4.2). Suppose that $\text{Var}(\mathbf{x}) = \sigma^2\mathbf{I}$. Then, the degrees of freedom for convex clustering is defined as $\frac{1}{\sigma^2}\sum_{\text{j}=1}^{np}\text{Cov}(\hat{u}_j, x_j)$ (see, e.g., [27]). An unbiased estimator of the degrees of freedom for convex clustering with $q = 1$ follows directly from [98].

Lemma 4.8. *[98] Assume that* $\mathbf{x} \sim \text{MVN}(\mathbf{u}, \sigma^2\mathbf{I})$*, and let* $\hat{\mathbf{u}}$ *be the solution to (4.2) with* $q = 1$*. Furthermore, let* $\hat{\mathcal{B}}_1 = \{j : |(\mathbf{D}\hat{\mathbf{u}})_j| \neq 0\}$*. We define the matrix* $\mathbf{D}_{-\hat{\mathcal{B}}_1}$ *by removing the rows of* $\mathbf{D}$ *that correspond to* $\hat{\mathcal{B}}_1$*. Then*

$$\begin{aligned}\hat{\text{df}}_1 &= \text{tr}\left(\mathbf{I} - \mathbf{D}^T_{-\hat{\mathcal{B}}_1}(\mathbf{D}_{-\hat{\mathcal{B}}_1}\mathbf{D}^T_{-\hat{\mathcal{B}}_1})^\dagger\mathbf{D}_{-\hat{\mathcal{B}}_1}\right)\\ &= \textit{number of unique elements in } \hat{\mathbf{u}}\end{aligned} \tag{4.11}$$

is an unbiased estimator of the degrees of freedom of convex clustering with $q = 1$*.*

There is an interesting interpretation of the degrees of freedom estimate for convex clustering with $q = 1$. Suppose that there are K estimated clusters, and all elements of the estimated mean corresponding to the K estimated clusters are unique. Then the unbiased estimate of the degrees of freedom is Kp, the product of the number of estimated clusters and the number of features. Next, we provide an unbiased estimator of the degrees of freedom for convex clustering with $q = 2$.

Lemma 4.9. *Assume that* $\mathbf{x} \sim \text{MVN}(\mathbf{u}, \sigma^2\mathbf{I})$*, and let* $\hat{\mathbf{u}}$ *be the solution to (4.2) with* $q = 2$*. Furthermore, let* $\hat{\mathcal{B}}_2 = \{(i, i') : \|\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}}\|_2 \neq 0\}$*. We define the matrix* $\mathbf{D}_{-\hat{\mathcal{B}}_2}$ *by removing rows of* $\mathbf{D}$ *that correspond to* $\hat{\mathcal{B}}_2$*. Let* $\mathbf{P} = \left(\mathbf{I} - \mathbf{D}^T_{-\hat{\mathcal{B}}_2}(\mathbf{D}_{-\hat{\mathcal{B}}_2}\mathbf{D}^T_{-\hat{\mathcal{B}}_2})^\dagger\mathbf{D}_{-\hat{\mathcal{B}}_2}\right)$ *be the projection matrix onto the complement of the space spanned by the rows of* $\mathbf{D}_{-\hat{\mathcal{B}}_2}$*. Then*

$$\hat{\text{df}}_2 = \text{tr}\left(\left[\mathbf{I} + \lambda\mathbf{P}\sum_{(i,i')\in\hat{\mathcal{B}}_2}\left(\frac{\mathbf{D}^T_{\mathcal{C}(i,i')}\mathbf{D}_{\mathcal{C}(i,i')}}{\|\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}}\|_2} - \frac{\mathbf{D}^T_{\mathcal{C}(i,i')}\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}}\hat{\mathbf{u}}^T\mathbf{D}^T_{\mathcal{C}(i,i')}\mathbf{D}_{\mathcal{C}(i,i')}}{\|\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}}\|_2^3}\right)\right]^{-1}\mathbf{P}\right) \tag{4.12}$$

is an unbiased estimator of the degrees of freedom of convex clustering with $q = 2$.

When $\lambda = 0$, $\|\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}}\|_2 \neq 0$ for all $i < i'$. Therefore, $\mathbf{P} = \mathbf{I} \in \mathbb{R}^{np \times np}$ and the degrees of freedom estimate is equal to $\text{tr}(\mathbf{I}) = np$. When λ is sufficiently large that $\hat{\mathcal{B}}_2$ is an empty set, one can verify that $\mathbf{P} = \mathbf{I} - \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^\dagger\mathbf{D}$ is a projection matrix of rank p, using the fact that $\text{rank}(\mathbf{D}) = p(n-1)$ from Lemma 4.1.1. Therefore $\hat{\text{df}}_2 = \text{tr}(\mathbf{P}) = p$.

We now assess the accuracy of the proposed unbiased estimates of the degrees of freedom. We simulate the data as described in Chapter 4.4 with $n = p = 20$ and $\sigma = 1$. We perform convex clustering with $q = 1$ and $q = 2$ across a fine grid of tuning parameters λ. For each λ, we compare the quantities (4.11) and (4.12) to

$$\frac{1}{\sigma^2}\sum_{j=1}^{np}(\hat{u}_j - u_j)(x_j - u_j), \tag{4.13}$$

which is an unbiased estimator of the true degrees of freedom, $\frac{1}{\sigma^2}\sum_{j=1}^{np}\text{Cov}(\hat{u}_j, x_j)$, averaged over 500 data sets. Note that (4.13) cannot be computed in practice, since it requires knowledge of the unknown quantity $\mathbf{u}$. Results are displayed in Figure 4.1. We see that the proposed estimators agree with the true degrees of freedom.

4.4 Simulation Studies

We compare convex clustering with $q = 2$ to the following proposals:

1. Single linkage clustering. Based on Chapter 4.2, we expect single linkage clustering to give similar results to convex clustering with $q = 2$.

2. The k-means clustering algorithm [64].

3. Average linkage hierarchical clustering [44].

In our simulation studies, we create $K = 2$ row clusters by randomly assigning each observation to a row cluster with equal probability. We generate an $n \times p$ data matrix $\mathbf{X}$ according

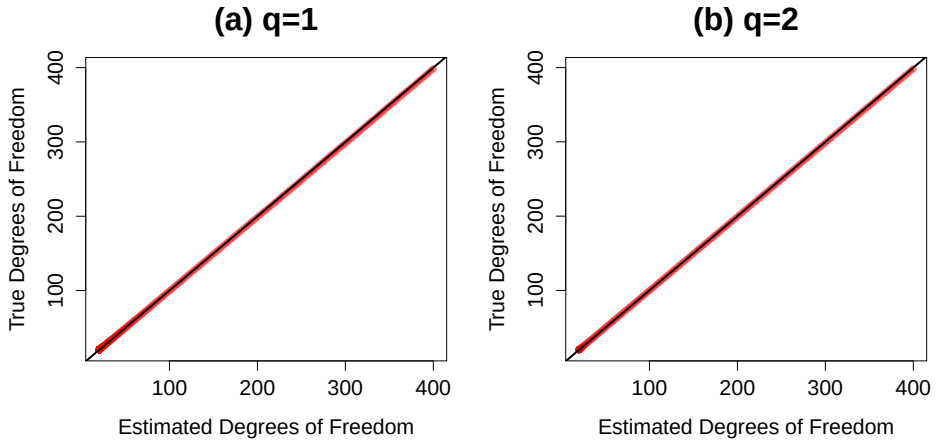

Figure 4.1: We compare the true degrees of freedom of convex clustering (y-axis), given in (4.13), to the proposed unbiased estimators of the degrees of freedom (x-axis), given in Lemmas 4.8 and 4.9. Panels (a) and (b) contain the results for convex clustering with $q = 1$ and $q = 2$, respectively. The red line is obtained by varying λ for convex clustering. The black line indicates $y = x$.

to $\mathbf{X}_{i\cdot} \sim \text{MVN}(\boldsymbol{\mu}_k, \sigma^2\mathbf{I})$ for $i \in D_k$, where $\boldsymbol{\mu}_1 = \mathbf{1}_p$ and $\boldsymbol{\mu}_2 = -\mathbf{1}_p$. We consider $n = p = 50$ and $\sigma = \{1, 2\}$.

We implement convex clustering (4.1) with $q = 2$ using the `R` package `cvxclustr`. In order to obtain the entire solution path for convex clustering, we use a fine grid of λ values for (4.2), in a range guided by Lemma 4.5. We apply the other methods by allowing the number of clusters to vary over a range from 1 to n clusters.

We use the Rand index to quantify the performance of the clustering methods [81]. A high value of the Rand index indicates good agreement between the true and estimated clusters. The Rand indices, averaged over 200 data sets, are summarized in Figure 4.2.

We see from Figure 4.2(a) that when the noise level is small, $\sigma = 1$, the performance of both convex clustering and single linkage clustering are approximately the same, and both of these methods outperform k-means clustering and average linkage clustering. However, when the noise level increases from $\sigma = 1$ to $\sigma = 2$, single linkage clustering outperforms con-

vex clustering (Figure 4.2(b)). Moreover, k-means clustering and average linkage clustering outperform the two methods. This suggests that the minimum signal needed for convex clustering to identify the correct clusters may be larger than that of average linkage hierarchical clustering and k-means clustering.

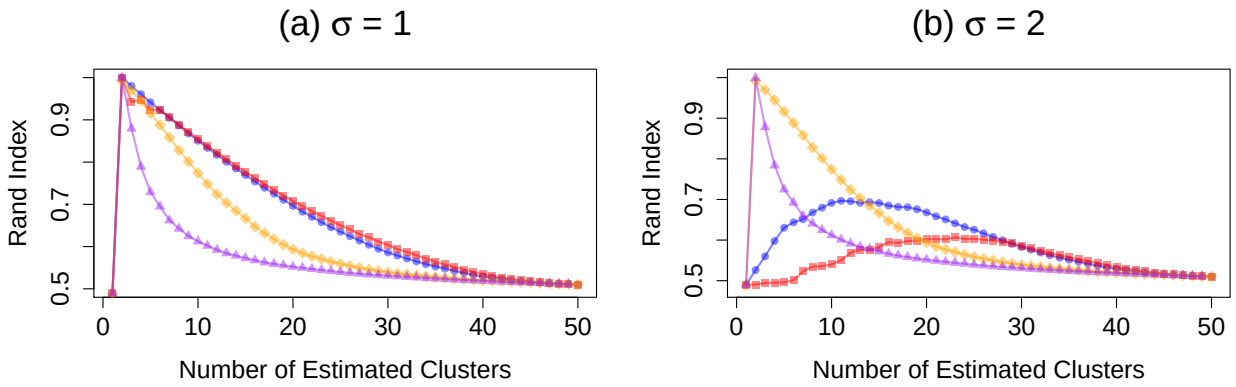

Figure 4.2: Simulation study for convex clustering and other clustering methods, $n = p = 50$, averaged over 200 data sets, for two noise levels. Colored lines correspond to single linkage clustering (), average linkage clustering (), k-means clustering (), and convex clustering ().

4.5 Discussion

Convex clustering recasts the clustering problem into a penalized regression problem. By studying its dual problem, we show that there is a connection between convex clustering and single linkage clustering. In addition, we establish several statistical properties of convex clustering. Through some numerical studies, we illustrate that the performance of convex clustering may not be appealing relative to traditional clustering methods when the signal-to-noise ratio is low.

Many authors have proposed a modification to the convex clustering problem (4.1),

$$\underset{\mathbf{U}\in\mathbb{R}^{n\times p}}{\text{minimize}}\ \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{X}_{i\cdot}-\mathbf{U}_{i\cdot}\|_2^2+\lambda Q_q(\mathbf{W},\mathbf{U}), \tag{4.14}$$

where $\mathbf{W}$ is an $n \times n$ symmetric matrix of positive weights, and $\mathrm{Q}_q(\mathbf{W}, \mathbf{U}) = \sum_{i<i'} W_{ii'} \|\mathbf{U}_{i.} - \mathbf{U}_{i'.}\|_q$ [16, 47, 60, 77]. For instance, the weights can be defined as $W_{ii'} = \exp\left(-\phi \|\mathbf{X}_{i.} - \mathbf{X}_{i'.}\|_2^2\right)$ for some constant $\phi > 0$, which may yield better empirical performance than (4.1). In future work, it would be interesting to study the statistical properties of (4.14) and to explore whether there is a connection between (4.14) and a modified version of single linkage clustering.

BIBLIOGRAPHY

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

[2] G.I. Allen and R. Tibshirani. Transposable regularized covariance models with an application to missing data imputation. *Annals of Applied Statistics*, 4(2):764–790, 2010.

[3] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. In *Optimization for Machine Learning*, pages 19–53. MIT Press, 2011.

[4] A.L. Barabási. Scale-free networks: A decade and beyond. *Science*, 325:412–413, 2009.

[5] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[6] J. Barzilai and J.M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[7] P.J. Bickel and E. Levina. Regularized estimation of large covariance matrices. *Annals of Statistics*, 36(1):199–227, 2008.

[8] J. Bien and R. Tibshirani. Sparse estimation of a covariance matrix. *Biometrika*, 98(4):807–820, 2011.

[9] L. Birgé. An alternative point of view on Lepski's method. *State of the art in probability and statistics*, 36:113–133, 2001.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the ADMM. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[12] T. Cai and W. Liu. Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, 106(494):672–684, 2011.

[13] A. Cardoso-Cachopo. 2009. "http://web.ist.utl.pt/acardoso/datasets/".

[14] S. Chaudhuri, M. Drton, and T. Richardson. Estimation of a covariance matrix with zeros. *Biometrika*, 94:199–216, 2007.

[15] Y. Cheng and G.M. Church. Biclustering of expression data. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, pages 93–103, 2000.

[16] E.C. Chi and K. Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics, in press*, 2014.

[17] H. Chipman and R. Tibshirani. Hybrid hierarchical clustering with applications to microarray data. *Biostatistics*, 7:286–301, 2005.

[18] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra. Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 114–125, 2004.

[19] H. Cho and I.S. Dhillon. Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 5(3):385–400, 2008.

[20] Y. Dai. A new analysis on the Barzilai-Borwein gradient method. *Journal of the Operations Research Society of China*, 1(2):187–198, 2013.

[21] P. Danaher, P. Wang, and D.M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B*, 76(2):373–397, 2014.

[22] A. Defazio and T.S. Caetano. A convex formulation for learning scale-free network via submodular relaxation. *Advances in Neural Information Processing Systems*, 2012.

[23] M. Drton and T.S. Richardson. A new algorithm for maximum likelihood estimation in Gaussian graphical models for marginal independence. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pages 184–191, 2003.

[24] M. Drton and T.S. Richardson. Graphical methods for efficient likelihood inference in Gaussian covariance models. *Journal of Machine Learning Research*, 9:893–914, 2008.

[25] J. Eckstein. Augmented Lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. *RUTCOR Research Reports*, 32, 2012.

[26] J. Eckstein and D.P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(3, Ser. A):293–318, 1992.

[27] B. Efron. How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394):461–470, 1986.

[28] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25):14863–14868, 1998.

[29] N. El Karoui. Operator norm consistent estimation of large-dimensional sparse covariance matrices. *The Annals of Statistics*, 36(6):2717–2756, 2008.

[30] P. Erdős and A. Rényi. On random graphs I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[31] T.S. Ferguson. *A Course in Large Sample Theory*. Chapman & Hall London, 1996.

