Uncovering Structure in High-Dimensions:
Networks and Multi-task Learning Problems

Mladen Kolar

July 2013
CMU-ML-13-106

Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Eric P. Xing, Chair
Aarti Singh
Larry Wasserman
Francis Bach (INRIA)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was funded in part by the grants NIH R01GM087694, AFOSR FA9550010247, ONR N000140910758, NSF DBI-0640543, NSF IIS-0713379, NSF Career DBI-0546594, NIH 1 R01 GM078622-01, Alfred P. Sloan Research Fellowship given to Eric P. Xing and a graduate fellowship from Facebook to Mladen Kolar.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Complex Systems; Dynamic Networks; Feature Selection; Gaussian Graphical Models; High-dimensional Inference; Markov Random Fields; Multi-task Learning; Semiparametric Estimation; Sparsity; Structure Learning; Undirected Graphical Models; Variable Screening; Varying Coefficient

To Gorana and Spomenka

Abstract

Extracting knowledge and providing insights into complex mechanisms underlying noisy high-dimensional data sets is of utmost importance in many scientific domains. Statistical modeling has become ubiquitous in the analysis of high dimensional functional data in search of better understanding of cognition mechanisms, in the exploration of large-scale gene regulatory networks in hope of developing drugs for lethal diseases, and in prediction of volatility in stock market in hope of beating the market. Statistical analysis in these high-dimensional data sets is possible only if an estimation procedure exploits hidden structures underlying data.

This thesis develops flexible estimation procedures with provable theoretical guarantees for uncovering unknown hidden structures underlying data generating process. Of particular interest are procedures that can be used on high dimensional data sets where the number of samples n is much smaller than the ambient dimension p. Learning in high-dimensions is difficult due to the curse of dimensionality, however, the special problem structure makes inference possible. Due to its importance for scientific discovery, we put emphasis on consistent structure recovery throughout the thesis. Particular focus is given to two important problems, semi-parametric estimation of networks and feature selection in multi-task learning.

Acknowledgments

Many people have helped me in completing this thesis. I would not be here at Carnegie Mellon University if it was not for my mother, Spomenka Kolar, who spent an enormous amount of time and effort in educating me at the earliest stages of my life. She provided a safety net that allowed me to freely explore different possibilities without ever worrying about what would happen in case I failed.

At Carnegie Mellon University, I was extremely fortunate to be advised by Eric Xing who taught me how to pick important problems and who provided me with freedom to create my own research agenda. His careful guidance has been invaluable. I am also very fortunate to be given the opportunity to work with many other brilliant researchers at the University: Larry Wasserman, Aarti Singh, Alessandro Rinaldo and John Lafferty. They have been mentors, advisors and collaborators. I spent a semester working with Francis Bach and Guillaume Obozinski in Paris. It was great fun to work with both of them. Matt Harrison got me excited about statistics by teaching an amazing intro course.

The academic environment at Carnegie Mellon University has been superb. It is an open and collaborative environment. I had the pleasure to write papers with Suyash Shringarpure, Pradipta Ray, Seyoung Kim, Xi Chen, Le Song, Ankur Parikh, James Sharpnack, Seunghak Lee, Sivaraman Balakrishnan, Amr Ahmed and Han Liu. My officemates Felipe Trevizan, Polo Chau, James Sharpnack, Charalampos (Babis) Tsourakakis, Yi Zhang, Austin McDonald, Chanwoo Kim and Oznur Tastan were amazing. A little known secret is that I started in the Language Technology Institute together with Jose Pablo González-Brenes, Kriti Puniyani, Sivaraman Balakrishnan and Ramnath Balasubramanian. I feel very lucky to have them as friends and peers. I thank all members of the SAILING lab who listened to my talks and contributed to forming research ideas.

Diane Stidle and Michelle Martin have always been willing to lend a helping hand whenever I needed it.

I also want to thank all my friends from far and near for many wonderful experiences and memories. I also want to thank the Smailagics, Asim, Brana, and Vedrana for their kindness and support.

Finally, I would like to acknowledge my sweetiepie, Gorana Smailagic, who has been a supporter and the love of my life for the past six years. Without her support, this would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, we have witnessed fast advancement of data-acquisition techniques in many areas, including biological domains, engineering and social sciences. As a result, new statistical and machine learning techniques are needed to help us develop a better understanding of complexities underlying large, noisy data sets.

Statistical inference in high-dimensions is challenging due to the curse of dimensionality. What makes the inference possible is that many real world systems have a special structure that can be represented with a much smaller number of parameters than the dimension of the ambient space. Even when a system cannot be represented exactly with few parameters, there are still good approximations that use few parameters and useful in providing insights into the system. This concept of parsimony commonly occurs in a number of scientific disciplines.

The main goal of this thesis is to develop flexible and principled statistical methods for uncovering hidden structure underlying high-dimensional, complex data sets with focus on scientific discovery. This thesis is naturally divided into two parts. In the first part, we focus on learning structure of time varying latent networks from nodal observations. The second part of the thesis focus on exploiting structure in multi-task learning.

1.1 Network Structure Estimation

Across the sciences, networks provide a fundamental setting for representing and interpreting information on the state of an entity, the structure and organization of communities, and changes in these over time. Traditional approaches to network analysis tend to make simplistic assumptions, such as assuming that there is only a single node or edge type, or ignoring the dynamics of the networks. Unfortunately, these classical approaches are not suitable for network data arising in contemporary applications. Modern network data can be large, dynamic, heterogeneous, noisy and incomplete. These characteristics add a degree of complexity to the interpretation and analysis of networks.

As a motivating example, let us consider estimation of cellular networks in systems biology. Studying biological networks is a difficult task, because in complex organisms, biological processes are often controlled by a large number of molecules that interact and exchange information in a spatial-temporally specific and context-dependent manner. Current approaches to studying

biological networks have primarily focused on creating a descriptive analysis of macroscopic properties, which include degree distribution, path length and motif profiles of the networks, or using graph mining tools to identify clusters and subgraphs. Such simple analysis offer limited insights into the remarkably complex functional and structural organization of a biological system, especially in a dynamic context. Furthermore, it is often common to completely ignore the dynamic context in which the data are collected. For example, in the analysis of microarray data collected over a time course it is common to infer a single static gene network. As a solution to this problem, we develop a flexible framework for inferring dynamic networks.

In this thesis, we develop flexible statistical procedures with rigorous theoretical guarantees for inferring unobservable dynamic network structure from nodal observations that are governed by the latent network. In particular, we build on the formalism of probabilistic graphical models in which we cast the problem of network learning as the problem of learning a graph structure from observational data. We develop methods for learning both undirected and directed graphical models. These estimation methods are developed for both gradually changing networks and networks with abrupt changes. Furthermore, we go beyond analysis dynamic systems only. Methods that are developed can be also used to learn conditional covariance structures, where a network depends on some other observed random variables.

Analysis of network data is an important problem in a number of disciplines [see, e.g., 53, for a textbook treatment of the topic]. However, these methods assume availability of network structure for performing a statistical analysis. In this thesis, we develop techniques that learn network structure from only nodal observations. Once a network structure is learned, any of the existing network analysis tools can be used to further investigate properties of the underlying system. Therefore, this thesis makes significant progress in advancing the boundary of what problems can be tackled using well developed network analysis tools.

1.2 Multi-task Learning

In different scientific fields, such as neuroscience and genetics, it has been empirically observed that learning jointly from related tasks (i.e., multi-task learning) improves estimation performance. For example, in biology, a genome-wide association mapping study aims to find a small set of causal single-nucleotide polymorphisms (SNPs) that account for genetic variations of a large number of genes. Identifying causal SNPs is a challenging problem for current statistical methods due to a large number of variables and low signal-to-noise ratio. However, genes in a biological pathway are co-expressed as a module and it is often assumed that a causal SNP affects multiple genes in one pathway. Therefore, once the whole biological pathway is examined, it is much easier to find the causal SNPs.

Prior to the work in this thesis, despite many investigations, the theory of variable selection in multi-task regression models was far from settled, and there was no clear picture that explained when variable selection can be done more efficiently by considering multiple tasks. Using the framework of the Normal means model, we are able to sharply characterize the theoretical properties of different estimation procedures. In particular, we provide a sharp characterization of the variable selection properties of two commonly used procedures for variable selection in high-dimensional problems, the lasso and group lasso. Interestingly, two distinct regimes emerge

showing that one or the other procedure is optimal, in the minimax sense, depending on the amount of relatedness between the tasks.

Although optimal in many settings, variable selection methods based on convex programming do not scale well to the setting when the number of variables is in the hundreds of thousands. For that reason, in this thesis, we study ways to identify relevant variables quickly. We prove that simultaneous orthogonal matching pursuit and marginal regression can be used to identify relevant variables quickly and under much less stringent conditions compared to the ones required for the lasso or group lasso.

1.3 Thesis Overview

The central focus of the thesis is uncovering unknown structure from high-dimensional data.

In Part I (Chapter 2 - Chapter 10), we focus on uncovering unknown latent networks:

- Chapter 2 reviews Markov random fields. The problem of uncovering networks is cast as a task of learning graph structure of a Markov random field. Two methods commonly used to learn graph structure in high-dimensions are reviewed. We will build on these methods in subsequent chapters.
- Chapter 3 introduces time-varying networks. These models are introduced as semi-parametric extensions of Markov random fields. Therefore, they are rather flexible in capturing real-world phenomena and at the same time easily interpretable by domain experts. We introduce general framework which will be used for estimation of time-varying networks in the subsequent Chapters.
- Chapter 4 presents algorihtms for recovery of time-varying network structure from discrete data. An algorithm for recovery of smoothly and abruptly changing networks is given. Using the algorithms, we reverse engineer the latent sequence of temporally rewiring political networks between Senators from the US Senate voting records and the latent evolving regulatory networks underlying 588 genes across the life cycle of *Drosophila melanogaster* from the microarray time course. The chapter is based on [112, 159].
- Chapter 5 establishes conditions under which the method proposed in Chapter 4, for recovery of smoothly varying networks, consistently recovers the structure of a network. This work complements previous empirical findings by providing sound theoretical guarantees for the proposed estimation procedure. The chapter is based on [103].
- Chapters 6 and 7 introduce and analyze procedures for recovery of graph structure of Gaussian graphical models. Again, sufficient conditions for consistent graph structure recovery are given, as well as efficient numerical algorithms. These chapters are based on [105, 106, 113].
- Chapter 8 is focused on conditional estimation of network structures. Unlike previous chapters, where the network structure changes as a function of time, in many applications, it is more natural to think of a network changing as a function of some other random variable. We motivate the problem with examples in portfolio selection and exploration of complex dependencies between assets. Efficient algorithms and their theoretical underpin-

nings are presented. This work was published in [111].

- Chapters 9 and 10 focus on estimation of static networks under more realistic assumptions than commonly studied in literature. Chapter 9 studies a simple two step procedure for estimating sparse precision matrices from data with missing values, which is tractable in high-dimensions and does not require imputation of the missing values [107]. Chapter 10 studies a principled framework for estimating structure of undirected graphical models from multivariate nodal data [109].

In Part II (Chapter 11 - Chapter 14), we focus on variable selection in multi-task learning:

- Chapter 11 reviews multi-task learning in the context of multiple output multivariate linear regression. The problem of variable selection in this setting is introduced.
- Chapter 12 analyzes commonly used penalties for variable selection in multi-task linear regression problems. We establish sharp bounds that characterize performance of these penalties. The chapter is based on [110].
- Chapter 13 and Chapter 14 focus on fast variable selection in multi-task problems. Problems that arise in genome-wide associations studies often involve hundred of thousands of single nucleotide polymorphisms, which are used as input variables. Problems of this size are not readily solvable using off-the-shelf solvers for convex programs. In these two chapters we analyze greedy methods that can quickly reduce the number of input variables. These chapters are based on [102, 104].

The conclusions and future directions are provided in Chapter 15.

1.4 Notation

We use $[n]$ to denote the set $\{1, \ldots, n\}$ and $[l : r]$ to denote the set $\{l, l+1, \ldots, r-1\}$. For a set $S \subset V$, we use the notation X_S to denote the set $\{X_a : a \in S\}$ of random variables. We use $\mathbf{X}$ to denote the $n \times p$ matrix whose rows consist of observations. The vector $\mathbf{X}_a = (x_{1,a}, \ldots, x_{n,a})'$ denotes a column of matrix $\mathbf{X}$ and, similarly, $\mathbf{X}_S = (\mathbf{X}_b \ : \ b \in S)$ denotes the $n \times |S|$ sub-matrix of $\mathbf{X}$ whose columns are indexed by the set S and $\mathbf{X}^{\mathcal{B}^j}$ denotes the sub-matrix $|\mathcal{B}^j| \times p$ whose rows are indexed by the set $\mathcal{B}^j$. For simplicity of notation, we will use $\setminus a$ to denote the index set $[p] \setminus \{a\}$, $\mathbf{X}_{\setminus a} = (\mathbf{X}_b \ : \ b \in [p] \setminus \{a\})$. For a vector $\mathbf{a} \in \mathbb{R}^p$, we let $S(\mathbf{a})$ denote the set of non-zero components of $\mathbf{a}$. Throughout the paper, we use $c_1, c_2, \ldots$ to denote positive constants whose value may change from line to line. For a vector $\mathbf{a} \in \mathbb{R}^n$, define $||\mathbf{a}||_1 = \sum_{i \in [n]} |a_i|$, $||\mathbf{a}||_2 = \sqrt{\sum_{i \in [n]} a_i^2}$ and $||\mathbf{a}||_\infty = \max_i |a_i|$. For a symmetric matrix $\mathbf{A}$, $\Lambda_{\min}(\mathbf{A})$ denotes the smallest and $\Lambda_{\max}(\mathbf{A})$ the largest eigenvalue. For a matrix $\mathbf{A}$ (not necessarily symmetric), we use $|||\mathbf{A}|||_\infty = \max_i \sum_j |A_{ij}|$. For two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, the dot product is denoted $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i \in [n]} a_i b_i$. For two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$, the dot product is denoted as $\langle\langle \mathbf{A}, \mathbf{B} \rangle\rangle = \mathrm{tr}(\mathbf{A}'\mathbf{B})$. Given two sequences $\{a_n\}$ and $\{b_n\}$, the notation $a_n = \mathcal{O}(b_n)$ means that there exists a constant c_1 such that $a_n \leq c_1 b_n$; the notation $a_n = \Omega(b_n)$ means that there exists a constant c_2 such that $a_n \geq c_2 b_n$ and the notation $a_n \asymp b_n$ means that $a_n = \mathcal{O}(b_n)$ and $b_n = \mathcal{O}(a_n)$. Similarly, we will use the notation $a_n = o_p(b_n)$ to denote that $b_n^{-1} a_n$ converges to 0 in probability.

Part I

Learning Network Structure

Chapter 2

Learning Network Structure

Network models have become popular as a way to abstract complex systems and gain insights into relational patterns among observed variables. For example, in a biological study, nodes of the network can represent genes in one organism and edges can represent associations or regulatory dependencies among genes. In a social domain, nodes of a network can represent actors and edges can represent interactions between actors. Recent popular techniques for modeling and exploring networks are based on the structure estimation in the probabilistic graphical models, specifically, Markov Random Fields (MRFs). These models represent conditional independence between variables, which are represented as nodes. Once the structure of the MRF is estimated, the network is drawn by connecting variables that are conditionally dependent. The hope is that this graphical representation is going to provide additional insight into the system under observation, for example, by showing how different parts of the system interact.

In this chapter, we review methods for learning structure of MRFs in high-dimensions with focus on the Ising model and the Gaussian graphical model (GGM). The Ising model represents a typical discrete MRF, while the GGMs are commonly used to represent continuous MRFs. We focus on these two models because they can be fully specified just with the first two moments. Even though they are quite simple, they are rich enough to be applicable in a number of domains and also provide an opportunity to succinctly present theoretical results. The statistical challenge is going to be structure estimation of a graphical model from a sample in a high-dimensional setting. Since the number of unknown model parameters exceeds the number of observations, classical tools, like the maximum likelihood estimator, are ill-posed in this high-dimensional setting. Therefore, additional assumption will be needed to make high-dimensional statistical inference possible. For example, we will need to assume that the parameter vector is sparse, that is, that only a few of the unknown model parameters are different from zero. Using penalized maximum likelihood (or pseudo-likelihood) estimation, we will see that the correct graph structure can be recovered consistently.

2.1 Preliminaries

In recent years, we have witnessed fast advancement of data-acquisition techniques in many areas, including biological domains, engineering and social sciences. As a result, new statistical

and machine learning techniques are needed to help us develop a better understanding of complexities underlying large, noisy data sets. Networks have been commonly used to abstract noisy data and provide an insight into regularities and dependencies between observed variables. For example, in a biological study, nodes of the network can represent genes in one organism and edges can represent associations or regulatory dependencies among genes. In a social domain, nodes of a network can represent actors and edges can represent interactions between actors. Recent popular techniques for modeling and exploring networks are based on the structure estimation in the probabilistic graphical models, specifically, Markov Random Fields (MRFs). These models represent conditional independence between variables, which are represented as nodes. Once the structure of the MRF is estimated, the network is drawn by connecting variables that are conditionally dependent.

Let $G = (V, E)$ represent a graph, of which V denotes the set of vertices, and E denotes the set of edges over vertices. Depending on the specific application of interest, a node $a \in V$ can represent a gene, a stock, or a social actor, and an edge $(a, b) \in E$ can represent a relationship (e.g., correlation, influence, friendship) between actors a and b. Let $\mathbf{X} = (X_1, \ldots, X_p)'$, where $p = |V|$, be a random vector of nodal states following a probability distribution indexed by $\boldsymbol{\theta} \in \Theta$. Under a MRF, the nodal states X_a's are assumed to be either discrete or continuous and the edge set $E \subseteq V \times V$ encodes certain conditional independence assumptions among components of the random vector $\mathbf{X}$, for example, the random variable X_a is conditionally independent of the random variable X_b given the rest of the variables if $(a, b) \notin E$. We focus on two types of MRFs: the Ising model and the Gaussian graphical models. We specify their forms below.

The Ising model arises as a special case of discrete MRFs, where each node takes binary nodal states. That is, under the Ising model, we have $X_a \in \mathcal{X} \equiv \{-1, 1\}$, for all $a \in V$ and the joint probability of $\mathbf{X} = \mathbf{x}$ can be expressed by a simple exponential family model:

$$\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left\{\sum_{a<b} \theta_{ab} x_a x_b\right\} \tag{2.1}$$

where $Z(\boldsymbol{\theta}) = \sum_{\mathbf{x}\in\{-1,1\}^p} \exp\left\{\sum_{a<b} \theta_{ab} x_a x_b\right\}$ denotes the partition function that is intractable to compute (even for moderately large p) and the weight potentials are given by θ_{ab} for all $(a, b) \in E$. Under the Ising model, the model is completely defined by the vector of parameters $(\theta_{ab})_{(a,b)\in V\times V}$. Furthermore, the parameters specify the graph structure, that is, we have that $\theta_{ab} = 0$ for all $(a, b) \notin E$.

The Gaussian graphical models are used as the simplest continuous MRFs, since the probability distribution under the GGM can be fully specified with the first two moments. Let

$$\mathbf{X} = (X_1, \ldots, X_p)' \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$$

be a p-dimensional multivariate Gaussian random variable with mean zero and covariance $\boldsymbol{\Sigma} = (\sigma_{ab})_{(a,b)\in V\times V}$. Associated with the vector $\mathbf{X}$ is a graph $G = (V, E)$ that encodes the conditional independence assumptions between the components of $\mathbf{X}$. Let $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1} = (\sigma_{ab})_{(a,b)\in V\times V}$ be the precision matrix. The precision matrix encodes the conditional independence assumptions as well, in the sense that variable X_a is conditionally independent of X_b given the rest of variables if and only if $\omega_{ab} = 0$. Therefore the graph G is specified directly by the positions of non-zero

elements of the precision matrix, that is, an edge $e_{ab} \in E$ only if $\omega_{ab} \neq 0$. An element ω_{ab} of the precision matrix is proportional to the partial correlation between random variables X_a and X_b. Indeed, we have

$$\rho_{ab|V\setminus\{a,b\}} = -\frac{\omega_{ab}}{\sqrt{\omega_{aa}\omega_{bb}}}.$$

This relationship will be used later to motivate the algorithms for learning structure of GGMs. All these properties are well known and can be found in a monograph on the Gaussian graphical models [130].

2.2 Structure Learning Procedures

One of the most important tasks in graphical models is that of learning the graph structure given a sample. Let $\mathcal{D}_n = \{\mathbf{x}_i \sim \mathbb{P}_{\boldsymbol{\theta}} \mid i \in [n]\}$ be a sample of n *i.i.d.* p-dimensional vectors drawn from the distribution $\mathbb{P}_{\boldsymbol{\theta}}$. The goal is to estimate conditional independence assumptions between the components of $\mathbf{X} \sim \mathbb{P}_{\boldsymbol{\theta}}$. In a high-dimensional setting, when $p \gg n$, it is common to use penalization or regularization methods in order to fit models. We will use the estimation procedures of the form

$$\arg\min_{\boldsymbol{\theta}} L(\mathcal{D}_n; \boldsymbol{\theta}) + \mathrm{pen}_\lambda(\boldsymbol{\theta}) \tag{2.2}$$

where $L(\cdot;\theta)$ is the convex loss function, $\mathrm{pen}_\lambda(\cdot)$ is the regularization term and λ is a tuning parameter. The first term in the objective is measuring the fit to data, while the second one measures the complexity of the model. The regularization term is used to encode some prior assumptions about the model, e.g., sparsity of the graph structure or the way the graph structure changes over time. The loss functions that is used will be problem specific. For example, in the case of the Gaussian graphical models, we will use the negative log-likelihood, while in the case of discrete MRFs a surrogate to the negative log-likelihood will be used.

2.2.1 Learning structure of an Ising model

In general, learning structure of an Ising model is hard [33] due to the combinatorial explosion of the search space of graphs. Therefore, score based searches are limited to restricted classes of models, such as, trees, polytrees and bounded tree-width hypertrees [27, 46, 163]. The computational complexity of search based procedures arises from two sources. First, there are $2^{\binom{p}{2}}$ potential graph structures to be evaluated. Second, computing a score for any fixed graph structure involves computing the normalization constant, which is intractable in general. Other methods for learning the graph structure include minimizing the Kullback-Leibler divergence [5] and other pseudo-likelihood methods [13, 29].

Ravikumar et al. [151] use an optimization approach to estimate the graph structure in a high-dimensional setting. This approach can be cast in the optimization framework outlined in (2.2), where the loss function is a node conditional likelihood and the ℓ_1 norm of a coefficient vector is used as a penalty function. Therefore, the optimization procedure decomposes across different nodes and as a result can be maximized efficiently. We describe the procedure in details below.

The estimation procedure in [151] is based on the neighborhood selection technique, where the graph structure is estimated by combining the local estimates of neighborhoods of each node. For each vertex $a \in V$, define the set of neighboring edges

$$S(a) = \{(a, b) \mid (a, b) \in E\}.$$

Under the model (2.1), the conditional distribution of X_a given other variables $\mathbf{X}_{\setminus a} = \{X_b \mid b \in V \setminus a\}$ takes the form

$$\mathbb{P}_{\boldsymbol{\theta}_a}(x_a \mid \mathbf{X}_{\setminus a} = \mathbf{x}_{\setminus a}) = \frac{\exp(2x_a\langle \boldsymbol{\theta}_a, \mathbf{x}_{\setminus a}\rangle)}{\exp(2x_a\langle \boldsymbol{\theta}_a, \mathbf{x}_{\setminus a}\rangle) + 1}, \tag{2.3}$$

where $\langle \mathbf{a}, \mathbf{b}\rangle = \mathbf{a}'\mathbf{b}$ denotes the dot product. Under the model (2.3) the log-likelihood, for one data-point, can be written in the following form

$$\begin{aligned}\gamma(\boldsymbol{\theta}_a; \mathbf{x}_i) &= \log \mathbb{P}_{\boldsymbol{\theta}_a}(x_{i,a} \mid \mathbf{x}_{i,\setminus a}) \\ &= x_{i,a}\langle \boldsymbol{\theta}_a, \mathbf{x}_{i,\setminus a}\rangle - \log\left(\exp(\langle \boldsymbol{\theta}_a, \mathbf{x}_{i,\setminus a}\rangle) + \exp(-\langle \boldsymbol{\theta}_a, \mathbf{x}_{i,\setminus a}\rangle)\right),\end{aligned}$$

where, for simplicity, we write $\mathbb{P}_{\boldsymbol{\theta}_a}(x_{i,a} \mid \mathbf{X}_{i,\setminus a} = \mathbf{x}_{i,\setminus u})$ as $\mathbb{P}_{\boldsymbol{\theta}_a}(x_{i,a} \mid \mathbf{x}_{i,\setminus u})$. The estimator $\widehat{\boldsymbol{\theta}}_a$ of the vector $\boldsymbol{\theta}_a$ is defined as the solution to the following convex program:

$$\widehat{\boldsymbol{\theta}}_a = \min_{\boldsymbol{\theta}_a \in \mathbb{R}^{p-1}} \left\{\ell\left(\boldsymbol{\theta}_a; \mathcal{D}_n\right) + \lambda||\boldsymbol{\theta}_a||_1\right\} \tag{2.4}$$

where $\ell(\boldsymbol{\theta}_a; \mathcal{D}_n) = -\sum_{i\in[n]} \gamma(\boldsymbol{\theta}_a; \mathbf{x}_i)$ is the logloss. Based on the vector $\widehat{\boldsymbol{\theta}}_a$, we have the following estimate of the neighborhood

$$\widehat{S}(a) = \left\{(a, b) \mid b \in V \setminus a,\ \widehat{\theta}_{ab} \neq 0\right\}.$$

The structure of graph G^τ is consistently estimated if every neighborhood is recovered, that is, $\widehat{S}(a) = S(a)$ for all $a \in V$. In §4 and §5, we build on this procedure to estimate time-varying networks from discrete nodal observations.

2.2.2 Learning structure of a Gaussian graphical model

A large amount of literature in both statistics and machine learning has been devoted to the problem of estimating sparse precision matrices, as they encode conditional independence structure between random variables. The problem of estimating precision matrices with zeros is known in statistics as *covariance selection* and was introduced in the seminal paper by [47]. An introduction to classical approaches, which are commonly based on identifying the correct set of non-zero elements and then estimating the non-zero elements, can be found in, for example, [56, 130]. [43] proposed a method that tests if partial correlations are different from zero. This and other classical methods can be applied when the number of dimensions p is small in comparison to the sample size n. However, due to the technological improvements of data collection processes, we have seen a surge in the number of high-dimensional data sets. As a result, more

recent literature on estimating sparse precision matrices is focused on methods suitable for high-dimensional problems where the number of variables p can be much larger than the sample size n.

[135] proposed a procedure based on *neighborhood selection* of each node via the ℓ_1 penalized regression. Leveraging the lasso [175] they efficiently estimate the non-zero pattern of the precision matrix. Like the approach in 2.2.1, this procedure uses a pseudo-likelihood, which decomposes across different nodes, to estimate graph edges and, although the estimated parameters are not consistent, the procedure recovers the graph structure consistently under a set of suitable conditions.

Let $\mathbf{\Omega} = (\omega_{ab})_{ab}$ be the precision matrix The neighborhood of the node a can be directly read of from the precision matrix as

$$S(a) = \{b \in V \setminus a \mid \omega_{ab} \neq 0\}.$$

It is a well known result for Gaussian graphical models that the elements of

$$\boldsymbol{\theta}^a = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^{p-1}} \mathbb{E}\left(X_a - \sum_{b \in \setminus a} X_b \theta_b\right)^2$$

are given by $\theta_b^a = -\omega_{ab}/\omega_{aa}$. Therefore, the neighborhood of a node a, $S(a)$, is equal to the set of non-zero coefficients of $\boldsymbol{\theta}^a$. Using the expression for $\boldsymbol{\theta}^a$, we can write $X_a = \sum_{b \in S_a} X_b \theta_b^a + \epsilon$, where ϵ is independent of $X_{\setminus a}$. The neighborhood selection procedure was motivated by the above relationship between the regression coefficients and the elements of the precision matrix. [135] proposed to solve the following optimization procedure

$$\widehat{\boldsymbol{\theta}}^a = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^{p-1}} \frac{1}{n} \sum_{i \in [n]} \left(x_{i,a} - \mathbf{x}'_{i,\setminus a} \boldsymbol{\theta}\right)^2 + \lambda ||\boldsymbol{\theta}||_1 \tag{2.5}$$

and proved that the non-zero coefficients of $\widehat{\boldsymbol{\theta}}^a$ consistently estimate the neighborhood of the node a, under a suitably chosen penalty parameter λ.

A related approach is proposed in [146] who consider a different neighborhood selection procedure for the structure estimation in which they estimate all neighborhoods jointly and as a result obtain a global estimate of the graph structure that empirically improves the performance on a number of networks. These neighborhood selection procedures are suitable for large-scale problems due to availability of fast solvers to ℓ_1 penalized problems [55, 73].

Another popular technique for estimating sparse precision matrix is based on ℓ_1-norm penalized maximum likelihood [195], which simultaneously estimates the graph structure and the elements of the covariance matrix. The penalized likelihood approach involves solving a semidefinite program (SDP)

$$\widehat{\mathbf{\Omega}} = \arg\min_{\mathbf{\Omega} \succ 0} \left\{ \operatorname{tr} \mathbf{\Omega} \widehat{\mathbf{\Sigma}} - \log |\mathbf{\Omega}| + \lambda ||\mathbf{\Omega}||_1 \right\}, \tag{2.6}$$

where $\widehat{\mathbf{\Sigma}}$ is a sample covariance matrix. A number of authors have worked on efficient solvers that exploit the special structure of the problem (see, for example, [19, 52, 71, 85, 154, 195]). Statistical properties of the above procedure were analyzed in [152, 154]. Some authors have proposed to use a nonconcave penalty instead of the ℓ_1 penalty, which tries to remedy the bias that the ℓ_1 penalty introduces [64, 67, 202]. See also [37, 39].

2.3 Discussion

In this chapter, we have discussed common approaches to estimation of graph structure in Markov random fields in a high-dimensional setting. The focus was on methods where the structure is estimated from i.i.d. data. Most of the work in the literature has been by the simplifying assumption of static network structure. In the next chapter, we motivate estimation of time-varying networks as a useful and flexible tool for exploring complex systems. Estimation framework will build on the methods presented here.

Chapter 3

Time Varying Networks

As discussed in Chapter 2, stochastic networks are a plausible representation of the relational information among entities in dynamic systems such as living cells or social communities. While there is a rich literature in estimating a static or temporally invariant network from observation data, little has been done toward estimating time-varying networks from time series of entity attributes. In this chapter, we introduce and motivate time-varying networks. A general estimation framework is presented, which is going to be used in subsequent chapters.

3.1 Motivation

In many problems arising from natural, social, and information sciences, it is often necessary to analyze a large quantity of random variables interconnected by a complex dependency network, such as the expressions of genes in a genome, or the activities of individuals in a community. Real-time analysis of such networks is important for understanding and predicting the organizational processes, modeling information diffusion, detecting vulnerability, and assessing the potential impact of interventions in various natural and built systems. It is not unusual for network data to be large, dynamic, heterogeneous, noisy, incomplete, or even unobservable. Each of these characteristics adds a degree of complexity to the interpretation and analysis of networks. One of the fundamental questions in this thesis is the following: how can one reverse engineer networks that are latent, and topologically evolving over time, from time series of nodal attributes?

Prior to our work, literature mainly focused on estimating a single static network underlying a complex system. However, in reality, many systems are inherently dynamic and can be better explained by a dynamic network whose structure evolves over time. We develop statistical methodology of dealing with the following real world problems:

- *Analysis of gene regulatory networks.* Suppose that we have a set of n microarray measurements of gene expression levels, obtained at different stages during the development of an organism or at different times during the cell cycle. Given this data, biologists would like to get insight into dynamic relationships between different genes and how these relations change at different stages of development. The problem is that at each time point there is only one or at most a few measurements of the gene expressions; and a naive approach to

estimating the gene regulatory network, which uses only the data at the time point in question to infer the network, would fail. To obtain a good estimate of the regulatory network at any time point, we need to leverage the data collected at other time points and extract some information from them.

- *Analysis of stock market.* In a finance setting, we have values of different stocks at each time point. Suppose, for simplicity, that we only measure whether the value of a particular stock is going up or down. We would like to find the underlying transient relational patterns between different stocks from these measurements and get insight into how these patterns change over time. Again, we only have one measurement at each time point and we need to leverage information from the data obtained at nearby time points.
- *Understanding social networks.* There are 100 Senators in the U.S. Senate and each can cast a vote on different bills. Suppose that we are given n voting records over some period of time. How can one infer the latent political liaisons and coalitions among different senators and the way these relationships change with respect to time and with respect to different issues raised in bills just from the voting records?

The aforementioned problems have commonality in estimating a sequence of time-specific latent relational structures between a fixed set of entities (i.e., variables), from a time series of observation data of entities states; and the relational structures between the entities are time evolving, rather than being invariant throughout the data collection period. A key technical hurdle preventing us from an in-depth investigation of the mechanisms underlying these complex systems is the unavailability of *serial snapshots* of the time-varying networks underlying these systems. For example, for a realistic biological system, it is impossible to experimentally determine time-specific networks for a series of time points based on current technologies such as two-hybrid or ChIP-chip systems. Usually, only time series measurements, such as microarray, stock price, etc., of the activity of the nodal entities, but not their linkage status, are available. Our goal is to recover the latent time-varying networks with temporal resolution up to every single time point based on time series measurements. Most of the existing work on structure estimation assumes that the data generating process is time-invariant and that the relational structure is fixed. (see, for example, [13, 19, 67, 71, 76, 135, 146, 151, 152, 154, 185, 195] and references therein), which may not be a suitable assumption for the described problems. Chapter 2 presents some of these methods. The focus of this chapter is to present a general framework for estimating dynamic network structure from a time series of entity attributes.

3.2 Estimation Framework

In the following few chapters, we will assume that we are given a sequence of observations

$$\mathcal{D}_n = \{\mathbf{x}^t \sim \mathbb{P}_{\boldsymbol{\theta}^t} \mid t \in \mathcal{T}_n\}$$

where $\mathcal{T}_n = \{1/n, 2/n, \ldots, 1\}$ is an index set. The observations are independent (but not identically distributed) samples from a series of time-evolving MRFs $\{\mathbb{P}_{\boldsymbol{\theta}^t}(\cdot)\}_{t\in\mathcal{T}_n}$. The goal is to estimate the parameters of the sequence of probability distributions $\{\mathbb{P}_{\boldsymbol{\theta}^t}\}_{t\in\mathcal{T}_n}$ or more specifically conditional independence assumptions encoded by a sequence of graphs $\{G^t\}_{t\in\mathcal{T}_n}$. The problem

of dynamic structure estimation is of high importance in domains that lack prior knowledge or measurement techniques about the interactions between different actors; and such estimates can provide desirable information about the details of relational changes in a complex system. It might seem that the problem is ill-defined, since for any time point we have at most one observation; however, as we will show shortly, under a set of suitable assumptions the problem is indeed well defined and the series of underlying graph structures can be estimated. For example, we may assume that the probability distributions are changing *smoothly* over time, or there exists a partition of the interval $[0, 1]$ into segments where the graph structure within each segment is invariant.

The estimation procedure we use to estimate the structure of a time-varying MRF will depend on the assumptions we make on the network dynamics. The general form of the estimation procedure will be as in §2.2. In the case that the network parameters change smoothly, we will use estimation procedures of the form

$$\widehat{\boldsymbol{\theta}}^{\tau} = \arg\min_{\boldsymbol{\theta}} \sum_{t \in \mathcal{T}_n} w_t^{\tau} \gamma(\boldsymbol{\theta}; \mathbf{x}^t) + \mathrm{pen}_{\lambda}(\boldsymbol{\theta}). \tag{3.1}$$

The first term is the local log-likelihood (or pseudo-likelihood), with $\gamma(\boldsymbol{\theta}; \mathbf{x}^t)$ being the log-likelihood (or pseudo-likelihood) and the weight w_t^{τ} defines the contribution of the point $\mathbf{x}^t$ at a time point $\tau \in [0, 1]$. The regularization term $\mathrm{pen}_{\lambda}(\boldsymbol{\theta})$ encourages sparsity of the estimated network at the time point $\tau \in [0, 1]$. Note that the above estimation procedure estimates the time-varying network only at one time point τ. In order to get insight into dynamics, we need to solve (3.1) for a number of time points τ, for example, for all $\tau \in \mathcal{T}_n$.

When the underlying network parameters are piecewise constant, we will use estimation procedures of the form

$$\{\widehat{\boldsymbol{\theta}}^t\}_{t \in \mathcal{T}_n} = \arg\min_{\{\boldsymbol{\theta}^t\}_{t \in \mathcal{T}_n}} \sum_{t \in \mathcal{T}_n} \gamma(\boldsymbol{\theta}; \mathbf{x}^t) + \mathrm{pen}_{\lambda}\left(\{\boldsymbol{\theta}^t\}_{t \in \mathcal{T}_n}\right). \tag{3.2}$$

Compared to the optimization problem in (3.1), here the whole dynamic network is estimated at once. The regularization term will encourage both sparsity of the parameter vector at each time point and the way parameters change over time.

In §4 and §5 we specialize optimization problems in (3.1) and (3.2) to problems of learning time-varying network structure from binary nodal observations. In §6 and §7, the two optimization are discussed in the context of learning network structure of Gaussian graphical models. In §8, a related problem of estimating conditional networks is discussed.

3.3 Related Work

In §2 we have discussed estimation of static networks from i.i.d. data. Here we discuss work related to estimation of dynamic networks. With few exceptions [75, 92, 166, 206], little has been done on modeling dynamical processes that guide topological rewiring and semantic evolution of networks over time. In particular, prior to our work, very little has been done toward estimating the time-varying graph topologies from observed nodal states, which represent attributes

of entities forming a network. [92] introduced a new class of models to capture dynamics of networks evolving over discrete time steps, called *temporal Exponential Random Graph Models* (tERGMs). This class of models uses a number of statistics defined on time-adjacent graphs, for example, "edge-stability," "reciprocity," "density," "transitivity," etc., to construct a log-linear graph transition model $P(G^t|G^{t-1})$ that captures dynamics of topological changes. [75] incorporate a hidden Markov process into the tERGMs, which imposes stochastic constraints on topological changes in graphs, and, in principle, show how to infer a time-specific graph structure from the posterior distribution of G^t, given the time series of node attributes. Unfortunately, even though this class of model is very expressive, the sampling algorithm for posterior inference scales only to small graphs with tens of nodes.

Other literature on inferring time inhomogeneous networks can be divided into two categories: estimation of directed graphical models and estimation of undirected graphical models. Literature on estimating time-inhomogeneous directed networks usually assumes a time-varying vector auto-regressive model for observed data [see, for example, 41, 42, 58, 79, 80, 81, 86, 99, 126, 145, 149, 158, 160, 187], a class of models that can be represented in the formalism of time-inhomogeneous Dynamic Bayesian Networks although not all authors use terminology commonly used in the Dynamic Bayesian Networks literature. Markov switching linear dynamical systems are another popular choice for modeling non-stationary time series [see, for example, 4, 44, 59, 95, 161, 196]. This body of work has focused on developing flexible models capable of capturing different assumptions on the underlying system, efficient algorithms and sampling schemes for fitting these models. Although a lot of work has been done in this area, little is known about finite sample and asymptotic properties regarding the consistent recovery of the underlying networks structures. Some asymptotic results are given in [160]. Due to the complexity of MCMC sampling procedures, existing work does not handle well networks with hundreds of nodes, which commonly arise in practice. Finally, the biggest difference from our work is that the estimated networks are directed. [178] point our that undirected models constitute the simplest class of models, whose understanding is crucial for the study of directed models and models with both, directed and undirected edges. [168] and [192] study estimation of time-varying Gaussian graphical models in a Bayesian setting. [168] use a reversible jump MCMC approach to estimate the time-varying variance structure of the data. [192] proposed an iterative procedure to segment the time-series using the dynamic programming approach developed by [69] and fit a Gaussian graphical model using the penalized maximum likelihood approach on each segment. To the best of our knowledge, [206] is the first work that focuses on consistent estimation, in the Frobenius norm, of covariance and concentration matrix under the assumption that the time-varying Gaussian graphical model changes smoothly over time. However, the problem of consistent estimation of the non-zero pattern in the concentration matrix, which corresponds to the graph structure estimation, is not addressed there. Note that the consistency of the graph structure recovery does not immediately follow from the consistency of the concentration matrix. Network estimation consistency for this smoothly changing model is established in [105]. Time-varying Gaussian graphical models with abrupt changes in network structure were studied in [106], where consistent network recovery is established using a completely different proof technique. A related problem is that of estimating conditional covariance matrices [111, 193], where in place of time, which is deterministic quantity, one has a random quantity. Methods for estimating time-varying discrete Markov random fields were given in [2] and [112], however, no results on the consis-

tency of the network structure were given. Note that a lot of the work appeared after our initial work was communicated [103].

3.4 Discussion

In this chapter, we have discussed a framework for estimating dynamic networks. This framework will be specialized to different models and assumptions on the way the structure changes over time in the following chapters. The framework extends the common estimation tools used for learning static networks.

Chapter 4

Estimating time-varying networks from binary nodal observations

In this chapter we present two new machine learning methods for estimating time-varying networks, which both build on a temporally smoothed ℓ_1-regularized logistic regression formalism that can be cast as a standard convex-optimization problem and solved efficiently using generic solvers scalable to large networks. We report promising results on recovering simulated time-varying networks. For real data sets, we reverse engineer the latent sequence of temporally rewiring political networks between Senators from the US Senate voting records and the latent evolving regulatory networks underlying 588 genes across the life cycle of *Drosophila melanogaster* from the microarray time course.

4.1 Preliminaries

Let $\mathcal{D}_n = \{\mathbf{x}^t \sim \mathbb{P}_{\boldsymbol{\theta}^t} \mid t \in \mathcal{T}_n\}$ be an independent sample of n observation from a time series, obtained at discrete time steps indexed by $\mathcal{T}_n = \{1/n, 2/n, \ldots, 1\}$ (for simplicity, we assume that the observations are equidistant in time). Each sample point comes from a different discrete time step and is distributed according to a distribution $\mathbb{P}_{\boldsymbol{\theta}^t}$ indexed by $\boldsymbol{\theta}^t \in \Theta$. In particular, we will assume that $\mathbf{X}^t$ is a p-dimensional random variable taking values from $\{-1, 1\}^p$ with a distribution of the following form:

$$\mathbb{P}_{\boldsymbol{\theta}^t}(\mathbf{x}) = \frac{1}{Z(\boldsymbol{\theta}^t)} \exp\left(\sum_{(u,v)\in E^t} \theta^t_{uv} x_u x_v \right), \tag{4.1}$$

where $Z(\boldsymbol{\theta}^t)$ is the partition function, $\boldsymbol{\theta}^t \in \mathbb{R}^{\binom{p}{2}}$ is the parameter vector, and $G^t = (V, E^t)$ is an undirected graph representing conditional independence assumptions among subsets of the p-dimensional random vector $\mathbf{X}^t$. Recall that $V = \{1, \ldots, p\}$ is the node set and each node corresponds with one component of the vector $\mathbf{X}^t$.

The model given in (4.1) can be thought of as a nonparametric extension of conventional MRFs, in the similar way as the varying-coefficient models [40, 96] are thought of as an extension to the linear regression models. The difference between the model given in (4.1) and an MRF

model is that our model allows for parameters to change, while in MRF the parameters are considered fixed. Allowing parameters to vary over time increases the expressiveness of the model, and make it more suitable for longitudinal network data. For simplicity of presentation, in this chapter we consider time-varying MRFs with only pairwise potentials as in (4.1). Note that in the case of discrete MRFs there is no loss of generality by considering only pairwise interactions, since any MRF with higher-order interactions can be represented with an equivalent MRF with pairwise interactions [191].

In this chapter, we are addressing the following graph structure estimation problem:

> Given any time point $\tau \in [0,1]$ estimate the graph structure associated with $\mathbb{P}_{\boldsymbol{\theta}^\tau}$, given the observations $\mathcal{D}_n$.

To obtain insight into the dynamics of changes in the graph structure, one only needs to estimate graph structure for multiple time-points, for example, for every $\tau \in \mathcal{T}_n$.

We specialize the general estimation framework described in §3 to binary nodal observations. Discussion that follows extends the setup introduced in §2.2.1 to allow for estimation of time-varying networks from binary observations.

The graph structure G^τ is encoded by the locations of the nonzero elements of the parameter vector $\boldsymbol{\theta}^\tau$, which we refer to as the nonzero pattern of the parameter $\boldsymbol{\theta}^\tau$. Components of the vector $\boldsymbol{\theta}^\tau$ are indexed by distinct pairs of nodes and a component of the vector θ^τ_{uv} is nonzero if and only if the corresponding edge $(u,v) \in E^\tau$. Throughout the rest of the chapter we will focus on estimation of the nonzero pattern of the vector $\boldsymbol{\theta}^\tau$ as a way to estimate the graph structure. Let $\boldsymbol{\theta}^\tau_u$ be the $(p-1)$-dimensional subvector of parameters

$$\boldsymbol{\theta}^\tau_u := \{\theta^\tau_{uv} \mid v \in V \setminus u\}$$

associated with each node $u \in V$, and let $S^\tau(u)$ be the set of edges adjacent to a node u at a time point τ:

$$S^\tau(u) := \{(u,v) \in V \times V \mid \theta^\tau_{uv} \neq 0\}.$$

Observe that the graph structure G^τ can be recovered from the local information on neighboring edges $S^\tau(u)$, for each node $u \in V$, which can be obtained from the nonzero pattern of the subvector $\boldsymbol{\theta}^\tau_u$ alone. The main focus of this section is on obtaining node-wise estimators $\widehat{\boldsymbol{\theta}}^\tau_u$ of the nonzero pattern of the subvector $\boldsymbol{\theta}^\tau_u$, which are then used to create estimates

$$\widehat{S}^\tau(u) := \{(u,v) \in V \times V \mid \widehat{\theta}^\tau_{uv} \neq 0\}, \qquad u \in V.$$

Note that the estimated nonzero pattern might be asymmetric, for example, $\widehat{\theta}^\tau_{uv} = 0$, but $\widehat{\theta}^\tau_{vu} \neq 0$. We consider using the $\min$ and $\max$ operations to combine the estimators $\widehat{\theta}^\tau_{uv}$ and $\widehat{\theta}^\tau_{vu}$. Let $\widetilde{\boldsymbol{\theta}}^\tau$ denote the combined estimator. The estimator combined using the $\min$ operation has the following form:

$$\widetilde{\theta}_{uv} = \begin{cases} \widehat{\theta}_{uv}, & \text{if } |\widehat{\theta}_{uv}| < |\widehat{\theta}_{vu}|, \\ \widehat{\theta}_{vu}, & \text{if } |\widehat{\theta}_{uv}| \geq |\widehat{\theta}_{vu}|, \end{cases} \qquad \text{“min\_symmetrization,”} \tag{4.2}$$

which means that the edge (u,v) is included in the graph estimate only if it appears in both estimates $\widehat{S}^\tau(u)$ and $\widehat{S}^\tau(v)$. Using the $\max$ operation, the combined estimator can be expressed

as

$$\widetilde{\theta}_{uv} = \begin{cases} \widehat{\theta}_{uv}, & \text{if } |\widehat{\theta}_{uv}| > |\widehat{\theta}_{vu}|, \\ \widehat{\theta}_{vu}, & \text{if } |\widehat{\theta}_{uv}| \leq |\widehat{\theta}_{vu}|, \end{cases} \qquad \text{“max\_symmetrization,”} \tag{4.3}$$

and, as a result, the edge (u, v) is included in the graph estimate if it appears in at least one of the estimates $\widehat{S}^\tau(u)$ or $\widehat{S}^\tau(v)$.

A stronger notion of structure estimation is that of *signed edge recovery* in which an edge $(u, v) \in E^\tau$ is recovered together with the sign of the parameter $\mathrm{sign}(\theta^\tau_{uv})$. For each vertex $u \in V$, similar to the set $S^\tau(u)$, we define the set of *signed neighboring edges* $S^\tau_\pm(u) := \{(\mathrm{sign}(\theta^\tau_{uv}), (u, v)) \ : \ (u, v) \in S^\tau(u)\}$, which can be determined from the signs of elements of the $(p-1)$-dimensional subvector of parameters $\boldsymbol{\theta}^\tau_u$. Based on the vector $\widehat{\boldsymbol{\theta}}^\tau_u$, we have the following estimate of the signed neighborhood:

$$\widehat{S}^\tau_\pm(u) := \Big\{(\mathrm{sign}(\widehat{\theta}^\tau_{uv}), (u, v)) \ : \ v \in V \backslash u, \ \widehat{\theta}^\tau_{uv} \neq 0\Big\}. \tag{4.4}$$

An estimator $\widehat{\boldsymbol{\theta}}^\tau_u$ is obtained through the use of pseudo-likelihood based on the conditional distribution of X^τ_u given the other of variables $\mathbf{X}^\tau_{\backslash u} = \{X^\tau_v \mid v \in V \setminus u\}$. Although the use of pseudo-likelihood fails in certain scenarios, for example, estimation of Exponential Random Graphs (see [180] for a recent study), the graph structure of an Ising model can be recovered from an i.i.d. sample using the pseudo-likelihood, as shown in [151]. Under the model (4.1), the conditional distribution of X^τ_u given the other variables $\mathbf{X}^\tau_{\backslash u}$ takes the form

$$\mathbb{P}_{\boldsymbol{\theta}^\tau_u}(x^\tau_u | \mathbf{X}^\tau_{\backslash u} = \mathbf{x}^\tau_{\backslash u}) = \frac{\exp(x^\tau_u \langle \boldsymbol{\theta}^\tau_u, \mathbf{x}^\tau_{\backslash u} \rangle)}{\exp(x^\tau_u \langle \boldsymbol{\theta}^\tau_u, \mathbf{x}^\tau_{\backslash u} \rangle) + \exp(-x^\tau_u \langle \boldsymbol{\theta}^\tau_u, \mathbf{x}^\tau_{\backslash u} \rangle)}, \tag{4.5}$$

where $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}'\mathbf{b}$ denotes the dot product. For simplicity, we will write $\mathbb{P}_{\boldsymbol{\theta}^\tau_u}(x^\tau_u | \mathbf{X}^\tau_{\backslash u} = \mathbf{x}^\tau_{\backslash u})$ as $\mathbb{P}_{\boldsymbol{\theta}^\tau_u}(x^\tau_u | \mathbf{x}^\tau_{\backslash u})$. Observe that the model given in equation (4.5) can be viewed as expressing X^τ_u as the response variable in the generalized varying-coefficient models with $\mathbf{X}^\tau_{\backslash u}$ playing the role of covariates. Under the model given in equation (4.5), the conditional log-likelihood, for the node u at the time point $t \in \mathcal{T}_n$, can be written in the following form:

$$\begin{aligned} \gamma(\boldsymbol{\theta}_u; \mathbf{x}^t) &= \log \mathbb{P}_{\boldsymbol{\theta}_u}(x^t_u | \mathbf{x}^t_{\backslash u}) \\ &= x^t_u \langle \boldsymbol{\theta}_u, \mathbf{x}^t_{\backslash u} \rangle - \log\big(\exp(\langle \boldsymbol{\theta}_u, \mathbf{x}^t_{\backslash u} \rangle) + \exp(-\langle \boldsymbol{\theta}_u, \mathbf{x}^t_{\backslash u} \rangle)\big). \end{aligned} \tag{4.6}$$

The nonzero pattern of $\boldsymbol{\theta}^\tau_u$ can be estimated by maximizing the conditional log-likelihood given in equation (4.6). What is left to show is how to combine the information across different time points, which will depend on the assumptions that are made on the unknown vector $\boldsymbol{\theta}^t$.

The primary focus is to develop methods applicable to data sets with the total number of observations n small compared to the dimensionality $p = p_n$. Without assuming anything about $\boldsymbol{\theta}^t$, the estimation problem is ill-posed, since there can be more parameters than samples. A common way to deal with the estimation problem is to assume that the graphs $\{G^t\}_{t \in \mathcal{T}_n}$ are sparse, that is, the parameter vectors $\{\boldsymbol{\theta}^t\}_{t \in \mathcal{T}_n}$ have only few nonzero elements. In particular, we assume that each node u has a small number of neighbors, that is, there exists a number $s \ll p$ such that it upper bounds the number of edges $|S^\tau(u)|$ for all $u \in V$ and $\tau \in \mathcal{T}_n$. In many real data sets the

sparsity assumption holds quite well. For example, in a genetic network, rarely a regulator gene would control more than a handful of regulatees under a specific condition [51]. Furthermore, we will assume that the parameter vector $\boldsymbol{\theta}^t$ behaves "nicely" as a function of time. Intuitively, without any assumptions about the parameter $\boldsymbol{\theta}^t$, it is impossible to aggregate information from observations even close in time, because the underlying probability distributions for observations from different time points might be completely different. In this chapter, we will consider two ways of constraining the parameter vector $\boldsymbol{\theta}^t$ as a function of time:

- *Smooth changes in parameters.* We first consider that the distribution generating the observation changes smoothly over the time, that is, the parameter vector $\boldsymbol{\theta}^t$ is a smooth function of time. Formally, we assume that there exists a constant $M > 0$ such that it upper bounds the following quantities:

$$\max_{u,v \in V \times V} \sup_{t \in [0,1]} \left| \frac{\partial}{\partial t} \theta^t_{uv} \right| < M, \qquad \max_{u,v \in V \times V} \sup_{t \in [0,1]} \left| \frac{\partial^2}{\partial t^2} \theta^t_{uv} \right| < M.$$

  Under this assumption, as we get more and more data (i.e., we collect data in higher and higher temporal resolution within interval $[0, 1]$), parameters, and graph structures, corresponding to any two adjacent time points will differ less and less.

- *Piecewise constant with abrupt structural changes in parameters.* Next, we consider that there are a number of change points at which the distribution generating samples changes abruptly. Formally, we assume that, for each node u, there is a partition

$$\mathcal{B}_u = \{0 = B_{u,0} < B_{u,1} < \cdots < B_{u,k_u} = 1\}$$

  of the interval $[0, 1]$, such that each element of $\boldsymbol{\theta}^t_u$ is constant on each segment of the partition. At change points some of the elements of the vector $\boldsymbol{\theta}^t_u$ may become zero, while some others may become nonzero, which corresponds to a change in the graph structure. If the number of change points is small, that is, the graph structure changes infrequently, then there will be enough samples at a segment of the partition to estimate the nonzero pattern of the vector $\boldsymbol{\theta}^\tau$.

In the following two sections we propose two estimation methods, each suitable for one of the assumptions discussed above.

4.2 Smooth changes in parameters

Under the assumption that the elements of $\boldsymbol{\theta}^t$ are smooth functions of time, as described in the previous section, we use a kernel smoothing approach to estimate the nonzero pattern of $\boldsymbol{\theta}^\tau_u$ at the time point of interest $\tau \in [0, 1]$, for each node $u \in V$. These node-wise estimators are then combined using either equation (4.2) or equation (4.3) to obtain the estimator of the nonzero pattern of $\boldsymbol{\theta}^\tau$. The estimator $\widehat{\boldsymbol{\theta}}^\tau_u$ is defined as a minimizer of the following objective:

$$\widehat{\boldsymbol{\theta}}^\tau_u := \arg \min_{\boldsymbol{\theta}_u \in \mathbb{R}^{p-1}} \{l(\boldsymbol{\theta}_u; \mathcal{D}_n) + \lambda_1 ||\boldsymbol{\theta}_u||_1\}, \tag{4.7}$$

where

$$l(\boldsymbol{\theta}_u; \mathcal{D}_n) = -\sum_{t \in \mathcal{T}_n} w^\tau_t \gamma(\boldsymbol{\theta}_u; \mathbf{x}^t)$$

is a weighted log-likelihood, with weights defined as $w_t^\tau = \frac{K_h(t-\tau)}{\sum_{t' \in \mathcal{T}_n} K_h(t'-\tau)}$ and $K_h(\cdot) = K(\cdot/h)$ is a symmetric, nonnegative kernel function. We will refer to this approach of obtaining an estimator as `smooth`. The ℓ_1 norm of the parameter is used to regularize the solution and, as a result, the estimated parameter has a lot of zeros. The number of the nonzero elements of $\widehat{\boldsymbol{\theta}}_u^\tau$ is controlled by the user-specified regularization parameter $\lambda_1 \geq 0$. The bandwidth parameter h is also a user defined parameter that effectively controls the number of observations around τ used to obtain $\widehat{\boldsymbol{\theta}}_u^\tau$. In §4.5 we discuss how to choose the parameters λ_1 and h. Note how (4.7) extends the optimization problem in (2.4) to allow for non-i.i.d. data.

The optimization problem (4.7) is the well-known objective of the ℓ_1 penalized logistic regression and there are many ways of solving it, for example, the interior point method of [101], the projected subgradient descent method of [52], or the fast coordinate-wise descent method of [70]. From our limited experience, the specialized first order methods work faster than the interior point methods and we briefly describe the iterative coordinate-wise descent method:

1. Set initial values: $\widehat{\boldsymbol{\theta}}_u^{\tau,0} \leftarrow \mathbf{0}$.
2. For each $v \in V \setminus u$, set the current estimate $\widehat{\theta}_{uv}^{\tau,iter+1}$ as a solution to the following optimization procedure:

$$\min_{\theta \in \mathbb{R}} \left\{ \sum_{t \in \mathcal{T}_n} \gamma(\widehat{\theta}_{u,1}^{\tau,iter+1}, \ldots, \widehat{\theta}_{u,v-1}^{\tau,iter+1}, \theta, \widehat{\theta}_{u,v+1}^{\tau,iter}, \ldots, \widehat{\theta}_{u,p-1}^{\tau,iter}; \mathbf{x}^t) + \lambda_1 |\theta| \right\}. \tag{4.8}$$

3. Repeat step 2 until convergence

For an efficient way of solving (4.8) refer to [70]. In our experiments, we find that the neighborhood of each node can be estimated in a few seconds even when the number of covariates is up to a thousand. A nice property of our algorithm is that the overall estimation procedure decouples to a collection of separate neighborhood estimation problems, which can be trivially parallelized. If we treat the neighborhood estimation as an atomic operation, the overall algorithm scales linearly as a product of the number of covariates p and the number of time points n, that is, $\mathcal{O}(pn)$. For instance, the Drosophila data set in the application section contains 588 genes and 66 time points. The method `smooth` can estimate the neighborhood of one node, for all points in a regularization plane, in less than 1.5 hours.[1]

4.3 Structural changes in parameters

In this section we give the estimation procedure of the nonzero pattern of $\{\boldsymbol{\theta}^t\}_{t \in \mathcal{T}_n}$ under the assumption that the elements of $\boldsymbol{\theta}_u^t$ are a piecewise constant function, with pieces defined by the partition $\mathcal{B}_u$. Again, the estimation is performed node-wise and the estimators are combined using either equation (4.2) or equation (4.3). As opposed to the kernel smoothing estimator defined in equation (4.7), which gives the estimate at one time point τ, the procedure described below simultaneously estimates $\{\widehat{\boldsymbol{\theta}}_u^t\}_{t \in \mathcal{T}_n}$. The estimators $\{\widehat{\boldsymbol{\theta}}_u^t\}_{t \in \mathcal{T}_n}$ are defined as a minimizer

[1] We have used a server with dual core 2.6GHz processor and 2GB RAM.

of the following convex optimization objective:

$$\arg \min_{\boldsymbol{\theta}_u^t \in \mathbb{R}^{p-1}, t \in \mathcal{T}_n} \left\{ \sum_{t \in \mathcal{T}_n} \gamma(\boldsymbol{\theta}_u^t; \mathbf{x}^t) + \lambda_1 \sum_{t \in \mathcal{T}_n} ||\boldsymbol{\theta}_u^t||_1 + \lambda_{\mathrm{TV}} \sum_{v \in V \setminus u} \mathrm{TV}(\{\theta_{uv}^t\}_{t \in \mathcal{T}_n}) \right\}, \quad (4.9)$$

where $\mathrm{TV}(\{\theta_{uv}^t\}_{t \in \mathcal{T}_n}) := \sum_{i=2}^{n} |\theta_{uv}^{i/n} - \theta_{uv}^{(i-1)/n}|$ is the total variation penalty. We will refer to this approach of obtaining an estimator as TV. The penalty is structured as a combination of two terms. As mentioned before, the ℓ_1 norm of the parameters is used to regularize the solution toward estimators with lots of zeros and the regularization parameter λ_1 controls the number of nonzero elements. The second term penalizes the difference between parameters that are adjacent in time and, as a result, the estimated parameters have infrequent changes across time. This composite penalty, known as the "fused" Lasso penalty, was successfully applied in a slightly different setting of signal denoising (see, for example, [148]) where it creates an estimate of the signal that is piecewise constant.

The optimization problem given in equation (4.9) is convex and can be solved using an off-the-shelf interior point solver (for example, the CVX package [84]). However, for large scale problems (i.e., both p and n are large), the interior point method can be computationally expensive, and we do not know of any specialized algorithm that can be used to solve (4.9) efficiently. Therefore, we propose a block-coordinate descent procedure which is much more efficient than the existing off-the-shelf solvers for large scale problems. Observe that the loss function can be decomposed as

$$\mathcal{L}(\{\boldsymbol{\theta}_u^t\}_{t \in \mathcal{T}_n}) = f_1(\{\boldsymbol{\theta}_u^t\}_{t \in \mathcal{T}_n}) + \sum_{v \in V \setminus u} f_2(\{\theta_{uv}^t\}_{t \in \mathcal{T}_n})$$

for a smooth differentiable convex function

$$f_1(\{\boldsymbol{\theta}_u^t\}_{t \in \mathcal{T}_n}) = \sum_{t \in \mathcal{T}_n} \gamma(\boldsymbol{\theta}_u^t; \mathbf{x}^t)$$

and a convex function

$$f_2(\{\theta_{uv}^t\}_{t \in \mathcal{T}_n}) = \lambda_1 \sum_{t \in \mathcal{T}_n} |\theta_{uv}^t| + \lambda_{\mathrm{TV}} \, \mathrm{TV}(\{\theta_{uv}^t\}_{t \in \mathcal{T}_n}).$$

Tseng [169] established that the block-coordinate descent converges for loss functions with such structure. Based on this observation, we propose the following algorithm:

1. Set initial values: $\widehat{\boldsymbol{\theta}}_u^{t,0} \leftarrow \mathbf{0}, \forall t \in \mathcal{T}_n$.
2. For each $v \in V \setminus u$, set the current estimates $\{\widehat{\theta}_{uv}^{t,iter+1}\}_{t \in \mathcal{T}_n}$ as a solution to the following optimization procedure:

$$\min_{\{\theta^t \in \mathbb{R}\}_{t \in \mathcal{T}_n}} \left\{ \sum_{t \in \mathcal{T}_n} \gamma(\widehat{\theta}_{u,1}^{t,iter+1}, \ldots, \widehat{\theta}_{u,v-1}^{t,iter+1}, \theta^t, \widehat{\theta}_{u,v+1}^{t,iter}, \ldots, \widehat{\theta}_{u,p-1}^{t,iter}; \mathbf{x}^t) + \lambda_1 \sum_{t \in \mathcal{T}^n} |\theta^t| + \lambda_{\mathrm{TV}} \, \mathrm{TV}(\{\theta^t\}_{t \in \mathcal{T}_n}) \right\}. \quad (4.10)$$

3. Repeat step 2 until convergence.

Using the proposed block-coordinate descent algorithm, we solve a sequence of optimization problems each with only n variables given in equation (4.10), instead of solving one big optimization problem with $n(n-1)$ variables given in equation (4.9). In our experiments, we find that the optimization in equation (4.9) can be estimated in an hour when the number of covariates is up to a few hundred and when the number of time points is also in the hundreds. Here, the bottleneck is the number of time points. Observe that the dimensionality of the problem in equation (4.10) grows linearly with the number of time points. Again, the overall estimation procedure decouples to a collection of smaller problems which can be trivially parallelized. If we treat the optimization in equation (4.9) as an atomic operation, the overall algorithm scales linearly as a function of the number of covariates p, that is, $\mathcal{O}(p)$. For instance, the Senate data set in the application section contains 100 Senators and 542 time points. It took about a day to solve the optimization problem in equation (4.9) for all points in the regularization plane.