[32] H. Firouzi and A.O. Hero. Local hub screening in sparse correlation graphs. *Proceedings of SPIE, volume 8858, Wavelets and Sparsity XV, 88581H*, 2013.

[33] R. Foygel and M. Drton. Extended Bayesian information criteria for Gaussian graphical models. *Advances in Neural Information Processing Systems*, 2010.

[34] C. Fraley and A. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

[35] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9:432–441, 2007.

[36] J.E. Gentle. *Matrix Algebra: Theory, Computations, and Applications in Statistics*. Springer, New York, 2007.

[37] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering of gene microarray data. *PNAS*, 97:12079–12084, 2000.

[38] J. Gu and J.S. Liu. Bayesian biclustering of gene expression data. *BMC Genomics*, 9:S4, 2008.

[39] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint structure estimation for categorical Markov networks. Submitted, available at http://www.stat.lsa.umich.edu/˜elevina, 2010.

[40] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Asymptotic properties of the joint neighborhood selection method for estimating categorical Markov networks. `arXiv: math.PR/0000000`, 2011.

[41] A.K. Gupta and D.K. Nagar. *Matrix Variate Distributions*. CRC Press, Boca Raton, FL, 1999.

[42] D. Hao, C. Ren, and C. Li. Revisiting the variation of clustering coefficient of biological networks suggests new modular structure. *BMC System Biology*, 6(34):1–10, 2012.

[43] J. A Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 6:123–129, 1972.

[44] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. Springer Verlag, New York, 2009.

[45] A. Hero and B. Rajaratnam. Hub discovery in partial correlation graphs. *IEEE Transactions on Information Theory*, 58:6064–6078, 2012.

[46] S. Hochreiter, U. Bodenhofer, M. Heusel, A. Mayr, A. Mitterecker, A. Kasim, T. Khamiakova, S. Sanden, D. Lin, W. Talloen, L. Bijnens, H. Gohlmann, Z. Shkedy, and D. Clevert. FABIA: factor analysis for bicluster acquisition. *Bioinformatics*, 26(12):1520–1527, 2010.

[47] T.D. Hocking, A. Joulin, F. Bach, and Jean.-P. Vert. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th International Conference on Machine Learning*, 2011.

[48] H. Höfling and R. Tibshirani. Estimation of sparse binary pairwise Markov networks using pseudo-likelihoods. *Journal of Machine Learning Research*, 10:883–906, 2009.

[49] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, New York, NY, 1985.

[50] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

[51] H. Jeong, S.P. Mason, A.L. Barabási, and Z.N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

[52] S. Kaiser, R. Santamaria, T. Khamiakova, M. Sill, R. Theron, L. Quintales, and F. Leisch. *biclust: BiCluster algorithms*, 2011. R package version 1.0.1.

[53] B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, 28(5):1302–1338, 2000.

[54] S.L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[55] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–86, 2002.

[56] M. Lee, H. Shen, J.Z. Huang, and J.S. Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095, 2010.

[57] S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using ℓ_1-regularization. *Advances in Neural Information Processing Systems*, 2007.

[58] L. Li, D. Alderson, J.C. Doyle, and W. Willinger. Towards a theory of scale-free graphs: definition, properties, and implications. *Internet Mathematics*, 2(4):431–523, 2005.

[59] F. Liljeros, C.R. Edling, L.A.N. Amaral, H.E. Stanley, and Aberg Y. The web of human sexual contacts. *Nature*, 411:907–908, 2001.

[60] F. Lindsten, H. Ohlsson, and L. Ljung. Clustering using sum-of-norms regularization: With application to particle filter output computation. In *Statistical Signal Processing Workshop (SSP)*, pages 201–204. IEEE, 2011.

[61] J. Liu, L. Yuan, and J. Ye. Guaranteed sparse recovery under linear transformation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 91–99, 2013.

[62] Q. Liu and A.T. Ihler. Learning scale free networks by reweighed ℓ_1 regularization. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, volume 15, pages 40–48, 2011.

[63] Y. Liu, D. Hayes, A. Nobel, and J. Marron. Statistical significance of clustering for high-dimension, low-sample size data. *Journal of the American Statistical Association*, 103(483):1281–1293, 2008.

[64] Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[65] S. Ma, L. Xue, and H. Zou. Alternating direction methods for latent variable Gaussian graphical model selection. *Neural Computation*, 28(8):2172–2198, 2013.

[66] S.C. Madeira and A.L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[67] Maglott et al. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, 33(D):54–58, 2004.

[68] K.V. Mardia, J. Kent, and J.M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[69] R. Mazumder and T. Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, 13:781–794, 2012.

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

[71] N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B*, 72:417–473, 2010.

[72] B.G. Mirkin. *Mathematical Classification and Clustering*. Springer, New York, 1996.

[73] K. Mohan, P. London, M. Fazel, D.M. Witten, and S.-I. Lee. Node-based learning of Gaussian graphical models. *Journal of Machine Learning Research*, 15:445–488, 2014.

[74] M.E.J. Newman. The structure of scientific collaboration networks. *PNAS*, 98:404–409, 2000.

[75] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 2006.

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

[77] K. Pelckmans, J. De Brabanter, J.A.K Suykens, and B. De Moor. Convex clustering shrinkage. In *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*, 2005.

[78] J. Peng, P. Wang, N. Zhou, and J. Zhu. Partial correlation estimation by joint sparse regression model. *Journal of the American Statistical Association*, 104(486):735–746, 2009.

[79] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 2006.

[80] P. Radchenko and G. Mukherjee. Consistent clustering using ℓ_1 fusion penalty. *arXiv preprint arXiv:1412.0753*, 2014.

[81] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[82] Rappaport et al. MalaCards: an integrated compendium for diseases and their annotation. *Database (Oxford)*, 2013.

[83] P. Ravikumar, M. Wainwright, and J. Lafferty. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.

[84] P. Ravikumar, M.J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

[85] M. Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7:26–33, 1997.

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

[87] A. Rothman, E. Levina, and J. Zhu. Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, 104:177–186, 2009.

[88] A. Shabalin, V. Weigman, C. Perou, and A. Nobel. Finding large average submatrices in high dimensional data. *Annals of Applied Statistics*, 3(3):985–1012, 2009.

[89] M. Sill and S. Kaiser. *s4vd: Biclustering via sparse singular value decomposition incorporating stability selection*, 2011. R package version 1.0.

[90] N. Simon, J.H. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[91] K.M. Tan, P. London, S.-I. Lee, M. Fazel, and D. Witten. Learning graphical models with hubs. *Journal of Machine Learning Research*, 15:3297–3331, 2014.

[92] K.M. Tan and D. Witten. Sparse biclustering of transposable data. *Journal of Computational and Graphical Statistics*, 23(4):985–1008, 2014.

[93] C. Tang, L. Zhang, A. Zhang, and M. Ramanathan. Interrelated two-way clustering: An unsupervised approach for gene expression data analysis. In *Proceedings of 2nd IEEE International Symposium on Bioinformatics and Bioengineering, Bethesda*, 2001.

[94] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[95] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, pages 267–288, 1996.

[96] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67(1):91–108, 2005.

[97] R.J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335–1371, 2011.

[98] R.J. Tibshirani and J. Taylor. Degrees of freedom in lasso problems. *The Annals of Statistics*, 40(2):1198–1232, 2012.

[99] H. Turner, T. Bailey, and W. Krzanowski. Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational Statistics and Data Analysis*, 48:235–254, 2005.

[100] S. Vaiter, C.-A. Deledalle, G. Peyré, J.M. Fadili, and C. Dossal. The degrees of freedom of partly smooth regularizers. *arXiv preprint arXiv:1404.5557*, 2014.

[101] A.W. Van der Vaart. *Asymptotic Statistics*. Cambridge university press, 2000.

[102] Verhaak et al. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, 17(1):98–110, 2010.

[103] S. Wang and J. Zhu. Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics*, 64:440–448, 2008.

[104] D.M. Witten, J.H. Friedman, and N. Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900, 2011.

[105] D.M. Witten and R. Tibshirani. Covariance-regularized regression and classification for high-dimensional problems. *Journal of the Royal Statistical Society: Series B*, 71(3):615–636, PMCID:PMC2806603, 2009.

[106] D.M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

[107] D.M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

[108] S.J. Wright, R.D. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

[109] B. Xie, W. Pan, and X. Shen. Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. *Electronic Journal of Statistics*, 2:168–212, 2008.

[110] L. Xue, S. Ma, and H. Zou. Positive definite ℓ_1 penalized estimation of large covariance matrices. *Journal of the American Statistical Association*, 107(500):1480–1491, 2012.

[111] E. Yang, G.I. Allen, Z. Liu, and P.K. Ravikumar. Graphical models via generalized linear models. *Advances in Neural Information Processing Systems*, 2012.

[112] S. Yang, Z. Pan, X. Shen, P. Wonka, and J. Ye. Fused multiple graphical lasso. *arXiv preprint arXiv:1209.2139*, 2012.

[113] M. Yuan. Efficient computation of ℓ_1 regularized estimates in Gaussian graphical models. *Journal of Computational and Graphical Statistics*, 17(4):809–826, 2008.

[114] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68:49–67, 2007.

[115] M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(10):19–35, 2007.

[116] H. Zha, X. He, G. Ding, H. Simon, and M. Gu. Spectral relaxation for k-means clustering. *Neural Information Processing Systems*, 14:1057–1064, 2001.

[117] C. Zhu, H. Xu, C. Leng, and S. Yan. Convex optimization procedure for clustering: theoretical revisit. In *Advances in Neural Information Processing Systems*. 2014.

[118] H. Zou, T. Hastie, and R. Tibshirani. On the "degrees of freedom" of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.

Appendix A

APPENDIX FOR CHAPTER 2

A.1 Derivation of Algorithm 1

Recall that the scaled augmented Lagrangian for (2.4) takes the form

$$\begin{aligned} L(\mathbf{B}, \tilde{\mathbf{B}}, \mathbf{W}) = \ell(\mathbf{X}, \boldsymbol{\Theta}) + \lambda_1 \|\mathbf{Z} - \mathrm{diag}(\mathbf{Z})\|_1 + \lambda_2 \|\mathbf{V} - \mathrm{diag}(\mathbf{V})\|_1 \\ + \lambda_3 \sum_{j=1}^{p} \|(\mathbf{V} - \mathrm{diag}(\mathbf{V}))_j\|_2 + g(\tilde{\mathbf{B}}) + \frac{\rho}{2} \|\mathbf{B} - \tilde{\mathbf{B}} + \mathbf{W}\|_F^2. \end{aligned} \tag{A.1}$$

The proposed ADMM algorithm requires the following updates:

1. $\mathbf{B}^{(t+1)} \leftarrow \underset{\mathbf{B}}{\mathrm{argmin}}\ L(\mathbf{B}, \tilde{\mathbf{B}}^t, \mathbf{W}^t)$,
2. $\tilde{\mathbf{B}}^{(t+1)} \leftarrow \underset{\tilde{\mathbf{B}}}{\mathrm{argmin}}\ L(\mathbf{B}^{(t+1)}, \tilde{\mathbf{B}}, \mathbf{W}^t)$,
3. $\mathbf{W}^{(t+1)} \leftarrow \mathbf{W}^t + \mathbf{B}^{(t+1)} - \tilde{\mathbf{B}}^{(t+1)}$.

We now proceed to derive the updates for $\mathbf{B}$ and $\tilde{\mathbf{B}}$.

Updates for B

To obtain updates for $\mathbf{B} = (\boldsymbol{\Theta}, \mathbf{V}, \mathbf{Z})$, we exploit the fact that (A.1) is separable in $\boldsymbol{\Theta}, \mathbf{V}$, and $\mathbf{Z}$. Therefore, we can simply update with respect to $\boldsymbol{\Theta}, \mathbf{V}$, and $\mathbf{Z}$ one-at-a-time. Update for $\boldsymbol{\Theta}$ depends on the form of the convex loss function, and is addressed in the main text. Updates for $\mathbf{V}$ and $\mathbf{Z}$ can be easily seen to take the form given in Algorithm 1.

Updates for $\tilde{\mathbf{B}}$

Minimizing the function in (A.1) with respect to $\tilde{\mathbf{B}}$ is equivalent to

$$\begin{array}{ll} \underset{\tilde{\mathbf{\Theta}},\tilde{\mathbf{V}},\tilde{\mathbf{Z}}}{\text{minimize}} & \left\{\frac{\rho}{2}\|\mathbf{\Theta}-\tilde{\mathbf{\Theta}}+\mathbf{W}_1\|_F^2+\frac{\rho}{2}\|\mathbf{V}-\tilde{\mathbf{V}}+\mathbf{W}_2\|_F^2+\frac{\rho}{2}\|\mathbf{Z}-\tilde{\mathbf{Z}}+\mathbf{W}_3\|_F^2\right\} \\ \text{subject to} & \tilde{\mathbf{\Theta}}=\tilde{\mathbf{Z}}+\tilde{\mathbf{V}}+\tilde{\mathbf{V}}^T. \end{array} \tag{A.2}$$

Let $\mathbf{\Gamma}$ be the $p \times p$ Lagrange multiplier matrix for the equality constraint. Then, the Lagrangian for (A.2) is

$$\frac{\rho}{2}\|\mathbf{\Theta}-\tilde{\mathbf{\Theta}}+\mathbf{W}_1\|_F^2+\frac{\rho}{2}\|\mathbf{V}-\tilde{\mathbf{V}}+\mathbf{W}_2\|_F^2+\frac{\rho}{2}\|\mathbf{Z}-\tilde{\mathbf{Z}}+\mathbf{W}_3\|_F^2+\langle\mathbf{\Gamma},\tilde{\mathbf{\Theta}}-\tilde{\mathbf{Z}}-\tilde{\mathbf{V}}-\tilde{\mathbf{V}}^T\rangle.$$

A little bit of algebra yields

$$\tilde{\mathbf{\Theta}}=\mathbf{\Theta}+\mathbf{W}_1-\frac{1}{\rho}\mathbf{\Gamma},$$

$$\tilde{\mathbf{V}}=\frac{1}{\rho}(\mathbf{\Gamma}+\mathbf{\Gamma}^T)+\mathbf{V}+\mathbf{W}_2,$$

$$\tilde{\mathbf{Z}}=\frac{1}{\rho}\mathbf{\Gamma}+\mathbf{Z}+\mathbf{W}_3,$$

where $\mathbf{\Gamma}=\frac{\rho}{6}[(\mathbf{\Theta}+\mathbf{W}_1)-(\mathbf{V}+\mathbf{W}_2)-(\mathbf{V}+\mathbf{W}_2)^T-(\mathbf{Z}+\mathbf{W}_3)]$.