4.4 Multiple observations

In the discussion so far, it is assumed that at any time point in $\mathcal{T}_n$ only one observation is available. There are situations with multiple observations at each time point, for example, in a controlled repeated microarray experiment two samples obtained at a certain time point could be regarded as independent and identically distributed, and we discuss below how to incorporate such observations into our estimation procedures. Later, in §4.6 we empirically show how the estimation procedures benefit from additional observations at each time point.

For the estimation procedure given in equation (4.7), there are no modifications needed to accommodate multiple observations at a time point. Each additional sample will be assigned the same weight through the kernel function $K_h(\cdot)$. On the other hand, we need a small change in equation (4.9) to allow for multiple observations. The estimators $\{\widehat{\boldsymbol{\theta}}_u^t\}_{t\in\mathcal{T}_n}$ are defined as follows:

$$\{\widehat{\boldsymbol{\theta}}_u^t\}_{t\in\mathcal{T}_n} = \arg\min_{\boldsymbol{\theta}_u^t\in\mathbb{R}^{p-1}, t\in\mathcal{T}_n} \left\{ \sum_{t\in\mathcal{T}_n}\sum_{\mathbf{x}\in\mathcal{D}_n^t} \gamma(\boldsymbol{\theta}_u^t;\mathbf{x}) + \lambda_1 \sum_{t\in\mathcal{T}_n} ||\boldsymbol{\theta}_u^t||_1 + \lambda_{\text{TV}} \sum_{v\in V\setminus u} \text{TV}(\{\theta_{uv}^t\}_{t\in\mathcal{T}_n}) \right\},$$

where the set $\mathcal{D}_n^t$ denotes elements from the sample $\mathcal{D}_n$ observed at a time point t.

4.5 Choosing tuning parameters

Estimation procedures discussed in §4.2 and §4.3, `smooth` and `TV` respectively, require a choice of tuning parameters. These tuning parameters control sparsity of estimated graphs and the way the graph structure changes over time. The tuning parameter λ_1, for both `smooth` and `TV`, controls the sparsity of the graph structure. Large values of the parameter λ_1 result in estimates with lots of zeros, corresponding to sparse graphs, while small values result in dense models. Dense models will have a higher pseudo-likelihood score, but will also have more degrees of freedom. A good choice of the tuning parameters is essential in obtaining a good estimator that does not overfit the data, and balances between the pseudo-likelihood and the degrees of

freedom. The bandwidth parameter h and the penalty parameter λ_{TV} control how similar are estimated networks that are close in time. Intuitively, the bandwidth parameter controls the size of a window around time point τ from which observations are used to estimate the graph G^τ. Small values of the bandwidth result in estimates that change often with time, while large values produce estimates that are almost time invariant. The penalty parameter λ_{TV} biases the estimates $\{\widehat{\boldsymbol{\theta}}_u^t\}_{t\in\mathcal{T}_n}$ that are close in time to have similar values; large values of the penalty result in graphs whose structure changes slowly, while small values allow for more changes in estimates.

We discuss how to choose the penalty parameters λ_1 and λ_{TV} for the method `TV`. Observe that $\gamma(\boldsymbol{\theta}_u^t; \mathbf{x}^t)$ represents a logistic regression loss function when regressing a node u onto the other nodes $V \setminus u$. Hence, problems defined in equation (4.7) and equation (4.9) can be regarded as *supervised* classification problems, for which a number of techniques can be used to select the tuning parameters, for example, cross-validation or held-out data sets can be used when enough data is available, otherwise, the BIC score can be employed. In this paper we focus on the BIC score defined for $\{\boldsymbol{\theta}_u^t\}_{t\in\mathcal{T}_n}$ as

$$\mathrm{BIC}(\{\boldsymbol{\theta}_u^t\}_{t\in\mathcal{T}_n}) := \sum_{t\in\mathcal{T}_n} \gamma(\boldsymbol{\theta}_u^t; \mathbf{x}^t) - \frac{\log n}{2}\,\mathrm{Dim}(\{\boldsymbol{\theta}_u^t\}_{t\in\mathcal{T}_n}),$$

where $\mathrm{Dim}(\cdot)$ denotes the degrees of freedom of the estimated model. Similar to [177], we adopt the following approximation to the degrees of freedom:

$$\mathrm{Dim}(\{\boldsymbol{\theta}_u^t\}_{t\in\mathcal{T}_n}) = \sum_{t\in\mathcal{T}_n}\sum_{v\in V\setminus u} \mathbb{I}[\mathrm{sign}(\theta_{uv}^t) \neq \mathrm{sign}(\theta_{uv}^{t-1})] \times \mathbb{I}[\mathrm{sign}(\theta_{uv}^t) \neq 0], \tag{4.11}$$

which counts the number of blocks on which the parameters are constant and not equal to zero. In practice, we average the BIC scores from all nodes and choose models according to the average.

Next, we address the way to choose the bandwidth h and the penalty parameter λ_1 for the method `smooth`. As mentioned earlier, the tuning of bandwidth parameter h should trade off the smoothness of the network changes and the coverage of samples used to estimate the network. Using a wider bandwidth parameter provides more samples to estimate the network, but this risks missing sharper changes in the network; using a narrower bandwidth parameter makes the estimate more sensitive to sharper changes, but this also makes the estimate subject to larger variance due to the reduced effective sample size. In this paper we adopt a heuristic for tuning the inital scale of the bandwidth parameter: we set it to be the median of the distance between pairs of time points. That is, we first form a matrix (d_{ij}) with its entries $d_{ij} := (t_i - t_j)^2$ ($t_i, t_j \in \mathcal{T}_n$). Then the scale of the bandwidth parameter is set to the median of the entries in (d_{ij}). In our later simulation experiments, we find that this heuristic provides a good initial guess for h, and it is quite close to the value obtained via exhaustive grid search. For the method `smooth`, the BIC score for $\{\boldsymbol{\theta}_u^t\}_{t\in\mathcal{T}_n}$ is defined as

$$\mathrm{BIC}(\{\boldsymbol{\theta}_u^t\}_{t\in\mathcal{T}_n}) := \sum_{\tau\in\mathcal{T}_n}\sum_{t\in\mathcal{T}_n} w_t^\tau \gamma(\boldsymbol{\theta}_u^\tau; \mathbf{x}^t) - \frac{\log n}{2}\,\mathrm{Dim}(\{\boldsymbol{\theta}_u^t\}_{t\in\mathcal{T}_n}), \tag{4.12}$$

where $\mathrm{Dim}(\cdot)$ is defined in equation (4.11).

4.6 Simulation studies

We have conducted a small empirical study of the performance of methods `smooth` and `TV`. Our idea was to choose parameter vectors $\{\boldsymbol{\theta}^t\}_{t\in\mathcal{T}_n}$, generate data according to the model in equation (4.1) using Gibbs sampling, and try to recover the nonzero pattern of $\boldsymbol{\theta}^t$ for each $t \in \mathcal{T}_n$. Parameters $\{\boldsymbol{\theta}^t\}_{t\in\mathcal{T}_n}$ are considered to be evaluations of the function $\boldsymbol{\theta}^t$ at $\mathcal{T}_n$ and we study two scenarios, as discussed in §4.1: $\boldsymbol{\theta}^t$ is a smooth function, $\boldsymbol{\theta}^t$ is a piecewise constant function. In addition to the methods `smooth` and `TV`, we will use the method of [151] to estimate a time-invariant graph structure, which we refer to as `static`. All of the three methods estimate the graph based on node-wise neighborhood estimation, which, as discussed in §4.1, may produce asymmetric estimates. Solutions combined with the min operation in equation (4.2) are denoted as `****.MIN`, while those combined with the max operation in equation (4.3) are denoted as `****.MAX`.

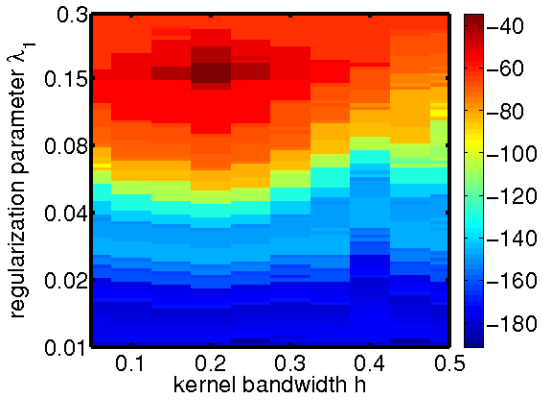
we took the number of nodes $p = 20$, the maximum node degree $s = 4$, the number of edges $e = 25$, and the sample size $n = 500$. The parameter vectors $\{\boldsymbol{\theta}^t\}_{t\in\mathcal{T}_n}$ and observation sequences are generated as follows:

1. Generate a random graph $\widetilde{G}^0$ with 20 nodes and 15 edges: edges are added, one at a time, between random pairs of nodes that have the node degree less than 4. Next, randomly add 10 edges and remove 10 edges from $\widetilde{G}^0$, taking care that the maximum node degree is still 4, to obtain $\widetilde{G}^1$. Repeat the process of adding and removing edges from $\widetilde{G}^1$ to obtain $\widetilde{G}^2, \ldots, \widetilde{G}^5$. We refer to these 6 graphs as the anchor graphs. We will randomly generate the prototype parameter vectors $\widetilde{\boldsymbol{\theta}}^0, \ldots, \widetilde{\boldsymbol{\theta}}^5$, corresponding to the anchor graphs, and then interpolate between them to obtain the parameters $\{\boldsymbol{\theta}^t\}_{t\in\mathcal{T}_n}$.
2. Generate a prototype parameter vector $\widetilde{\boldsymbol{\theta}}^i$ for each anchor graph $\widetilde{G}^i$, $i \in \{0, \ldots, 5\}$, by sampling nonzero elements of the vector independently from $\mathsf{Unif}([0.5, 1])$. Then generate $\{\boldsymbol{\theta}^t\}_{t\in\mathcal{T}_n}$ according to one of the following two cases:
   - Smooth function: The parameters $\{\boldsymbol{\theta}^t\}_{t\in((i-1)/5, i/5]\cap\mathcal{T}_n}$ are obtained by linearly interpolating 100 points between $\widetilde{\boldsymbol{\theta}}^{i-1}$ and $\widetilde{\boldsymbol{\theta}}^i$, $i \in \{1, \ldots, 5\}$.
   - Piecewise constant function: The parameters $\{\boldsymbol{\theta}^t\}_{t\in((i-1)/5, i/5]\cap\mathcal{T}_n}$ are set to be equal to $(\widetilde{\boldsymbol{\theta}}^{i-1} + \widetilde{\boldsymbol{\theta}}^i)/2$, $i \in \{1, \ldots, 5\}$.

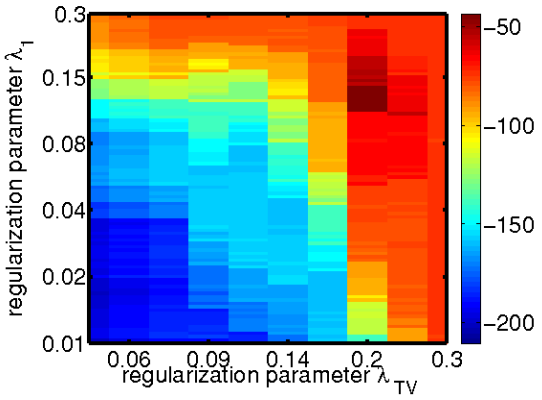
   Observe that after interpolating between the prototype parameters, a graph corresponding to $\boldsymbol{\theta}^t$ has 25 edges and the maximum node degree is 4.
3. Generate 10 independent samples at each $t \in \mathcal{T}_n$ according to $\mathbb{P}_{\boldsymbol{\theta}^t}$, given in equation (4.1), using Gibbs sampling.

We estimate $\widehat{G}^t$ for each $t \in \mathcal{T}_n$ with our `smooth` and `TV` methods, using $k \in \{1, \ldots, 10\}$ samples at each time point. The results are expressed in terms of the precision (Pre) and the recall (Rec) and $F1$ score, which is the harmonic mean of precision and recall, that is, $F1 := 2 * \mathsf{Pre} * \mathsf{Rec}/(\mathsf{Pre} + \mathsf{Rec})$. Let $\widehat{E}^t$ denote the estimated edge set of $\widehat{G}^t$, then the precision is calculated as $\mathsf{Pre} := 1/n \sum_{t\in\mathcal{T}_n} |\widehat{E}^t \cap E^t|/|\widehat{E}^t|$ and the recall as $\mathsf{Rec} := 1/n \sum_{t\in\mathcal{T}_n} |\widehat{E}^t \cap E^t|/|E^t|$. Furthermore, we report results averaged over 20 independent runs.

We discuss the estimation results when the underlying parameter vector changes smoothly. See Figure 4.2 for results. It can be seen that as the number of the i.i.d. observations at each time



(a) Average BIC score



(b) Average BIC score

Figure 4.1: Plot of the $\mathrm{BIC}_{\mathrm{avg}}$ score over the regularization plane. The parameter vector θ^t is a smooth function of time and at each time point there is one observation. (a) The graph structure recovered using the method smooth. (b) Recovered using the method TV.

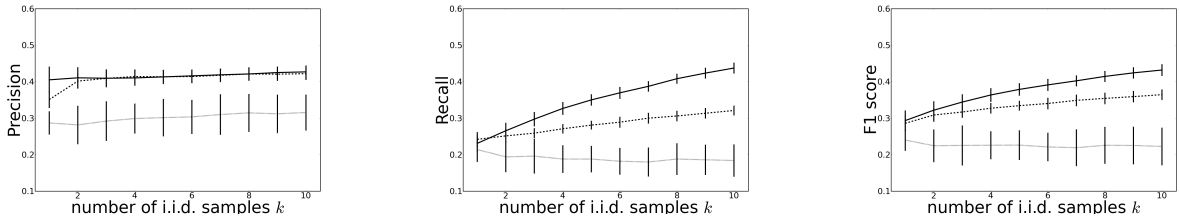
point increases, the performance of both methods `smooth` and `TV` increases. On the other hand, the performance of the method `static` does not benefit from additional i.i.d. observations. This observation should not be surprising as the time-varying network models better fit the data generating process. When the underlying parameter vector $\boldsymbol{\theta}^t$ is a smooth function of time, we expect that the method `smooth` would have a faster convergence and better performance, which can be seen in Figure 4.2. There are some differences between the estimates obtained through `MIN` and `MAX` symmetrization. In our limited numerical experience, we have seen that `MAX` symmetrization outperforms `MIN` symmetrization. `MIN` symmetrization is more conservative in including edges to the graph and seems to be more susceptible to noise.

Next, we discuss the estimation results when then the underlying parameter vector is a piecewise constant function. See Figure 4.3 for results. Again, both performance of the method `smooth` and of the method `TV` improve as there are more independent samples at different time points, as opposed to the method `static`. It is worth noting that the empirical performance of `smooth` and `TV` is very similar in the setting when $\boldsymbol{\theta}^t$ is a piecewise constant function of time, with the method `TV` performing marginally better. This may be a consequence of the way we present results, averaged over all time points in $\mathcal{T}_n$. A closer inspection of the estimated graphs shows that the method `smooth` poorly estimates graph structure close to the time point at which the parameter vector changes abruptly (results not shown).

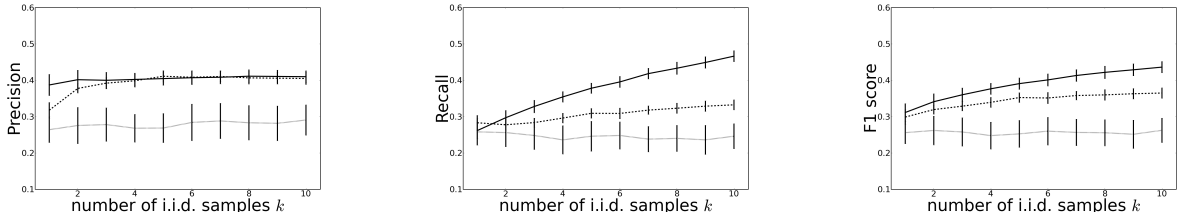
The tuning parameters h and λ_1 for `smooth`, and λ_1 and λ_{TV} for `TV`, are chosen by maximizing the average BIC score,

$$\mathrm{BIC}_{\mathrm{avg}} := 1/p \sum_{u \in V} \mathrm{BIC}(\{\boldsymbol{\theta}_u^t\}_{t \in \mathcal{T}_n}),$$

over a grid of parameters. The bandwidth parameter h is searched over $\{0.05, 0.1, \ldots, 0.45, 0.5\}$ and the penalty parameter λ_{TV} over 10 points, equidistant on the log-scale, from the interval $[0.05, 0.3]$. The penalty parameter is searched over 100 points, equidistant on the log-scale, from the interval $[0.01, 0.3]$ for both `smooth` and `TV`. The same range is used to select the penalty

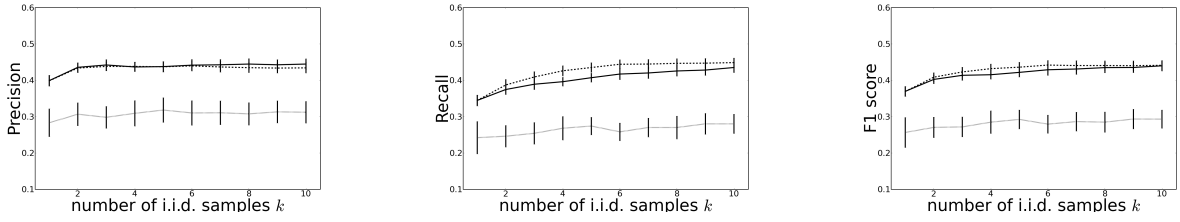


(a) Precision (b) Recall (c) F1 score

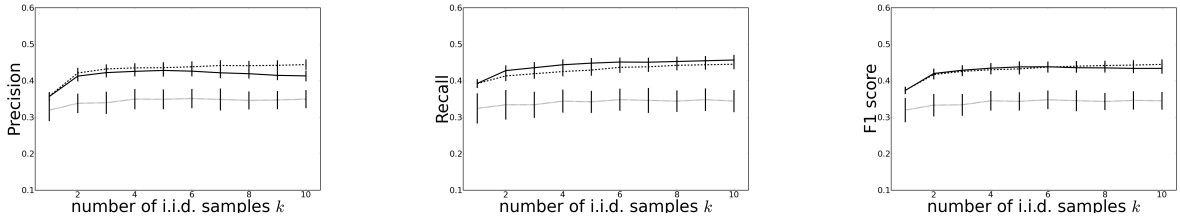


(d) Precision (e) Recall (f) F1 score

Figure 4.2: Results of estimation when the underlying parameter $\{\boldsymbol{\theta}^t\}_{t\in\mathcal{T}_n}$ changes smoothly with time. The upper row consists of results when the graph is estimated combining the neighborhoods using the $\min$ operation, while the lower row consists of results when the $\max$ operation is used to combine neighborhoods. Precision, recall, and F1 score are plotted as the number of i.i.d. samples k at each time point increases from 1 to 10. The solid, dashed, and dotted lines denote results for smooth, TV, and static, respectively.



(a) Precision (b) Recall (c) F1 score



(d) Precision (e) Recall (f) F1 score

Figure 4.3: Results of estimation when the underlying parameter $\{\boldsymbol{\theta}^t\}_{t\in\mathcal{T}_n}$ is a piecewise constant function of time. The upper row consists of results when the graph is estimated combining the neighborhoods using the $\min$ operation, while the lower row consists of results when the $\max$ operation is used to combine neighborhoods. Precision, recall, and F1 score are plotted as the number of i.i.d. samples k at each time point increases from 1 to 10. The solid, dashed, and dotted lines denote results for smooth, TV, and static, respectively.

parameter λ for the method `static` that estimates a time-invariant network. In our experiments, we use the Epanechnikov kernel $K(z) = 3/4 * (1 - z^2)\, \mathbb{1}\{|z| \leq 1\}$ and we remind our reader that $K_h(\cdot) = K(\cdot/h)$. For illustrative purposes, in Figure 4.1 we plot the $\mathrm{BIC}_{\mathrm{avg}}$ score over the grid of tuning parameters.

We have decided to perform simulation studies on Erdös–Rényi graphs, while real-world graphs are likely to have different properties, such as a scale-free network with a long tail in its degree distribution. From a theoretical perspective, our method can still recover the true structure of these networks regardless of the degree distribution, although for a more complicated model, we may need more samples in order to achieve this. [146] proposed a joint sparse regression model, which performs better than the neighborhood selection method when estimating networks with hubs (nodes with very high degree) and scale-free networks. For such networks, we can extend their model to our time-varying setting, and potentially make more efficient use of the samples, however, we do not pursue this direction here.

4.7 Applications to real data

In this section we present the analysis of two real data sets using the algorithms presented in §4.1. First, we present the analysis of the senate data consisting of Senators' votes on bills during the 109th Congress. The second data set consists of expression levels of more than 4000 genes from the life cycle of *Drosophila melanogaster*.

4.7.1 Senate voting records data

The US senate data consists of voting records from 109th congress (2005–2006). There are 100 senators whose votes were recorded on the 542 bills. Each senator corresponds to a variable, while the votes are samples recorded as -1 for no and 1 for yes. This data set was analyzed in [19], where a static network was estimated. Here, we analyze this data set in a time-varying framework in order to discover how the relationship between senators changes over time.

This data set has many missing values, corresponding to votes that were not cast. We follow the approach of [19] and fill those missing values with (-1). Bills were mapped onto the $[0, 1]$ interval, with 0 representing Jan 1st, 2005 and 1 representing Dec 31st, 2006. We use the Epanechnikov kernel for the method `smooth`. The tuning parameters are chosen optimizing the average BIC score over the same range as used for the simulations in §4.6. For the method `smooth`, the bandwidth parameter was selected as $h = 0.174$ and the penalty parameter $\lambda_1 = 0.195$, while penalty parameters $\lambda_1 = 0.24$ and $\lambda_{\mathrm{TV}} = 0.28$ were selected for the method `TV`. In the figures in this section, we use pink square nodes to represent republican Senators and blue circle nodes to represent democrat Senators.

A first question is whether the learned network reflects the political division between Republicans and Democrats. Indeed, at any time point t, the estimated network contains few clusters of nodes. These clusters consist of either Republicans or Democrats connected to each others; see Figure 4.4. Furthermore, there are very few links connecting different clusters. We observe that most Senators vote similarly to other members of their party. Links connecting different clusters usually go through senators that are members of one party, but have views more similar to the

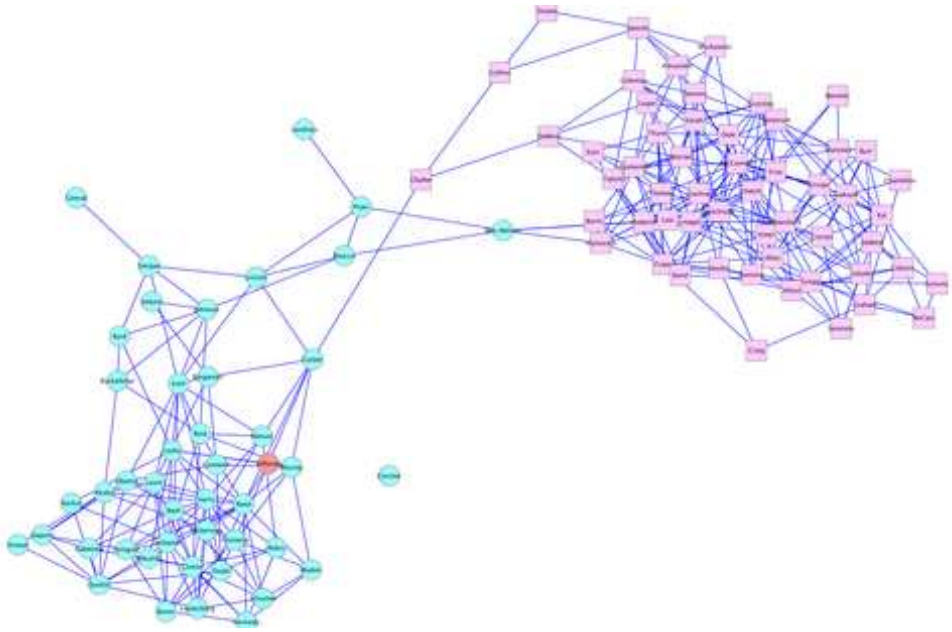


Figure 4.4: 109th Congress, Connections between Senators in April 2005. Democrats are represented with blue circles, Republicans with pink squares, and the red circle represents independent Senator Jeffords.

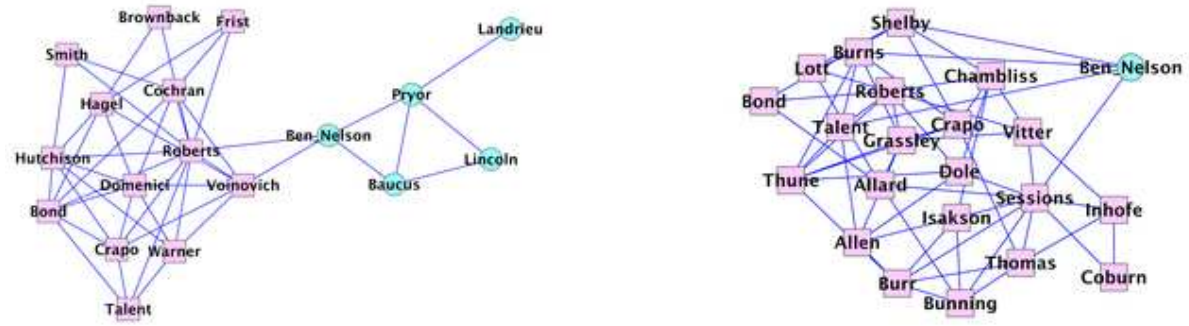
other party, for example, Senator Ben Nelson or Senator Chafee. Note that we do not necessarily need to estimate a time evolving network to discover this pattern of political division, as they can also be observed from a time-invariant network (see, for example, [19]).

Therefore, what is more interesting is whether there is any time evolving pattern. To show this, we examine neighborhoods of Senators Jon Corzine and Bob Menendez. Senator Corzine stepped down from the Senate at the end of the 1st Session in the 109th Congress to become the Governor of New Jersey. His place in the Senate was filled by Senator Menendez. This dynamic change of interactions can be well captured by the time-varying network (Figure 4.5). Interestingly, we can see that Senator Lautenberg who used to interact with Senator Corzine switches to Senator Menendez in response to this event.

Another interesting question is whether we can discover senators with swaying political stance based on time evolving networks. We discover that Senator Ben Nelson and Lincoln Chafee fall into this category. Although Senator Ben Nelson is a Democrat from Nebraska, he is considered to be one of the most conservative Democrats in the Senate. Figure 4.6 presents

(a) March 2005 (b) August 2005 (c) March 2006 (d) August 2006

Figure 4.5: Direct neighbors of the node that represent Senator Corzine and Senator Menendez at four different time points. Senator Corzine stepped down at the end of the 1st Session and his place was taken by Senator Menendez, which is reflected in the graph structure.

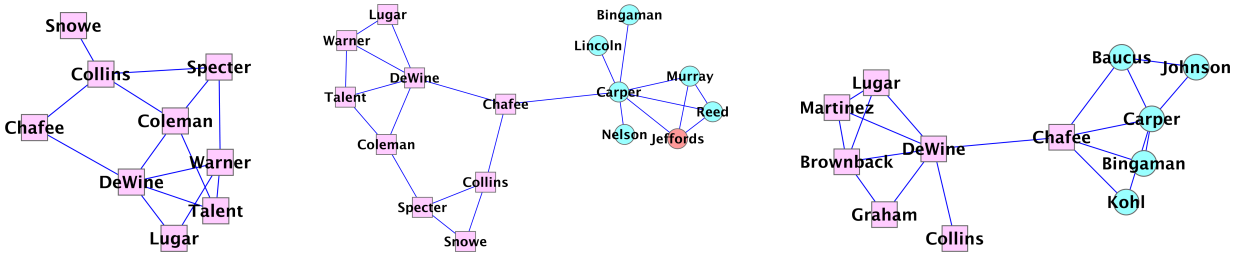


(a) May 2005 (b) August 2006

Figure 4.6: Neighbors of Senator Ben Nelson (distance two or lower) at the beginning of the 109th Congress and at the end of the 109th Congress. Democrats are represented with blue circles, Republicans with pink squares. The estimated neighborhood in August 2006 consists only of Republicans, which may be due to the type of bills passed around that time on which Senator Ben Nelson had similar views as other Republicans.

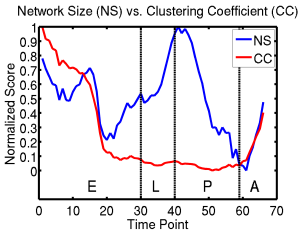
neighbors at distance two or less of Senator Ben Nelson at two time points, one during the 1st Session and one during the 2nd Session. As a conservative Democrat, he is connected to both Democrats and Republicans since he shares views with both parties. This observation is supported by Figure 4.6(a) which presents his neighbors during the 1st Session. It is also interesting to note that during the second session, his views drifted more toward the Republicans [Figure 4.6(b)]. For instance, he voted against abortion and withdrawal of most combat troops from Iraq, which are both Republican views.

In contrast, although Senator Lincoln Chafee is a Republican, his political view grew increasingly Democratic. Figure 4.7 presents neighbors of Senator Chafee at three time points during the 109th Congress. We observe that his neighborhood includes an increasing amount of Democrats as time progresses during the 109th Congress. Actually, Senator Chafee later left the Republican Party and became an independent in 2007. Also, his view on abortion, gay rights, and environmental policies are strongly aligned with those of Democrats, which is also consistently reflected in the estimated network. We emphasize that these patterns about Senator Nelson and Chafee could not be observed in a static network.

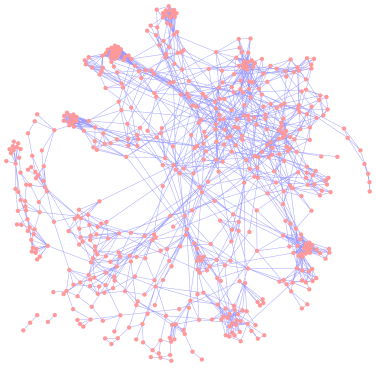


(a) (b) (c)

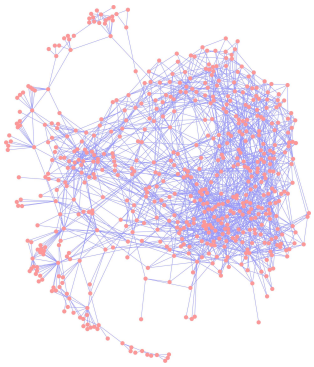
Figure 4.7: Neighbors of Senator Chafee (distance two or lower) at different time points during the 109th Congress. Democrats are represented with blue circles, Republicans with pink squares, and the red circle represents independent Senator Jeffords.



(a) Network statistics



(b) Mid-embryonic stage



(c) Mid-pupal stage

Figure 4.8: Characteristic of the dynamic networks estimated for the genes related to the developmental process. (a) Plot of two network statistics as functions of the development time line. Network size ranges between 1712 and 2061 over time, while local clustering coefficient ranges between 0.23 and 0.53 over time; To focus on relative activity over time, both statistics are normalized to the range between 0 and 1. (b) and (c) are the visualization of two examples of networks from different time points. We can see that network size can evolve in a very different way from the local clustering coefficient.

4.7.2 Gene regulatory networks of Drosophila melanogaster

In this section we used the kernel reweighting approach to reverse engineer the gene regulatory networks of *Drosophila melanogaster* from a time series of gene expression data measured during its full life cycle. Over the developmental course of *Drosophila melanogaster*, there exist multiple underlying "themes" that determine the functionalities of each gene and their relationships to each other, and such themes are dynamical and stochastic. As a result, the gene regulatory networks at each time point are context-dependent and can undergo systematic rewiring, rather than being invariant over time. In a seminal study by [127], it was shown that the "active regulatory paths" in the gene regulatory networks of *Saccharomyces cerevisiae* exhibit topological changes and hub transience during a temporal cellular process, or in response to diverse stimuli. We expect similar properties can also be observed for the gene regulatory networks of *Drosophila melanogaster*.

We used microarray gene expression measurements from [10] as our input data. In such an experiment, the expression levels of 4028 genes are simultaneously measured at various developmental stages. Particularly, 66 time points are chosen during the full developmental cycle of *Drosophila melanogaster*, spanning across four different stages, *that is*, embryonic (1–30 time point), larval (31–40 time point), pupal (41–58 time points), and adult stages (59–66 time points). In this study we focused on 588 genes that are known to be related to the developmental process based on their gene ontologies.

Usually, the samples prepared for microarray experiments are a mixture of tissues with possibly different expression levels. This means that microarray experiments only provide rough estimates of the average expression levels of the mixture. Other sources of noise can also be introduced into the microarray measurements during, for instance, the stage of hybridization and digitization. Therefore, microarray measurements are far from the exact values of the expression levels, and it will be more robust if we only consider the binary state of the gene expression:

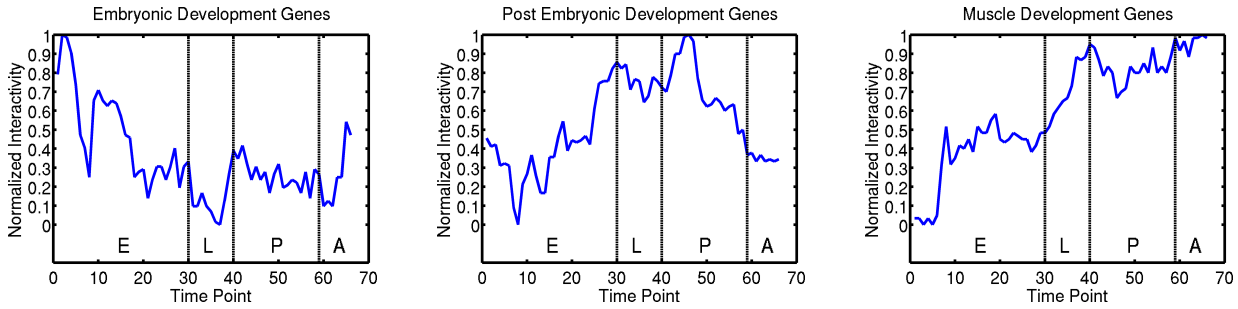


Figure 4.9: Interactivity of 3 groups of genes related to (a) embryonic development (ranging between 169 and 241), (b) post-embryonic development (ranging between 120 and 210), and (c) muscle development (ranging between 29 and 89). To focus on the relative activity over time, we normalize the score to $[0, 1]$. The higher the interactivity, the more active the group of genes. The interactivities of these three groups are very consistent with their functional annotations.

either being up-regulated or down-regulated. For this reason, we binarize the gene expression levels into $\{-1, 1\}$ (-1 for down-regulated and 1 for up-regulated). We learned a sequence of binary MRFs from these time series.

First, we study the global pattern of the time evolving regulatory networks. In Figure 4.8(a) we plotted two different statistics of the reversed engineered gene regulatory networks as a function of the developmental time point (1–66). The first statistic is the network size as measured by the number of edges; and the second is the average local clustering coefficient as defined by [188]. For comparison, we normalized both statistics to the range between $[0, 1]$. It can be seen that the network size and its local clustering coefficient follow very different trajectories during the developmental cycle. The network size exhibits a wave structure featuring two peaks at mid-embryonic stage and the beginning of the pupal stage. A similar pattern of gene activity has also been observed by [10]. In contrast, the clustering coefficients of the dynamic networks drop sharply after the mid-embryonic stage, and they stay low until the start of the adult stage. One explanation is that at the beginning of the development process, genes have a more fixed and localized function, and they mainly interact with other genes with similar functions. However, after mid-embryonic stage, genes become more versatile and involved in more diverse roles to serve the need of rapid development; as the organism turns into an adult, its growth slows down and each gene is restored to its more specialized role. To illustrate how the network properties change over time, we visualized two networks from mid-embryonic stage (time point 15) and mid-pupal stage (time point 45) using the spring layout algorithm in Figure 4.8(b) and (c) respectively. Although the size of the two networks are comparable, tight local clusters of interacting genes are more visible during mid-embryonic stage than mid-pupal stage, which is consistent with the evolution local clustering coefficient in Figure 4.8(a).

To judge whether the learned networks make sense biologically, we zoom into three groups of genes functionally related to different stages of the development process. In particular, the first group (30 genes) is related to embryonic development based on their functional ontologies; the second group (27 genes) is related to post-embryonic development; and the third group (25 genes) is related to muscle development. For each group, we use the number of within group connections plus all its outgoing connections to describe the activitiy of each group of genes (for short, we call it interactivity). In Figure 4.9 we plotted the time courses of interactivity for the

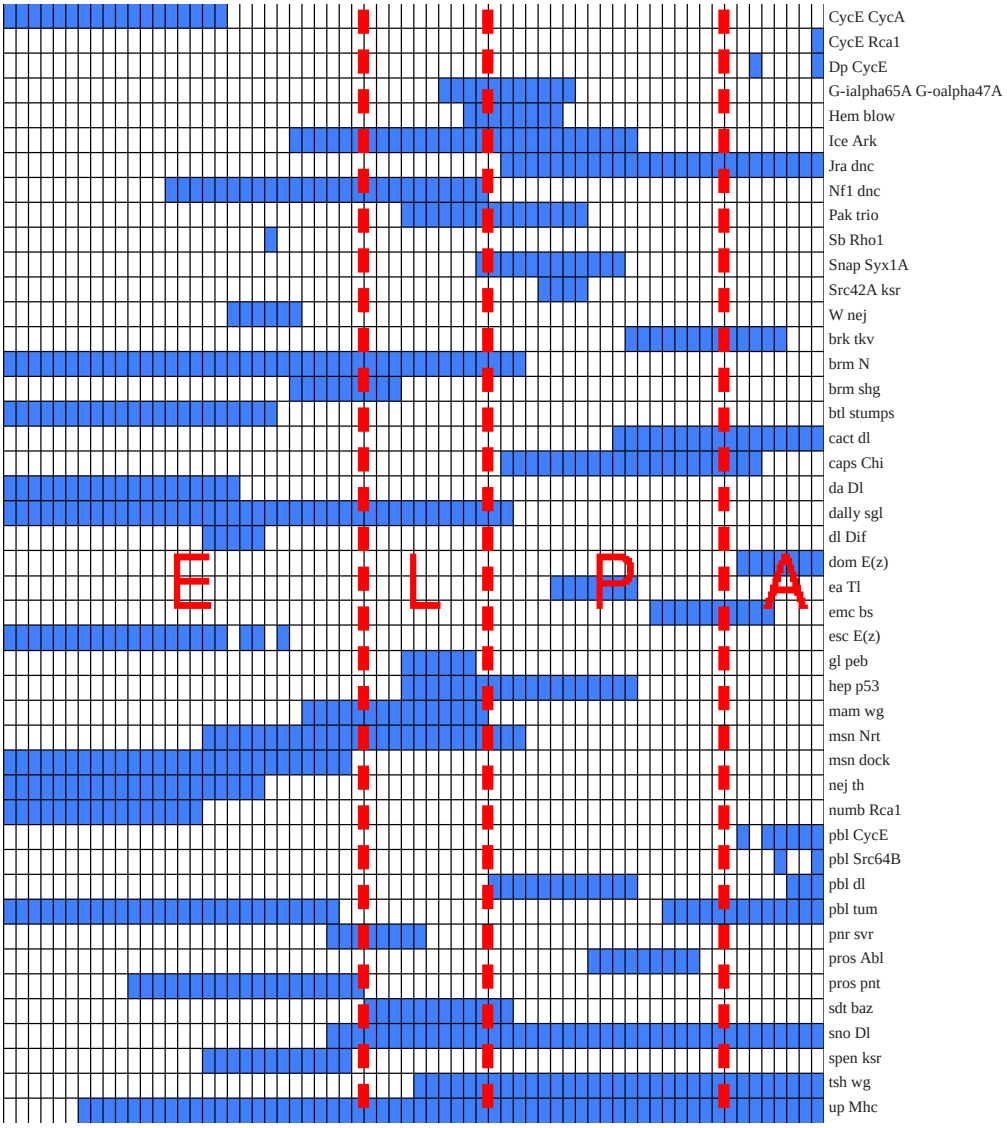
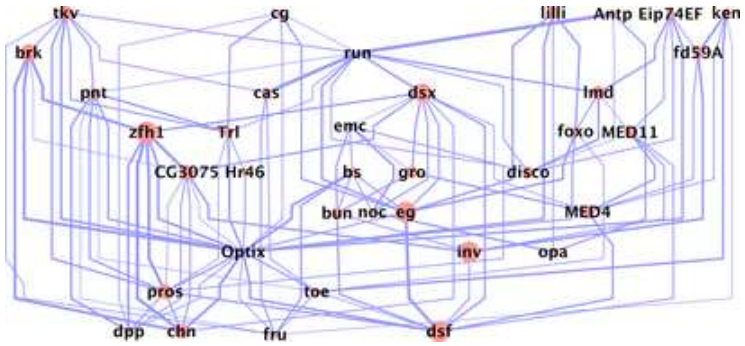
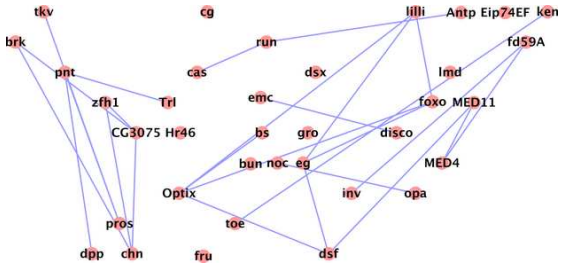


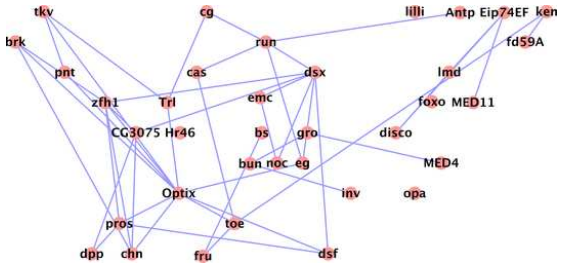
Figure 4.10: Timeline of 45 known gene interactions. Each cell in the plot corresponds to one gene pair of gene interaction at one specific time point. The cells in each row are ordered according to their time point, ranging from embryonic stage (E) to larval stage (L), to pupal stage (P), and to adult stage (A). Cells colored blue indicate the corresponding interaction listed in the right column is present in the estimated network; blank color indicates the interaction is absent.



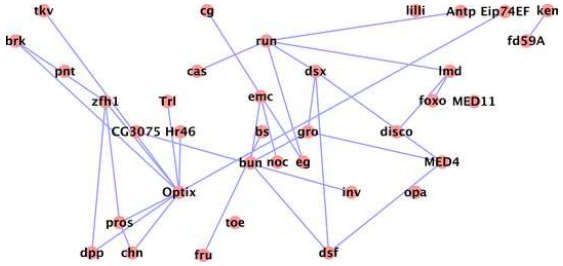
(a) Summary network



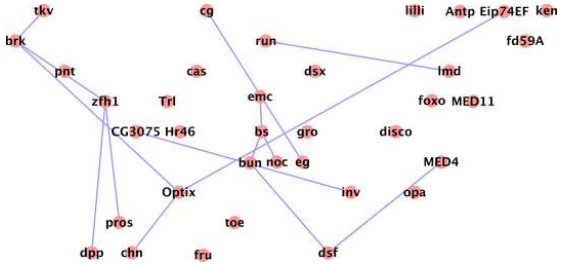
(b) Time point 15 (mid-embryonic stage)



(c) Time point 35 (mid-larval stage)



(d) Time point 49 (mid-pupal stage)

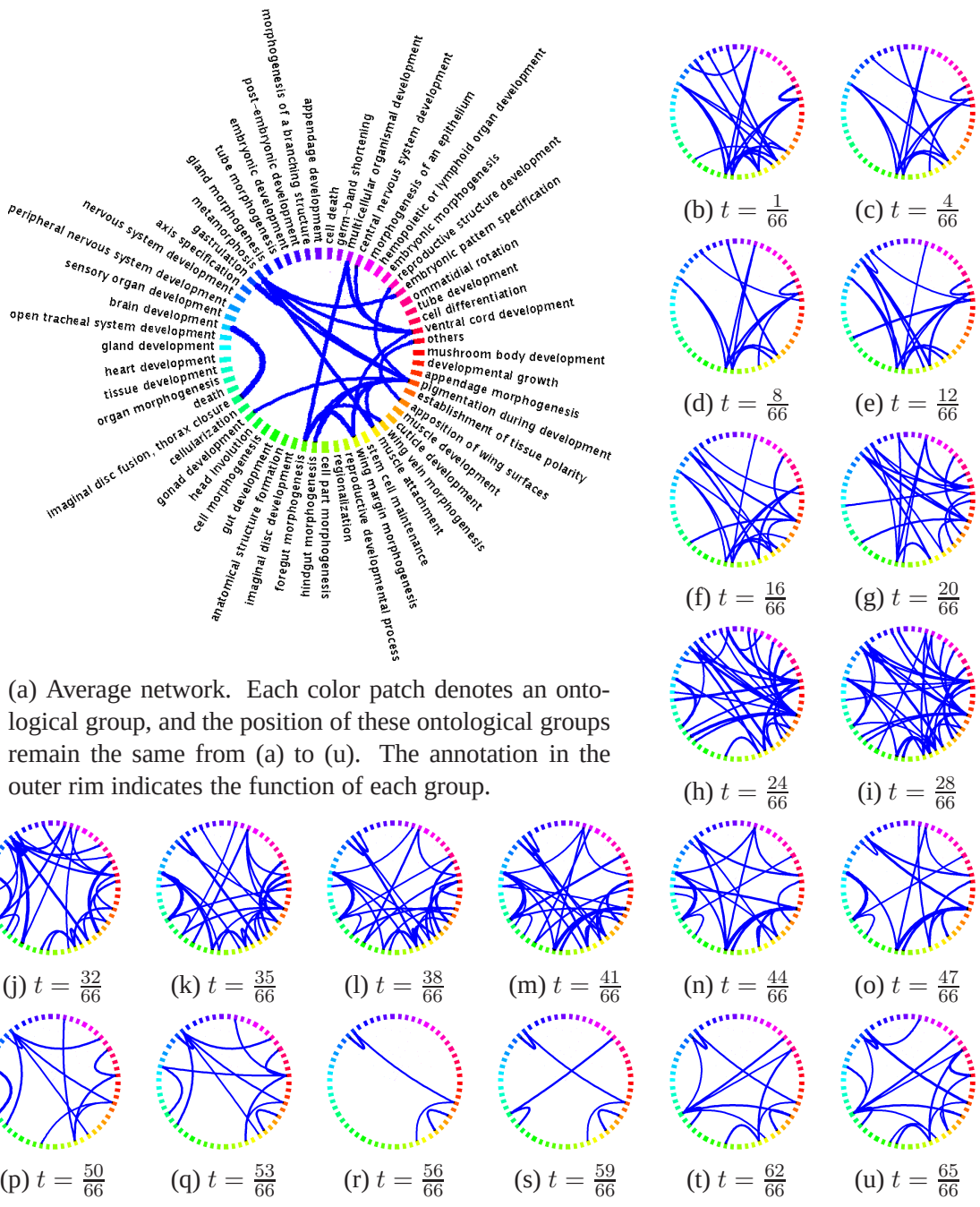


(e) Time point 62 (mid-adult stage)

Figure 4.11: The largest transcriptional factors (TF) cascade involving 36 transcriptional factors. (a) The summary network is obtained by summing the networks from all time points. Each node in the network represents a transcriptional factor, and each edge represents an interaction between them. On different stages of the development, the networks are different, (b), (c), (d), (e) shows representative networks for the embryonic, larval, pupal, and adult stage of the development respectively.

three groups respectively. For comparison, we normalize all scores to the range of $[0, 1]$. We see that the time courses have a nice correspondence with their supposed roles. For instance, embryonic development genes have the highest interactivity during embryonic stage, and post-embryonic genes increase their interactivity during the larval and pupal stages. The muscle development genes are less specific to certain developmental stages, since they are needed across the developmental cycle. However, we see its increased activity when the organism approaches its adult stage where muscle development becomes increasingly important.

The estimated networks also recover many known interactions between genes. In recovering these known interactions, the dynamic networks also provide additional information as to when



(a) Average network. Each color patch denotes an ontological group, and the position of these ontological groups remain the same from (a) to (u). The annotation in the outer rim indicates the function of each group.

(b) $t = \frac{1}{66}$ (c) $t = \frac{4}{66}$ (d) $t = \frac{8}{66}$ (e) $t = \frac{12}{66}$ (f) $t = \frac{16}{66}$ (g) $t = \frac{20}{66}$ (h) $t = \frac{24}{66}$ (i) $t = \frac{28}{66}$ (j) $t = \frac{32}{66}$ (k) $t = \frac{35}{66}$ (l) $t = \frac{38}{66}$ (m) $t = \frac{41}{66}$ (n) $t = \frac{44}{66}$ (o) $t = \frac{47}{66}$ (p) $t = \frac{50}{66}$ (q) $t = \frac{53}{66}$ (r) $t = \frac{56}{66}$ (s) $t = \frac{59}{66}$ (t) $t = \frac{62}{66}$ (u) $t = \frac{65}{66}$

Figure 4.12: Interactions between gene ontological groups related to the developmental process undergo dynamic rewiring. The weight of an edge between two ontological groups is the total number of connections between genes in the two groups. In the visualization, the width of an edge is proportional to its edge weight. We thresholded the edge weight at 30 in (b)–(u) so that only those interactions exceeding this number are displayed. The average network in (a) is produced by averaging the networks underlying (b)–(u). In this case, the threshold is set to 20 instead.

interactions occur during development. In Figure 4.10 we listed these recovered known interactions and the precise time when they occur. This also provides a way to check whether the learned networks are biologically plausible given the prior knowledge of the actual occurrence of gene interactions. For instance, the interaction between genes msn and dock is related to the regulation of embryonic cell shape, correct targeting of photoreceptor axons. This is very consistent with the timeline provided by the dynamic networks. A second example is the interaction between genes sno and Dl which is related to the development of compound eyes of *Drosophila*. A third example is between genes caps and Chi which are related to wing development during pupal stage. What is most interesting is that the dynamic networks provide timelines for many other gene interactions that have not yet been verified experimentally. This information will be a useful guide for future experiments.

We further studied the relations between 130 transcriptional factors (TF). The network contains several clusters of transcriptional cascades, and we will present the detail of the largest transcriptional factor cascade involving 36 transcriptional factors (Figure 4.11). This cascade of TFs is functionally very coherent, and many TFs in this network play important roles in the nervous system and eye development. For example, Zn finger homeodomain 1 (zhf1), brinker (brk), charlatan (chn), decapentaplegic (dpp), invected (inv), forkhead box, subgroup 0 (foxo), Optix, eagle (eg), prospero (pros), pointed (pnt), thickveins (tkv), extra macrochaetae (emc), lilliputian (lilli), and doublesex (dsx) are all involved in nervous and eye development. Besides functional coherence, the network also reveals the dynamic nature of gene regulation: some relations are persistent across the full developmental cycle, while many others are transient and specific to certain stages of development. For instance, five transcriptional factors, brk-pnt-zfh1-pros-dpp, form a long cascade of regulatory relations which are active across the full developmental cycle. Another example is gene Optix which is active across the full developmental cycle and serves as a hub for many other regulatory relations. As for transience of the regulatory relations, TFs to the right of the Optix hub reduced in their activity as development proceeds to a later stage. Furthermore, Optix connects two disjoint cascades of gene regulations to its left and right side after embryonic stage.

The dynamic networks also provide an overview of the interactions between genes from different functional groups. In Figure 4.12 we grouped genes according to 58 ontologies and visualized the connectivity between groups. We can see that large topological changes and network rewiring occur between functional groups. Besides expected interactions, the figure also reveals many seemingly unexpected interactions. For instance, during the transition from pupa stage to adult stage, *Drosophila* is undergoing a huge metamorphosis. One major feature of this metamorphosis is the development of the wing. As can be seen from Figure 4.12(r) and (s), genes related to metamorphosis, wing margin morphogenesis, wing vein morphogenesis, and apposition of wing surfaces are among the most active group of genes, and they carry their activity into adult stage. Actually, many of these genes are also very active during early embryonic stage [for example, Figure 4.12(b) and (c)]; though the difference is they interact with different groups of genes. On one hand, the abundance of the transcripts from these genes at embryonic stage is likely due to maternal deposit [10]; on the other hand, this can also be due to the diverse functionalities of these genes. For instance, two genes related to wing development, held out wings (how) and tolloid (td), also play roles in embryonic development.

4.8 Discussion

We have presented two algorithms for an important problem of structure estimation of time-varying networks. While the structure estimation of the static networks is an important problem in itself, in certain cases static structures are of limited use. More specifically, a static structure only shows connections and interactions that are persistent throughout the whole time period and, therefore, time-varying structures are needed to describe dynamic interactions that are transient in time. Although the algorithms presented in this paper for learning time-varying networks are simple, they can already be used to discover some patterns that would not be discovered using a method that estimates static networks. However, the ability to learn time-varying networks comes at a price of extra tuning parameters: the bandwidth parameter h or the penalty parameter λ_{TV}.

Throughout the chapter, we assume that the observations at different points in time are independent. An important future direction is the analysis of the graph structure estimation from a general time series, with dependent observations. In our opinion, this extension will be straightforward but with great practical importance. Furthermore, we have worked with the assumption that the data are binary, however, extending the procedure to work with multi-category data is also straightforward. One possible approach is explained in [151] and can be directly used here.

There are still ways to improve the methods presented here. For instance, more principled ways of selecting tuning parameters are definitely needed. Selecting the tuning parameters in the neighborhood selection procedure for static graphs is not an easy problem, and estimating time-varying graphs makes the problem more challenging. Furthermore, methods presented here do not allow for the incorporation of existing knowledge on the network topology into the algorithm. In some cases, the data are very scarce and we would like to incorporate as much prior knowledge as possible, so developing Bayesian methods seems very important.

The method `smooth` and the method `TV` represent two different ends of the spectrum: one algorithm is able to estimate smoothly changing networks, while the other one is tailored toward estimation of structural changes in the model. It is important to bring the two methods together in the future work. There is a great amount of work on nonparametric estimation of change points and it would be interesting to incorporate those methods for estimating time-varying networks.

Chapter 5

Sparsistent estimation of smoothly varying Ising model

In the previous chapter, we proposed a method based on kernel-smoothing ℓ_1-penalized logistic regression for estimating time-varying networks from nodal observations collected from a time-series of observational data. In this chapter, we establish conditions under which the proposed method consistently recovers the structure of a time-varying network. This work complements previous empirical findings by providing sound theoretical guarantees for the proposed estimation procedure. Theoretical findings are illustrated through numerical simulations.

5.1 Introduction

In this chapter, we study the problem of estimating a sequence of high-dimensional MRFs that slowly evolve over time from observational data. Recall the setup introduced in the previous chapter. We are given a sequence of n nodal states $\mathcal{D}_n = \{\mathbf{x}^t \sim \mathbb{P}_{\boldsymbol{\theta}^t} \mid t \in \mathcal{T}_n\}$, with the time index defined as $\mathcal{T}_n = \{1/n, 2/n, \ldots, 1\}$. For simplicity of presentation, we assume that the observations are equidistant in time and only one observation is available at each time point from distribution $\mathbb{P}_{\boldsymbol{\theta}^t}$ indexed by $\boldsymbol{\theta}^t$. Specifically, we assume that the p-dimensional random vector $\mathbf{X}^t$ takes values in $\{-1, 1\}^p$ and the probability distribution takes the following form:

$$\mathbb{P}_{\boldsymbol{\theta}^t}(x) = \frac{1}{Z(\boldsymbol{\theta}^t)} \exp\left(\sum_{(u,v)\in E^t} \theta^t_{uv} x_u x_v\right), \quad \forall t \in \mathcal{T}_n,$$

where $Z(\boldsymbol{\theta}^t)$ is the partition function, $\boldsymbol{\theta}^t \in \mathbb{R}^{\binom{p}{2}}$ is the parameter vector and $G^t = (V, E^t)$ is an undirected graph representing certain conditional independence assumptions among subsets of the p-dimensional random vector $\mathbf{X}^t$. For any given time point $\tau \in [0, 1]$, we are interested in estimating the graph G^τ associated with $\mathbb{P}_{\boldsymbol{\theta}^\tau}$, given the observations $\mathcal{D}_n$. Since we are primarily interested in a situation where the total number of observation n is small compared to the dimension p, our estimation task is going to be feasible only under some regularity conditions. We impose two natural assumptions: the *sparsity* of the graphs $\{G^t\}_{t\in\mathcal{T}_n}$, and the *smoothness* of the parameters $\boldsymbol{\theta}^t$ as functions of time. These assumptions are precisely stated in §5.2. Intuitively,

Input: Dataset $\mathcal{D}_n$, time point of interest $\tau \in [0,1]$, penalty parameter λ_n, bandwidth parameter h
Output: Estimate of the graph structure $\widehat{G}^\tau$

foreach $u \in V$ **do**

Estimate $\widehat{\boldsymbol{\theta}}_u$ by solving the convex program (4.7)

Estimate the set of signed neighboring edges $\widehat{S}^\tau_\pm(u)$ using (4.4)

end

Combine sets $\{\widehat{S}^\tau_\pm(u)\}_{u\in V}$ to obtain $\widehat{G}^\tau$.

Algorithm 1: Graph structure estimation

the smoothness assumption is required so that a graph structure at the time point τ can be estimated from samples close in time to τ. On the other hand, the sparsity assumption is required to avoid the curse of dimensionality and to ensure that a the graph structure can be identified from a small sample.

The main contribution of this chapter is to establish theoretical guarantees for the estimation procedure discussed in §4.2. The estimation procedure is based on temporally smoothed ℓ_1-regularized logistic regression formalism, as summarized in Algorithm 1. An application to real world data was given in [159], where the procedure was used to infer the latent evolving regulatory network underlying 588 genes across the life cycle of *Drosophila melanogaster* from microarray time course. Although the true regulatory network is not known for this organism, the procedure recovers a number of interactions that were previously experimentally validated. Since in most real world problems the ground truth is not known, we emphasize the importance of simulation studies to evaluate the estimation procedure.

It is noteworthy that the problem of the graph structure estimation is quite different from the problem of (value-) consistent estimation of the unknown parameter $\boldsymbol{\theta}$ that indexes the distribution. In general, the graph structure estimation requires a more stringent assumptions on the underlying distribution and the parameter values. For example, observe that a consistent estimator of $\boldsymbol{\theta}$ in the Euclidean distance does not guarantee a consistent estimation of the graph structure, encoded by the non-zero patter of the estimator. In the motivating problems that we gave in §2 and §3, the main goal is to understand the interactions between different actors. These interactions are more easily interpreted by a domain expert than the numerical values of the parameter vector $\boldsymbol{\theta}$ and have potential to reveal more information about the underlying process of interest. This is especially true in situations where there is little or no domain knowledge and one is interested in obtaining casual, preliminary information.

5.2 Main theoretical result

In this section, we provide conditions under which the estimation procedure detailed in §4.2 consistently recovers the graph structure. In particular, we show that under suitable conditions

$\mathbb{P}[\forall u\ \widehat{S}^{\tau}_{\pm}(u) = S^{\tau}_{\pm}(u)] \xrightarrow{n\to\infty} 1$, the property known as *sparsistency*. We are mainly interested in the high-dimensional case, where the dimension $p = p_n$ is comparable or even larger than the sample size n. It is of great interest to understand the performance of the estimator under this assumption, since in many real world scenarios the dimensionality of data is large. Our analysis is asymptotic and we consider the model dimension $p = p_n$ to grow at a certain rate as the sample size grows. This essentially allows us to consider more "complicated" models as we observe more data points. Another quantity that will describe the complexity of the model is the maximum node degree $s = s_n$, which is also considered as a function of the sample size. Under the assumption that the true-graph structure is sparse, we will require that the maximum node degree is small, $s \ll n$. The main result describes the scaling of the triple (n, p_n, s_n) under which the estimation procedure given in the previous section estimates the graph structure consistently.

We will need certain regularity conditions to hold in order to prove the sparsistency result. These conditions are expressed in terms of the Hessian of the log-likelihood function as evaluated at the true model parameter, i.e., the Fisher information matrix. The Fisher information matrix $\mathbf{Q}^{\tau}_u \in \mathbb{R}^{(p-1)\times(p-1)}$ is a matrix defined for each node $u \in V$ as:

$$\begin{aligned}\mathbf{Q}^{\tau}_u &:= \mathbb{E}[\nabla^2 \log \mathbb{P}_{\boldsymbol{\theta}^{\tau}_u}[X_u|\mathbf{X}_{\backslash u}]] \\ &= \mathbb{E}[\eta(\mathbf{X}; \boldsymbol{\theta}^{\tau}_u)\mathbf{X}_{\backslash u}\mathbf{X}'_{\backslash u}],\end{aligned}$$

where

$$\eta(\mathbf{x}; \boldsymbol{\theta}_u) := \frac{4\exp(2x_u\langle \boldsymbol{\theta}_u, \mathbf{x}_{\backslash u}\rangle)}{(\exp(2x_u\langle \boldsymbol{\theta}_u, \mathbf{x}_{\backslash u}\rangle) + 1)^2}$$

is the variance function and ∇^2 denotes the operator that computes the matrix of second derivatives. We write $\mathbf{Q}^{\tau} := \mathbf{Q}^{\tau}_u$ and assume that the following assumptions hold for each node $u \in V$.