A.2 Conditions for HGL Solution to be Block-Diagonal

We begin by introducing some notation. Let $\|\mathbf{V}\|_{u,v}$ be the ℓ_u/ℓ_v norm of a matrix $\mathbf{V}$. For instance, $\|\mathbf{V}\|_{1,q}=\sum_{j=1}^p\|\mathbf{V}_j\|_q$. We define the support of a matrix $\mathbf{\Theta}$ as follows: $\text{supp}(\mathbf{\Theta})=\{(i,j):\Theta_{ij}\neq 0\}$. We say that $\mathbf{\Theta}$ is supported on a set $\mathcal{G}$ if $\text{supp}(\mathbf{\Theta})\subseteq\mathcal{G}$. Let $\{C_1,\ldots,C_K\}$ be a partition of the index set $\{1,\ldots,p\}$, and let $\mathcal{T}=\cup_{k=1}^K\{C_k\times C_k\}$. We let $\mathbf{A}_\mathcal{T}$ denote the restriction of the matrix $\mathbf{A}$ to the set $\mathcal{T}$: that is, $(\mathbf{A}_\mathcal{T})_{ij}=0$ if $(i,j)\notin\mathcal{T}$ and $(\mathbf{A}_\mathcal{T})_{ij}=A_{ij}$ if $(i,j)\in\mathcal{T}$. Note that any matrix supported on $\mathcal{T}$ is block-diagonal with K blocks, subject to some permutation of its rows and columns. Also, let $S_{\max}=\max_{(i,j)\in\mathcal{T}^c}|S_{ij}|$. Define

$$\begin{array}{rcll} \tilde{\mathbf{P}}(\mathbf{\Theta}) & = & \min\limits_{\mathbf{V},\mathbf{Z}} & \|\mathbf{Z}-\text{diag}(\mathbf{Z})\|_1+\hat{\lambda}_2\|\mathbf{V}-\text{diag}(\mathbf{V})\|_1+\hat{\lambda}_3\|\mathbf{V}-\text{diag}(\mathbf{V})\|_{1,q} \\ & & \text{subject to} & \mathbf{\Theta}=\mathbf{Z}+\mathbf{V}+\mathbf{V}^T, \end{array} \tag{A.3}$$

where $\hat{\lambda}_2 = \frac{\lambda_2}{\lambda_1}$ and $\hat{\lambda}_3 = \frac{\lambda_3}{\lambda_1}$. Then, optimization problem (2.6) is equivalent to

$$\underset{\boldsymbol{\Theta}\in\mathcal{S}}{\text{minimize}} \quad -\log\det(\boldsymbol{\Theta}) + \langle\boldsymbol{\Theta}, \mathbf{S}\rangle + \lambda_1\tilde{\mathbf{P}}(\boldsymbol{\Theta}), \tag{A.4}$$

where $\mathcal{S} = \{\boldsymbol{\Theta} : \boldsymbol{\Theta} \succ 0, \boldsymbol{\Theta} = \boldsymbol{\Theta}^T\}$.

Proof of Theorem 2.1 (Sufficient Condition)

Proof. First, we note that if $(\boldsymbol{\Theta}, \mathbf{V}, \mathbf{Z})$ is a feasible solution to (2.6), then $(\boldsymbol{\Theta}_{\mathcal{T}}, \mathbf{V}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{T}})$ is also a feasible solution to (2.6). Assume that $(\boldsymbol{\Theta}, \mathbf{V}, \mathbf{Z})$ is not supported on $\mathcal{T}$. We want to show that the objective value of (2.6) evaluated at $(\boldsymbol{\Theta}_{\mathcal{T}}, \mathbf{V}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{T}})$ is smaller than the objective value of (2.6) evaluated at $(\boldsymbol{\Theta}, \mathbf{V}, \mathbf{Z})$. By Fischer's inequality [49],

$$-\log\det(\boldsymbol{\Theta}) \geq -\log\det(\boldsymbol{\Theta}_{\mathcal{T}}).$$

Therefore, it remains to show that

$$\langle\boldsymbol{\Theta}, \mathbf{S}\rangle + \lambda_1\|\mathbf{Z} - \operatorname{diag}(\mathbf{Z})\|_1 + \lambda_2\|\mathbf{V} - \operatorname{diag}(\mathbf{V})\|_1 + \lambda_3\|\mathbf{V} - \operatorname{diag}(\mathbf{V})\|_{1,q} \quad >$$
$$\langle\boldsymbol{\Theta}_{\mathcal{T}}, \mathbf{S}\rangle + \lambda_1\|\mathbf{Z}_{\mathcal{T}} - \operatorname{diag}(\mathbf{Z}_{\mathcal{T}})\|_1 + \lambda_2\|\mathbf{V}_{\mathcal{T}} - \operatorname{diag}(\mathbf{V}_{\mathcal{T}})\|_1 + \lambda_3\|\mathbf{V}_{\mathcal{T}} - \operatorname{diag}(\mathbf{V}_{\mathcal{T}})\|_{1,q},$$

or equivalently, that

$$\langle\boldsymbol{\Theta}_{\mathcal{T}^c}, \mathbf{S}\rangle + \lambda_1\|\mathbf{Z}_{\mathcal{T}^c}\|_1 + \lambda_2\|\mathbf{V}_{\mathcal{T}^c}\|_1 + \lambda_3(\|\mathbf{V} - \operatorname{diag}(\mathbf{V})\|_{1,q} - \|\mathbf{V}_{\mathcal{T}} - \operatorname{diag}(\mathbf{V}_{\mathcal{T}})\|_{1,q}) > 0.$$

Since $\|\mathbf{V} - \operatorname{diag}(\mathbf{V})\|_{1,q} \geq \|\mathbf{V}_{\mathcal{T}} - \operatorname{diag}(\mathbf{V}_{\mathcal{T}})\|_{1,q}$, it suffices to show that

$$\langle\boldsymbol{\Theta}_{\mathcal{T}^c}, \mathbf{S}\rangle + \lambda_1\|\mathbf{Z}_{\mathcal{T}^c}\|_1 + \lambda_2\|\mathbf{V}_{\mathcal{T}^c}\|_1 > 0. \tag{A.5}$$

Note that $\langle\boldsymbol{\Theta}_{\mathcal{T}^c}, \mathbf{S}\rangle = \langle\boldsymbol{\Theta}_{\mathcal{T}^c}, \mathbf{S}_{\mathcal{T}^c}\rangle$. By the sufficient condition, $S_{\max} < \lambda_1$ and $2S_{\max} < \lambda_2$. In addition, we have that

$$\begin{aligned}
|\langle\boldsymbol{\Theta}_{\mathcal{T}^c}, \mathbf{S}\rangle| &= |\langle\boldsymbol{\Theta}_{\mathcal{T}^c}, \mathbf{S}_{\mathcal{T}^c}\rangle| \\
&= |\langle\mathbf{V}_{\mathcal{T}^c} + \mathbf{V}_{\mathcal{T}^c}^T + \mathbf{Z}_{\mathcal{T}^c}, \mathbf{S}_{\mathcal{T}^c}\rangle| \\
&= |\langle 2\mathbf{V}_{\mathcal{T}^c} + \mathbf{Z}_{\mathcal{T}^c}, \mathbf{S}_{\mathcal{T}^c}\rangle| \\
&\leq (2\|\mathbf{V}_{\mathcal{T}^c}\|_1 + \|\mathbf{Z}_{\mathcal{T}^c}\|_1)S_{\max} \\
&< \ \lambda_2\|\mathbf{V}_{\mathcal{T}^c}\|_1 + \lambda_1\|\mathbf{Z}_{\mathcal{T}^c}\|_1,
\end{aligned}$$

where the last inequality follows from the sufficient condition. We have shown (A.5) as desired. □

Proof of Theorem 2.2 (Necessary Condition)

We first present a simple lemma for proving Theorem 2.2. Throughout the proof of Theorem 2.2, $\|\cdot\|_\infty$ indicates the maximal absolute element of a matrix and $\|\cdot\|_{\infty,s}$ indicates the dual norm of $\|\cdot\|_{1,q}$.

Lemma A.1. *The dual representation of* $\tilde{\mathbf{P}}(\mathbf{\Theta})$ *in (A.3) is*

$$\begin{aligned} \tilde{\mathbf{P}}^*(\mathbf{\Theta}) \;=\; & \max_{\mathbf{X},\mathbf{Y},\mathbf{\Lambda}} \quad \langle \mathbf{\Lambda}, \mathbf{\Theta} \rangle \\ & \text{subject to} \quad \mathbf{\Lambda} + \mathbf{\Lambda}^T = \hat{\lambda}_2 \mathbf{X} + \hat{\lambda}_3 \mathbf{Y} \\ & \qquad\qquad \|\mathbf{X}\|_\infty \le 1, \|\mathbf{\Lambda}\|_\infty \le 1, \|\mathbf{Y}\|_{\infty,s} \le 1 \\ & \qquad\qquad X_{ii} = 0, Y_{ii} = 0, \Lambda_{ii} = 0 \textit{ for } i = 1, \ldots, p, \end{aligned} \tag{A.6}$$

where $\frac{1}{s} + \frac{1}{q} = 1$.

Proof. We first state the dual representations for the norms in (A.3):

$$\begin{aligned} \|\mathbf{Z} - \mathrm{diag}(\mathbf{Z})\|_1 \;=\; & \max_{\mathbf{\Lambda}} \quad \langle \mathbf{\Lambda}, \mathbf{Z} \rangle \\ & \text{subject to} \quad \|\mathbf{\Lambda}\|_\infty \le 1, \Lambda_{ii} = 0 \text{ for } i = 1, \ldots, p, \end{aligned}$$

$$\begin{aligned} \|\mathbf{V} - \mathrm{diag}(\mathbf{V})\|_1 \;=\; & \max_{\mathbf{X}} \quad \langle \mathbf{X}, \mathbf{V} \rangle \\ & \text{subject to} \quad \|\mathbf{X}\|_\infty \le 1, X_{ii} = 0 \text{ for } i = 1, \ldots, p, \end{aligned}$$

$$\begin{aligned} \|\mathbf{V} - \mathrm{diag}(\mathbf{V})\|_{1,q} \;=\; & \max_{\mathbf{Y}} \quad \langle \mathbf{Y}, \mathbf{V} \rangle \\ & \text{subject to} \quad \|\mathbf{Y}\|_{\infty,s} \le 1, Y_{ii} = 0 \text{ for } i = 1, \ldots, p. \end{aligned}$$

Then,

$$\begin{aligned}
\tilde{\mathbf{P}}(\boldsymbol{\Theta}) \;=\; & \min_{\mathbf{V},\mathbf{Z}} && \|\mathbf{Z}-\mathrm{diag}(\mathbf{Z})\|_1+\hat{\lambda}_2\|\mathbf{V}-\mathrm{diag}(\mathbf{V})\|_1+\hat{\lambda}_3\|\mathbf{V}-\mathrm{diag}(\mathbf{V})\|_{1,q} \\
& \text{subject to} && \boldsymbol{\Theta}=\mathbf{Z}+\mathbf{V}+\mathbf{V}^T \\
= \; & \min_{\mathbf{V},\mathbf{Z}} && \max_{\boldsymbol{\Lambda},\mathbf{X},\mathbf{Y}} \langle\boldsymbol{\Lambda},\mathbf{Z}\rangle+\hat{\lambda}_2\langle\mathbf{X},\mathbf{V}\rangle+\hat{\lambda}_3\langle\mathbf{Y},\mathbf{V}\rangle \\
& \text{subject to} && \|\boldsymbol{\Lambda}\|_\infty\le 1,\|\mathbf{X}\|_\infty\le 1,\|\mathbf{Y}\|_{\infty,s}\le 1 \\
& && \Lambda_{ii}=0,X_{ii}=0,Y_{ii}=0 \text{ for } i=1,\ldots,p \\
& && \boldsymbol{\Theta}=\mathbf{Z}+\mathbf{V}+\mathbf{V}^T \\
= \; & \max_{\boldsymbol{\Lambda},\mathbf{X},\mathbf{Y}} && \min_{\mathbf{V},\mathbf{Z}} \langle\boldsymbol{\Lambda},\mathbf{Z}\rangle+\hat{\lambda}_2\langle\mathbf{X},\mathbf{V}\rangle+\hat{\lambda}_3\langle\mathbf{Y},\mathbf{V}\rangle \\
& \text{subject to} && \|\boldsymbol{\Lambda}\|_\infty\le 1,\|\mathbf{X}\|_\infty\le 1,\|\mathbf{Y}\|_{\infty,s}\le 1 \\
& && \Lambda_{ii}=0,X_{ii}=0,Y_{ii}=0 \text{ for } i=1,\ldots,p \\
& && \boldsymbol{\Theta}=\mathbf{Z}+\mathbf{V}+\mathbf{V}^T \\
= \; & \max_{\boldsymbol{\Lambda},\mathbf{X},\mathbf{Y}} && \langle\boldsymbol{\Lambda},\boldsymbol{\Theta}\rangle \\
& \text{subject to} && \boldsymbol{\Lambda}+\boldsymbol{\Lambda}^T=\hat{\lambda}_2\mathbf{X}+\hat{\lambda}_3\mathbf{Y} \\
& && \|\mathbf{X}\|_\infty\le 1,\|\boldsymbol{\Lambda}\|_\infty\le 1,\|\mathbf{Y}\|_{\infty,s}\le 1 \\
& && X_{ii}=0,Y_{ii}=0,\Lambda_{ii}=0 \text{ for } i=1,\ldots,p.
\end{aligned}$$

The third equality holds since the constraints on $(\mathbf{V},\mathbf{Z})$ and on $(\boldsymbol{\Lambda},\mathbf{X},\mathbf{Y})$ are both compact convex sets and so by the minimax theorem, we can swap max and min. The last equality follows from the fact that

$$\begin{aligned}
& \min_{\mathbf{V},\mathbf{Z}} && \langle\boldsymbol{\Lambda},\mathbf{Z}\rangle+\hat{\lambda}_2\langle\mathbf{X},\mathbf{V}\rangle+\hat{\lambda}_3\langle\mathbf{Y},\mathbf{V}\rangle \\
& \text{subject to} && \boldsymbol{\Theta}=\mathbf{Z}+\mathbf{V}+\mathbf{V}^T \\
= \; & && \begin{cases} \langle\boldsymbol{\Lambda},\boldsymbol{\Theta}\rangle & \text{if } \boldsymbol{\Lambda}+\boldsymbol{\Lambda}^T=\hat{\lambda}_2\mathbf{X}+\hat{\lambda}_3\mathbf{Y} \\ -\infty & \text{otherwise.} \end{cases}
\end{aligned}$$

□

We now present the proof of Theorem 2.2.

Proof. The optimality condition for (A.4) is given by

$$\mathbf{0}=-\boldsymbol{\Theta}^{-1}+\mathbf{S}+\lambda_1\boldsymbol{\Lambda}, \tag{A.7}$$

where $\mathbf{\Lambda}$ is a subgradient of $\tilde{\mathbf{P}}(\mathbf{\Theta})$ in (A.3) and the left-hand side of the above equation is a zero matrix of size $p \times p$.

Now suppose that $\mathbf{\Theta}^*$ that solves (A.7) is supported on $\mathcal{T}$, i.e., $\mathbf{\Theta}^*_{\mathcal{T}^c} = 0$. Then for any $(i,j) \in \mathcal{T}^c$, we have that

$$0 = S_{ij} + \lambda_1 \Lambda^*_{ij}, \tag{A.8}$$

where $\mathbf{\Lambda}^*$ is a subgradient of $\tilde{\mathbf{P}}(\mathbf{\Theta}^*)$. Note that $\mathbf{\Lambda}^*$ must be an optimal solution to the optimization problem (A.6). Therefore, it is also a feasible solution to (A.6), implying that

$$\begin{aligned} |\Lambda^*_{ij} + \Lambda^*_{ji}| &\leq \hat{\lambda}_2 + \hat{\lambda}_3, \\ |\Lambda^*_{ij}| &\leq 1. \end{aligned}$$

From (A.8), we have that $\Lambda^*_{ij} = -\frac{S_{ij}}{\lambda_1}$ and thus,

$$\begin{aligned} \lambda_1 &\geq \lambda_1 \max_{(i,j)\in\mathcal{T}^c} |\Lambda^*_{ij}| \\ &= \lambda_1 \max_{(i,j)\in\mathcal{T}^c} \frac{|S_{ij}|}{\lambda_1} \\ &= S_{\max}. \end{aligned}$$

Also, recall that $\hat{\lambda}_2 = \frac{\lambda_2}{\lambda_1}$ and $\hat{\lambda}_3 = \frac{\lambda_3}{\lambda_1}$. We have that

$$\begin{aligned} \lambda_2 + \lambda_3 &\geq \lambda_1 \max_{(i,j)\in\mathcal{T}^c} |\Lambda^*_{ij} + \Lambda^*_{ji}| \\ &= \lambda_1 \max_{(i,j)\in\mathcal{T}^c} \frac{2|S_{ij}|}{\lambda_1} \\ &= 2S_{\max}. \end{aligned}$$

Hence, we obtain the desired result. □

A.3 Some Properties of HGL

Proof of Lemma 2.1

Proof. Let $(\mathbf{\Theta}^*, \mathbf{Z}^*, \mathbf{V}^*)$ be the solution to (2.6) and suppose that $\mathbf{Z}^*$ is not a diagonal matrix. Note that $\mathbf{Z}^*$ is symmetric since $\mathbf{\Theta} \in \mathcal{S} \equiv \{\mathbf{\Theta} : \mathbf{\Theta} \succ 0 \text{ and } \mathbf{\Theta} = \mathbf{\Theta}^T\}$. Let $\hat{\mathbf{Z}} = \text{diag}(\mathbf{Z}^*)$, a matrix that contains the diagonal elements of the matrix $\mathbf{Z}^*$. Also, construct $\hat{\mathbf{V}}$ as follows,

$$\hat{\mathbf{V}}_{ij} = \begin{cases} \mathbf{V}^*_{ij} + \frac{\mathbf{Z}^*_{ij}}{2} & \text{if } i \neq j \\ \mathbf{V}^*_{jj} & \text{otherwise.} \end{cases}$$

Then, we have that $\mathbf{\Theta}^* = \hat{\mathbf{Z}} + \hat{\mathbf{V}} + \hat{\mathbf{V}}^T$. Thus, $(\mathbf{\Theta}^*, \hat{\mathbf{Z}}, \hat{\mathbf{V}})$ is a feasible solution to (2.6). We now show that $(\mathbf{\Theta}^*, \hat{\mathbf{Z}}, \hat{\mathbf{V}})$ has a smaller objective than $(\mathbf{\Theta}^*, \mathbf{Z}^*, \mathbf{V}^*)$ in (2.6), giving us a contradiction. Note that

$$\begin{aligned} \lambda_1\|\hat{\mathbf{Z}} - \text{diag}(\hat{\mathbf{Z}})\|_1 + \lambda_2\|\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}})\|_1 &= \lambda_2\|\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}})\|_1 \\ &= \lambda_2 \textstyle\sum_{i\neq j} |\mathbf{V}^*_{ij} + \frac{\mathbf{Z}^*_{ij}}{2}| \\ &\leq \lambda_2\|\mathbf{V}^* - \text{diag}(\mathbf{V}^*)\|_1 + \tfrac{\lambda_2}{2}\|\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*)\|_1, \end{aligned}$$

and

$$\begin{aligned} & \lambda_3 \textstyle\sum_{j=1}^p \|(\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}}))_j\|_q \\ \leq\ & \lambda_3 \textstyle\sum_{j=1}^p \|(\mathbf{V}^* - \text{diag}(\mathbf{V}^*))_j\|_q + \tfrac{\lambda_3}{2} \textstyle\sum_{j=1}^p \|(\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*))_j\|_q \\ \leq\ & \lambda_3 \textstyle\sum_{j=1}^p \|(\mathbf{V}^* - \text{diag}(\mathbf{V}^*))_j\|_q + \tfrac{\lambda_3}{2}\|\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*)\|_1, \end{aligned}$$

where the last inequality follows from the fact that for any vector $\mathbf{x} \in \mathbb{R}^p$ and $q \geq 1$, $\|\mathbf{x}\|_q$ is a nonincreasing function of q [36].

Summing up the above inequalities, we get that

$$\begin{aligned} \lambda_1\|\hat{\mathbf{Z}} - \text{diag}(\hat{\mathbf{Z}})\|_1 + \lambda_2\|\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}})\|_1 + \lambda_3 \textstyle\sum_{j=1}^p \|(\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}}))_j\|_q &\leq \\ \tfrac{\lambda_2+\lambda_3}{2}\|\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*)\|_1 + \lambda_2\|\mathbf{V}^* - \text{diag}(\mathbf{V}^*)\|_1 + \lambda_3 \textstyle\sum_{j=1}^p \|(\mathbf{V}^* - \text{diag}(\mathbf{V}^*))_j\|_q &< \\ \lambda_1\|\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*)\|_1 + \lambda_2\|\mathbf{V}^* - \text{diag}(\mathbf{V}^*)\|_1 + \lambda_3 \textstyle\sum_{j=1}^p \|(\mathbf{V}^* - \text{diag}(\mathbf{V}^*))_j\|_q, & \end{aligned}$$

where the last inequality uses the assumption that $\lambda_1 > \frac{\lambda_2+\lambda_3}{2}$. We arrive at a contradiction and therefore the result holds. □

Proof of Lemma 2.2

Proof. Let $(\mathbf{\Theta}^*, \mathbf{Z}^*, \mathbf{V}^*)$ be the solution to (2.6) and suppose $\mathbf{V}^*$ is not a diagonal matrix. Let $\hat{\mathbf{V}} = \text{diag}(\mathbf{V}^*)$, a diagonal matrix that contains the diagonal elements of $\mathbf{V}^*$. Also construct $\hat{\mathbf{Z}}$ as follows,

$$\hat{\mathbf{Z}}_{ij} = \begin{cases} \mathbf{Z}^*_{ij} + \mathbf{V}^*_{ij} + \mathbf{V}^*_{ji} & \text{if } i \neq j \\ \mathbf{Z}^*_{ij} & \text{otherwise.} \end{cases}$$

Then, we have that $\mathbf{\Theta}^* = \hat{\mathbf{V}} + \hat{\mathbf{V}}^T + \hat{\mathbf{Z}}$. We now show that $(\mathbf{\Theta}^*, \hat{\mathbf{Z}}, \hat{\mathbf{V}})$ has a smaller objective value than $(\mathbf{\Theta}^*, \mathbf{Z}^*, \mathbf{V}^*)$ in (2.6), giving us a contradiction. We start by noting that

$$\begin{aligned} \lambda_1\|\hat{\mathbf{Z}} - \text{diag}(\hat{\mathbf{Z}})\|_1 + \lambda_2\|\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}})\|_1 &= \lambda_1\|\hat{\mathbf{Z}} - \text{diag}(\hat{\mathbf{Z}})\|_1 \\ &\leq \lambda_1\|\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*)\|_1 + 2\lambda_1\|\mathbf{V}^* - \text{diag}(\mathbf{V}^*)\|_1. \end{aligned}$$

By Holder's Inequality, we know that $\mathbf{x}^T\mathbf{y} \leq \|\mathbf{x}\|_q\|\mathbf{y}\|_s$ where $\frac{1}{s} + \frac{1}{q} = 1$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{p-1}$. Setting $\mathbf{y} = \text{sign}(\mathbf{x})$, we have that $\|\mathbf{x}\|_1 \leq (p-1)^{\frac{1}{s}}\|\mathbf{x}\|_q$. Consequently,

$$\frac{\lambda_3}{(p-1)^{\frac{1}{s}}}\|\mathbf{V}^* - \text{diag}(\mathbf{V}^*)\|_1 \leq \lambda_3 \sum_{j=1}^{p} \|(\mathbf{V}^* - \text{diag}(\mathbf{V}^*))_j\|_q.$$

Combining these results, we have that

$$\begin{aligned} &\lambda_1\|\hat{\mathbf{Z}} - \text{diag}(\hat{\mathbf{Z}})\|_1 + \lambda_2\|\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}})\|_1 + \lambda_3 \sum_{j=1}^{p} \|(\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}}))_j\|_q \\ &\leq \lambda_1\|\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*)\|_1 + 2\lambda_1\|\mathbf{V}^* - \text{diag}(\mathbf{V}^*)\|_1 \\ &< \lambda_1\|\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*)\|_1 + \left(\lambda_2 + \frac{\lambda_3}{(p-1)^{\frac{1}{s}}}\right)\|\mathbf{V}^* - \text{diag}(\mathbf{V}^*)\|_1 \\ &\leq \lambda_1\|\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*)\|_1 + \lambda_2\|\mathbf{V}^* - \text{diag}(\mathbf{V}^*)\|_1 + \lambda_3 \sum_{j=1}^{p} \|(\mathbf{V}^* - \text{diag}(\mathbf{V}^*))_j\|_q, \end{aligned}$$

where we use the assumption that $\lambda_1 < \frac{\lambda_2}{2} + \frac{\lambda_3}{2(p-1)^{\frac{1}{s}}}$. This leads to a contradiction. □

Proof of Lemma 2.3

In this proof, we consider the case when $\lambda_1 > \frac{\lambda_2+\lambda_3}{2}$. A similar proof technique can be used to prove the case when $\lambda_1 < \frac{\lambda_2+\lambda_3}{2}$.

Proof. Let $f(\mathbf{\Theta}, \mathbf{V}, \mathbf{Z})$ denote the objective of (2.6) with $q = 1$, and $(\mathbf{\Theta}^*, \mathbf{V}^*, \mathbf{Z}^*)$ the optimal solution. By Lemma 2.1, the assumption that $\lambda_1 > \frac{\lambda_2+\lambda_3}{2}$ implies that $\mathbf{Z}^*$ is a diagonal matrix. Now let $\hat{\mathbf{V}} = \frac{1}{2}\left(\mathbf{V}^* + (\mathbf{V}^*)^T\right)$. Then

$$\begin{aligned}
& f(\mathbf{\Theta}^*, \hat{\mathbf{V}}, \mathbf{Z}^*) \\
= \; & -\log\det\mathbf{\Theta}^* + \langle\mathbf{\Theta}^*, \mathbf{S}\rangle + \lambda_1\|\mathbf{Z}^* - \mathrm{diag}(\mathbf{Z}^*)\|_1 + (\lambda_2+\lambda_3)\|\hat{\mathbf{V}} - \mathrm{diag}(\hat{\mathbf{V}})\|_1 \\
= \; & -\log\det\mathbf{\Theta}^* + \langle\mathbf{\Theta}^*, \mathbf{S}\rangle + \frac{\lambda_2+\lambda_3}{2}\|\mathbf{V}^* + \mathbf{V}^{*T} - \mathrm{diag}(\mathbf{V}^* + \mathbf{V}^{*T})\|_1 \\
\leq \; & -\log\det\mathbf{\Theta}^* + \langle\mathbf{\Theta}^*, \mathbf{S}\rangle + (\lambda_2+\lambda_3)\|\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*)\|_1 \\
= \; & f(\mathbf{\Theta}^*, \mathbf{V}^*, \mathbf{Z}^*) \\
\leq \; & f(\mathbf{\Theta}^*, \hat{\mathbf{V}}, \mathbf{Z}^*),
\end{aligned}$$

where the last inequality follows from the assumption that $(\mathbf{\Theta}^*, \mathbf{V}^*, \mathbf{Z}^*)$ solves (2.6). By strict convexity of f, this means that $\mathbf{V}^* = \hat{\mathbf{V}}$, i.e., $\mathbf{V}^*$ is symmetric. This implies that

$$\begin{aligned}
f(\mathbf{\Theta}^*, \mathbf{V}^*, \mathbf{Z}^*) &= -\log\det\mathbf{\Theta}^* + \langle\mathbf{\Theta}^*, \mathbf{S}\rangle + \frac{\lambda_2+\lambda_3}{2}\|\mathbf{V}^* + \mathbf{V}^{*T} - \mathrm{diag}(\mathbf{V}^* + \mathbf{V}^{*T})\|_1 \\
&= -\log\det\mathbf{\Theta}^* + \langle\mathbf{\Theta}^*, \mathbf{S}\rangle + \frac{\lambda_2+\lambda_3}{2}\|\mathbf{\Theta}^* - \mathrm{diag}(\mathbf{\Theta}^*)\|_1 \qquad \text{(A.9)} \\
&= g(\mathbf{\Theta}^*),
\end{aligned}$$

where $g(\mathbf{\Theta})$ is the objective of the graphical lasso optimization problem, evaluated at $\mathbf{\Theta}$, with tuning parameter $\frac{\lambda_2+\lambda_3}{2}$. Suppose that $\tilde{\mathbf{\Theta}}$ minimizes $g(\mathbf{\Theta})$, and $\mathbf{\Theta}^* \neq \tilde{\mathbf{\Theta}}$. Then, by (A.9) and strict convexity of g, $g(\mathbf{\Theta}^*) = f(\mathbf{\Theta}^*, \mathbf{V}^*, \mathbf{Z}^*) \leq f(\tilde{\mathbf{\Theta}}, \tilde{\mathbf{\Theta}}/2, \mathbf{0}) = g(\tilde{\mathbf{\Theta}}) < g(\mathbf{\Theta}^*)$, giving us a contradiction. Thus it must be that $\tilde{\mathbf{\Theta}} = \mathbf{\Theta}^*$. □

A.4 Simulation Study for Hub Covariance Graph

In this section, we present the results for the simulation study described in Section 2.4.2 with $n = 100$, $p = 200$, and $|\mathcal{H}| = 4$. We calculate the proportion of correctly estimated hub nodes with $r = 40$. The results are shown in Figure A.1. As we can see from Figure A.1, our proposal outperforms [8]. In particular, we can see from Figure A.1(c) that [8] fails to identify hub nodes.

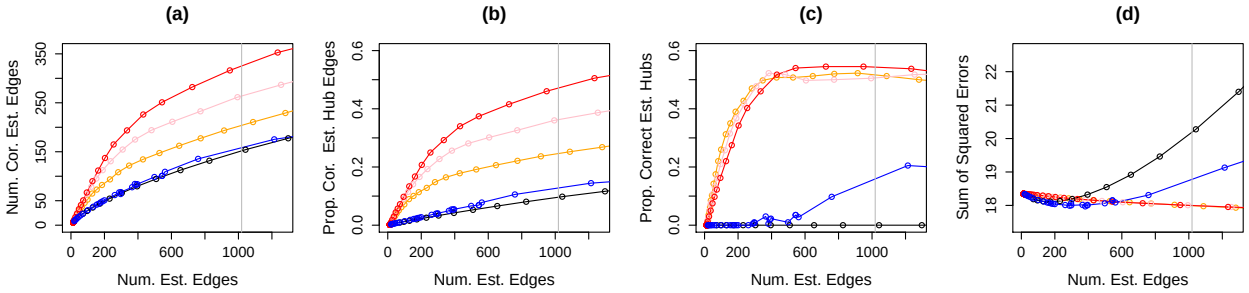

Figure A.1: Covariance graph simulation with $n = 100$ and $p = 200$. Details of the axis labels are as in Figure 2.3. The colored lines correspond to the proposal of [110] (——); HCG with $\lambda_3 = 1$ (——), $\lambda_3 = 1.5$ (——), and $\lambda_3 = 2$ (——); and the proposal of [8] (——).