A1: Dependency condition There exist constants $C_{\min}, D_{\min}, D_{\max} > 0$ such that

$$\Lambda_{\min}(\mathbf{Q}^{\tau}_{SS}) \geq C_{\min}$$

and

$$\Lambda_{\min}\left(\boldsymbol{\Sigma}^{\tau}\right) \geq D_{\min}, \quad \Lambda_{\max}\left(\boldsymbol{\Sigma}^{\tau}\right) \leq D_{\max},$$

where $\boldsymbol{\Sigma}^{\tau} = \mathbb{E}_{\boldsymbol{\theta}^{\tau}}[\mathbf{X}\mathbf{X}']$. Here $\Lambda_{\min}(\cdot)$ and $\Lambda_{\max}(\cdot)$ denote the minimum and maximum eigenvalue of a matrix.

A2: Incoherence condition There exists an incoherence parameter $\alpha \in (0, 1]$ such that

$$|||\mathbf{Q}^{\tau}_{S^cS}(\mathbf{Q}^{\tau}_{SS})^{-1}|||_{\infty} \leq 1 - \alpha,$$

where, for a matrix $A \in \mathbb{R}^{a\times b}$, the ℓ_{∞} matrix norm is defined as

$$|||A|||_{\infty} := \max_{i\in\{1,\ldots,a\}} \sum_{j=1}^{b} |a_{ij}|.$$

The set S^c denotes the complement of the set S in $\{1, \ldots, p\}$, that is, $S^c = \{1, \ldots, p\}\backslash S$. With some abuse of notation, when defining assumptions A1 and A2, we use the index set $S := S^{\tau}(u)$

to denote nodes adjacent to the node u at time τ. For example, if $s = |S|$, then $\mathbf{Q}^{\tau}_{SS} \in \mathbb{R}^{s\times s}$ denotes the sub-matrix of $\mathbf{Q}^{\tau}$ indexed by S.

Condition A1 assures that the relevant features are not too correlated, while condition A2 assures that the irrelevant features do not have to strong effect onto the relevant features. Similar conditions are common in other literature on high-dimensional estimation (see, for example, [76, 135, 146, 151] and references therein). The difference here is that we assume the conditions hold for the time point of interest τ at which we want to recover the graph structure.

Next, we assume that the distribution $\mathbb{P}_{\boldsymbol{\theta}^{\tau}}$ changes smoothly over time, which we express in the following form, for every node $u \in V$.

A3: Smoothness conditions Let $\boldsymbol{\Sigma}^t = [\sigma^t_{uv}]$. There exists a constant $M > 0$ such that it upper bounds the following quantities:

$$\max_{u,v\in V\times V} \sup_{t\in[0,1]} |\frac{\partial}{\partial t}\sigma^t_{uv}| < M, \quad \max_{u,v\in V\times V} \sup_{t\in[0,1]} |\frac{\partial^2}{\partial t^2}\sigma^t_{uv}| < M$$

$$\max_{u,v\in V\times V} \sup_{t\in[0,1]} |\frac{\partial}{\partial t}\theta^t_{uv}| < M, \quad \max_{u,v\in V\times V} \sup_{t\in[0,1]} |\frac{\partial^2}{\partial t^2}\theta^t_{uv}| < M.$$

The condition A3 captures our notion of the distribution that changes smoothly over time. If we consider the elements of the covariance matrix and the elements of the parameter vector as a function of time, then these functions have bounded first and second derivatives. From these assumptions, it is not too hard to see that elements of the Fisher information matrix are also smooth functions of time.

A4: Kernel The kernel $K : \mathbb{R} \mapsto \mathbb{R}$ is a symmetric function, supported in $[-1, 1]$. There exists a constant $M_K \geq 1$ which upper bounds the quantities $\max_{z\in\mathbb{R}} |K(z)|$ and $\max_{z\in\mathbb{R}} K(z)^2$.

The condition A4 gives some regularity conditions on the kernel used to define the weights. For example, the assumption is satisfied by the box kernel $K(z) = \frac{1}{2}\,\mathbb{I}\{z \in [-1, 1]\}$.

With the assumptions made above, we are ready to state the theorem that characterizes the consistency of the method given in §4.2 for recovering the unknown time-varying graph structure. An important quantity, appearing in the statement, is the minimum value of the parameter vector that is different from zero

$$\theta_{\min} = \min_{(u,v)\in E^{\tau}} |\theta^{\tau}_{uv}|.$$

Intuitively, the success of the recovery should depend on how hard it is to distinguish the true non-zero parameters from noise.

Theorem 5.1. *Assume that the dependency condition A1 holds with $C_{\min}$, $D_{\min}$ and $D_{\max}$, that for each node $u \in V$, the Fisher information matrix $\mathbf{Q}^{\tau}$ satisfies the incoherence condition A2 with parameter α, the smoothness assumption A3 holds with parameter M, and that the kernel function used in* (4.7) *satisfies assumption A4 with parameter M_K. Let the regularization parameter satisfy*

$$\lambda_n \geq C\frac{\sqrt{\log p}}{n^{1/3}}$$

for a constant $C > 0$ independent of (n, p, s). Furthermore, assume that the following conditions hold:

1. $h = \mathcal{O}(n^{-\frac{1}{3}})$
2. $s = o(n^{1/3})$, $\frac{s^3 \log p}{n^{2/3}} = o(1)$
3. $\theta_{\min} = \Omega(\frac{\sqrt{s \log p}}{n^{1/3}})$.

Then for a fixed $\tau \in [0,1]$ the estimated graph $\widehat{G}^{\tau}(\lambda_n)$ obtained through neighborhood selection satisfies

$$\mathbb{P}\left[\widehat{G}^{\tau}(\lambda_n) \neq G^{\tau}\right] = \mathcal{O}\left(\exp\left(-C\frac{n^{2/3}}{s^3} + C' \log p\right)\right) \to 0,$$

for some constants C', C'' independent of (n, p, s).

This theorem guarantees that the procedure in Algorithm 1 asymptotically recovers the sequence of graphs underlying all the nodal-state measurements in a time series, and the snapshot of the evolving graph at any time point during measurement intervals, under appropriate regularization parameter λ_n as long as the ambient dimensionality p and the maximum node degree s are not too large, and minimum $\boldsymbol{\theta}$ values do not tend to zero too fast.

Remarks:

1. The bandwidth parameter h is chosen so that it balances variance and squared bias of estimation of the elements of the Fisher information matrix.
2. Theorem 5.1 states that the tuning parameter λ can be set as $\lambda_n \geq Cn^{-1/3}\sqrt{\log p}$. In practice, one can use the Bayesian information criterion to select the tuning parameter λ_n is a data dependent way, as explained in §4.5. We conjecture that this approach would lead to asymptotically consistent model selection, however, this claim needs to be proven.
3. Condition 2 requires that the size of the neighborhood of each node remains smaller than the size of the samples. However, the model ambient dimension p is allowed to grow exponentially in n.
4. Condition 3 is crucial to be able to distinguish true elements in the neighborhood of a node. We require that the size of the minimum element of the parameter vector stays bounded away from zero.
5. The rate of convergence is dictated by the rate of convergence of the sample Fisher information matrix to the true Fisher information matrix, as shown in Lemma 5.3. Using a local linear smoother, instead of the kernel smoother, to estimate the coefficients in the model (4.5) one could get a faster rate of convergence.
6. Theorem 5.1 provides sufficient conditions for reliable estimation of the sequence of graphs when the sample size is large enough. In order to improve small sample properties of the procedure, one could adapt the approach of [76] to the time-varying setting, to incorporate sharing between nodes. [76] estimate all the local neighborhoods simultaneously, as opposed to estimating each neighborhood individually, effectively reducing the number of parameters needed to be inferred from data. This is especially beneficial in networks with prominent hubs and scale-free networks.

In order to obtain insight into the network dynamics one needs to estimate the graph structure at multiple time points. A common choice is to estimate the graph structure for every $\tau \in \mathcal{T}_n$ and obtain a sequence of graph structures $\{\widehat{G}^{\tau}\}_{\tau \in \mathcal{T}_n}$. We a have the following immediate consequence of Theorem 5.1.

Corollary 5.1. *Under the assumptions of Theorem 5.1, we have that*

$$\mathbb{P}\left[\forall \tau \in \mathcal{T}_n \ : \ \widehat{G}^{\tau}(\lambda_n) = G^{\tau}\right] \xrightarrow{n \to \infty} 1.$$

In the sequel, we set out to prove Theorem 5.1. First, we show that the minimizer $\widehat{\boldsymbol{\theta}}_u^{\tau}$ of (4.7) is unique under the assumptions given in Theorem 5.1. Next, we show that with high probability the estimator $\widehat{\boldsymbol{\theta}}_u^{\tau}$ recovers the true neighborhood of a node u. Repeating the procedure for all nodes $u \in V$ we obtain the result stated in Theorem 5.1. The proof uses the results that the empirical estimates of the Fisher information matrix and the covariance matrix are close elementwise to their population versions.

5.3 Proof of the main result

In this section we give the proof of Theorem 5.1. The proof is given through a sequence of technical lemmas. We build on the ideas developed in [151]. Note that in what follows, we use C, C' and C'' to denote positive constants independent of (n, p, s) and their value my change from line to line.

The main idea behind the proof is to characterize the minimum obtained in (4.7) and show that the correct neighborhood of one node at an arbitrary time point can be recovered with high probability. Next, using the union bound over the nodes of a graph, we can conclude that the whole graph is estimated sparsistently at the time points of interest.

We first address the problem of uniqueness of the solution to (4.7). Note that because the objective in (4.7) is not strictly convex, it is necessary to show that the non-zero pattern of the parameter vector is unique, since otherwise the problem of sparsistent graph estimation would be meaningless. Under the conditions of Theorem 5.1 we have that the solution is unique. This is shown in Lemma 5.1 and Lemma 5.2. Lemma 5.1 gives conditions under which two solutions to the problem in (4.7) have the same pattern of non-zero elements. Lemma 5.2 then shows, that with probability tending to 1, the solution is unique. Once we have shown that the solution to the problem in (4.7) is unique, we proceed to show that it recovers the correct pattern of non-zero elements. To show that, we require the sample version of the Fisher information matrix to satisfy certain conditions. Under the assumptions of Theorem 5.1, Lemma 5.3 shows that the sample version of the Fisher information matrix satisfies the same conditions as the true Fisher information matrix, although with worse constants. Next we identify two events, related to the Karush-Kuhn-Tucker optimality conditions, on which the vector $\widehat{\boldsymbol{\theta}}_u$ recovers the correct neighborhood the node u. This is shown in Proposition 5.1. Finally, Proposition 5.2 shows that the event, on which the neighborhood of the node u is correctly identified, occurs with probability tending to 1 under the assumptions of Theorem 5.1. Table 5.1 provides a summary of different parts of the proof.

Let us denote the set of all solution to (4.7) as $\Theta(\lambda_n)$. We define the objective function in (4.7) by

$$F(\boldsymbol{\theta}_u) := -\sum_{t \in \mathcal{T}_n} w_t^{\tau} \gamma(\boldsymbol{\theta}_u; \mathbf{x}^t) + \lambda_n ||\boldsymbol{\theta}_u||_1 \tag{5.1}$$

Table 5.1: Outline of the proof strategy.

Result	Description of the result
Lemma 5.1 and Lemma 5.2	These two lemmas establish the uniqueness of the solution to the optimization problem in (4.7).
Lemma 5.3	Shows that the sample version of the Fisher information matrix satisfies the similar conditions to the population version of the Fisher information matrix.
Proposition 5.1	Shows that on an event, related to the KKT conditions, the vector $\widehat{\boldsymbol{\theta}}_u$ recovers the correct neighborhood the node u.
Proposition 5.2	Shows that the event in Proposition 5.1 holds with probability tending to 1.

and we say that $\boldsymbol{\theta}_u \in \mathbb{R}^{p-1}$ satisfies the system $(\mathcal{S})$ when

$$\forall v = 1, \ldots, p-1, \begin{cases} \sum_{t \in \mathcal{T}_n} w_t^\tau (\nabla \gamma(\boldsymbol{\theta}_u; \mathbf{x}^t))_v = \lambda_n \operatorname{sign}(\theta_{uv}) & \text{if } \theta_{uv} \neq 0 \\ |\sum_{t \in \mathcal{T}_n} w_t^\tau (\nabla \gamma(\boldsymbol{\theta}_u; \mathbf{x}^t))_v| \leq \lambda_n & \text{if } \theta_{uv} = 0, \end{cases} \tag{5.2}$$

where

$$\nabla \gamma(\boldsymbol{\theta}_u; \mathbf{x}^t) = \mathbf{x}^t_{\setminus u} \left\{ x_u^t + 1 - 2\mathbb{P}_{\boldsymbol{\theta}_u}[x_u^t = 1 | \mathbf{x}^t_{\setminus u}] \right\} \tag{5.3}$$

is the score function. Eq. (5.2) is obtained by taking the sub-gradient of $F(\boldsymbol{\theta})$ and equating it to zero. From the Karush-Kuhn-Tucker (KKT) conditions it follows that $\boldsymbol{\theta}_u \in \mathbb{R}^{p-1}$ belongs to $\Theta(\lambda_n)$ if and only if $\boldsymbol{\theta}_u$ satisfies the system $(\mathcal{S})$. The following Lemma shows that any two solutions have the same non-zero pattern.

Lemma 5.1. *Consider a node $u \in V$. If $\overline{\boldsymbol{\theta}}_u \in \mathbb{R}^{p-1}$ and $\widetilde{\boldsymbol{\theta}}_u \in \mathbb{R}^{p-1}$ both belong to $\Theta(\lambda_n)$ then $\langle \mathbf{x}^t_{\setminus u}, \overline{\boldsymbol{\theta}}_u \rangle = \langle \mathbf{x}^t_{\setminus u}, \widetilde{\boldsymbol{\theta}}_u \rangle$, $t \in \mathcal{T}_n$. Furthermore, solutions $\overline{\boldsymbol{\theta}}_u$ and $\widetilde{\boldsymbol{\theta}}_u$ have non-zero elements in the same positions.*

We now use the result of Lemma 5.1 to show that with high probability the minimizer in (4.7) is unique. We consider the following event:

$$\Omega_{01} = \{ D_{\min} - \delta \leq \mathbf{y}' \widehat{\boldsymbol{\Sigma}}^\tau_{SS} \mathbf{y} \leq D_{\max} + \delta \ : \ \mathbf{y} \in \mathbb{R}^s, ||\mathbf{y}||_2 = 1 \}.$$

Lemma 5.2. *Consider a node $u \in V$. Assume that the conditions of Lemma 5.6 are satisfied. Assume also that the dependency condition A1 holds. There are constants $C, C', C'' > 0$ depending on M and M_K only, such that*

$$\mathbb{P}[\Omega_{01}] \geq 1 - 4 \exp(-Cnh(\frac{\delta}{s} - C'h)^2 + C'' \log(s)).$$

Moreover, on the event Ω_{01}, the minimizer of (4.7) *is unique.*

We have shown that the estimate $\widehat{\boldsymbol{\theta}}_u^\tau$ is unique on the event Ω_{01}, which under the conditions of Theorem 5.1 happens with probability converging to 1 exponentially fast. To finish the proof of Theorem 5.1 we need to show that the estimate $\widehat{\boldsymbol{\theta}}_u^\tau$ has the same non-zero pattern as the true parameter vector $\boldsymbol{\theta}_u^\tau$. In order to show that we consider a few "good" events, which happen with high probability and on which the estimate $\widehat{\boldsymbol{\theta}}_u^\tau$ has the desired properties. We start by characterizing the sample version of the Fisher information matrix, defined in (5.10). Consider the following events:

$$\Omega_{02} := \{C_{\min} - \delta \leq \mathbf{y}' \widehat{\mathbf{Q}}_{SS}^\tau \mathbf{y} \ : \ \mathbf{y} \in \mathbb{R}^s, ||\mathbf{y}||_2 = 1\}$$

and

$$\Omega_{03} := \{|\!|\!| \widehat{\mathbf{Q}}_{S^cS}^\tau (\widehat{\mathbf{Q}}_{SS}^\tau)^{-1} |\!|\!|_\infty \leq 1 - \frac{\alpha}{2}\}.$$

Lemma 5.3. *Assume that the conditions of Lemma 5.6 are satisfied. Assume also that the dependency condition A1 holds and the incoherence condition A2 holds with the incoherence parameter α. There are constants $C, C', C'' > 0$ depending on M, M_K and α only, such that*

$$\mathbb{P}[\Omega_{02}] \geq 1 - 2\exp(-C\frac{nh\delta^2}{s^2} + C'\log(s))$$

and

$$\mathbb{P}[\Omega_{03}] \geq 1 - \exp(-C\frac{nh}{s^3} + C''\log(p)).$$

Lemma 5.3 guarantees that the sample Fisher information matrix satisfies "good" properties with high probability, under the appropriate scaling of quantities n, p, s and h.

We are now ready to analyze the optimum to the convex program (4.7). To that end we apply the mean-value theorem coordinate-wise to the gradient of the weighted logloss

$$\sum_{t \in \mathcal{T}_n} w_t^\tau \nabla \gamma(\boldsymbol{\theta}_u; \mathbf{x}^t)$$

x and obtain

$$\sum_{t \in \mathcal{T}_n} w_t^\tau (\nabla \gamma(\widehat{\boldsymbol{\theta}}_u^\tau; \mathbf{x}^t) - \nabla \gamma(\boldsymbol{\theta}_u^\tau; \mathbf{x}^t)) = [\sum_{t \in \mathcal{T}_n} w_t^\tau \nabla^2 \gamma(\boldsymbol{\theta}_u^\tau; \mathbf{x}^t)](\widehat{\boldsymbol{\theta}}_u^\tau - \boldsymbol{\theta}_u^\tau) + \boldsymbol{\Delta}^\tau, \qquad (5.4)$$

where $\boldsymbol{\Delta}^\tau \in \mathbb{R}^{p-1}$ is the remainder term of the form

$$\Delta_v^\tau \ = \ [\sum_{t \in \mathcal{T}_n} w_t^\tau (\nabla^2 \gamma(\overline{\boldsymbol{\theta}}_u^{(v)}; \mathbf{x}^t) - \nabla^2 \gamma(\boldsymbol{\theta}_u^\tau; \mathbf{x}^t))]_v' (\widehat{\boldsymbol{\theta}}_u^\tau - \boldsymbol{\theta}_u^\tau)$$

and $\overline{\boldsymbol{\theta}}_u^{(v)}$ is a point on the line between $\boldsymbol{\theta}_u^\tau$ and $\widehat{\boldsymbol{\theta}}_u^\tau$, and $[\cdot]_v'$ denoting the v-th row of the matrix. Recall that $\widehat{\mathbf{Q}}^\tau = \sum_{t \in \mathcal{T}_n} w_t^\tau \nabla^2 \gamma(\boldsymbol{\theta}_u^\tau; \mathbf{x}^t)$. Using the expansion (5.4), we write the KKT conditions given in (5.2) in the following form, $\forall v = 1, \ldots, p-1$,

$$\begin{cases} \widehat{\mathbf{Q}}_v^\tau(\boldsymbol{\theta}_u - \boldsymbol{\theta}_u^\tau) + \sum_{t \in \mathcal{T}_n} w_t^\tau (\nabla \gamma(\boldsymbol{\theta}_u^\tau; \mathbf{x}^t))_v + \Delta_v^\tau = \lambda_n \operatorname{sign}(\theta_{uv}) & \text{if } \theta_{uv} \neq 0 \\ |\widehat{\mathbf{Q}}_v^\tau(\boldsymbol{\theta}_u - \boldsymbol{\theta}_u^\tau) + \sum_{t \in \mathcal{T}_n} w_t^\tau (\nabla \gamma(\boldsymbol{\theta}_u^\tau; \mathbf{x}^t))_v + \Delta_v^\tau| \leq \lambda_n & \text{if } \theta_{uv} = 0. \end{cases} \qquad (5.5)$$

We consider the following events

$$\Omega_0 = \Omega_{01} \cap \Omega_{02} \cap \Omega_{03},$$

$$\Omega_1 = \{\forall v \in S \ : \ |\lambda_n((\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\operatorname{sign}(\boldsymbol{\theta}^\tau_S))_v - ((\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\mathbf{W}^\tau_S)_v| < |\theta^\tau_{uv}|\}$$

and

$$\Omega_2 = \{\forall v \in S^c \ : \ |(\mathbf{W}^\tau_{S^c} - \widehat{\mathbf{Q}}^\tau_{S^cS}(\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\mathbf{W}^\tau_S)_v| < \frac{\alpha}{2}\lambda_n\}$$

where

$$\mathbf{W}^\tau = \sum_{t\in\mathcal{T}_n} w^\tau_t \nabla\gamma(\boldsymbol{\theta}^\tau_u; \mathbf{x}^t) + \boldsymbol{\Delta}^\tau.$$

We will work on the event Ω_0 on which the minimum eigenvalue of $\widehat{\mathbf{Q}}^\tau_{SS}$ is strictly positive and, so, $\widehat{\mathbf{Q}}^\tau_{SS}$ is regular and $\Omega_0 \cap \Omega_1$ and $\Omega_0 \cap \Omega_2$ are well defined.

Proposition 5.1. *Assume that the conditions of Lemma 5.3 are satisfied. The event*

$$\{\forall \widehat{\boldsymbol{\theta}}^\tau_u \in \mathbb{R}^{p-1} \textit{ solution of } (\mathcal{S}),\ \textit{we have } \operatorname{sign}(\widehat{\boldsymbol{\theta}}^\tau_u) = \operatorname{sign}(\boldsymbol{\theta}^\tau_u)\} \cap \Omega_0$$

contains event $\Omega_0 \cap \Omega_1 \cap \Omega_2$.

Proof. We consider the following linear functional

$$G : \left\{ \begin{array}{ccc} \mathbb{R}^s & \to & \mathbb{R}^s \\ \boldsymbol{\theta} & \mapsto & \boldsymbol{\theta} - \boldsymbol{\theta}^\tau_S + (\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\mathbf{W}^\tau_S - \lambda_n(\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\operatorname{sign}(\boldsymbol{\theta}^\tau_S). \end{array} \right.$$

For any two vectors $\mathbf{y} = (y_1, \ldots, y_s)' \in \mathbb{R}^s$ and $\mathbf{r} = (r_1, \ldots, r_s)' \in \mathbb{R}^s_+$, define the following set centered at $\mathbf{y}$ as

$$\mathcal{B}(\mathbf{y}, \mathbf{r}) = \prod_{i=1}^s (y_i - r_i, y_i + r_i).$$

Now, we have

$$G\left(\mathcal{B}(\boldsymbol{\theta}^\tau_S, |\boldsymbol{\theta}^\tau_S|)\right) = \mathcal{B}\left((\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\mathbf{W}^\tau_S - \lambda_n(\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\operatorname{sign}(\boldsymbol{\theta}^\tau_S), |\boldsymbol{\theta}^\tau_S|\right).$$

On the event $\Omega_0 \cap \Omega_1$,

$$0 \in \mathcal{B}\left((\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\mathbf{W}^\tau_S - \lambda_n(\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\operatorname{sign}(\boldsymbol{\theta}^\tau_S), |\boldsymbol{\theta}^\tau_S|\right),$$

which implies that there exists a vector $\overline{\boldsymbol{\theta}}^\tau_S \in \mathcal{B}(\boldsymbol{\theta}^\tau_S, |\boldsymbol{\theta}^\tau_S|)$ such that $G(\overline{\boldsymbol{\theta}}^\tau_S) = 0$. For $\overline{\boldsymbol{\theta}}^\tau_S$ it holds that $\overline{\boldsymbol{\theta}}^\tau_S = \boldsymbol{\theta}^\tau_S + \lambda_n(\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\operatorname{sign}(\boldsymbol{\theta}^\tau_S) - (\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\mathbf{W}^\tau_S$ and $|\overline{\boldsymbol{\theta}}^\tau_S - \boldsymbol{\theta}^\tau_S| < |\boldsymbol{\theta}^\tau_S|$. Thus, the vector $\overline{\boldsymbol{\theta}}^\tau_S$ satisfies

$$\operatorname{sign}(\overline{\boldsymbol{\theta}}^\tau_S) = \operatorname{sign}(\boldsymbol{\theta}^\tau_S)$$

and

$$\widehat{\mathbf{Q}}_{SS}(\overline{\boldsymbol{\theta}}^\tau_S - \boldsymbol{\theta}^\tau_S) + \mathbf{W}^\tau_S = \lambda_n \operatorname{sign}(\overline{\boldsymbol{\theta}}^\tau_S). \tag{5.6}$$

Next, we consider the vector $\overline{\boldsymbol{\theta}}^{\tau} = \begin{pmatrix} \overline{\boldsymbol{\theta}}^{\tau}_{S} \\ \overline{\boldsymbol{\theta}}^{\tau}_{S^c} \end{pmatrix}$ where $\overline{\boldsymbol{\theta}}^{\tau}_{S^c}$ is the null vector of $\mathbb{R}^{p-1-s}$. On event Ω_0, from Lemma 5.3 we know that $\|\widehat{\mathbf{Q}}^{\tau}_{S^cS}(\widehat{\mathbf{Q}}^{\tau}_{SS})^{-1}\|_{\infty} \leq 1 - \frac{\alpha}{2}$. Now, on the event $\Omega_0 \cap \Omega_2$ it holds

$$\begin{aligned} &||\widehat{\mathbf{Q}}^{\tau}_{S^cS}(\overline{\boldsymbol{\theta}}^{\tau}_{S} - \boldsymbol{\theta}^{\tau}_{S}) + \mathbf{W}^{\tau}_{S^c}||_{\infty} = \\ &\quad || - \widehat{\mathbf{Q}}^{\tau}_{S^cS}(\widehat{\mathbf{Q}}^{\tau}_{SS})^{-1}\mathbf{W}^{\tau}_{S} + \mathbf{W}^{\tau}_{S^c} + \lambda_n \widehat{\mathbf{Q}}^{\tau}_{S^cS}(\widehat{\mathbf{Q}}^{\tau}_{SS})^{-1}\operatorname{sign}(\overline{\boldsymbol{\theta}}^{\tau}_{S})||_{\infty} < \lambda_n. \end{aligned} \tag{5.7}$$

Note that for $\overline{\boldsymbol{\theta}}^{\tau}$, equations (5.6) and (5.7) are equivalent to saying that $\overline{\boldsymbol{\theta}}^{\tau}$ satisfies conditions (5.5) or (5.2), i.e., saying that $\overline{\boldsymbol{\theta}}^{\tau}$ satisfies the KKT conditions. Since $\operatorname{sign}(\overline{\boldsymbol{\theta}}^{\tau}_{S}) = \operatorname{sign}(\boldsymbol{\theta}^{\tau}_{S})$, we have $\operatorname{sign}(\overline{\boldsymbol{\theta}}^{\tau}) = \operatorname{sign}(\boldsymbol{\theta}^{\tau}_{u})$. Furthermore, because of the uniqueness of the solution to (4.7) on the event Ω_0 , we conclude that $\widehat{\boldsymbol{\theta}}^{\tau}_{u} = \overline{\boldsymbol{\theta}}^{\tau}$. □

Proposition 5.1 implies Theorem 5.1 if we manage to show that the event $\Omega_0 \cap \Omega_1 \cap \Omega_2$ occurs with high probability under the assumptions stated in Theorem 5.1. Proposition 5.2 characterizes the probability of that event, which concludes the proof of Theorem 5.1.

Proposition 5.2. *Assume that the conditions of Theorem 5.1 are satisfied. Then there are constants $C, C' > 0$ depending on M, M_K, $D_{\max}$, $C_{\min}$ and α only, such that the following holds:*

$$\mathbb{P}[\Omega_0 \cap \Omega_1 \cap \Omega_2] \geq 1 - 2\exp(-Cnh(\lambda_n - sh)^2 + \log(p)).$$

Proof. We start the proof of the proposition by giving a technical lemma, which characterizes the distance between vectors $\widehat{\boldsymbol{\theta}}^{\tau}_{u} = \overline{\boldsymbol{\theta}}^{\tau}$ and $\boldsymbol{\theta}^{\tau}_{u}$ under the assumptions of Theorem 5.1, where $\overline{\boldsymbol{\theta}}^{\tau}$ is constructed in the proof of Proposition 5.1. The following lemma gives a bound on the distance between the vectors $\widehat{\boldsymbol{\theta}}^{\tau}_{S}$ and $\boldsymbol{\theta}^{\tau}_{S}$, which we use in the proof of the proposition. The proof of the lemma is given in Appendix.

Lemma 5.4. *Assume that the conditions of Theorem 5.1 are satisfied. There are constants $C, C' > 0$ depending on $M, M_K, D_{\max}, C_{\min}$ and α only, such that*

$$||\widehat{\boldsymbol{\theta}}^{\tau}_{S} - \boldsymbol{\theta}^{\tau}_{S}||_2 \leq C\frac{\sqrt{s\log p}}{n^{1/3}} \tag{5.8}$$

with probability at least $1 - \exp(-C'\log p)$.

Using Lemma 5.4 we can prove Proposition 5.2. We start by studying the probability of the event Ω_2. We have

$$\Omega_2^C \subset \cup_{v \in S^c}\{\mathbf{W}_v + (\widehat{\mathbf{Q}}^{\tau}_{S^cS}(\widehat{\mathbf{Q}}^{\tau}_{SS})^{-1}\mathbf{W}^{\tau}_{S})_v \geq \frac{\alpha}{2}\lambda_n\}.$$

Recall that $\mathbf{W}^{\tau} = \sum_{t \in \mathcal{T}_n} w^{\tau}_t \nabla\gamma(\boldsymbol{\theta}^{\tau}_{u}; \mathbf{x}^t) + \boldsymbol{\Delta}^{\tau}$. Let us define the event

$$\Omega_3 = \{\max_{1 \leq v \leq p-1} |\mathbf{e}'_v \sum_{t \in \mathcal{T}_n} w^{\tau}_t \nabla\gamma(\boldsymbol{\theta}^{\tau}_{u}; \mathbf{x}^t)| < \frac{\alpha\lambda_n}{4(2-\alpha)}\},$$

where $\mathbf{e}_v \in \mathbb{R}^{p-1}$ is a unit vector with one at the position v and zeros elsewhere. From the proof of Lemma 5.4 available in the appendix we have that $\mathbb{P}[\Omega_3] \geq 1 - 2\exp(-C\log(p))$ and on that event the bound given in (5.8) holds.