A.5 Run Time Study for the ADMM algorithm for HGL

In this section, we present a more extensive run time study for the ADMM algorithm for HGL. We ran experiments with $p = 100, 200, 300$ and with $n = p/2$ on a 2.26GHz Intel Core 2 Duo machine. Results averaged over 10 replications are displayed in Figures A.2(a)-(b), where the panels depict the run time and number of iterations required for the algorithm to converge, as a function of λ_1, with $\lambda_2 = 0.5$ and $\lambda_3 = 2$ fixed. The number of iterations required for the algorithm to converge is computed as the total number of iterations in Step 2 of Algorithm 1. We see from Figure A.2(a) that as p increases from 100 to 300, the run times increase substantially, but never exceed several minutes. Note that these results are without using the block diagonal condition in Theorem 2.1.

A.6 Update for Θ in Step 2(a)i in Algorithm 1 for Binary Ising Model using Barzilai-Borwein Method

We consider updating $\mathbf{\Theta}$ in Step 2(a)i of Algorithm 1 for binary Ising model. Let

$$h(\mathbf{\Theta}) = -\sum_{j=1}^{p}\sum_{j'=1}^{p}\theta_{jj'}(\mathbf{X}^T\mathbf{X})_{jj'} + \sum_{i=1}^{p}\sum_{j=1}^{p}\log\left(1+\exp\left[\theta_{jj}+\sum_{j'\neq j}\theta_{jj'}x_{ij'}\right]\right) + \frac{\rho}{2}\|\mathbf{\Theta}-\tilde{\mathbf{\Theta}}+\mathbf{W}_1\|_F^2.$$

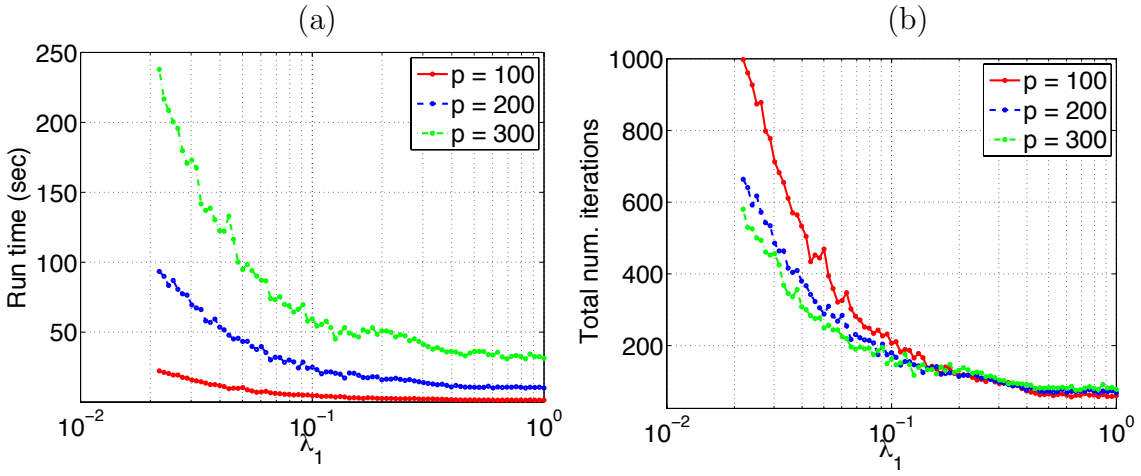

Figure A.2: (a): Run time (in seconds) of the ADMM algorithm for HGL, as a function of λ_1, for fixed values of λ_2 and λ_3. (b): The total number of iterations required for the ADMM algorithm for HGL to converge, as a function of λ_1. All results are averaged over 10 simulated data sets. These results are without using the block diagonal condition in Theorem 2.1.

Then, the optimization problem for Step 2(a)i of Algorithm 1 is

$$\underset{\mathbf{\Theta}\in\mathcal{S}}{\text{minimize}} \quad h(\mathbf{\Theta}), \tag{A.10}$$

where $\mathcal{S} = \{\mathbf{\Theta} : \mathbf{\Theta} = \mathbf{\Theta}^T\}$. In solving (A.10), we will treat $\mathbf{\Theta} \in \mathcal{S}$ as an implicit constraint.

The Barzilai-Borwein method is a gradient descent method with the step-size chosen to mimic the secant condition of the BFGS method (see, for example, [6, 75]). The convergence of the Barzilai-Borwein method for unconstrained minimization using a non-monotone line search was shown in [85]. Recent convergence results for a quadratic cost function can be found in [20]. To implement the Barzilai-Borwein method, we need to evaluate the gradient of $h(\mathbf{\Theta})$. Let $\nabla h(\mathbf{\Theta})$ be a $p \times p$ matrix, where the (j, j') entry is the gradient of $h(\mathbf{\Theta})$ with respect to $\theta_{jj'}$, computed under the constraint $\mathbf{\Theta} \in \mathcal{S}$, that is, $\theta_{jj'} = \theta_{j'j}$. Then,

$$(\nabla h(\mathbf{\Theta}))_{jj} = -(\mathbf{X}^T\mathbf{X})_{jj} + \sum_{i=1}^{n}\left[\frac{\exp(\theta_{jj} + \sum_{j'\neq j}\theta_{jj'}x_{ij'})}{1+\exp(\theta_{jj} + \sum_{j'\neq j}\theta_{jj'}x_{ij'})}\right] + \rho(\theta_{jj} - \tilde{\theta}_{jj} + (\mathbf{W}_1)_{jj}),$$

and

$$(\nabla h(\mathbf{\Theta}))_{jj'} = -2(\mathbf{X}^T\mathbf{X})_{jj} + 2\rho(\theta_{jj'} - \tilde{\theta}_{jj'} + (\mathbf{W}_1)_{jj'}) \\ + \sum_{i=1}^{n}\left[\frac{x_{ij'}\exp(\theta_{jj} + \sum_{j'\neq j}\theta_{jj'}x_{ij'})}{1+\exp(\theta_{jj} + \sum_{j'\neq j}\theta_{jj'}x_{ij'})} + \frac{x_{ij}\exp(\theta_{j'j'} + \sum_{j\neq j'}\theta_{jj'}x_{ij})}{1+\exp(\theta_{j'j'} + \sum_{j\neq j'}\theta_{jj'}x_{ij})}\right].$$

A simple implementation of the Barzilai-Borwein algorithm for solving (A.10) is detailed in Algorithm 5. We note that the Barzilai-Borwein algorithm can be improved (see, for example, [6, 108]). We leave such improvement for future work.

Algorithm 5 Barzilai-Borwein Algorithm for Solving (A.10).

1. **Initialize** the parameters:
   (a) $\mathbf{\Theta}_1 = \mathbf{I}$ and $\mathbf{\Theta}_0 = 2\mathbf{I}$.
   (b) constant $\tau > 0$.
2. **Iterate** until the stopping criterion $\frac{\|\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1}\|_F^2}{\|\mathbf{\Theta}_{t-1}\|_F^2} \leq \tau$ is met, where $\mathbf{\Theta}_t$ is the value of $\mathbf{\Theta}$ obtained at the tth iteration:
   (a) $\alpha_t = \text{trace}\left[(\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1})^T(\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1})\right] / \text{trace}\left[(\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1})^T(\nabla h(\mathbf{\Theta}_t) - \nabla h(\mathbf{\Theta}_{t-1}))\right]$.
   (b) $\mathbf{\Theta}_{t+1} = \mathbf{\Theta}_t - \alpha_t \nabla h(\mathbf{\Theta}_t)$.

Appendix B

APPENDIX FOR CHAPTER 3

B.1 Proof of Lemma 3.1

Proof. Let $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ denote the SVD of $\mathbf{X}$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal $n \times n$ and $p \times p$ matrices and $\mathbf{D}$ is a $n \times p$ matrix with decreasing nonnegative diagonal elements. Note that any $n \times K$ orthogonal matrix $\mathbf{A}$ can be written as $\mathbf{A} = \mathbf{U}\boldsymbol{\alpha}$ for some orthogonal $n \times K$ matrix $\boldsymbol{\alpha}$, and any orthogonal $p \times K$ matrix $\mathbf{B}$ can be written as $\mathbf{B} = \mathbf{V}\boldsymbol{\beta}$ for some orthogonal $p \times K$ matrix $\boldsymbol{\beta}$. Thus, instead of solving (3.4), we can solve

$$\underset{\boldsymbol{\alpha}^T\boldsymbol{\alpha}=\mathbf{I}_K, \boldsymbol{\beta}^T\boldsymbol{\beta}=\mathbf{I}_K}{\text{maximize}} ||\boldsymbol{\alpha}^T\mathbf{D}\boldsymbol{\beta}||_F^2. \tag{B.1}$$

By inspection, (B.1) is solved by $\boldsymbol{\alpha} = \mathbf{I}_{n\times K}\mathbf{Q}_1$ and $\boldsymbol{\beta} = \mathbf{I}_{p\times K}\mathbf{Q}_2$, where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are any $K \times K$ orthogonal matrix, and where $\mathbf{I}_{n\times K}$ and $\mathbf{I}_{p\times K}$ and $n \times K$ are $p \times K$ identity matrices. Therefore, the solution to (3.4) takes the form $\mathbf{A} = \mathbf{U}\mathbf{I}_{n\times K}\mathbf{Q}_1 = \mathbf{U}_{1:K}\mathbf{Q}_1$, and $\mathbf{B} = \mathbf{V}\mathbf{I}_{p\times K}\mathbf{Q}_2 = \mathbf{V}_{1:K}\mathbf{Q}_2$. □

B.2 Proof of Lemma 3.2

Proof. We must minimize the quantity

$$\text{tr}(\boldsymbol{\Sigma}_k^{-1}(\mathbf{X}_{k,r} - \mu_{kr})\boldsymbol{\Delta}_r^{-1}(\mathbf{X}_{k,r} - \mu_{kr})^T) + 2\lambda|\mu_{kr}|$$

with respect to μ_{kr}. This amounts to minimizing

$$\mu_{kr}^2\text{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{1}^T) - 2\mu_{kr}\text{tr}(\boldsymbol{\Sigma}_k^{-1}\mathbf{1}\boldsymbol{\Delta}_r^{-1}\mathbf{X}_{k,r}^T) + 2\lambda|\mu_{kr}|,$$

where $\mathbf{1}$ is a $|C_k| \times |D_r|$ matrix with all entries equal to 1. Completing the square, we see that this is equivalent to minimizing

$$\left(\mu_{kr} \sqrt{\mathrm{tr}(\mathbf{\Sigma}_k^{-1} \mathbf{1} \mathbf{\Delta}_r^{-1} \mathbf{1}^T)} - \frac{\mathrm{tr}(\mathbf{\Sigma}_k^{-1} \mathbf{1} \mathbf{\Delta}_r^{-1} \mathbf{X}_{k,r}^T)}{\sqrt{\mathrm{tr}(\mathbf{\Sigma}_k^{-1} \mathbf{1} \mathbf{\Delta}_r^{-1} \mathbf{1}^T)}} \right)^2 + 2\lambda |\mu_{kr}|$$

with respect to μ_{kr}. The result follows directly. □

B.3 Proof of Theorem 3.1

Before we prove Theorem 3.1, we present a simple lemma.

Lemma B.1. *Let $\bar{X}$ denote the mean of the elements in $\mathbf{X}$. Then,*

$$\sum_{i=1}^{n} \sum_{j=1}^{p} (X_{ij} - \bar{X})^2 = \sum_{i=1}^{n} \sum_{j=1}^{p} X_{ij}^2 - np(\bar{X})^2 = \frac{1}{2np} \sum_{i=1}^{n} \sum_{j=1}^{p} \sum_{i'=1}^{n} \sum_{j'=1}^{p} (X_{ij} - X_{i'j'})^2. \quad \text{(B.2)}$$

Now we proceed with a proof of Theorem 3.1.

Proof. Problem (3.4) is equivalent to the problem

$$\underset{\mathbf{A}^T\mathbf{A}=\mathbf{I}_K, \mathbf{B}^T\mathbf{B}=\mathbf{I}_K}{\text{minimize}} \left\{ ||\mathbf{X}||_F^2 - ||\mathbf{A}^T \mathbf{X} \mathbf{B}||_F^2 \right\}, \quad \text{(B.3)}$$

which is equivalent to

$$\underset{\mathbf{A}^T\mathbf{A}=\mathbf{I}_K, \mathbf{B}^T\mathbf{B}=\mathbf{I}_K}{\text{minimize}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{p} X_{ij}^2 - \sum_{k=1}^{K} \sum_{r=1}^{K} (\sum_{i=1}^{n} \sum_{j=1}^{p} A_{ik} X_{ij} B_{jr})^2 \right\}. \quad \text{(B.4)}$$

Since (3.4) constrains $\mathbf{A}$ to be orthogonal, the two additional constraints in the theorem statement imply that the kth column of $\mathbf{A}$ contains exactly n_k elements that are equal to $\frac{1}{\sqrt{n_k}}$, and $n - n_k$ elements that equal zero. Moreover, the non-zero elements of each column of $\mathbf{A}$ are non-overlapping. A similar claim holds for $\mathbf{B}$. Let C_k denote the indices of the non-zero elements in the kth column of $\mathbf{A}$, and similarly let D_r denote the indices of the non-zero elements in the rth column of $\mathbf{B}$. Then (B.4) leads to

$$\underset{C_1,\ldots,C_K,D_1,\ldots,D_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \sum_{r=1}^{K} \left(\sum_{i \in C_k} \sum_{j \in D_r} X_{ij}^2 - n_k p_r (\frac{1}{n_k p_r} \sum_{i \in C_k} \sum_{j \in D_r} X_{ij})^2 \right) \right\}. \quad \text{(B.5)}$$

Finally, applying Lemma B.1 reveals that this is equivalent to

$$\underset{C_1,\ldots,C_K,D_1,\ldots,D_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \sum_{r=1}^{K} \sum_{i \in C_k} \sum_{j \in D_r} (X_{ij} - \bar{X}_{kr})^2 \right\}. \tag{B.6}$$

Now one can easily show that this is equivalent to the biclustering optimization problem in equation 3.1 in the case that $K = R$. □

Appendix C

APPENDIX FOR CHAPTER 4

C.1 Proof of Lemma 4.2

Proof. We rewrite problem (4.2) as

$$\underset{\mathbf{u}\in\mathbb{R}^{np},\boldsymbol{\gamma}\in\mathbb{R}^{\left[p\cdot\binom{n}{2}\right]}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2+\lambda \mathrm{P}_q(\boldsymbol{\gamma}) \qquad \text{subject to } \boldsymbol{\gamma}=\mathbf{D}\mathbf{u},$$

with the Lagrangian function

$$\mathcal{L}(\mathbf{u},\boldsymbol{\gamma},\boldsymbol{\nu})=\frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2+\lambda \mathrm{P}_q(\boldsymbol{\gamma})+\boldsymbol{\nu}^T(\mathbf{D}\mathbf{u}-\boldsymbol{\gamma}), \tag{C.1}$$

where $\boldsymbol{\nu}\in\mathbb{R}^{\left[p\cdot\binom{n}{2}\right]}$ is the Lagrangian dual variable. In order to derive the dual problem, we need to minimize the Lagrangian function over the primal variables $\mathbf{u}$ and $\boldsymbol{\gamma}$. Recall from Chapter 4.1 that $\mathrm{P}_q^*(\cdot)$ is the dual norm of $\mathrm{P}_q(\cdot)$. It can be shown that

$$\inf_{\boldsymbol{\gamma}\in\mathbb{R}^{\left[p\cdot\binom{n}{2}\right]}} \mathcal{L}(\mathbf{u},\boldsymbol{\gamma},\boldsymbol{\nu})=\begin{cases}\frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2+\boldsymbol{\nu}^T\mathbf{D}\mathbf{u} & \text{if} \quad \mathrm{P}_q^*(\boldsymbol{\nu})\le\lambda,\\ -\infty & \text{otherwise,}\end{cases}$$

and

$$\inf_{\boldsymbol{\gamma}\in\mathbb{R}^{\left[p\cdot\binom{n}{2}\right]},\mathbf{u}\in\mathbb{R}^{np}} \mathcal{L}(\mathbf{u},\boldsymbol{\gamma},\boldsymbol{\nu})=\begin{cases}-\frac{1}{2}\|\mathbf{x}-\mathbf{D}^T\boldsymbol{\nu}\|_2^2 & \text{if} \quad \mathrm{P}_q^*(\boldsymbol{\nu})\le\lambda.\\ -\infty & \text{otherwise.}\end{cases}$$

Therefore, the dual problem for (4.2) is

$$\underset{\boldsymbol{\nu}\in\mathbb{R}^{\left[p\cdot\binom{n}{2}\right]}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{x}-\mathbf{D}^T\boldsymbol{\nu}\|_2^2 \qquad \text{subject to } \mathrm{P}_q^*(\boldsymbol{\nu})\le\lambda. \tag{C.2}$$

We now establish an explicit relationship between the solution to convex clustering and its dual problem. Differentiating the Lagrangian function (C.1) with respect to $\mathbf{u}$ and setting it equal to zero, we obtain

$$\hat{\mathbf{u}}=\mathbf{x}-\mathbf{D}^T\hat{\boldsymbol{\nu}},$$

where $\hat{\boldsymbol{\nu}}$ is the solution to the dual problem, which satisfies $\mathrm{P}_q^*(\hat{\boldsymbol{\nu}}) \leq \lambda$ by (C.2). Multiplying both sides by $\mathbf{D}$, we obtain the relationship (4.4). □

C.2 Proof of Lemma 4.3

Proof. We rewrite (4.5) as

$$\underset{\mathbf{u}' \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}, \boldsymbol{\eta} \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}}{\text{minimize}} \frac{1}{2}\|\mathbf{D}\mathbf{x} - \mathbf{u}'\|_2^2 + \lambda \mathrm{P}_q(\boldsymbol{\eta}) \qquad \text{subject to } \boldsymbol{\eta} = \mathbf{u}',$$

with the Lagrangian function

$$\mathcal{L}(\mathbf{u}', \boldsymbol{\eta}, \boldsymbol{\nu}') = \frac{1}{2}\|\mathbf{D}\mathbf{x} - \mathbf{u}'\|_2^2 + \lambda \mathrm{P}_q(\boldsymbol{\eta}) + (\boldsymbol{\nu}')^T(\mathbf{u}' - \boldsymbol{\eta}), \tag{C.3}$$

where $\boldsymbol{\nu}' \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}$ is the Lagrangian dual variable. In order to derive the dual problem, we minimize the Lagrangian function over the primal variables $\mathbf{u}'$ and $\boldsymbol{\eta}$. It can be shown that

$$\inf_{\boldsymbol{\eta} \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}} \mathcal{L}(\mathbf{u}', \boldsymbol{\eta}, \boldsymbol{\nu}') = \begin{cases} \frac{1}{2}\|\mathbf{D}\mathbf{x} - \mathbf{u}'\|_2^2 + (\boldsymbol{\nu}')^T\mathbf{u}' & \text{if} \quad \mathrm{P}_q^*(\boldsymbol{\nu}') \leq \lambda, \\ -\infty & \text{otherwise,} \end{cases}$$

and

$$\inf_{\boldsymbol{\eta} \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}, \mathbf{u}' \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}} \mathcal{L}(\mathbf{u}', \boldsymbol{\eta}, \boldsymbol{\nu}') = \begin{cases} -\frac{1}{2}\|\mathbf{D}\mathbf{x} - \boldsymbol{\nu}'\|_2^2 & \text{if} \quad \mathrm{P}_q^*(\boldsymbol{\nu}') \leq \lambda. \\ -\infty & \text{otherwise.} \end{cases}$$

Therefore, the dual problem for (4.5) is

$$\underset{\boldsymbol{\nu}' \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}}{\text{minimize}} \frac{1}{2}\|\mathbf{D}\mathbf{x} - \boldsymbol{\nu}'\|_2^2 \qquad \text{subject to } \mathrm{P}_q^*(\boldsymbol{\nu}') \leq \lambda. \tag{C.4}$$

We now establish an explicit relationship between the solution to (4.5) and its dual problem. Differentiating the Lagrangian function (C.3) with respect to $\mathbf{u}'$ and setting it equal to zero, we obtain

$$\hat{\mathbf{u}}' = \mathbf{D}\mathbf{x} - \hat{\boldsymbol{\nu}}',$$

where $\hat{\boldsymbol{\nu}}'$ is the solution to the dual problem, which we know from (C.4) satisfies $\mathrm{P}_q^*(\hat{\boldsymbol{\nu}}') \leq \lambda$. □

C.3 Proof of Theorem 4.1

In order to prove Theorem 4.1, we need two lemmas on the tail bounds for Chi-square and non-central Chi-square distributions.

Lemma C.1. *[53] For all $c > 0$,*

$$\Pr\left(\chi^2_\nu \geq \nu + 2\sqrt{\nu c} + 2c\right) \leq e^{-c}.$$

Lemma C.2. *[9] Let $\chi^2_\nu(\delta)$ denote a χ^2_ν random variable with non-centrality parameter δ. For all $c > 0$,*

$$\Pr\left(\chi^2_\nu(\delta) \leq \nu + \delta - 2\sqrt{(\nu + 2\delta)c}\right) \leq e^{-c}.$$

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1:

Proof. Recall from Chapter 4.2 that $\mathcal{S} = \{(i, i') : i, i' \in D_k, i < i'\}$. Also, recall from (4.8) that $\hat{\mathcal{S}}(\lambda) = \{(i, i') : \|\mathbf{X}_{i.} - \mathbf{X}_{i'.}\|_2 < \lambda, i < i'\}$. Note that the event

$$\begin{aligned}\{\hat{\mathcal{S}}(\delta/2) \neq \mathcal{S}\} &= \{\exists\ (i, i') \in \mathcal{S} \text{ such that } \|\mathbf{X}_{i.} - \mathbf{X}_{i'.}\|_2 \geq \delta/2\} \\ &\quad \bigcup \{\exists\ (i, i') \notin \mathcal{S} \text{ such that } \|\mathbf{X}_{i.} - \mathbf{X}_{i'.}\|_2 < \delta/2\} \\ &= \left\{ \bigcup_{(i,i') \in \mathcal{S}} (\|\mathbf{X}_{i.} - \mathbf{X}_{i'.}\|_2 \geq \delta/2) \right\} \bigcup \left\{ \bigcup_{(i,i') \notin \mathcal{S}} (\|\mathbf{X}_{i.} - \mathbf{X}_{i'.}\|_2 < \delta/2) \right\}\end{aligned}$$

By the union bound,

$$\Pr\left(\hat{\mathcal{S}}(\delta/2) \neq \mathcal{S}\right) \leq \Pr\left(\bigcup_{(i,i') \in \mathcal{S}} (\|\mathbf{X}_{i.} - \mathbf{X}_{i'.}\|_2 \geq \delta/2) \right) + \Pr\left(\bigcup_{(i,i') \notin \mathcal{S}} (\|\mathbf{X}_{i.} - \mathbf{X}_{i'.}\|_2 < \delta/2) \right).$$

We now provide an upper bound for each term separately.

Upper bound for $\Pr\left(\bigcup_{(i,i')\in\mathcal{S}}(\|\mathbf{X}_{i.}-\mathbf{X}_{i'.}\|_2\geq\delta/2)\right)$: For $(i,i')\in\mathcal{S}$, the ith and i'th observations have the same mean. Therefore, $\|\mathbf{X}_{i.}-\mathbf{X}_{i'.}\|_2^2\sim 2\sigma^2\chi_p^2$. By the union bound,

$$\begin{aligned}\Pr\left(\bigcup_{(i,i')\in\mathcal{S}}(\|\mathbf{X}_{i.}-\mathbf{X}_{i'.}\|_2\geq\delta/2)\right)&\leq\sum_{(i,i')\in\mathcal{S}}\Pr\left(\|\mathbf{X}_{i.}-\mathbf{X}_{i'.}\|_2\geq\delta/2\right)\\&\leq n^2\Pr\left(\chi_p\geq\frac{\delta}{2\sqrt{2}\sigma}\right),\end{aligned}$$

where the last inequality follows from the fact that $|\mathcal{S}|<n^2$. By Assumption 4.1 and taking $c=3\log n$ in Lemma C.1, we obtain

$$n^2\Pr\left(\chi_p\geq\frac{\delta}{2\sqrt{2}\sigma}\right)\leq n^2\Pr\left(\chi_p\geq\sqrt{p}\sqrt{1+\sqrt{\frac{12\log n}{p}}+6\frac{\log n}{p}}\right)\leq\frac{1}{n}.$$

Therefore, $\Pr\left(\bigcup_{(i,i')\in\mathcal{S}}(\|\mathbf{X}_{i.}-\mathbf{X}_{i'.}\|_2\geq\delta/2)\right)\leq\frac{1}{n}$.

Upper bound for $\Pr\left(\bigcup_{(i,i')\notin\mathcal{S}}(\|\mathbf{X}_{i.}-\mathbf{X}_{i'.}\|_2<\delta/2)\right)$: We first recall that a random variable A is said to be stochastically larger than the random variable B if $\Pr(A<t)\leq\Pr(B<t)$ for all t. For non-centrality parameters $\delta_1>\delta_2\geq 0$, $\chi_p^2(\delta_1)$ is stochastically larger than $\chi_p^2(\delta_2)$ [101]. For $(i,i')\notin\mathcal{S}$, $\|\mathbf{X}_{i.}-\mathbf{X}_{i'.}\|_2^2\sim 2\sigma^2\chi_p^2\left(\frac{\|\boldsymbol{\mu}_{k(i)}-\boldsymbol{\mu}_{k(i')}\|_2^2}{2\sigma^2}\right)$, where $\boldsymbol{\mu}_{k(i)}$ is the mean vector corresponding to the ith observation. By the union bound,

$$\begin{aligned}\Pr\left(\bigcup_{(i,i')\notin\mathcal{S}}(\|\mathbf{X}_{i.}-\mathbf{X}_{i'.}\|_2<\delta/2)\right)&\leq\sum_{(i,i')\notin\mathcal{S}}\Pr\left(\|\mathbf{X}_{i.}-\mathbf{X}_{i'.}\|_2<\delta/2\right)\\&=\sum_{(i,i')\notin\mathcal{S}}\Pr\left(\sqrt{2}\sigma\chi_p\left(\frac{\|\boldsymbol{\mu}_{k(i)}-\boldsymbol{\mu}_{k(i')}\|_2^2}{2\sigma^2}\right)<\delta/2\right)\\&=\sum_{(i,i')\notin\mathcal{S}}\Pr\left(\chi_p\left(\frac{\|\boldsymbol{\mu}_{k(i)}-\boldsymbol{\mu}_{k(i')}\|_2^2}{2\sigma^2}\right)<\frac{\delta}{2\sqrt{2}\sigma}\right).\end{aligned}$$

By the definition of δ in Assumption 4.1, $\chi_p\left(\frac{\|\boldsymbol{\mu}_{k(i)}-\boldsymbol{\mu}_{k(i')}\|_2^2}{2\sigma^2}\right)$ is stochastically larger than $\chi_p\left(\frac{\delta^2}{2\sigma^2}\right)$. Using the above fact and the fact that there are at most n^2 possible pairs of (i,i')

not in $\mathcal{S}$, we obtain

$$\sum_{(i,i')\notin\mathcal{S}} \Pr\left(\chi_p\left(\frac{\|\boldsymbol{\mu}_{k(i)}-\boldsymbol{\mu}_{k(i')}\|_2^2}{2\sigma^2}\right)<\frac{\delta}{2\sqrt{2}\sigma}\right)\leq n^2\Pr\left(\chi_p\left(\frac{\delta^2}{2\sigma^2}\right)<\frac{\delta}{2\sqrt{2}\sigma}\right).$$

We now present two facts needed to obtain an upper bound for $n^2\Pr\left(\chi_p\left(\frac{\delta^2}{2\sigma^2}\right)<\frac{\delta}{2\sqrt{2}\sigma}\right)$. By Lemma C.2 and taking $c=3\log n$, we have

$$n^2\Pr\left(\chi_p^2\left(\frac{\delta^2}{2\sigma^2}\right)<p+\frac{\delta^2}{2\sigma^2}-2\sqrt{3(p+\delta^2/\sigma^2)\log n}\right)\leq\frac{1}{n}.$$

Now, note that the inequality $p+\frac{\delta^2}{2\sigma^2}-2\sqrt{3(p+\delta^2/\sigma^2)\log n}>\frac{\delta^2}{8\sigma^2}$ holds by choosing n,p such that $\frac{\log n}{p}\leq\frac{1}{16}$. Therefore,

$$\begin{aligned}
n^2\Pr\left(\chi_p\left(\frac{\delta^2}{2\sigma^2}\right)<\frac{\delta}{2\sqrt{2}\sigma}\right) &= n^2\Pr\left(\chi_p^2\left(\frac{\delta^2}{2\sigma^2}\right)<\frac{\delta^2}{8\sigma^2}\right)\\
&\leq n^2\Pr\left(\chi_p^2\left(\frac{\delta^2}{2\sigma^2}\right)<p+\frac{\delta^2}{2\sigma^2}-2\sqrt{3(p+\delta^2/\sigma^2)\log n}\right)\\
&\leq\frac{1}{n}.
\end{aligned}$$

We have shown that $\Pr\left(\bigcup_{(i,i')\notin\mathcal{S}}(\|\mathbf{X}_{i\cdot}-\mathbf{X}_{i'\cdot}\|_2<\delta/2)\right)\leq\frac{1}{n}$.

Combining the results, we obtain

$$\Pr\left(\hat{\mathcal{S}}(\delta/2)=\mathcal{S}\right)=1-\Pr\left(\hat{\mathcal{S}}(\delta/2)\neq\mathcal{S}\right)\geq 1-\frac{2}{n}.$$

□

C.4 Proof of Lemma 4.5

Proof. Since $\mathbf{D}$ is not of full rank by Lemma 4.1.1, the solution to (4.3) in the absence of constraint is not unique, and takes the form

$$\begin{aligned}
\hat{\boldsymbol{\nu}} &= (\mathbf{D}\mathbf{D}^T)^\dagger\mathbf{D}\mathbf{x}+(\mathbf{I}-\mathbf{D}(\mathbf{D}^T\mathbf{D})^\dagger\mathbf{D}^T)\boldsymbol{\omega}\\
&= (\mathbf{D}^T)^\dagger\mathbf{x}+(\mathbf{I}-\mathbf{D}\mathbf{D}^\dagger)\boldsymbol{\omega} \qquad \text{(C.5)}\\
&= \frac{1}{n}\mathbf{D}\mathbf{x}+(\mathbf{I}-\frac{1}{n}\mathbf{D}\mathbf{D}^T)\boldsymbol{\omega},
\end{aligned}$$

for $\boldsymbol{\omega} \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}$. The second equality follows from Lemma 4.1.3 and the last equality follows from Lemma 4.1.2.