On the event Ω_3, we bound the remainder term $\boldsymbol{\Delta}^\tau$. Let $g : \mathbb{R} \mapsto \mathbb{R}$ be defined as

$$g(z) = \frac{4\exp(2z)}{(1+\exp(2z))^2}.$$

Then $\eta(\mathbf{x}; \boldsymbol{\theta}_u) = g(x_u\langle \boldsymbol{\theta}_u, \mathbf{x}_{\setminus u}\rangle)$. For $v \in \{1, \ldots, p-1\}$, using the mean value theorem it follows that

$$\begin{aligned}
\Delta_v &= [\sum_{t\in\mathcal{T}_n} w_t^\tau(\nabla^2\gamma(\overline{\boldsymbol{\theta}}_u^{(v)}; \mathbf{x}^t) - \nabla^2\gamma(\boldsymbol{\theta}_u^\tau; \mathbf{x}^t))]_v'(\widehat{\boldsymbol{\theta}}_u^\tau - \boldsymbol{\theta}_u^\tau) \\
&= \sum_{t\in\mathcal{T}_n} w_t^\tau[\eta(\mathbf{x}^t; \overline{\boldsymbol{\theta}}_u^{(v)}) - \eta(\mathbf{x}^t; \boldsymbol{\theta}_u^\tau)][\mathbf{x}_{\setminus u}^t \mathbf{x}_{\setminus u}^{t'}]_v'[\widehat{\boldsymbol{\theta}}_u^\tau - \boldsymbol{\theta}_u^\tau] \\
&= \sum_{t\in\mathcal{T}_n} w_t^\tau g'(\mathbf{x}_u^t\langle \overline{\overline{\boldsymbol{\theta}}}_u^{(v)}, \mathbf{x}_{\setminus u}^t\rangle)[x_u^t \mathbf{x}_{\setminus u}^t]'[\overline{\boldsymbol{\theta}}_u^{(v)} - \boldsymbol{\theta}_u^\tau][x_v^t \mathbf{x}_{\setminus u}^{t'}][\widehat{\boldsymbol{\theta}}_u^\tau - \boldsymbol{\theta}_u^\tau] \\
&= \sum_{t\in\mathcal{T}_n} w_t^\tau\{g'(x_u^t\langle \overline{\overline{\boldsymbol{\theta}}}_u^{(v)}, \mathbf{x}_{\setminus u}^t\rangle)x_u^t x_v^t\}\{[\overline{\boldsymbol{\theta}}_u^{(v)} - \boldsymbol{\theta}_u^\tau]'\mathbf{x}_{\setminus u}^t \mathbf{x}_{\setminus u}^{t'}[\widehat{\boldsymbol{\theta}}_u^\tau - \boldsymbol{\theta}_u^\tau]\},
\end{aligned}$$

where $\overline{\overline{\boldsymbol{\theta}}}_u^{(v)}$ is another point on the line joining $\widehat{\boldsymbol{\theta}}_u^\tau$ and $\boldsymbol{\theta}_u^\tau$. A simple calculation shows that $|g'(x_u^t\langle \overline{\overline{\boldsymbol{\theta}}}_u^{(v)}, \mathbf{x}_{\setminus u}^t\rangle)x_u^t x_v^t| \leq 1$, for all $t \in \mathcal{T}_n$, so we have

$$\begin{aligned}
|\Delta_v| &\leq [\overline{\boldsymbol{\theta}}_u^{(v)} - \boldsymbol{\theta}_u^\tau]'\{\sum_{t\in\mathcal{T}_n} w_t^\tau \mathbf{x}_{\setminus u}^t \mathbf{x}_{\setminus u}^{t'}\}[\widehat{\boldsymbol{\theta}}_u^\tau - \boldsymbol{\theta}_u^\tau] \\
&\leq [\widehat{\boldsymbol{\theta}}_u^\tau - \boldsymbol{\theta}_u^\tau]'\{\sum_{t\in\mathcal{T}_n} w_t^\tau \mathbf{x}_{\setminus u}^t \mathbf{x}_{\setminus u}^{t'}\}[\widehat{\boldsymbol{\theta}}_u^\tau - \boldsymbol{\theta}_u^\tau] \\
&= [\widehat{\boldsymbol{\theta}}_S^\tau - \boldsymbol{\theta}_S^\tau]'\{\sum_{t\in\mathcal{T}_n} w_t^\tau \mathbf{x}_S^t \mathbf{x}_S^{t'}\}[\widehat{\boldsymbol{\theta}}_S^\tau - \boldsymbol{\theta}_S^\tau] \\
&\leq D_{\max}||\widehat{\boldsymbol{\theta}}_S^\tau - \boldsymbol{\theta}_S^\tau||_2^2.
\end{aligned} \tag{5.9}$$

Combining the equations (5.9) and (5.8), we have that on the event Ω_3

$$\max_{1\leq v\leq p-1} |\Delta_v| \leq C\lambda_n^2 s < \frac{\lambda_n \alpha}{4(2-\alpha)}$$

where C is a constant depending on $D_{\max}$ and $C_{\min}$ only.

On the event $\Omega_0 \cap \Omega_3$, we have

$$W_v^\tau + (\widehat{\mathbf{Q}}_{S^cS}^\tau(\widehat{\mathbf{Q}}_{SS}^\tau)^{-1}\mathbf{W}_S^\tau)_v < \frac{\alpha\lambda_n}{2(2-\alpha)} + (1-\alpha)\frac{\alpha\lambda_n}{2(2-\alpha)} \leq \frac{\alpha\lambda_n}{2}$$

and we can conclude that $\mathbb{P}[\Omega_2] \geq 1 - 2\exp(-C\log(p))$ for some constant C depending on $M, M_K, C_{\min}, D_{\max}$ and α only.

Next, we study the probability of the event Ω_1. We have

$$\Omega_1^C \subset \cup_{v\in S}\{\lambda_n((\widehat{\mathbf{Q}}_{SS}^\tau)^{-1}\operatorname{sign}(\boldsymbol{\theta}_S^\tau))_v + ((\widehat{\mathbf{Q}}_{SS}^\tau)^{-1}W_S^\tau)_v \geq \theta_{uv}^\tau\}.$$

Again, we will consider the event Ω_3. On the event $\Omega_0 \cap \Omega_3$ we have that

$$\lambda_n((\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\operatorname{sign}(\boldsymbol{\theta}^\tau_S))_v + ((\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\mathbf{W}^\tau_S)_v \leq \frac{\lambda_n\sqrt{s}}{C_{\min}} + \frac{\lambda_n}{2C_{\min}} \leq C\lambda_n\sqrt{s},$$

for some constant C. When $\theta_{\min} > C\lambda_n\sqrt{s}$, we have that $\mathbb{P}[\Omega_1] \geq 1 - 2\exp(-C\log(p))$ for some constant C that depends on $M, M_K, C_{\min}, D_{\max}$ and α only. □

In summary, under the assumptions of Theorem 5.1, the probability of event $\Omega_0 \cap \Omega_1 \cap \Omega_2$ converges to one exponentially fast. On this event, we have shown that the estimator $\widehat{\boldsymbol{\theta}}^\tau_u$ is the unique minimizer of (4.7) and that it consistently estimates the signed non-zero pattern of the true parameter vector $\boldsymbol{\theta}^\tau_u$, i.e., it consistently estimates the neighborhood of a node u. Applying the union bound over all nodes $u \in V$, we can conclude that the estimation procedure consistently estimates the graph structure at a time point τ.

5.4 Numerical simulation

In this section, we demonstrate numerical performance of Algorithm 1. A detailed comparison with other estimation procedures and an application to biological data has been reported in [112]. We will use three different types of graph structures: a chain, a nearest-neighbor and a random graph. Each graph has $p = 50$ nodes and the maximum node degree is bounded by $s = 4$. These graphs are detailed below:

Example 1: Chain graph. First a random permutation π of $\{1, \ldots, p\}$ is chosen. Then a graph structure is created by connecting consecutive nodes in the permutation, that is,

$$(\pi(1), \pi(2)), \ldots, (\pi(p-1), \pi(p)) \in E.$$

Example 2: Nearest neighbor graph. A nearest neighbor graph if generated following the procedure outlined in [119]. For each node, we draw a point uniformly at random on a unit square and compute the pairwise distances between nodes. Each node is then connected to 4 closest neighbors. Since some of nodes will have more than 4 adjacent edges, we remove randomly edges from nodes that have degree larger than 4 until the maximum degree of a node in a graph is 4.

Example 3: Random graph. To generate a random graph with $e = 45$ edges, we add each edges one at a time, between random pairs of nodes that have the node degree less than 4.

We use the above described procedure to create the first random graph $\widetilde{G}^0$. Next, we randomly add 10 edges and remove 10 edges from $\widetilde{G}^0$, taking care that the maximum node degree is still 4, to obtain $\widetilde{G}^1$. Repeat the process of adding and removing edges from $\widetilde{G}^1$ to obtain $\widetilde{G}^2, \ldots, \widetilde{G}^5$. We refer to these 6 graphs as the anchor graphs. We will randomly generate the prototype parameter vectors $\widetilde{\boldsymbol{\theta}}^0, \ldots, \widetilde{\boldsymbol{\theta}}^5$, corresponding to the anchor graphs, and then interpolate 200 points between them to obtain the parameters $\{\boldsymbol{\theta}^t\}_{t\in\mathcal{T}_n}$, which gives us $n = 1000$. We generate a prototype parameter vector $\widetilde{\boldsymbol{\theta}}^i$ for each anchor graph $\widetilde{G}^i$, $i \in \{0, \ldots, 5\}$, by sampling non-zero elements of the vector independently from $\mathsf{Unif}([-1, 0.5] \cup [0.5, 1])$. Now, for each $t \in \mathcal{T}_n$ we generate 10 i.i.d. samples using Gibbs sampling from the distribution $\mathbb{P}_{\boldsymbol{\theta}^t}$. Specifically, we discard samples from the first 10^4 iterations and collect samples every 100 iterations.

Table 5.2: Summary of simulation results. The number of nodes $p = 50$ and the number of discrete time points $n = 1000$.

		Number of independent samples									
		1	2	3	4	5	6	7	8	9	10
Precision	Chain	0.75	0.95	0.96	0.96	0.97	0.98	0.99	0.99	0.99	0.99
	NN	0.84	0.98	0.97	0.96	0.98	0.98	0.98	0.98	0.97	0.98
	Random	0.55	0.57	0.65	0.71	0.75	0.79	0.83	0.84	0.85	0.85
Recall	Chain	0.59	0.65	0.69	0.72	0.73	0.73	0.73	0.73	0.73	0.73
	NN	0.48	0.57	0.61	0.63	0.63	0.64	0.64	0.64	0.65	0.65
	Random	0.50	0.52	0.55	0.56	0.56	0.58	0.60	0.60	0.63	0.66
F1 score	Chain	0.66	0.76	0.80	0.82	0.83	0.84	0.84	0.84	0.85	0.84
	NN	0.61	0.72	0.74	0.76	0.77	0.77	0.77	0.77	0.77	0.78
	Random	0.52	0.54	0.60	0.63	0.64	0.67	0.70	0.70	0.72	0.74

We estimate $\widehat{G}^t$ for each $t \in \mathcal{T}_n$ using $k \in \{1, \ldots, 10\}$ samples at each time point. The results are expressed in terms of the precision (Pre) and the recall (Rec) and $F1$ score, which is the harmonic mean of precision and recall, i.e., $F1 := 2 * \mathsf{Pre} * \mathsf{Rec}/(\mathsf{Pre} + \mathsf{Rec})$. Let $\widehat{E}^t$ denote the estimated edge set of $\widehat{G}^t$, then the precision is calculated as $\mathsf{Pre} := 1/n \sum_{t \in \mathcal{T}_n} |\widehat{E}^t \cap E^t|/|\widehat{E}^t|$ and the recall as $\mathsf{Rec} := 1/n \sum_{t \in \mathcal{T}_n} |\widehat{E}^t \cap E^t|/|E^t|$. Furthermore, we report results averaged over 100 independent runs. The tuning parameters are selected by maximizing the BIC score over a grid of regularization parameters as described in §4.5. Table 5.2 contains a summary of simulation results.

We perform an additional simulation that illustrates that the conditions of Theorem 5.1 can be satisfied. We will use the random chain graph and the nearest neighbor graph for two simulation settings. In each setting, we generate two anchor graphs with p nodes and create two prototype parameter vectors, as described above. Then we interpolate these two parameters over n points. Theorem 5.1 predicts the scaling for the sample size n, as a function of other parameters, required to successfully recover the graph at a time point τ. Therefore, if our theory correctly predicts the behavior of the estimation procedure and we plot the hamming distance between the true and recovered graph structure against appropriately rescaled sample size, we expect the curves to reach zero distance for different problem sizes at a same point. The bandwidth parameter h is set as $h = 4.8n^{-1/3}$ and the penalty parameter λ_n as $\lambda_n = 2\sqrt{n^{-2/3}\log(p)}$ as suggested by the theory. Figure 5.1 shows the hamming distance against the scaled sample size $n/(s^{4.5}\log^{1.5}(p))$. Each point is averaged over 100 independent runs.

5.5 Discussion

In the chapter, we focus on sparsistent estimation of the time-varying high-dimensional graph structure in Markov Random Fields from a small size sample. An interesting open direction is estimation of the graph structure from a general time-series, where observations are dependent.

In our opinion, the graph structure that changes with time creates the biggest technical difficulties. Incorporating dependent observations would be an easier problem to address, however, the one of great practical importance, since samples in the real data sets are likely to be dependent. Another open direction is to establish necessary conditions, to complement sufficient conditions established here, under which it is possible to estimate a time-varying graph structure. Another research direction may be to use non-convex penalties introduced by [64] in place of the ℓ_1 penalty. The idea would be to relax the condition imposed in the assumption A2, since it is well known that the SCAD penalties improve performance when the variables are correlated.

5.6 Technical results

5.6.1 Large deviation inequalities

In this section we characterize the deviation of elements of the sample Fisher information matrix $\widehat{\mathbf{Q}}^\tau := \widehat{\mathbf{Q}}_u^\tau$ at time point τ, defined as

$$\widehat{\mathbf{Q}}^\tau = \sum_t w_t^\tau \eta(\mathbf{x}^t; \boldsymbol{\theta}_u^\tau) \mathbf{x}_{\setminus u}^t \mathbf{x}_{\setminus u}^{t'}, \tag{5.10}$$

and the sample covariance matrix $\widehat{\boldsymbol{\Sigma}}^\tau$ from their population versions $\mathbf{Q}^\tau$ and $\boldsymbol{\Sigma}^\tau$. These results are crucial for the proof of the main theorem, where the consistency result depends on the bounds on the difference $\widehat{\mathbf{Q}}^\tau - \mathbf{Q}^\tau$ and $\widehat{\boldsymbol{\Sigma}}^\tau - \boldsymbol{\Sigma}^\tau$. In the following, we use C, C' and C'' as generic positive constants independent of (n, p, s).

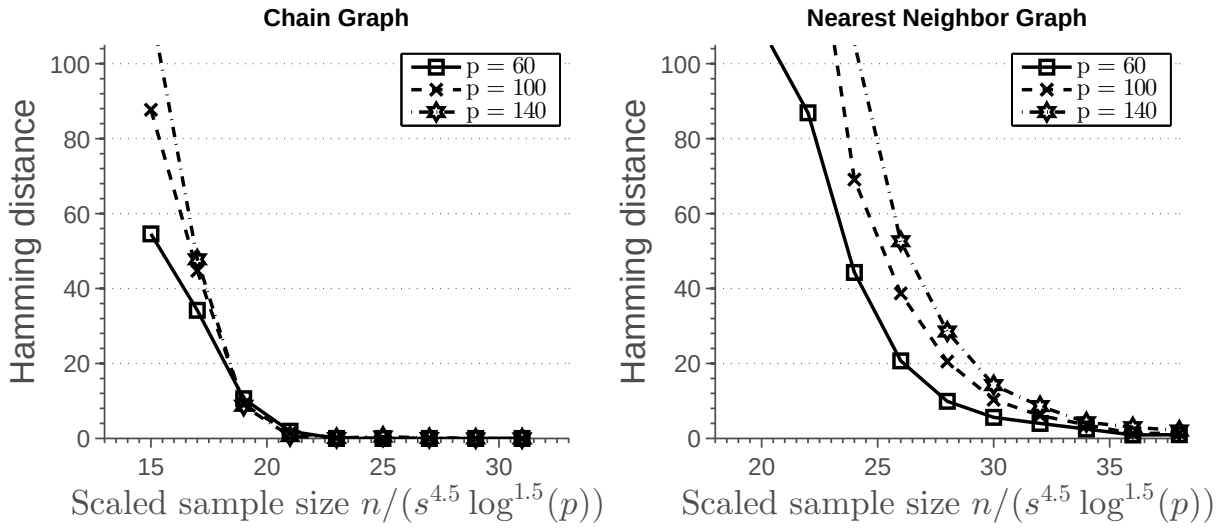


Figure 5.1: Average hamming distance plotted against the rescaled sample size. Each column represents one simulation setting. Results are averaged over 100 independent runs.

Sample Fisher information matrix

To bound the deviation between elements of $\widehat{\mathbf{Q}}^\tau = [\widehat{q}^\tau_{vv'}]$ and $\mathbf{Q}^\tau = [q^\tau_{vv'}]$, $v, v' \in V\backslash u$, we will use the following decomposition:

$$\begin{aligned}|\widehat{q}^\tau_{vv'} - q^\tau_{vv'}| \leq & |\sum_{t\in\mathcal{T}_n} w^\tau_t \eta(\mathbf{x}^t;\boldsymbol{\theta}^\tau_u)x^t_v x^t_{v'} - \sum_{t\in\mathcal{T}_n} w^\tau_t \eta(\mathbf{x}^t;\boldsymbol{\theta}^t_u)x^t_v x^t_{v'}| \\ & + |\sum_{t\in\mathcal{T}_n} w^\tau_t \eta(\mathbf{x}^t;\boldsymbol{\theta}^t_u)x^t_v x^t_{v'} - \mathbb{E}[\sum_{t\in\mathcal{T}_n} w^\tau_t \eta(\mathbf{x}^t;\boldsymbol{\theta}^t_u)x^t_v x^t_{v'}]| \\ & + |\mathbb{E}[\sum_{t\in\mathcal{T}_n} w^\tau_t \eta(\mathbf{x}^t;\boldsymbol{\theta}^t_u)x^t_v x^t_{v'}] - q^\tau_{vv'}|.\end{aligned} \tag{5.11}$$

The following lemma gives us bounds on the terms in (5.11).

Lemma 5.5. *Assume that the smoothness condition A3 is satisfied and that the kernel function $K(\cdot)$ satisfies A4. Furthermore, assume*

$$\max_{t\in[0,1]} |\{v \in \{1,\dots,p\} \ : \ \theta^t_{uv} \neq 0\}| < s,$$

i.e., the number of non-zero elements of the parameter vector is bounded by s. There exist constants $C, C', C'' > 0$, depending on M and M_K only, which are the constants quantifying assumption A3 and A4, respectively , such that for any $\tau \in [0,1]$, we have

$$\max_{v,v'} \ |\widehat{q}^\tau_{vv'} - \sum_{t\in\mathcal{T}_n} w^\tau_t \eta(\mathbf{x}^t;\boldsymbol{\theta}^t_u)x^t_v x^t_{v'}| \ = \ Csh \tag{5.12}$$

$$\max_{v,v'} \ |\mathbb{E}[\sum_{t\in\mathcal{T}_n} w^\tau_t \eta(\mathbf{x}^t;\boldsymbol{\theta}^t_u)x^t_v x^t_{v'}] - q^\tau_{vv'}| \ = \ C'h. \tag{5.13}$$

Furthermore,

$$|\sum_{t\in\mathcal{T}_n}(w^\tau_t \eta(\mathbf{x}^t;\boldsymbol{\theta}^t_u)x^t_v x^t_{v'} - \mathbb{E}[w^\tau_t \eta(\mathbf{x}^t;\boldsymbol{\theta}^t_u)X^t_v X^t_{v'}])| < \epsilon \tag{5.14}$$

with probability at least $1 - 2\exp(-C''nh\epsilon^2)$.

Proof. We start the proof by bounding the difference $|\eta(\mathbf{x};\boldsymbol{\theta}^{t+\delta}_u) - \eta(\mathbf{x};\boldsymbol{\theta}^t_u)|$ which will be useful later on. By applying the mean value theorem to $\eta(\mathbf{x};\cdot)$ and the Taylor expansion on $\boldsymbol{\theta}^t_u$ we obtain:

$$\begin{aligned}|\eta(\mathbf{x};\boldsymbol{\theta}^{t+\delta}_u) - \eta(\mathbf{x};\boldsymbol{\theta}^t_u)| &= |\sum_{v=1}^{p-1}(\theta^{t+\delta}_{uv} - \theta^t_{uv})\eta'(\mathbf{x};\overline{\boldsymbol{\theta}}^{(v)}_u)| \qquad \left(\begin{array}{c}\overline{\boldsymbol{\theta}}^{(v)}_u \text{ is a point on the line} \\ \text{between } \boldsymbol{\theta}^{t+\delta}_u \text{ and } \boldsymbol{\theta}^t_u\end{array}\right) \\ &\leq \sum_{v=1}^{p-1}|\theta^{t+\delta}_{uv} - \theta^t_{uv}| \qquad (\ |\eta'(\mathbf{x};\cdot)| \leq 1\ ) \\ &= \sum_{v=1}^{p-1}|\delta\frac{\partial}{\partial t}\theta^t_{uv} + \frac{\delta^2}{2}\frac{\partial^2}{\partial t^2}\theta^t_{uv}\Big|_{t=\beta_v}| \qquad \left(\begin{array}{c}\beta_v \text{ is a point on the line} \\ \text{between } t \text{ and } t+\delta\end{array}\right)\end{aligned}$$

Without loss of generality, let $\tau = 1$. Using the above equation, and the Riemann integral to approximate the sum, we have

$$\begin{aligned}
&|\sum_{t\in\mathcal{T}_n} w_t^\tau \eta(\mathbf{x}^t;\boldsymbol{\theta}_u^\tau)x_v^t x_{v'}^t - \sum_{t\in\mathcal{T}_n} w_t^\tau \eta(\mathbf{x}^t;\boldsymbol{\theta}_u^t)x_v^t x_{v'}^t| \\
&\approx |\int \frac{2}{h}K(\frac{z-\tau}{h})[\eta(\mathbf{x}^z;\boldsymbol{\theta}_u^\tau) - \eta(\mathbf{x}^z;\boldsymbol{\theta}_u^z)]x_v^z x_{v'}^z dz| \\
&\leq 2\int_{-\frac{1}{h}}^0 K(z')|\eta(\mathbf{x}^{\tau+z'h};\boldsymbol{\theta}_u^\tau) - \eta(\mathbf{x}^{\tau+z'h};\boldsymbol{\theta}_u^{\tau+z'h})|dz' \\
&\leq 2\int_{-1}^0 K(z')[\sum_{v=1}^{p-1} |z'h\frac{\partial}{\partial t}\theta_{uv}^t\Big|_{t=\tau} + \frac{(z'h)^2}{2}\frac{\partial^2}{\partial t^2}\theta_{uv}^t\Big|_{t=\beta_v}|]dz' \\
&\leq Csh,
\end{aligned}$$

for some constant $C > 0$ depending on M from A3 which bounds the derivatives in the equation above, and M_K from A4 which bounds the kernel. The last inequality follows from the assumption that the number of non-zero components of the vector $\boldsymbol{\theta}_u^t$ is bounded by s.

Next, we prove equation (5.13). Using the Taylor expansion, for any fixed $1 \leq v, v' \leq p-1$ we have

$$\begin{aligned}
&|\mathbb{E}[\sum_{t\in\mathcal{T}_n} w_t^\tau \eta(\mathbf{x}^t;\boldsymbol{\theta}_u^t)x_v^t x_{v'}^t] - q_{vv'}^\tau| \\
&\quad = |\sum_{t\in\mathcal{T}_n} w_t^\tau (q_{vv'}^t - q_{vv'}^\tau)| \\
&\quad = |\sum_{t\in\mathcal{T}_n} w_t^\tau ((t-\tau)\frac{\partial}{\partial t}q_{vv'}^t\Big|_{t=\tau} + \frac{(t-\tau)^2}{2}\frac{\partial^2}{\partial t^2}q_{vv'}^t\Big|_{t=\xi}|,
\end{aligned}$$

where $\xi \in [t, \tau]$. Since $w_t^\tau = 0$ for $|t-\tau| > h$, we have

$$\max_{v,v'} |\mathbb{E}[\sum_{t\in\mathcal{T}_n} w_t^\tau \eta(\mathbf{x}^t;\boldsymbol{\theta}_u^t)x_v^t x_{v'}^t] - q_{vv'}^\tau| \leq C'h$$

for some constant $C > 0$ depending on M and M_K only.

Finally, we prove equation (5.14). Observe that

$$w_t^\tau \eta(\mathbf{x}^t;\boldsymbol{\theta}_u^t)x_v^t x_{v'}^t$$

are independent and bounded random variables $[-w_t^\tau, w_t^\tau]$. The equation simply follows from the Hoeffding's inequality. □

Using results of Lemma 5.5 we can obtain the rate at which the element-wise distance between the true and sample Fisher information matrix decays to zero as a function of the bandwidth parameter h and the size of neighborhood s. In the proof of the main theorem, the bandwidth parameter will be chosen so that the bias and variance terms are balanced.

Sample covariance matrix

The deviation of the elements of the sample covariance matrix is bounded in a similar way as the deviation of elements of the sample Fisher information matrix, given in Lemma 5.5. Denoting the sample covariance matrix at time point τ as

$$\widehat{\mathbf{\Sigma}}^{\tau} = \sum_{t} w_t^{\tau} \mathbf{x}^t \mathbf{x}^{t'},$$

and the difference between the elements of $\widehat{\mathbf{\Sigma}}^{\tau}$ and $\mathbf{\Sigma}^{\tau}$ can be bounded as

$$\begin{aligned} |\widehat{\sigma}_{uv}^{\tau} - \sigma_{uv}^{\tau}| \;&=\; |\sum_{t\in\mathcal{T}_n} w_t^{\tau} x_u^t x_v^t - \sigma_{uv}^{\tau}| \\ &\leq\; |\sum_{t\in\mathcal{T}_n} w_t^{\tau} x_u^t x_v^t - \mathbb{E}[\sum_{t\in\mathcal{T}_n} w_t^{\tau} x_u^t x_v^t]| \\ &+\; |\mathbb{E}[\sum_{t\in\mathcal{T}_n} w_t^{\tau} x_u^t x_v^t] - \sigma_{uv}^{\tau}|. \end{aligned} \tag{5.15}$$

The following lemma gives us bounds on the terms in (5.15).

Lemma 5.6. *Assume that the smoothness condition A3 is satisfied and that the kernel function $K(\cdot)$ satisfies A4. There are constants $C, C' > 0$ depending on M and M_K only such that for any $\tau \in [0,1]$, we have*

$$\max_{u,v} |\mathbb{E}[\sum_{t\in\mathcal{T}_n} w_t^{\tau} x_u^t x_v^t] - \sigma_{uv}^{\tau}| \leq Ch. \tag{5.16}$$

and

$$|\sum_{t\in\mathcal{T}_n} w_t^{\tau} x_u^t x_v^t - \mathbb{E}[\sum_{t\in\mathcal{T}_n} w_t^{\tau} x_u^t x_v^t]| \leq \epsilon \tag{5.17}$$

with probability at least $1 - 2\exp(-C'nh\epsilon^2)$.

Proof. To obtain the Lemma, we follow the same proof strategy as in the proof of Lemma 5.5. In particular, (5.16) is proved in the same way as (5.13) and (5.17) in the same way as (5.14). The details of this derivation are omitted. □

5.6.2 Proof of Lemma 5.1

The set of minima $\Theta(\lambda_n)$ of a convex function is convex. So, for two distinct points of minima, $\overline{\boldsymbol{\theta}}_u$ and $\widetilde{\boldsymbol{\theta}_u}$, every point on the line connecting two points also belongs to minima, i.e. $\xi\overline{\boldsymbol{\theta}_u} + (1-\xi)\widetilde{\boldsymbol{\theta}_u} \in \Theta(\lambda_n)$, for any $\xi \in (0,1)$. Let $\boldsymbol{\eta} = \overline{\boldsymbol{\theta}}_u - \widetilde{\boldsymbol{\theta}}_u$ and now any point on the line can be written as $\widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta}$. The value of the objective at any point of minima is constant and we have

$$F(\widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta}) = c, \quad \xi \in (0,1),$$

where c is some constant. By taking the derivative with respect to ξ of $F(\widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta})$ we obtain

$$\sum_{t\in\mathcal{T}_n} w_t^\tau \left[-x_u^t + \frac{\exp(\langle \widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta}, \mathbf{x}_{\setminus u}^t\rangle) - \exp(-\langle \widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta}, \mathbf{x}_{\setminus u}^t\rangle)}{\exp(\langle \widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta}, \mathbf{x}_{\setminus u}^t\rangle) + \exp(-\langle \widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta}, \mathbf{x}_{\setminus u}^t\rangle)} \right] \langle \boldsymbol{\eta}, \mathbf{x}_{\setminus u}^t\rangle + \lambda_n \sum_{v=1}^{p-1} \eta_v \operatorname{sign}(\widetilde{\theta}_{uv} + \xi\eta_v) = 0. \tag{5.18}$$

On a small neighborhood of ξ the sign of $\widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta}$ is constant, for each component v, since the function $\widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta}$ is continuous in ξ. By taking the derivative with respect to ξ of (5.18) and noting that the last term is constant on a small neighborhood of ξ we have

$$4\sum_{t\in\mathcal{T}_n} w_t^\tau \langle \boldsymbol{\eta}, \mathbf{x}_{\setminus u}^t\rangle^2 \frac{\exp(-2\langle \widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta}, \mathbf{x}_{\setminus u}^t\rangle)}{\left(1 + \exp(-2\langle \widetilde{\boldsymbol{\theta}}_u + \xi\boldsymbol{\eta}, \mathbf{x}_{\setminus u}^t\rangle)\right)^2} = 0.$$

This implies that $\langle \boldsymbol{\eta}, \mathbf{x}_{\setminus u}^t\rangle = 0$ for every $t \in \mathcal{T}_n$, which implies that $\langle \mathbf{x}_{\setminus u}^t, \overline{\boldsymbol{\theta}}_u\rangle = \langle \mathbf{x}_{\setminus u}^t, \widetilde{\boldsymbol{\theta}}_u\rangle$, $t \in \mathcal{T}_n$, for any two solutions $\overline{\boldsymbol{\theta}}_u$ and $\widetilde{\boldsymbol{\theta}}_u$. Since $\overline{\boldsymbol{\theta}}_u$ and $\widetilde{\boldsymbol{\theta}}_u$ were two arbitrary elements of $\Theta(\lambda_n)$ we can conclude that $\langle \mathbf{x}_{\setminus u}^t, \boldsymbol{\theta}_u\rangle$, $t \in \mathcal{T}_n$ is constant for all elements $\boldsymbol{\theta}_u \in \Theta(\lambda_n)$.