Let $\hat{\mathbf{u}}$ be the solution to (4.2). Substituting $\hat{\boldsymbol{\nu}}$ given in (C.5) into (4.4), we obtain

$$\begin{aligned}
\mathbf{D}\hat{\mathbf{u}} &= \mathbf{D}\mathbf{x} - \mathbf{D}\mathbf{D}^T\hat{\boldsymbol{\nu}} \\
&= \mathbf{D}\mathbf{x} - \frac{1}{n}\mathbf{D}\mathbf{D}^T\mathbf{D}\mathbf{x} - \mathbf{D}\mathbf{D}^T\boldsymbol{\omega} + \frac{1}{n}\mathbf{D}\mathbf{D}^T\mathbf{D}\mathbf{D}^T\boldsymbol{\omega} \\
&= \mathbf{D}\mathbf{x} - \mathbf{D}\mathbf{x} - \mathbf{D}\mathbf{D}^T\boldsymbol{\omega} + \mathbf{D}\mathbf{D}^T\boldsymbol{\omega} \\
&= \mathbf{0}.
\end{aligned}$$

Recall from Definition 4.1 that all observations are estimated to belong to the same cluster if $\mathbf{D}\hat{\mathbf{u}} = \mathbf{0}$. For any $\hat{\boldsymbol{\nu}}$ in (C.5), picking $\lambda = \mathrm{P}_q^*(\hat{\boldsymbol{\nu}})$ guarantees that the constraint on the dual problem (4.3) is inactive, and therefore that convex clustering has a trivial solution of $\mathbf{D}\hat{\mathbf{u}} = \mathbf{0}$.

Since $\hat{\boldsymbol{\nu}}$ is not unique, $\mathrm{P}_q^*(\hat{\boldsymbol{\nu}})$ is not unique. In order to obtain the smallest tuning parameter λ such that $\mathbf{D}\hat{\mathbf{u}} = \mathbf{0}$, we take

$$\lambda_{\text{upper}} := \min_{\boldsymbol{\omega} \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right]}} \mathrm{P}_q^* \left(\frac{1}{n}\mathbf{D}\mathbf{x} + \left(\mathbf{I} - \frac{1}{n}\mathbf{D}\mathbf{D}^T \right) \boldsymbol{\omega} \right).$$

Any tuning parameter $\lambda \geq \lambda_{\text{upper}}$ results in an estimate for which all observations belong to a single cluster. □

C.5 Proof of Lemma 4.6

In order to simplify our analysis, we start by reformulating (4.2) as in [61]. Let $\mathbf{D} = \mathbf{A}\Lambda\mathbf{V}_\beta^T$ be the *singular value decomposition* of $\mathbf{D}$, where $\mathbf{A} \in \mathbb{R}^{\left[p \cdot \binom{n}{2}\right] \times p(n-1)}$, $\Lambda \in \mathbb{R}^{p(n-1) \times p(n-1)}$, and $\mathbf{V}_\beta \in \mathbb{R}^{np \times p(n-1)}$. Construct $\mathbf{V}_\alpha \in \mathbb{R}^{np \times p}$ such that $\mathbf{V} = [\mathbf{V}_\alpha, \mathbf{V}_\beta] \in \mathbb{R}^{np \times np}$ is an orthogonal matrix, that is, $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$. Note that $\mathbf{V}_\alpha^T\mathbf{V}_\beta = 0$.

Let $\boldsymbol{\beta} = \mathbf{V}_\beta^T\mathbf{u} \in \mathbb{R}^{p(n-1)}$ and $\boldsymbol{\alpha} = \mathbf{V}_\alpha^T\mathbf{u} \in \mathbb{R}^p$. Also, let $\lambda' = \frac{\lambda}{np}$. Optimization problem (4.2) then becomes

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^p, \boldsymbol{\beta} \in \mathbb{R}^{p(n-1)}}{\text{minimize}} \ \frac{1}{2np}\|\mathbf{x} - \mathbf{V}_\alpha\boldsymbol{\alpha} - \mathbf{V}_\beta\boldsymbol{\beta}\|^2 + \lambda'\mathrm{P}_q(\mathbf{Z}\boldsymbol{\beta}), \tag{C.6}$$

where $\mathbf{Z} = \mathbf{A}\Lambda \in \mathbb{R}^{\left[p\cdot\binom{n}{2}\right]\times p(n-1)}$. Note that $\text{rank}(\mathbf{Z}) = p(n-1)$ and therefore, there exists a pseudo-inverse $\mathbf{Z}^\dagger \in \mathbb{R}^{p(n-1)\times\left[p\cdot\binom{n}{2}\right]}$ such that $\mathbf{Z}^\dagger\mathbf{Z} = \mathbf{I}$. Recall that the set $\mathcal{C}(i,i')$ contains the row indices of $\mathbf{D}$ such that $\mathbf{D}_{\mathcal{C}(i,i')}\mathbf{u} = \mathbf{U}_{i\cdot} - \mathbf{U}_{i'\cdot}$. Let the submatrices $\mathbf{Z}_{\mathcal{C}(i,i')}$ and $\mathbf{Z}^\dagger_{\mathcal{C}(i,i')}$ denote the rows of $\mathbf{Z}$ and the columns of $\mathbf{Z}^\dagger$, respectively, corresponding to the indices in the set $\mathcal{C}(i,i')$. By Lemma 4.1.5,

$$\Lambda_{\min}(\mathbf{Z}) = \Lambda_{\min}(\mathbf{D}) = \frac{1}{\Lambda_{\max}(\mathbf{Z}^\dagger)} = \sqrt{n} \qquad \text{and} \qquad \Lambda_{\max}(\mathbf{Z}) = \Lambda_{\max}(\mathbf{D}) = \frac{1}{\Lambda_{\min}(\mathbf{Z}^\dagger)} = \sqrt{n}. \tag{C.7}$$

Let $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ denote the solution to (C.6).

Proof of Lemma 4.6:

Proof. We establish a finite sample bound for the prediction error of convex clustering with $q = 1$ by analyzing (C.6). First, note that $\hat{\mathbf{u}} = \mathbf{V}_\alpha\hat{\boldsymbol{\alpha}} + \mathbf{V}_\beta\hat{\boldsymbol{\beta}}$ and $\mathbf{u} = \mathbf{V}_\alpha\boldsymbol{\alpha} + \mathbf{V}_\beta\boldsymbol{\beta}$. Thus, $\frac{1}{2np}\|\hat{\mathbf{u}} - \mathbf{u}\|^2 = \frac{1}{2np}\|\mathbf{V}_\alpha(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}) + \mathbf{V}_\beta(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2$. Recall that $\mathrm{P}_1(\mathbf{Z}\boldsymbol{\beta}) = \|\mathbf{Z}\boldsymbol{\beta}\|_1$. By the definition of $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$, we have

$$\frac{1}{2np}\|\mathbf{x} - (\mathbf{V}_\alpha\hat{\boldsymbol{\alpha}} + \mathbf{V}_\beta\hat{\boldsymbol{\beta}})\|^2 + \lambda'\|\mathbf{Z}\hat{\boldsymbol{\beta}}\|_1 \leq \frac{1}{2np}\|\mathbf{x} - (\mathbf{V}_\alpha\boldsymbol{\alpha} + \mathbf{V}_\beta\boldsymbol{\beta})\|^2 + \lambda'\|\mathbf{Z}\boldsymbol{\beta}\|_1,$$

implying

$$\frac{1}{2np}\|\mathbf{V}_\alpha(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}) + \mathbf{V}_\beta(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2 + \lambda'\|\mathbf{Z}\hat{\boldsymbol{\beta}}\|_1 \leq \frac{1}{np}\mathrm{G}(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}) + \lambda'\|\mathbf{Z}\boldsymbol{\beta}\|_1, \tag{C.8}$$

where $\mathrm{G}(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}) = \boldsymbol{\epsilon}^T\left[\mathbf{V}_\alpha(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}) + \mathbf{V}_\beta(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right]$. Recall that $\mathbf{V}_\alpha^T\mathbf{V}_\alpha = \mathbf{I}$ and $\mathbf{V}_\alpha^T\mathbf{V}_\beta = \mathbf{0}$. By the optimality condition of (C.6),

$$\begin{aligned}
\hat{\boldsymbol{\alpha}} &= \mathbf{V}_\alpha^T(\mathbf{x} - \mathbf{V}_\beta\hat{\boldsymbol{\beta}}) \\
&= \mathbf{V}_\alpha^T\left(\mathbf{V}_\alpha\boldsymbol{\alpha} + \mathbf{V}_\beta\boldsymbol{\beta} + \boldsymbol{\epsilon} - \mathbf{V}_\beta\hat{\boldsymbol{\beta}}\right) \\
&= \boldsymbol{\alpha} + \mathbf{V}_\alpha^T\boldsymbol{\epsilon}.
\end{aligned}$$

Therefore, substituting $\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha} = \mathbf{V}_\alpha^T \boldsymbol{\epsilon}$ into $\mathrm{G}(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})$, we obtain

$$\begin{aligned}
\frac{1}{np}\left|\mathrm{G}(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})\right| &= \frac{1}{np}\left|\boldsymbol{\epsilon}^T\left[\mathbf{V}_\alpha(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}) + \mathbf{V}_\beta(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right]\right| \\
&= \frac{1}{np}\left|\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^T\mathbf{V}_\beta(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right| \\
&\leq \frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon} + \frac{1}{np}\left|\boldsymbol{\epsilon}^T\mathbf{V}_\beta(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right| \\
&= \frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon} + \frac{1}{np}\left|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger\mathbf{Z}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right| \\
&\leq \frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon} + \frac{1}{np}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger\|_\infty\|\mathbf{Z}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|_1.
\end{aligned}$$

We now establish bounds for $\frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}$ and $\frac{1}{np}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger\|_\infty$ that hold with high probability.

Bound for $\frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}$:

From Lemma 9.3 of [31], $\frac{1}{\sigma^2}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon} \sim \chi_p^2$ since $\mathbf{V}_\alpha\mathbf{V}_\alpha^T$ is a projection matrix of rank p. By Lemma C.1 and taking $\nu = p$ and $c = \log(np)$, we have that

$$\Pr\left(\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon} \geq \sigma^2\left[p + 2\sqrt{p\log(np)} + 2\log(np)\right]\right) \leq \frac{1}{np}. \tag{C.9}$$

Bound for $\frac{1}{np}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger\|_\infty$:

Let e_j be an indicator vector of length $p \cdot \binom{n}{2}$ with a one in the jth entry and zeroes in the remaining entries. Let $v_j = e_j^T(\mathbf{Z}^\dagger)^T\mathbf{V}_\beta^T\boldsymbol{\epsilon}$. Using the fact that $\Lambda_{\max}(\mathbf{V}_\beta) = 1$ and $\Lambda_{\max}(\mathbf{Z}^\dagger) = \frac{1}{\sqrt{n}}$ (C.7), we know that each v_j is a Gaussian random variable with mean zero and variance at most $\frac{\sigma^2}{n}$. Therefore,

$$\Pr\left(\sqrt{n}\cdot v_j \geq 2\sigma\sqrt{\log\left(p\cdot\binom{n}{2}\right)}\right) \leq \Pr\left(N(0,\sigma^2) \geq 2\sigma\sqrt{\log\left(p\cdot\binom{n}{2}\right)}\right).$$

Thus,

$$\begin{aligned}
\Pr\left(\sqrt{n}\cdot\max_j|v_j| \geq 2\sigma\sqrt{\log\left(p\cdot\binom{n}{2}\right)}\right) &\leq 2p\binom{n}{2}\Pr\left(N(0,\sigma^2) \geq 2\sigma\sqrt{\log\left(p\cdot\binom{n}{2}\right)}\right) \\
&\leq \frac{2}{p\cdot\binom{n}{2}},
\end{aligned}$$

which follows from an application of the union bound and the fact that

$$\begin{aligned}\Pr\left(N(0,1) \geq 2\sqrt{\log\left(p\cdot\binom{n}{2}\right)}\right) &= \int_{2\sqrt{\log\left(p\cdot\binom{n}{2}\right)}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt \\ &\leq \int_{2\sqrt{\log\left(p\cdot\binom{n}{2}\right)}}^{\infty} t \exp\left(-\frac{t^2}{2}\right) dt \\ &= \frac{1}{\left(p\cdot\binom{n}{2}\right)^2}.\end{aligned}$$

Using the above facts, we obtain

$$\Pr\left(\|\boldsymbol{\epsilon}^T \mathbf{V}_\beta \mathbf{Z}^\dagger\|_\infty \geq 2\sigma\sqrt{\frac{\log(p\cdot\binom{n}{2})}{n}}\right) \leq \frac{2}{p\cdot\binom{n}{2}}. \tag{C.10}$$

Combining the two upper bounds: Setting $\lambda' > 4\sigma\sqrt{\frac{\log\left(p\cdot\binom{n}{2}\right)}{n^3p^2}}$ and combining the results from (C.9) and (C.10), we obtain

$$\frac{1}{np}\mathrm{G}(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}) \leq \sigma^2\left[\frac{1}{n} + 2\sqrt{\frac{\log(np)}{n^2p}} + 2\frac{\log(np)}{np}\right] + \frac{\lambda'}{2}\|\mathbf{Z}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|_1 \tag{C.11}$$

with probability at least $1 - \frac{2}{p\cdot\binom{n}{2}} - \frac{1}{np}$. Substituting (C.11) into (C.8), we obtain

$$\begin{aligned}&\frac{1}{2np}\|\mathbf{V}_\alpha(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}) + \mathbf{V}_\beta(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2 + \lambda'\|\mathbf{Z}\hat{\boldsymbol{\beta}}\|_1 \\ &\leq \sigma^2\left[\frac{1}{n} + 2\sqrt{\frac{\log(np)}{n^2p}} + 2\frac{\log(np)}{np}\right] + \frac{\lambda'}{2}\|\mathbf{Z}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|_1 + \lambda'\|\mathbf{Z}\boldsymbol{\beta}\|_1.\end{aligned}$$

We get Lemma 4.6 by an application of the triangle inequality and by rearranging the terms. □