Next, we need to show that the conclusion from above implies that any two solutions have non-zero elements in the same position. From equation (5.2), it follows that the set of non-zero components of the solution is given by

$$S = \left\{ 1 \le v \le p-1 \ : \ \left| \sum_{t\in\mathcal{T}_n} w_t^\tau (\nabla\gamma(\boldsymbol{\theta}_u; \mathbf{x}^t))_v \right| = \lambda \right\}.$$

Using equation (5.3) we have that

$$\sum_{t\in\mathcal{T}_n} w_t^\tau (\nabla\gamma(\boldsymbol{\theta}_u^\tau; \mathbf{x}^t))_v = \sum_{t\in\mathcal{T}_n} w_t^\tau (\mathbf{x}_{\setminus u}^t \{x_u^t + 1 - 2\frac{\exp(2x_u^t\langle \boldsymbol{\theta}_u^\tau, \mathbf{x}_{\setminus u}^t\rangle)}{\exp(2x_u^t\langle \boldsymbol{\theta}_u^\tau, \mathbf{x}_{\setminus u}^\tau\rangle) + 1}\})_v,$$

which is constant across different elements $\boldsymbol{\theta}_u \in \Theta(\lambda_n)$, since $\langle \mathbf{x}_{\setminus u}^t, \boldsymbol{\theta}_u\rangle$, $t \in \mathcal{T}_n$ is constant for all $\boldsymbol{\theta}_u \in \Theta(\lambda_n)$. This implies that the set of non-zero components is the same for all solutions. □

5.6.3 Proof of Lemma 5.2

Under the assumptions given in the Lemma, we can apply the result of Lemma 5.6. Let $\mathbf{y} \in \mathbb{R}^s$ be a unit norm minimal eigenvector of $\widehat{\boldsymbol{\Sigma}}_{SS}^\tau$. We have

$$\begin{aligned} \Lambda_{\min}(\boldsymbol{\Sigma}_{SS}^\tau) &= \min_{||\mathbf{x}||_2=1} \mathbf{x}'\boldsymbol{\Sigma}_{SS}^\tau\mathbf{x} \\ &= \min_{||\mathbf{x}||_2=1} \{\mathbf{x}'\widehat{\boldsymbol{\Sigma}}_{SS}^\tau\mathbf{x} + \mathbf{x}'(\boldsymbol{\Sigma}_{SS}^\tau - \widehat{\boldsymbol{\Sigma}}_{SS}^\tau)\mathbf{x}\} \\ &\le \mathbf{y}'\widehat{\boldsymbol{\Sigma}}_{SS}^\tau\mathbf{y} + \mathbf{y}'(\boldsymbol{\Sigma}_{SS}^\tau - \widehat{\boldsymbol{\Sigma}}_{SS}^\tau)\mathbf{y}, \end{aligned}$$

which implies

$$\Lambda_{\min}(\widehat{\mathbf{\Sigma}}^{\tau}_{SS}) \geq D_{\min} - |||(\mathbf{\Sigma}^{\tau}_{SS} - \widehat{\mathbf{\Sigma}}^{\tau}_{SS})|||_2.$$

Let $\mathbf{\Sigma}^{\tau} = [\sigma^{\tau}_{uv}]$ and $\widehat{\mathbf{\Sigma}}^{\tau} = [\widehat{\sigma}^{\tau}_{uv}]$. We have the following bound on the spectral norm

$$|||\mathbf{\Sigma}^{\tau}_{SS} - \widehat{\mathbf{\Sigma}}^{\tau}_{SS}|||_2 \leq \left(\sum_{u=1}^{s} \sum_{v=1}^{s} (\widehat{\sigma}^{\tau}_{uv} - \sigma^{\tau}_{uv})^2 \right)^{1/2} \leq \delta,$$

with the probability at least $1 - 2\exp(-Cnh(\frac{\delta}{s} - C'h)^2 + C''\log(s))$, for some fixed constants $C, C', C'' > 0$ depending on M and M_K only.

Similarly, we have that

$$\Lambda_{\max}(\widehat{\mathbf{\Sigma}}^{\tau}_{SS}) \leq D_{\max} + \delta,$$

with probability at least $1 - 2\exp(-Cnh(\frac{\delta}{s} - C'h)^2 + C''\log(s))$, for some fixed constants $C, C', C'' > 0$ depending on M and M_K only.

From Lemma 5.1, we know that any two solutions $\overline{\boldsymbol{\theta}}_u, \widetilde{\boldsymbol{\theta}}_u \in \Theta(\lambda_n)$ of the optimization problem (4.7) have non-zero elements in the same position. So, for any two solutions $\overline{\boldsymbol{\theta}}_u, \widetilde{\boldsymbol{\theta}}_u \in \Theta(\lambda_n)$, it holds

$$\mathbf{X}_{\setminus u}(\overline{\boldsymbol{\theta}}_u - \widetilde{\boldsymbol{\theta}}_u) = \mathbf{X}_{\setminus u,S}(\overline{\boldsymbol{\theta}}_u - \widetilde{\boldsymbol{\theta}}_u)_S + \mathbf{X}_{\setminus u,S^c}(\overline{\boldsymbol{\theta}}_u - \widetilde{\boldsymbol{\theta}}_u)_{S^c} = \mathbf{X}_{\setminus u,S}(\overline{\boldsymbol{\theta}}_u - \widetilde{\boldsymbol{\theta}}_u)_S.$$

Furthermore, from Lemma 5.1 we know that the two solutions are in the kernel of $\mathbf{X}_{\setminus u,S}$. On the event Ω_{01}, kernel of $\mathbf{X}_{\setminus u,S}$ is $\{0\}$. Thus, the solution is unique on Ω_{01}. □

5.6.4 Proof of Lemma 5.3

We first analyze the probability of the event Ω_{02}. Using the same argument to those in the proof of Lemma 5.2, we obtain

$$\Lambda_{\min}(\widehat{\mathbf{Q}}^{\tau}_{SS}) \geq C_{\min} - |||\mathbf{Q}^{\tau}_{SS} - \widehat{\mathbf{Q}}^{\tau}_{SS}|||_2.$$

Next, using results of Lemma 5.5, we have the following bound

$$|||\mathbf{Q}^{\tau}_{SS} - \widehat{\mathbf{Q}}^{\tau}_{SS}|||_2 \leq \left(\sum_{u=1}^{s} \sum_{v=1}^{s} (\widehat{q}^{\tau}_{uv} - q^{\tau}_{uv})^2 \right)^{1/2} \leq \delta, \tag{5.19}$$

with probability at least $1 - 2\exp(-C\frac{nh\delta^2}{s^2} + 2\log(s))$, for some fixed constants $C, C' > 0$ depending on M and M_K only.

Next, we deal with the event Ω_{03}. We are going to use the following decomposition

$$\begin{aligned}
\widehat{\mathbf{Q}}^{\tau}_{S^cS}(\widehat{\mathbf{Q}}^{\tau}_{SS})^{-1} &= \mathbf{Q}^{\tau}_{S^cS}[(\widehat{\mathbf{Q}}^{\tau}_{SS})^{-1} - (\mathbf{Q}^{\tau}_{SS})^{-1}] \\
&+ [\widehat{\mathbf{Q}}^{\tau}_{S^cS} - \mathbf{Q}^{\tau}_{S^cS}](\mathbf{Q}^{\tau}_{SS})^{-1} \\
&+ [\widehat{\mathbf{Q}}^{\tau}_{S^cS} - \mathbf{Q}^{\tau}_{S^cS}][(\widehat{\mathbf{Q}}^{\tau}_{SS})^{-1} - (\mathbf{Q}^{\tau}_{SS})^{-1}] \\
&+ \mathbf{Q}^{\tau}_{S^cS}(\mathbf{Q}^{\tau}_{SS})^{-1} \\
&= T_1 + T_2 + T_3 + T_4.
\end{aligned}$$

Under the assumption A2, we have that $\|T_4\|_\infty \leq 1-\alpha$. The lemma follows if we prove that for all the other terms we have $\|\cdot\|_\infty \leq \frac{\alpha}{6}$. Using the submultiplicative property of the norm, we have for the first term:

$$\begin{aligned}\|T_1\|_\infty &\leq \|\mathbf{Q}^\tau_{S^cS}\left(\mathbf{Q}^\tau_{SS}\right)^{-1}\|_\infty \|\widehat{\mathbf{Q}}^\tau_{SS} - \mathbf{Q}^\tau_{SS}\|_\infty \|(\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\|_\infty \\ &\leq (1-\alpha)\|\widehat{\mathbf{Q}}^\tau_{SS} - \mathbf{Q}^\tau_{SS}\|_\infty \sqrt{s}\|(\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\|_2.\end{aligned} \tag{5.20}$$

Using (5.19), we can bound the term $\|\left(\widehat{\mathbf{Q}}^\tau_{SS}\right)^{-1}\|_2 \leq C''$, for some constant depending on $C_{\min}$ only, with probability at least $1-2\exp(-C\frac{nh}{s}+2\log(s))$, for some fixed constant $C>0$. The bound on the term $\|\widehat{\mathbf{Q}}^\tau_{SS} - \mathbf{Q}^\tau_{SS}\|_\infty$ follows from application of Lemma 5.5. Observe that

$$\begin{aligned}\mathbb{P}[\|\widehat{\mathbf{Q}}^\tau_{SS} - \mathbf{Q}^\tau_{SS}\|_\infty \geq \delta] &= \mathbb{P}[\max_{v\in S}\{\sum_{v'\in S}|\widehat{q}^\tau_{vv'} - q^\tau_{vv'}|\} \geq \delta] \\ &\leq 2\exp(-Cnh(\frac{\delta}{s} - C'sh)^2 + 2\log(s)),\end{aligned} \tag{5.21}$$

for some fixed constants $C, C' > 0$. Combining all the elements, we obtain the bound on the first term $\|T_1\|_\infty \leq \frac{\alpha}{6}$, with probability at least $1 - C\exp(C'\frac{nh}{s^3} + C''\log(s))$, for some constants $C, C', C'' > 0$.

Next, we analyze the second term. We have that

$$\begin{aligned}\|T_2\|_\infty &\leq \|\widehat{\mathbf{Q}}^\tau_{S^cS} - \mathbf{Q}^\tau_{S^cS}\|_\infty \sqrt{s}\|\left(\mathbf{Q}^\tau_{SS}\right)^{-1}\|_2 \\ &\leq \frac{\sqrt{s}}{C_{\min}}\|\widehat{\mathbf{Q}}^\tau_{S^cS} - \mathbf{Q}^\tau_{S^cS}\|_\infty.\end{aligned} \tag{5.22}$$

The bound on the term $\|\widehat{\mathbf{Q}}^\tau_{SS} - \mathbf{Q}^\tau_{SS}\|_\infty$ follows in the same way as the bound in (5.21) and we can conclude that $\|T_3\|_\infty \leq \frac{\alpha}{6}$ with probability at least $1 - C\exp(C'\frac{nh}{s^3} + C''\log(p))$, for some constants $C, C', C'' > 0$.

Finally, we bound the third term T_3. We have the following decomposition

$$\begin{aligned}&\|[\widehat{\mathbf{Q}}^\tau_{S^cS} - \mathbf{Q}^\tau_{S^cS}][(\widehat{\mathbf{Q}}^\tau_{SS})^{-1} - (\mathbf{Q}^\tau_{SS})^{-1}]\|_\infty \\ &\quad\leq \|\widehat{\mathbf{Q}}^\tau_{S^cS} - \mathbf{Q}^\tau_{S^cS}\|_\infty \sqrt{s}\|(\mathbf{Q}^\tau_{SS})^{-1}[\mathbf{Q}^\tau_{SS} - \widehat{\mathbf{Q}}^\tau_{SS}](\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\|_2 \\ &\quad\leq \frac{\sqrt{s}}{C_{\min}}\|\widehat{\mathbf{Q}}^\tau_{S^cS} - \mathbf{Q}^\tau_{S^cS}\|_\infty \|\mathbf{Q}^\tau_{SS} - \widehat{\mathbf{Q}}^\tau_{SS}\|_2 \|(\widehat{\mathbf{Q}}^\tau_{SS})^{-1}\|_2.\end{aligned}$$

Bounding the remaining terms as in equations (5.22), (5.21) and (5.20), we obtain that $\|T_3\|_\infty \leq \frac{\alpha}{6}$ with probability at least $1 - C\exp(C'\frac{nh}{s^3} + C''\log(p))$.

Bound on the probability of event Ω_{03} follows from combining the bounds on all terms. □

5.6.5 Proof of Lemma 5.4

To prove this Lemma, we use a technique of [154] applied to the problem of consistency of the penalized covariance matrix estimator. Let us define the following function

$$H : \left\{\begin{array}{ccc}\mathbb{R}^p & \to & \mathbb{R} \\ \mathbf{D} & \mapsto & F(\boldsymbol{\theta}^\tau_u + \mathbf{D}) - F(\boldsymbol{\theta}^\tau_u),\end{array}\right.$$

where the function $F(\cdot)$ is defined in equation (5.1). The function $H(\cdot)$ takes the following form

$$H(\mathbf{D}) = \sum_{t\in\mathcal{T}_n} w_t^\tau(\gamma(\boldsymbol{\theta}_u^\tau;\mathbf{x}^t) - \gamma(\boldsymbol{\theta}_u^\tau + \mathbf{D};\mathbf{x}^t)) \\ + \ \lambda_n(||\boldsymbol{\theta}_u^\tau + \mathbf{D}||_1 - ||\boldsymbol{\theta}_u^\tau||_1).$$

Recall the minimizer of (4.7) constructed in the proof of Proposition 5.1, $\widehat{\boldsymbol{\theta}}_u^\tau = (\overline{\boldsymbol{\theta}}_S', 0_{S^c}')'$. The minimizer of the function $H(\cdot)$ is $\widehat{\mathbf{D}} = \widehat{\boldsymbol{\theta}}_u^\tau - \boldsymbol{\theta}_u^\tau$. Function $H(\cdot)$ is convex and $H(0) = 0$ by construction. Therefor $H(\widehat{\mathbf{D}}) \leq 0$. If we show that for some radius $B > 0$, and $\mathbf{D} \in \mathbb{R}^p$ with $||\mathbf{D}||_2 = B$ and $\mathbf{D}_{S^c} = \mathbf{0}$, we have $H(\mathbf{D}) > 0$, then we claim that $||\widehat{\mathbf{D}}||_2 \leq B$. This follows from the convexity of $H(\cdot)$.

We proceed to show strict positivity of $H(\cdot)$ on the boundary of the ball with radius $B = K\lambda_n\sqrt{s}$, where $K > 0$ is a parameter to be chosen wisely later. Let $\mathbf{D} \in \mathbb{R}^p$ be an arbitrary vector with $||\mathbf{D}||_2 = B$ and $\mathbf{D}_{S^c} = \mathbf{0}$, then by the Taylor expansion of $\gamma(\cdot;\mathbf{x}^t)$ we have

$$\begin{aligned} H(\mathbf{D}) \ &= \ -(\sum_{t\in\mathcal{T}_n} w_t^\tau \nabla\gamma(\boldsymbol{\theta}_u^\tau;\mathbf{x}^t))'\mathbf{D} \\ &\quad - \mathbf{D}'[\sum_{t\in\mathcal{T}_n} w_t^\tau \eta(\mathbf{x}^t;\boldsymbol{\theta}_u^\tau + \alpha\mathbf{D})\mathbf{x}_{\backslash u}^t \mathbf{x}_{\backslash u}^{t'}]\mathbf{D} \\ &\quad + \lambda_n(||\boldsymbol{\theta}_u^\tau + \mathbf{D}||_1 - ||\boldsymbol{\theta}_u^\tau||_1) \\ &= \ (I) + (II) + (III), \end{aligned} \tag{5.23}$$

for some $\alpha \in [0,1]$.

We start from the term (I). Let $\mathbf{e}_v \in \mathbb{R}^p$ be a unit vector with one at the position v and zeros elsewhere. Then random variables $-\mathbf{e}_v' \sum_{t\in\mathcal{T}_n} w_t^\tau \nabla\gamma(\boldsymbol{\theta}_u^\tau;\mathbf{x}^t)$ are bounded $[-\frac{C}{nh}, \frac{C}{nh}]$ for all $1 \leq v \leq p-1$, with constant $C > 0$ depending on M_K only. Using the Hoeffding inequality and the union bound, we have

$$\max_{1\leq v\leq p-1} |\mathbf{e}_v'(\sum_{t\in\mathcal{T}_n} w_t^\tau \nabla\gamma(\boldsymbol{\theta}_u^\tau;\mathbf{x}^t) - \mathbb{E}[\sum_{t\in\mathcal{T}_n} w_t^\tau \nabla\gamma(\boldsymbol{\theta}_u^\tau;\mathbf{x}^t)])| \leq \delta,$$

with probability at least $1 - 2\exp(-Cnh\delta^2 + \log(p))$, where $C > 0$ is a constant depending on M_K only. Moreover, denoting

$$p(\boldsymbol{\theta}_u^t) = \mathbb{P}_{\boldsymbol{\theta}_u^t}[x_u^t = 1 \mid \mathbf{x}_{\backslash u}^t]$$

to simplify the notation, we have for all $1 \leq v \leq p-1$,

$$\begin{aligned} &|\mathbb{E}[\mathbf{e}_v' \sum_{t\in\mathcal{T}_n} w_t^\tau \nabla\gamma(\boldsymbol{\theta}_u^\tau;\mathbf{x}^t) \mid \{\mathbf{x}_{\backslash u}^t\}_{t\in\mathcal{T}_n}]| \\ &= \ |\mathbb{E}[\sum_{t\in\mathcal{T}_n} w_t^\tau x_v^t[x_u^t + 1 - 2p(\boldsymbol{\theta}_u^\tau)] \mid \{\mathbf{x}_{\backslash u}^t\}_{t\in\mathcal{T}_n}]| \\ &= \ |2\sum_{t\in\mathcal{T}_n} w_t^\tau x_v^t[p(\boldsymbol{\theta}_u^t) - p(\boldsymbol{\theta}_u^\tau)]| \\ &\leq \ 4\int_{-\frac{1}{h}}^{0} K(z)|p(\boldsymbol{\theta}_u^{\tau+zh}) - p(\boldsymbol{\theta}_u^\tau)|dz. \end{aligned} \tag{5.24}$$

Next, we apply the mean value theorem on $p(\cdot)$ and the Taylor's theorem on $\boldsymbol{\theta}_u^t$. Under the assumption A3, we have

$$\begin{aligned}
&|p(\boldsymbol{\theta}_u^{\tau+zh}) - p(\boldsymbol{\theta}_u^{\tau})| \\
&\quad\leq \sum_{v=1}^{p-1} |\boldsymbol{\theta}_{uv}^{\tau+zh} - \theta_{uv}^{\tau}| \qquad (|\, p'(\cdot)| \leq 1\,) \\
&\quad= \sum_{v=1}^{p-1} |zh\frac{\partial}{\partial t}\theta_{uv}^t\Big|_{t=\tau} + \frac{(zh)^2}{2}\frac{\partial^2}{\partial t^2}\theta_{uv}^t\Big|_{t=\alpha_v}| \qquad (\,\alpha_v \in [\tau+zh, \tau]\,) \\
&\quad\leq Cs|zh + \frac{(zh)^2}{2}|,
\end{aligned} \tag{5.25}$$

for some $C > 0$ depending only on M. Combining (5.25) and (5.24) we have that

$$|\mathbb{E}[\mathbf{e}_v' \sum_{t\in\mathcal{T}_n} w_t^{\tau} \nabla\gamma(\boldsymbol{\theta}_u^{\tau}; \mathbf{x}^t)| \leq Csh$$

for all $1 \leq v \leq p-1$. Thus, with probability greater than

$$1 - 2\exp(-Cnh(\lambda_n - sh)^2 + \log(p))$$

for some constant $C > 0$ depending only on M_K, M and α, which under the conditions of Theorem 5.1 goes to 1 exponentially fast, we have

$$\max_{1\leq v\leq p-1} |\mathbf{e}_v' \sum_{t\in\mathcal{T}_n} w_t^{\tau} \nabla\gamma(\boldsymbol{\theta}_u^{\tau}; \mathbf{x}^t)| \leq \frac{\alpha\lambda_n}{4(2-\alpha)} < \frac{\lambda_n}{4}.$$

On that event, using Hölder's inequality, we have

$$\begin{aligned}
|(\sum_{t\in\mathcal{T}_n} w_t^{\tau} \nabla\gamma(\boldsymbol{\theta}_u^{\tau}; \mathbf{x}^t))'\mathbf{D}| &\leq ||\mathbf{D}||_1 \max_{1\leq v\leq p-1} |\mathbf{e}_v' \sum_{t\in\mathcal{T}_n} w_t^{\tau} \nabla\gamma(\boldsymbol{\theta}_u^{\tau}; \mathbf{x}^t)| \\
&\leq \frac{\lambda_n}{4}\sqrt{s}||\mathbf{D}||_2 \leq (\lambda_n\sqrt{s})^2\frac{K}{4}.
\end{aligned}$$

The triangle inequality applied to the term (III) of equation (5.23) yields:

$$\begin{aligned}
\lambda_n(||\boldsymbol{\theta}_u^{\tau} + \mathbf{D}||_1 - ||\boldsymbol{\theta}_u^{\tau}||_1) &\geq -\lambda_n||\mathbf{D}_S||_1 \\
&\geq -\lambda_n\sqrt{s}||\mathbf{D}_S||_2 \geq -K(\lambda_n\sqrt{s})^2.
\end{aligned}$$

Finally, we bound the term (II) of equation (5.23). Observe that since $\mathbf{D}_{S^c} = 0$, we have

$$\begin{aligned}
&\mathbf{D}'[\sum_{t\in\mathcal{T}_n} w_t^{\tau} \eta(\mathbf{x}^t; \boldsymbol{\theta}_u^{\tau} + \alpha\mathbf{D})\mathbf{x}_{\setminus u}^t \mathbf{x}_{\setminus u}^{t'}]\mathbf{D} \\
&\quad= \mathbf{D}_S'[\sum_{t\in\mathcal{T}_n} w_t^{\tau} \eta(\mathbf{x}^t; \boldsymbol{\theta}_u^{\tau} + \alpha\mathbf{D})\mathbf{x}_S^t \mathbf{x}_S^{t'}]\mathbf{D}_S \\
&\quad\geq K^2\Lambda_{\min}(\sum_{t\in\mathcal{T}_n} w_t^{\tau} \eta(\mathbf{x}^t; \boldsymbol{\theta}_u^{\tau} + \alpha\mathbf{D})\mathbf{x}_S^t \mathbf{x}_S^{t'})
\end{aligned}$$

Let $g : \mathbb{R} \mapsto \mathbb{R}$ be defined as $g(z) = \frac{4\exp(2z)}{(1+\exp(2z))^2}$. Now, $\eta(\mathbf{x}; \boldsymbol{\theta}_u) = g(x_u \langle \boldsymbol{\theta}_u, \mathbf{x}_{\setminus u} \rangle)$ and we have

$$\begin{aligned}
&\Lambda_{\min}(\sum_{t \in \mathcal{T}_n} w_t^\tau \eta(\mathbf{x}^t; \boldsymbol{\theta}_u^\tau + \alpha \mathbf{D}) \mathbf{x}_S^t \mathbf{x}_S^{t'}) \\
&\geq \min_{\alpha \in [0,1]} \Lambda_{\min}(\sum_{t \in \mathcal{T}_n} w_t \eta(\mathbf{x}^t; \boldsymbol{\theta}_u^\tau + \alpha \mathbf{D}) \mathbf{x}_S^t \mathbf{x}_S^{t'}) \\
&\geq \Lambda_{\min}(\sum_{t \in \mathcal{T}_n} w_t^\tau \eta(\mathbf{x}^t; \boldsymbol{\theta}_u^\tau) \mathbf{x}_S^t \mathbf{x}_S^{t'}) \\
&\quad - \max_{\alpha \in [0,1]} \| \sum_{t \in \mathcal{T}_n} w_t^\tau g'(x_u^t \langle \boldsymbol{\theta}_u^\tau + \alpha \mathbf{D}, \mathbf{x}_S^t \rangle)(x_u^t \mathbf{D}'_S \mathbf{x}_S^t) \mathbf{x}_S^t \mathbf{x}_S^{t'} \|_2 \\
&\geq C_{\min} - \max_{\alpha \in [0,1]} \| \sum_{t \in \mathcal{T}_n} w_t^\tau g'(x_u^t \langle \boldsymbol{\theta}_u^\tau + \alpha \mathbf{D}, \mathbf{x}_S^t \rangle)(x_u^t \mathbf{D}'_S \mathbf{x}_S^t) \mathbf{x}_S^t \mathbf{x}_S^{t'} \|_2
\end{aligned}$$

To bound the spectral norm, we observe that for any fixed $\alpha \in [0,1]$ and $y \in \mathbb{R}^s, ||\mathbf{y}||_2 = 1$ we have:

$$\begin{aligned}
&\mathbf{y}' \{ \sum_{t \in \mathcal{T}_n} w_t^\tau g'(x_u^t \langle \boldsymbol{\theta}_u^\tau + \alpha \mathbf{D}, \mathbf{x}_S^t \rangle)(x_u^t \mathbf{D}'_S \mathbf{x}_S^t) \mathbf{x}_S^t \mathbf{x}_S^{t'} \} \mathbf{y} \\
&= \sum_{t \in \mathcal{T}_n} w_t^\tau g'(x_u^t \langle \boldsymbol{\theta}_u^\tau + \alpha \mathbf{D}, \mathbf{x}_S^t \rangle)(x_u^t \mathbf{D}'_S \mathbf{x}_S^t)(\mathbf{x}_S^{t'} \mathbf{y})^2 \\
&\leq \sum_{t \in \mathcal{T}_n} w_t^\tau |g'(x_u^t \langle \boldsymbol{\theta}_u^\tau + \alpha \mathbf{D}, \mathbf{x}_S^t \rangle)(x_u^t \mathbf{D}'_S \mathbf{x}_S^t)| (\mathbf{x}_S^{t'} \mathbf{y})^2 \\
&\leq \sqrt{s} ||\mathbf{D}||_2 \| \sum_t w_t^\tau \mathbf{x}_S^t \mathbf{x}_S^{t'} \|_2 \qquad (\ |g'(\cdot)| \leq 1\ ) \\
&\leq D_{\max} K \lambda_n s \leq \frac{C_{\min}}{2}.
\end{aligned}$$

The last inequality follows as long as $\lambda_n s \leq \frac{C_{\min}}{2 D_{\max} K}$. We have shown that

$$\Lambda_{\min}(\sum_{t \in \mathcal{T}_n} w_t^\tau \eta(\mathbf{x}^t; \boldsymbol{\theta}_u^\tau + \alpha \mathbf{D}) \mathbf{x}_S^t \mathbf{x}_S^{t'}) \geq \frac{C_{\min}}{2},$$

with high probability.

Putting the bounds on the three terms together, we have

$$H(\mathbf{D}) \ \geq \ (\lambda_n \sqrt{s})^2 \left\{ -\frac{1}{4} K + \frac{C_{\min}}{2} K^2 - K \right\},$$

which is strictly positive for $K = \frac{5}{C_{\min}}$. For this choice of K, we have that $\lambda_n s \leq \frac{C_{\min}^2}{10 D_{\max}}$, which holds under the conditions of Theorem 5.1 for n large enough. $\square$

Chapter 6

Sparsistent Estimation Of Smoothly Varying Gaussian Graphical Models

The time-varying multivariate Gaussian distribution and the undirected graph associated with it, as introduced in [206], provide a useful statistical framework for modeling complex dynamic networks. In many application domains, it is of high importance to estimate the graph structure of the model consistently for the purpose of scientific discovery. In this chapter, we show that under suitable technical conditions, the structure of the undirected graphical model can be consistently estimated in the high dimensional setting, when the dimensionality of the model is allowed to diverge with the sample size. The model selection consistency is shown for the procedure proposed in [206] and for the modified neighborhood selection procedure of [135].

6.1 Preliminaries

In this chapter, we study consistent graph structure estimation in a time-varying Gaussian graphical model [206]. Let

$$\mathbf{x}^t \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}^t\right), \quad t \in \mathcal{T}_n = \{1/n, 2/n, \ldots, 1\} \tag{6.1}$$

be an independent sequence of p-dimensional observations distributed according to a multivariate Gaussian distribution whose covariance matrix changes smoothly over time. A graph $G^t = (V, E^t)$ is associated with each observation $\mathbf{x}^t$ and it represents the non-zero elements of the precision matrix $\mathbf{\Omega}^t = (\mathbf{\Sigma}^t)^{-1}$ (recall that $e_{ab} \in E^t$ only if $\omega_{ab}^t \neq 0$). With changing precision matrix $\mathbf{\Omega}^t$, the associated graphs change as well, which allows for modelling of dynamic networks. The model given in (6.1) can be thought of as a special case of the varying coefficient models introduced in [96]. In particular, the model in (6.1), inherits flexibility and modelling power from the class of nonparametric models, but at the same time it retains interpretability of parametric models. Indeed, there are no assumptions on the parametric form of the elements of the covariance matrix $\mathbf{\Sigma}^t$ as a function of time.

Under the model (6.1), [206] studied the problem of the consistent recovery in the Frobenius norm of $\mathbf{\Omega}^\tau$ for some $\tau \in [0, 1]$, as well as the predictive performance of the fitted model. While those results are very interesting and important in statistics, in many application areas, it is the

graph structure that provides most insight into complex systems by allowing visualiziation of relational structures and mechanisms that explain the data. For example, in computational biology, a graph estimated from a gene expression microarray profile can reveal the topology of genetic regulation circuitry, while in sociocultural analysis, a graph structure helps identify communities and communication patterns among actors. Unfortunately, the consistent estimation of the graph structure does not follow immediately from the consistent estimation of the precision matrix $\mathbf{\Omega}$. We address the problem of the consistent graph structure recovery under the model (6.1). Our work has applications in many disciplines, including computational biology and computational finance, where the assumptions that the data are distributed i.i.d. are not satisfied. For example, a gene regulatory network is assumed to change throughout the developmental process of the organism, and a plausible way to model the longitudinal gene expression levels is by using the multivariate Gaussian distribution with a time-evolving structure.

The main contributions of the chapter include establishing sufficient conditions for the penalized likelihood procedure, proposed in [206], to estimate the graph structure consistently. Furthermore, we modify the neighborhood selection procedure of [135] to estimate the graph structure under the model (6.1) and provide sufficient conditions for the graph recovery.

6.2 Penalized likelihood estimation

In this section, we show that, under some technical conditions, the procedure proposed in [206] is able to consistently estimate the set of non-zero elements of the precision matrix $\mathbf{\Omega}^\tau$ at a given time point $\tau \in [0, 1]$. Under the model (6.1), an estimator of the precision matrix can be obtained by minimizing the following objective

$$\widehat{\mathbf{\Omega}}^\tau = \underset{\mathbf{\Omega} \succ 0}{\operatorname{argmin}} \left\{ \operatorname{tr} \mathbf{\Omega} \widehat{\mathbf{\Sigma}}^\tau - \log |\mathbf{\Omega}| + \lambda ||\mathbf{\Omega}^-||_1 \right\}, \tag{6.2}$$

where $\mathbf{\Omega}^-$ has off-diagonal elements equal to those of $\mathbf{\Omega}$ and diagonal elements equal to zero, $\widehat{\mathbf{\Sigma}}^\tau = \sum_{t \in \mathcal{T}_n} w_t^\tau \mathbf{x}^t (\mathbf{x}^t)'$ is the weighted sample covariance matrix, with weights defined as

$$w_t^\tau = \frac{K_h(t - \tau)}{\sum_{t \in \mathcal{T}_n} K_h(t - \tau)}, \tag{6.3}$$

$K : \mathbb{R} \mapsto \mathbb{R}$ being the kernel function and $K_h(\cdot) = K(\cdot/h)$. Note that (6.2) extends the penalized maximum likelihood estimation procedure given (2.6) for learning network structure from i.i.d. data. The tuning parameter λ controls the number of non-zero pattern of the estimated precision matrix, while the bandwidth parameter h controls the smoothness over time of the estimated precision matrix and the effective sample size. These tuning parameters depend on the sample size n, but we will omit this dependence in our notation. In practice, the parameters are chosen using standard model selection techniques in data dependent way, for example, using cross-validation or Bayesian information criterion. The kernel K is taken such that the following set of assumptions holds.