C.6 Proof of Lemma 4.7

Proof. We establish a finite sample bound for the prediction error of convex clustering with $q = 2$ by analyzing (C.6). Recall that $\mathrm{P}_2(\mathbf{Z}\boldsymbol{\beta}) = \sum_{i<i'} \|\mathbf{Z}_{\mathcal{C}(i,i')}\boldsymbol{\beta}\|_2$. By the definition of $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$, we have

$$\frac{1}{2np}\|\mathbf{x} - (\mathbf{V}_\alpha\hat{\boldsymbol{\alpha}} + \mathbf{V}_\beta\hat{\boldsymbol{\beta}})\|^2 + \lambda'\sum_{i<i'}\|\mathbf{Z}_{\mathcal{C}(i,i')}\hat{\boldsymbol{\beta}}\|_2 \leq \frac{1}{2np}\|\mathbf{x} - (\mathbf{V}_\alpha\boldsymbol{\alpha} + \mathbf{V}_\beta\boldsymbol{\beta})\|^2 + \lambda'\sum_{i<i'}\|\mathbf{Z}_{\mathcal{C}(i,i')}\boldsymbol{\beta}\|_2,$$

implying

$$\frac{1}{2np}\|\mathbf{V}_\alpha(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha})+\mathbf{V}_\beta(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|^2+\lambda'\sum_{i<i'}\|\mathbf{Z}_{\mathcal{C}(i,i')}\hat{\boldsymbol{\beta}}\|_2 \leq \frac{1}{np}\mathrm{G}(\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{\beta}})+\lambda'\sum_{i<i'}\|\mathbf{Z}_{\mathcal{C}(i,i')}\boldsymbol{\beta}\|_2, \quad \text{(C.12)}$$

where $\mathrm{G}(\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{\beta}}) = \boldsymbol{\epsilon}^T\left[\mathbf{V}_\alpha(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha})+\mathbf{V}_\beta(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\right]$. Again, by the optimality condition of (C.6), we have that $\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha} = \mathbf{V}_\alpha^T\boldsymbol{\epsilon}$. Substituting this into $\frac{1}{np}\mathrm{G}(\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{\beta}})$, we obtain

$$\begin{aligned}
\frac{1}{np}\left|\mathrm{G}(\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{\beta}})\right| &= \frac{1}{np}\left|\boldsymbol{\epsilon}^T\left[\mathbf{V}_\alpha(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha})+\mathbf{V}_\beta(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\right]\right| \\
&= \frac{1}{np}\left|\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}+\boldsymbol{\epsilon}^T\mathbf{V}_\beta(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\right| \\
&\leq \frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}+\frac{1}{np}\left|\boldsymbol{\epsilon}^T\mathbf{V}_\beta(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\right| \\
&= \frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}+\frac{1}{np}\left|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger\mathbf{Z}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\right| \\
&= \frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}+\frac{1}{np}\left|\sum_{i<i'}(\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}_{\mathcal{C}(i,i')}^\dagger)(\mathbf{Z}_{\mathcal{C}(i,i')}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}))\right| \\
&\leq \frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}+\frac{1}{np}\sum_{i<i'}\left|(\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}_{\mathcal{C}(i,i')}^\dagger)(\mathbf{Z}_{\mathcal{C}(i,i')}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}))\right| \\
&\leq \frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}+\frac{1}{np}\sum_{i<i'}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}_{\mathcal{C}(i,i')}^\dagger\|_2\|\mathbf{Z}_{\mathcal{C}(i,i')}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|_2 \\
&\leq \frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}+\frac{1}{np}\cdot\max_{i<i'}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}_{\mathcal{C}(i,i')}^\dagger\|_2\sum_{i<i'}\|\mathbf{Z}_{\mathcal{C}(i,i')}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|_2,
\end{aligned}$$

where the second inequality follows from an application of the triangle inequality and the third inequality from an application of the Cauchy-Schwarz inequality. We now establish bounds for $\frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}$ and $\frac{1}{np}\cdot\max_{i<i'}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}_{\mathcal{C}(i,i')}^\dagger\|_2$ that hold with large probability.

Bound for $\frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon}$:

This is established in the proof of Lemma 4.6 in (C.9), i.e.,

$$\Pr\left(\frac{1}{np}\boldsymbol{\epsilon}^T\mathbf{V}_\alpha\mathbf{V}_\alpha^T\boldsymbol{\epsilon} \geq \sigma^2\left[\frac{1}{n}+2\sqrt{\frac{\log(np)}{n^2p}}+2\frac{\log(np)}{np}\right]\right) \leq \frac{1}{np}.$$

Bound for $\frac{1}{np}\cdot\max_{i<i'}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}_{\mathcal{C}(i,i')}^\dagger\|_2$:

First, note that there are p indices in each set $\mathcal{C}(i,i')$. Therefore, for each set $\mathcal{C}(i,i')$, we obtain

$$\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger_{\mathcal{C}(i,i')}\|_2 \leq \sqrt{p}\cdot\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger_{\mathcal{C}(i,i')}\|_\infty.$$

Note that

$$\frac{1}{np}\cdot\max_{i<i'}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger_{\mathcal{C}(i,i')}\|_2 \leq \sqrt{\frac{1}{n^2p}}\cdot\max_{i<i'}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger_{\mathcal{C}(i,i')}\|_\infty = \sqrt{\frac{1}{n^2p}}\cdot\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger\|_\infty. \quad \text{(C.13)}$$

Therefore, using (C.13),

$$\begin{aligned}
&\Pr\left(\frac{1}{np}\cdot\max_{i<i'}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger_{\mathcal{C}(i,i')}\|_2 \geq 2\sigma\sqrt{\frac{\log\left(p\cdot\binom{n}{2}\right)}{n^3p}}\right)\\
&\leq \Pr\left(\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger\|_\infty \geq 2\sigma\sqrt{\frac{\log\left(p\cdot\binom{n}{2}\right)}{n}}\right) \qquad \text{(C.14)}\\
&\leq \frac{2}{p\cdot\binom{n}{2}},
\end{aligned}$$

where the last inequality follows from (C.10) in the proof of Lemma 4.6.

Therefore, for $\lambda' > 4\sigma\sqrt{\frac{\log\left(p\cdot\binom{n}{2}\right)}{n^3p}}$, we have $\frac{\lambda'}{2} < \frac{1}{np}\cdot\max_{i<i'}\|\boldsymbol{\epsilon}^T\mathbf{V}_\beta\mathbf{Z}^\dagger_{\mathcal{C}(i,i')}\|_2$ with probability at most $\frac{2}{p\cdot\binom{n}{2}}$. Combining the results from (C.9) and (C.14), we have that

$$\frac{1}{np}\mathrm{G}(\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{\beta}}) \leq \sigma^2\left[\frac{1}{n}+2\sqrt{\frac{\log(np)}{n^2p}}+2\frac{\log(np)}{np}\right]+\frac{\lambda'}{2}\sum_{i<i'}\|\mathbf{Z}_{\mathcal{C}(i,i')}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|_2 \qquad \text{(C.15)}$$

with probability at least $1-\frac{2}{p\cdot\binom{n}{2}}-\frac{1}{np}$. Substituting (C.15) into (C.12) , we obtain

$$\begin{aligned}
&\frac{1}{2np}\|\mathbf{V}_\alpha(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha})+\mathbf{V}_\beta(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|^2+\lambda'\sum_{i<i'}\|\mathbf{Z}_{\mathcal{C}(i,i')}\hat{\boldsymbol{\beta}}\|_2\\
&\leq \sigma^2\left[\frac{1}{n}+2\sqrt{\frac{\log(np)}{n^2p}}+2\frac{\log(np)}{np}\right]+\frac{\lambda'}{2}\sum_{i<i'}\|\mathbf{Z}_{\mathcal{C}(i,i')}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|_2+\lambda'\sum_{i<i'}\|\mathbf{Z}_{\mathcal{C}(i,i')}\boldsymbol{\beta}\|_2.
\end{aligned}$$

We get Lemma 4.7 by an application of the triangle inequality and by rearranging the terms. □

C.7 Proof of Lemma 4.9

Proof. We derive the degrees of freedom for $\hat{\mathbf{u}}$ for (4.2) when $\mathbf{x} \sim \text{MVN}(\mathbf{u}, \sigma^2\mathbf{I})$. Directly from the dual problem (4.3), we can see that the solution $\mathbf{D}^T\hat{\boldsymbol{\nu}}$ is the projection of $\mathbf{x}$ onto the convex set $K = \left\{\mathbf{D}^T\boldsymbol{\nu} : \text{P}_2^*(\boldsymbol{\nu}) \leq \lambda\right\}$. Using the primal-dual relationship $\hat{\mathbf{u}} = \mathbf{x} - \mathbf{D}^T\hat{\boldsymbol{\nu}}$, we see that $\hat{\mathbf{u}}$ is the residual from projecting $\mathbf{x}$ onto the convex set K. By Lemma 1 of [98], $\hat{\mathbf{u}}$ is continuous and almost differentiable with respect to $\mathbf{x}$. Therefore, by Stein's formula, the degrees of freedom can be characterized as $\text{E}\left[\text{tr}\left(\frac{\partial\hat{\mathbf{u}}}{\partial\mathbf{x}}\right)\right]$.

Recall that $\mathbf{D}_{\mathcal{C}(i,i')}$ denotes the rows of $\mathbf{D}$ corresponding to the indices in the set $\mathcal{C}(i,i')$. Let $\hat{\mathcal{B}}_2 = \{(i,i') : \|\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}}\|_2 \neq 0\}$. By the optimality condition of (4.2) with $q = 2$, we obtain

$$(\mathbf{x} - \hat{\mathbf{u}}) = \lambda \sum_{i<i'} \mathbf{D}_{\mathcal{C}(i,i')}^T g_{\mathcal{C}(i,i')}, \tag{C.16}$$

where

$$g_{\mathcal{C}(i,i')} = \begin{cases} \frac{\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}}}{\|\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}}\|_2} & \text{if } (i,i') \in \hat{\mathcal{B}}_2. \\ \in \{\Gamma : \|\Gamma\|_2 \leq 1\} & \text{if } (i,i') \notin \hat{\mathcal{B}}_2. \end{cases}$$

We define the matrix $\mathbf{D}_{-\hat{\mathcal{B}}_2}$ by removing the rows of $\mathbf{D}$ that correspond to elements in $\hat{\mathcal{B}}_2$. Let $\mathbf{P} = \left(\mathbf{I} - \mathbf{D}_{-\hat{\mathcal{B}}_2}^T(\mathbf{D}_{-\hat{\mathcal{B}}_2}\mathbf{D}_{-\hat{\mathcal{B}}_2}^T)^\dagger\mathbf{D}_{-\hat{\mathcal{B}}_2}\right)$ be the projection matrix onto the complement of the space spanned by the rows of $\mathbf{D}_{-\hat{\mathcal{B}}_2}$.

By the definition of $\mathbf{D}_{-\hat{\mathcal{B}}_2}$, we obtain $\mathbf{D}_{-\hat{\mathcal{B}}_2}\hat{\mathbf{u}} = \mathbf{0}$. Therefore, $\mathbf{P}\hat{\mathbf{u}} = \hat{\mathbf{u}}$. Multiplying $\mathbf{P}$ onto both sides of (C.16), we obtain

$$\begin{aligned} \mathbf{P}\mathbf{x} - \hat{\mathbf{u}} &= \lambda\mathbf{P}\sum_{i<i'} \mathbf{D}_{\mathcal{C}(i,i')}^T g_{\mathcal{C}(i,i')} \\ &= \lambda\mathbf{P}\sum_{(i,i')\in\hat{\mathcal{B}}_2} \frac{\mathbf{D}_{\mathcal{C}(i,i')}^T\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}}}{\|\mathbf{D}_{\mathcal{C}(i,i')}\hat{\mathbf{u}}\|_2}, \end{aligned} \tag{C.17}$$

where the second equality follows from the fact that $\mathbf{P}\mathbf{D}_{\mathcal{C}(i,i')}^T = \mathbf{0}$ for any $(i,i') \notin \hat{\mathcal{B}}_2$.

[100] showed that there exists a neighborhood around almost every $\mathbf{x}$ such that the solution $\hat{\mathcal{B}}_2$ is locally constant with respect to $\mathbf{x}$. Therefore, the derivative of (C.17) with

respect to $\mathbf{x}$ is

$$\mathbf{P} - \frac{\partial \hat{\mathbf{u}}}{\partial \mathbf{x}} = \lambda \mathbf{P} \sum_{(i,i') \in \hat{\mathcal{B}}_2} \left(\frac{\mathbf{D}_{\mathcal{C}(i,i')}^T \mathbf{D}_{\mathcal{C}(i,i')}}{\|\mathbf{D}_{\mathcal{C}(i,i')} \hat{\mathbf{u}}\|_2} - \frac{\mathbf{D}_{\mathcal{C}(i,i')}^T \mathbf{D}_{\mathcal{C}(i,i')} \hat{\mathbf{u}} \hat{\mathbf{u}}^T \mathbf{D}_{\mathcal{C}(i,i')}^T \mathbf{D}_{\mathcal{C}(i,i')}}{\|\mathbf{D}_{\mathcal{C}(i,i')} \hat{\mathbf{u}}\|_2^3} \right) \frac{\partial \hat{\mathbf{u}}}{\partial \mathbf{x}}, \tag{C.18}$$

using the fact that for any matrix $\mathbf{A}$ with $\|\mathbf{A}\mathbf{v}\|_2 \neq 0$, $\frac{\partial}{\partial \mathbf{v}} \frac{\mathbf{A}^T \mathbf{A} \mathbf{v}}{\|\mathbf{A}\mathbf{v}\|_2} = \frac{\mathbf{A}^T \mathbf{A}}{\|\mathbf{A}\mathbf{v}\|_2} - \frac{\mathbf{A}^T \mathbf{A} \mathbf{v} \mathbf{v}^T \mathbf{A}^T \mathbf{A}}{\|\mathbf{A}\mathbf{v}\|_2^3}$.

Solving (C.18) for $\frac{\partial \hat{\mathbf{u}}}{\partial \mathbf{x}}$, we have

$$\frac{\partial \hat{\mathbf{u}}}{\partial \mathbf{x}} = \left[\mathbf{I} + \lambda \mathbf{P} \sum_{(i,i') \in \hat{\mathcal{B}}_2} \left(\frac{\mathbf{D}_{\mathcal{C}(i,i')}^T \mathbf{D}_{\mathcal{C}(i,i')}}{\|\mathbf{D}_{\mathcal{C}(i,i')} \hat{\mathbf{u}}\|_2} - \frac{\mathbf{D}_{\mathcal{C}(i,i')}^T \mathbf{D}_{\mathcal{C}(i,i')} \hat{\mathbf{u}} \hat{\mathbf{u}}^T \mathbf{D}_{\mathcal{C}(i,i')}^T \mathbf{D}_{\mathcal{C}(i,i')}}{\|\mathbf{D}_{\mathcal{C}(i,i')} \hat{\mathbf{u}}\|_2^3} \right) \right]^{-1} \mathbf{P}. \tag{C.19}$$

Therefore, an unbiased estimator of the degrees of freedom is of the form

$$\operatorname{tr}\left(\frac{\partial \hat{\mathbf{u}}}{\partial \mathbf{x}} \right) = \operatorname{tr}\left(\left[\mathbf{I} + \lambda \mathbf{P} \sum_{(i,i') \in \hat{\mathcal{B}}_2} \left(\frac{\mathbf{D}_{\mathcal{C}(i,i')}^T \mathbf{D}_{\mathcal{C}(i,i')}}{\|\mathbf{D}_{\mathcal{C}(i,i')} \hat{\mathbf{u}}\|_2} - \frac{\mathbf{D}_{\mathcal{C}(i,i')}^T \mathbf{D}_{\mathcal{C}(i,i')} \hat{\mathbf{u}} \hat{\mathbf{u}}^T \mathbf{D}_{\mathcal{C}(i,i')}^T \mathbf{D}_{\mathcal{C}(i,i')}}{\|\mathbf{D}_{\mathcal{C}(i,i')} \hat{\mathbf{u}}\|_2^3} \right) \right]^{-1} \mathbf{P} \right).$$

□