Assumption K: The kernel $K : \mathbb{R} \mapsto \mathbb{R}$ is symmetric, supported in $[-1, 1]$ and there exists a constant $M_K \geq 1$ which upper bounds the quantities $\max_{x \in \mathbb{R}} |K(x)|$ and $\max_{x \in \mathbb{R}} K(x)^2$.
For example, the assumption **K** is satisfied by the box kernel $K(x) = \frac{1}{2} \mathbb{1}\{x \in [-1, 1]\}$.

A similar estimator to the one given in (6.2) is analyzed in [206] and the convergence rate is established for $||\widehat{\mathbf{\Omega}}^\tau - \mathbf{\Omega}^\tau||_F$. However, establishing that the estimated edge set

$$\widehat{E}^\tau = \{(a,b) \mid a \neq b, \widehat{\omega}^\tau_{ab} \neq 0\}$$

consistently estimates the true edge set $E^t = \{(a,b) \mid a \neq b, \omega^t_{ab} \neq 0\}$ is a harder problem, which requires stronger conditions on the true model. Let $s = \max_i |E^{t_i}|$ denote the maximum number of edges in a graph and $d = \max_{t \in \mathcal{T}_n} \max_{a \in V} |\{b \in V \mid a \neq b, e_{ab} \in E^t\}|$ the maximum node degree. In the remainder of this section, we provide sufficient conditions on (n, p, d, h, λ) under which the estimator given by (6.2) recovers the graph structure with high probability. To that end, we will use some of the results established in [152].

We start by imposing some assumptions on the true model. The first assumption assures that the covariance matrix is not singular at any time point. Note that if the population covariance matrix was singular, the problem of recovering the true graph structure would be ill-defined, since there would be no unique graph structure associated with the probability distribution.

Assumption C: There exist constants $\Lambda_{\max}, M_\infty < \infty$ such that for all $t \in \mathcal{T}_n$ we have

$$\frac{1}{\Lambda_{\max}} \leq \Lambda_{\min}(\mathbf{\Sigma}^t) \leq \Lambda_{\max}(\mathbf{\Sigma}^t) \leq \Lambda_{\max} \quad \text{and} \quad |||\mathbf{\Sigma}^t|||_{\infty,\infty} \leq M_\infty.$$

Furthermore, we assume that $\sigma^\tau_{aa} = 1$ for all $a \in V$.

The next assumption captures the notion of the distribution changing smoothly over time.

Assumption S: Let $\mathbf{\Sigma}^t = (\sigma^t_{ab})$. There exists a constant $M_\Sigma > 0$ such that

$$\max_{a,b} \sup_{\tau \in [0,1]} |\dot{\sigma}^\tau_{ab}| \leq M_\Sigma, \text{ and}$$
$$\max_{a,b} \sup_{\tau \in [0,1]} |\ddot{\sigma}^\tau_{ab}| \leq M_\Sigma,$$

where $\dot{\sigma}^t_{ab}$ and $\ddot{\sigma}^t_{ab}$ denote the first and second derivative with respect to time.

Assumptions similar to **C** and **S** are also imposed in [206] in order to show consistency in the Frobenius norm. In particular, the rate of the convergence of $||\widehat{\mathbf{\Omega}}^\tau - \mathbf{\Omega}^\tau||_F$ depends on the quantities $\Lambda_{\max}, M_\infty$ and M_Σ. Assumption **S** captures our notion of a distribution that is smoothly changing over time and together with assumption **C** guarantees that the precision matrix $\mathbf{\Omega}^t$ changes smoothly over time as well. The common variance of the components is assumed for presentation simplicity and can be obtained through scaling.

Assumptions **C** and **S** are not enough to guarantee recovery of the non-zero pattern of the population precision matrix $\mathbf{\Omega}^\tau$. From the previous work on variable selection in generalized linear models (see, for example, [63], [151], [12]) we know that additional assumptions are needed on the Fisher information matrix in order to guarantee consistent model identification. In the case of the multivariate Gaussian distribution the Fisher information matrix at time $\tau \in [0,1]$ is given as

$$\mathcal{I}^\tau = \mathcal{I}(\mathbf{\Omega}^\tau) = (\mathbf{\Omega}^\tau)^{-1} \otimes (\mathbf{\Omega}^\tau)^{-1},$$

where $\otimes$ denotes the Kronecker product. The elements of the Fisher information matrix can be also expressed as $\mathcal{I}^\tau_{(a,b),(a',b')} = \text{Corr}(X^\tau_a X^\tau_b, X^\tau_{a'} X^\tau_{b'})$. Let $S = S^\tau = E^\tau \bigcup \{(a,a)\}_{a \in V}$ be an

index set of the non-zero elements of $\mathbf{\Omega}^\tau$ and S^C denotes its complement in $V \times V$. Let $\mathcal{I}^\tau_{SS}$ denote the $|S| \times |S|$ sub-matrix of $\mathcal{I}^\tau$ indexed by elements of S.

Assumption F: The sub-matrix $\mathcal{I}_{SS}$ is invertible. There exist constants $\alpha \in (0, 1]$ and $M_{\mathcal{I}} < \infty$ such that

$$|||\mathcal{I}^\tau_{S^C S}(\mathcal{I}^\tau_{SS})^{-1}|||_{\infty,\infty} \leq 1 - \alpha \quad \text{and} \quad |||(\mathcal{I}^\tau_{SS})^{-1}|||_{\infty,\infty} \leq M_{\mathcal{I}}.$$

The assumption **F** is identical to the assumptions made in [152]. We need to assume that it holds only for the time point of interest τ at which the precision matrix is being estimated.

With these assumptions, we have the following result.

Theorem 6.1. *Fix a time point of interest $\tau \in [0, 1]$. Let $\{\mathbf{x}^t\}_{t \in \mathcal{T}_n}$ be an independent sample according to the model* (6.1)*. Under the assumptions* **C**, **S**, **F** *and* **K** *there exists a constant $C > 0$ depending only on $\Lambda_{\max}, M_\infty, M_\Sigma, M_K, M_{\mathcal{I}}$ and α for which the following holds. Suppose that the weighted sample covariance matrix $\widehat{\mathbf{\Sigma}}^\tau$ is estimated using the kernel with the bandwidth parameter satisfying $h = \mathcal{O}\left(n^{-1/3}\right)$. If the penalty parameter λ in* (6.2) *scales as $\lambda = \mathcal{O}\left(n^{-1/3}\sqrt{\log p}\right)$ and the sample size satisfies $n > Cd^3(\log p)^{3/2}$, then the minimizer $\widehat{\mathbf{\Omega}}^\tau$ of* (6.2) *defines the edge set $\widehat{E}^\tau$ which satisfies*

$$\mathbb{P}[\widehat{E}^\tau \neq \{(a, b) \mid a \neq b, |\omega^\tau_{ab}| > \omega_{\min}\}] = \mathcal{O}(\exp(-c \log p)) \to 0,$$

for some constant $c > 0$, with $\omega_{\min} = M_\omega n^{-1/3}\sqrt{\log p}$ and M_ω being a sufficiently large constant.

The theorem states that all the non-zero elements of the population precision matrix $\mathbf{\Omega}^\tau$, which are larger in absolute value than $\omega_{\min}$, will be identified. Note that if the elements of the precision matrix are too small, then the estimation procedure is not able to distinguish them from zero. Furthermore, the estimation procedure does not falsely include zero elements into the estimated set of edges. The theorem guarantees consistent recovery of the set of sufficiently large non-zero elements of the precision matrix at the time point τ. In order to obtain insight into the network dynamics, the graph corresponding to $\mathbf{\Omega}^t$ needs to be estimated at multiple time points. Due to the slow rate of convergence of $\widehat{\mathbf{\Omega}}^t$, it is sufficient to estimate a graph at each time point $t \in \mathcal{T}_n$.

Comparing Theorem 6.1 to the results on the static graph structure estimation [152], we can observe a slower rate of convergence. The difference arises from the fact that using the kernel estimate, we effectively use only the sample that is "close" to the time point τ. Using a local linear smoother, instead of the kernel smoother to reduce the bias in the estimation, a better dependence on the sample size could be obtained. Finally we note that, for simplicity and ease of interpretation, Theorem 6.1 is stated without providing explicit dependence of the rate of convergence on the constants appearing in the assumptions.

6.2.1 Proof of Theorem 6.1

The proof of the theorem will be separated into several propositions to facilitate the exposition. Technical lemmas and some proofs are deferred to the end of chapter. Our proof uses some ideas introduced in [152].

We start by introducing the following function

$$G(\mathbf{\Omega}) = \operatorname{tr} \mathbf{\Omega}\widehat{\mathbf{\Sigma}}^{\tau} - \log|\mathbf{\Omega}| + \lambda||\mathbf{\Omega}^{-}||_1, \quad \forall \mathbf{\Omega} \succ 0$$

and we say that $\mathbf{\Omega} \in \mathbb{R}^{p\times p}$ satisfies the system $(\mathcal{S})$ when $\forall\, a \neq b \in V \times V$,

$$\begin{array}{rl} (\widehat{\mathbf{\Sigma}}^{\tau})_{ab} - (\mathbf{\Omega}^{-1})_{ab} = -\lambda \operatorname{sign}((\mathbf{\Omega}^{-1})_{ab}), & \text{if } (\mathbf{\Omega}^{-1})_{ab} \neq 0 \\ |(\widehat{\mathbf{\Sigma}^{\tau}})_{ab} - (\mathbf{\Omega}^{-1})_{ab}| \leq \lambda, & \text{if } (\mathbf{\Omega}^{-1})_{ab} = 0. \end{array} \tag{6.4}$$

It is known that $\mathbf{\Omega} \in \mathbb{R}^{p\times p}$ is the minimizer of Equation (6.2) if and only if it satisfies the system $(\mathcal{S})$. Since $G(\mathbf{\Omega})$ is strictly convex, the minimum, if attained, is unique. The assumption **C** guarantees that the minimum is attained. Therefore, we do not have to worry about the possibility of having several $\mathbf{\Omega}$ satisfying the system $(\mathcal{S})$.

Recall that we use the set S to index the non-zero elements of the population precision matrix. Without loss of generality we write

$$\mathcal{I} = \begin{pmatrix} \mathcal{I}_{SS} & \mathcal{I}_{SS^C} \\ \mathcal{I}_{S^CS} & \mathcal{I}_{S^CS^C} \end{pmatrix}, \quad \vec{\mathbf{\Sigma}} = \begin{pmatrix} \vec{\mathbf{\Sigma}}_S \\ \vec{\mathbf{\Sigma}}_{S^C} \end{pmatrix}.$$

Let $\mathbf{\Omega} = \mathbf{\Omega}^{\tau} + \mathbf{\Delta}$. Using the first-order Taylor expansion of the function $g(\mathbf{X}) = \mathbf{X}^{-1}$ around $\mathbf{\Omega}^{\tau}$ we have

$$\mathbf{\Omega}^{-1} = (\mathbf{\Omega}^{\tau})^{-1} - (\mathbf{\Omega}^{\tau})^{-1}\mathbf{\Delta}(\mathbf{\Omega}^{\tau})^{-1} + R(\mathbf{\Delta}), \tag{6.5}$$

where $R(\mathbf{\Delta})$ denotes the remainder term. We consider the following two events

$$\mathcal{E}_1 \;=\; \left\{|(\mathcal{I}_{SS})^{-1}[(\vec{\widehat{\mathbf{\Sigma}}}^{\tau} - \vec{\mathbf{\Sigma}}^{\tau}) - \overrightarrow{R(\mathbf{\Delta})} + \lambda\overrightarrow{\operatorname{sign}}(\mathbf{\Omega}^{\tau})]_S| < \omega(n,p)\right\}$$

and

$$\mathcal{E}_2 = \left\{|\mathcal{I}_{S^CS}(\mathcal{I}_{SS})^{-1}[(\vec{\mathbf{\Sigma}}^{\tau} - \vec{\widehat{\mathbf{\Sigma}}}^{\tau}) + \overrightarrow{R(\mathbf{\Delta})}]_S + (\vec{\widehat{\mathbf{\Sigma}}}^{\tau} - \vec{\mathbf{\Sigma}}^{\tau})_{S^C} - \overrightarrow{R(\mathbf{\Delta})}_{S^C}| < \alpha\lambda\right\},$$

where, in both events, inequalities hold element-wise.

Proposition 6.1. *Under the assumptions of Theorem 6.1, the event*

$$\left\{\widehat{\mathbf{\Omega}}^{\tau} \in \mathbb{R}^{p\times p} \textit{ minimizer of } (6.2), \operatorname{sign}(\widehat{\omega}_{ab}) = \operatorname{sign}(\omega^{\tau}_{ab}) \textit{ for all } |\omega_{ab}| \notin (0, \omega_{\min})\right\}$$

contains the event $\mathcal{E}_1 \bigcap \mathcal{E}_2$.

Proof. We start by manipulating the conditions given in (6.4). Using (6.5) and using the fact that $\operatorname{vec}((\mathbf{\Omega}^{\tau})^{-1}\mathbf{\Delta}(\mathbf{\Omega}^{\tau})^{-1}) = ((\mathbf{\Omega}^{\tau})^{-1} \otimes (\mathbf{\Omega}^{\tau})^{-1})\vec{\mathbf{\Delta}} = \mathcal{I}\vec{\mathbf{\Delta}}$, we can rewrite (6.4) in the equivalent form

$$\begin{array}{c} (\mathcal{I}\vec{\mathbf{\Delta}})_S + (\vec{\widehat{\mathbf{\Sigma}}}^{\tau} - \vec{\mathbf{\Sigma}}^{\tau})_S - (\overrightarrow{R(\mathbf{\Delta})})_S = -\lambda(\overrightarrow{\operatorname{sign}}(\mathbf{\Omega}))_S \\ |(\mathcal{I}\vec{\mathbf{\Delta}})_{S^C} + (\vec{\widehat{\mathbf{\Sigma}}}^{\tau} - \vec{\mathbf{\Sigma}}^{\tau})_{S^C} - (\overrightarrow{R(\mathbf{\Delta})})_{S^C}| \leq \lambda\, \mathbb{1}_{S^C}, \end{array} \tag{6.6}$$

where $\mathbb{1}_{S^C}$ is the vector of the form $(1, 1, \ldots, 1)'$ and the equations hold element-wise. Now consider the following linear functional, $F : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$,

$$\boldsymbol{\theta} \mapsto \boldsymbol{\theta} - \vec{\mathbf{\Omega}}^{\tau}_S + (\mathcal{I}_{SS})^{-1}\left[(\vec{\widehat{\mathbf{\Sigma}}}^{\tau} - \vec{\mathbf{\Sigma}}^{\tau}) - \overrightarrow{R(\mathbf{\Delta})}\right]_S + \lambda(\mathcal{I}_{SS})^{-1}\overrightarrow{\operatorname{sign}}(\boldsymbol{\theta}).$$

For any two vectors $\mathbf{x} = (x_1, \ldots, x_{|S|})' \in \mathbb{R}^{|S|}$ and $\mathbf{r} = (r_1, \ldots, r_{|S|})' \in \mathbb{R}_+^{|S|}$, define the set

$$\mathcal{B}(\mathbf{x}, \mathbf{r}) = \prod_{i=1}^{|S|} (x_i - r_i, x_i + r_i).$$

Now, we have

$$F(\mathcal{B}(\vec{\mathbf{\Omega}}_S^\tau, \omega_{\min})) = \mathcal{B}\big((\mathcal{I}_{SS})^{-1}[(\vec{\hat{\mathbf{\Sigma}}}^\tau - \vec{\mathbf{\Sigma}}^\tau) - \overrightarrow{R(\mathbf{\Delta})}]_S + \lambda(\mathcal{I}_{SS})^{-1}\overrightarrow{\text{sign}}(\mathbf{\Omega}_S^\tau), \omega_{\min}\big) = \mathcal{H}.$$

On the event $\mathcal{E}_1$, we have $\mathbf{0} \in \mathcal{H}$ and hence there exists $\overline{\vec{\mathbf{\Omega}}}_S \in \mathcal{B}(\vec{\mathbf{\Omega}}_S^\tau, \omega_{\min})$ such that $F(\overline{\vec{\mathbf{\Omega}}}_S) = \mathbf{0}$. Thus we have $\text{sign}(\overline{\omega}_{ab}) = \text{sign}(\omega_{ab}^\tau)$ for all elements $(a, b) \in S$ such that $|\omega_{ab}^\tau| > \omega_{\min}$ and

$$\mathcal{I}_{SS}\vec{\mathbf{\Delta}}_S + (\vec{\hat{\mathbf{\Sigma}}} - \vec{\mathbf{\Sigma}})_S - (\overrightarrow{R(\mathbf{\Delta})})_S = -\lambda(\overrightarrow{\text{sign}}(\overline{\mathbf{\Omega}}))_S. \tag{6.7}$$

Under the assumption on the Fisher information matrix $\mathbf{F}$ and on the event $\mathcal{E}_2$ it holds

$$\begin{aligned} -\lambda\, \mathbb{1}_{S^C} &< \mathcal{I}_{S^C S}\vec{\mathbf{\Delta}}_S + \left(\vec{\hat{\mathbf{\Sigma}}}^\tau - \vec{\mathbf{\Sigma}}^\tau\right)_{S^C} - \left(\overrightarrow{R(\mathbf{\Delta})}\right)_{S^C} \\ &= \mathcal{I}_{S^C S}(\mathcal{I}_{SS})^{-1}\left[(\vec{\mathbf{\Sigma}}^\tau - \vec{\hat{\mathbf{\Sigma}}}^\tau) + \overrightarrow{R(\mathbf{\Delta})}\right]_S + \left(\vec{\hat{\mathbf{\Sigma}}}^\tau - \vec{\mathbf{\Sigma}}^\tau\right)_{S^C} - \left(\overrightarrow{R(\mathbf{\Delta})}\right)_{S^C} \\ &\quad + \lambda\mathcal{I}_{S^C S}(\mathcal{I}_{SS})^{-1}\left(\overrightarrow{\text{sign}}\left(\overline{\mathbf{\Omega}}\right)\right)_S \\ &< \lambda\, \mathbb{1}_{S^C}\,. \end{aligned} \tag{6.8}$$

Now, we consider the vector $\overline{\vec{\mathbf{\Omega}}} = \begin{pmatrix} \overline{\vec{\mathbf{\Omega}}}_S \\ \vec{\mathbf{0}}_{S^C} \end{pmatrix} \in \mathbb{R}^{p^2}$. Note that for $\overline{\mathbf{\Omega}}$, equations (6.7) and (6.8) are equivalent to saying that $\overline{\mathbf{\Omega}}$ satisfies conditions (6.6) or (6.4), that is, saying that $\overline{\mathbf{\Omega}}$ satisfies the system $(\mathcal{S})$. We have that $\text{sign}(\overline{\omega}_{ab}) = \text{sign}(\omega_{ab}^\tau)$ for all (a, b) such that $|\omega_{ab}^\tau| \notin (0, \omega_{\min})$. Furthermore the solution to (6.2) is unique. □

Using Proposition 6.1, Theorem 6.1 follows if we show that events $\mathcal{E}_1$ and $\mathcal{E}_2$ occur with high probability. The following two propositions state that the events $\mathcal{E}_1$ and $\mathcal{E}_2$ occur with high probability.

Proposition 6.2. *Under the assumptions of Theorem 6.1, there exist constants $C_1, C_2 > 0$ depending on $\Lambda_{\max}$, M_∞, M_Σ, M_K, M_ω, $M_\mathcal{I}$ and α such that $\mathbb{P}[\mathcal{E}_1] \geq 1 - C_1 \exp(-C_2 \log p)$.*

Proof. We will perform analysis on the event

$$\mathcal{A} = \left\{ ||\widehat{\mathbf{\Sigma}}^\tau - \mathbf{\Sigma}^\tau||_\infty \leq \frac{\alpha\lambda}{8} \right\}. \tag{6.9}$$

Under the assumptions of the proposition, it follows from Lemma 11 in [105] that

$$\mathbb{P}[\mathcal{A}] \geq 1 - C_1 \exp(-C_2 \log p).$$

Also, under the assumptions of the proposition, Lemma can be applied to conclude that $R(\mathbf{\Delta}) \leq \frac{\alpha\lambda}{8}$. Let $e_j \in \mathbb{R}^{|S|}$ be a unit vector with 1 at position j and zeros elsewhere. On the event $\mathcal{A}$, it holds that

$$\begin{aligned}
&\max_{1\leq j\leq |S|} |e_j'(\mathcal{I}_{SS})^{-1}[(\vec{\hat{\mathbf{\Sigma}}}^{\tau} - \vec{\mathbf{\Sigma}}^{\tau}) - \overrightarrow{R(\mathbf{\Delta})} + \lambda\overrightarrow{\text{sign}}(\mathbf{\Omega}^{\tau})]_S| \\
&\qquad \leq |||(\mathcal{I}_{SS})^{-1}|||_{\infty,\infty}\big(||(\vec{\hat{\mathbf{\Sigma}}}^{\tau} - \vec{\mathbf{\Sigma}}^{\tau})_S||_\infty + ||\overrightarrow{R(\mathbf{\Delta})}_S||_\infty + \lambda||\overrightarrow{\text{sign}}(\mathbf{\Omega}^{\tau}_S)||_\infty\big) \\
&\qquad\quad \text{(using the Hölder's inequality)} \\
&\qquad \leq M_{\mathcal{I}}\frac{4+\alpha}{4}\lambda \leq C\frac{\sqrt{\log p}}{n^{1/3}} < \omega_{\min} = M_\omega\frac{\sqrt{\log p}}{n^{1/3}},
\end{aligned}$$

for a sufficiently large constant M_ω. □

Proposition 6.3. *Under the assumptions of Theorem 6.1, there exist $C_1, C_2 > 0$ depending on $\Lambda_{\max}$, M_∞, M_Σ, M_K, $M_{\mathcal{I}}$ and α such that $\mathbb{P}[\mathcal{E}_2] \geq 1 - C_1\exp(-C_2\log p)$.*

Proof. We will work on the event $\mathcal{A}$ defined in (6.9). Under the assumptions of the proposition, Lemma 12 in [105] gives $R(\mathbf{\Delta}) \leq \frac{\alpha\lambda}{8}$. Let $e_j \in \mathbb{R}^{p^2-|S|}$ be a unit vector with 1 at position j and zeros elsewhere. On the event $\mathcal{A}$, it holds that

$$\begin{aligned}
&\max_{1\leq j\leq (p^2-|S|)} \Big|e_j'\big(\mathcal{I}_{S^CS}(\mathcal{I}_{SS})^{-1}[(\vec{\mathbf{\Sigma}}^{\tau} - \vec{\hat{\mathbf{\Sigma}}}^{\tau}) + \overrightarrow{R(\mathbf{\Delta})}]_S + (\vec{\hat{\mathbf{\Sigma}}}^{\tau} - \vec{\mathbf{\Sigma}}^{\tau})_{S^C} - \overrightarrow{R(\mathbf{\Delta})}_{S^C}\big)\Big| \\
&\leq |||\mathcal{I}_{S^CS}(\mathcal{I}_{SS})^{-1}|||_{\infty,\infty}\left(||\vec{\mathbf{\Sigma}}^{\tau} - \vec{\hat{\mathbf{\Sigma}}}^{\tau}||_\infty + ||\overrightarrow{R(\mathbf{\Delta})}||_\infty\right) + ||\vec{\hat{\mathbf{\Sigma}}}^{\tau} - \vec{\mathbf{\Sigma}}^{\tau}||_\infty + ||\overrightarrow{R(\mathbf{\Delta})}||_\infty \\
&\leq (1-\alpha)\frac{\alpha\lambda}{4} + \frac{\alpha\lambda}{4} \leq \alpha\lambda,
\end{aligned}$$

which concludes the proof. □

Theorem 6.1 follows from Propositions 6.1, 6.2 and 6.3.

6.3 Neighborhood selection estimation

In this section, we discuss the neighborhood selection approach to selection of non-zero elements of the precision matrix $\mathbf{\Omega}^{\tau}$ under the model (6.1). The neighborhood selection procedure was proposed in [135] as a way to estimate the graph structure associated to a GGM from an i.i.d. sample. The method was applied to learn graph structure in more general settings as well (see, for example, [76, 112, 146, 151]). As opposed to optimizing penalized likelihood, the neighborhood selection method is based on optimizing penalized pseudo-likelihood on each node of the graph, which results in local estimation of the graph structure. While the procedure is very scalable and suitable for large problems, it does not result in consistent estimation of the precision matrix. On the other hand, as we will show, the non-zero pattern of the elements of the precision matrix can be recovered under weaker assumptions.

We start by describing the neighborhood selection method under the model (6.1). Here, we modify As mentioned in the introduction, the elements of the precision matrix are related to the

partial correlation coefficients as $\rho_{ab}^t = -\omega_{ab}^t/\sqrt{\omega_{aa}^t\omega_{bb}^t}$. A well known result [130] relates the partial correlation coefficients to a regression model where a variable X_a is regressed onto the rest of variables $\mathbf{X}_{\setminus a}$,

$$X_a = \sum_{b\in V\setminus\{a\}} X_b\theta_{ab}^t + \epsilon_a^t, \quad a \in V.$$

In the equation above, ϵ_a^t is independent of $\mathbf{X}_{\setminus a}$ if and only if $\theta_{ab}^t = \rho_{ab}^t\sqrt{\omega_{aa}^t/\omega_{bb}^t}$. The relationship between the elements of the precision matrix and the least square regression immediately suggests the following estimator for $\boldsymbol{\theta}_{\setminus a}^\tau = \{\theta_{ab}^\tau\}_{b\in V\setminus\{a\}}$,

$$\widehat{\boldsymbol{\theta}}_{\setminus a}^\tau = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{p-1}} \sum_{t\in\mathcal{T}_n}(x_a^t - \sum_{b\neq a} x_b^t\theta_b)^2 w_t^\tau + \lambda||\boldsymbol{\theta}||_1, \tag{6.10}$$

where the weight w_t^τ are defined in (6.3). Note how (6.10) modifies the objective in (2.5) to estimate changing neighborhoods. The estimator $\widehat{\boldsymbol{\theta}}_{\setminus a}^\tau$ defines the neighborhood of the node $a \in V$ at the time point τ as $\widehat{S}_a^\tau = S(\widehat{\boldsymbol{\theta}}_{\setminus a}^\tau)$. By estimating the neighborhood of each node and combining them, the whole graph structure can be obtained. There are two natural ways to combine the estimated neighborhoods, using the union, $\widehat{E}^{\tau,\cup} = \{(a,b) \mid b \in \widehat{S}_a^\tau \vee a \in \widehat{S}_b^\tau\}$, or intersection of different neighborhoods, $\widehat{E}^{\tau,\cap} = \{(a,b) \mid b \in \widehat{S}_a^\tau \wedge a \in \widehat{S}_b^\tau\}$. Asymptotically these two approaches are equivalent and we will denote the resulting set of edges as $\widehat{E}^\tau$.

The consistency of the graph estimation for the neighborhood selection procedure will be proven under similar assumptions to those of Theorem 6.1. However, the assumption **F** can be relaxed. Let $S = S_a^\tau = S(\boldsymbol{\theta}_{\setminus a}^\tau)$ denote the set of neighbors of the node a. Using the index set S, we write $\boldsymbol{\Sigma}_{SS}^\tau$ for the $|S| \times |S|$ submatrix of $\boldsymbol{\Sigma}^\tau$ whose rows and columns are indexed by the elements of S.

Assumption $\widetilde{\boldsymbol{F}}$: There exist constants $\gamma \in (0,1]$ such that

$$|||\boldsymbol{\Sigma}_{S^CS}^\tau(\boldsymbol{\Sigma}_{SS}^\tau)^{-1}|||_{\infty,\infty} \leq 1-\gamma$$

for all $a \in \{1,\ldots,p\}$ (recall that $S = S_a^\tau$).

The assumption $\widetilde{\boldsymbol{F}}$ is known in the literature as the irrepresentable condition [135, 181, 190, 205]. It is known that it is sufficient and almost necessary condition for the consistent variable selection in the Lasso setting. Compared to the assumption **F** that was sufficient for the consistent graph selection using penalized maximum likelihood estimator, the assumption $\widetilde{\boldsymbol{F}}$ is weaker, see for example, [134] and [152].

With these assumptions, we have the following result.

Theorem 6.2. *Fix a time point of interest $\tau \in [0,1]$. Let $\{\mathbf{x}^t\}_{t\in\mathcal{T}_n}$ be an independent sample according to the model* (6.1)*. Under the assumptions* **C**, **S**, $\widetilde{\boldsymbol{F}}$ *and* **K** *there exists a constant $C > 0$ depending only on $\Lambda_{\max}, M_\Sigma, M_K$ and γ for which the following holds. Suppose that the bandwidth parameter used in* (6.10) *satisfies $h = \mathcal{O}\left(n^{-1/3}\right)$. If the penalty parameter λ in* (6.10) *scales as $\lambda = \mathcal{O}\left(n^{-1/3}\sqrt{\log p}\right)$ and the sample size satisfies $n > Cd^{3/2}(\log p)^{3/2}$, then the neighborhood selection procedure defines the edge set $\widehat{E}^\tau$, by solving* (6.10) *for all $a \in V$, which satisfies*

$$\mathbb{P}[\widehat{E}^\tau \neq \{(a,b) \mid a \neq b, |\theta_{ab}^\tau| > \theta_{\min}\}] = \mathcal{O}(\exp(-cn^{2/3}(d\log p)^{-1})) \to 0,$$

for some constant $c > 0$, with $\theta_{\min} = M_\theta n^{-1/3}\sqrt{d \log p}$ and M_θ being a sufficiently large constant.

The theorem states that the neighborhood selection procedure can be used to estimate the pattern of non-zero elements of the matrix $\mathbf{\Omega}^\tau$ that are sufficiently large, as defined by $\theta_{\min}$ and the relationship between $\boldsymbol{\theta}^\tau_{\setminus a}$ and the elements of $\mathbf{\Omega}^\tau$. Similarly to the procedure defined in §6.2, in order to gain insight into the network dynamics, the graph structure needs to be estimated at multiple time points.

The advantage of the neighborhood selection procedure over the penalized likelihood procedure is that it allows for very simple parallel implementation, since the neighborhood of each node can be estimated independently. Furthermore, the assumptions under which the neighborhood selection procedure consistently estimates the structure of the graph are weaker. Therefore, since the network structure is important in many problems, it seems that the neighborhood selection procedure should be the method of choice. However, in problems where the estimated coefficients of the precision matrix are also of importance, the penalized likelihood approach has the advantage over the neighborhood selection procedure. In order to estimate the precision matrix using the neighborhood selection, one needs first to estimate the structure and then fit the parameters subject to the structural constraints. However, it was pointed out by [17] that such two step procedures are not stable.

6.3.1 Proof of Theorem 6.2

There has been a lot of work on the analysis of the Lasso and related procedure (see for example [12, 16, 190, 205]). We will adapt some of the standard tools to prove our theorem. We will prove that the estimator $\widehat{\boldsymbol{\theta}}^\tau_{\setminus a}$ defined in (6.10) consistently defines the neighborhood of the node a. Using the union bound over all the nodes in the graph, we will then conclude the theorem.

Unlike the optimization problem (6.2), the problem defined in (6.10) is not strongly convex. Let $\widehat{\Theta}$ be the set of all minimizers of (6.10). To simplify the notation, we introduce $\widetilde{\mathbf{X}}_a \in \mathbb{R}^{p-1}$ with components $\widetilde{x}^t_a = \sqrt{w^\tau_t} x^t_a$ and $\widetilde{\mathbf{X}}_{\setminus a} \in \mathbb{R}^{n \times p-1}$ with rows equal to $\widetilde{\mathbf{x}}^t_{\setminus a} = \sqrt{w^\tau_t}\mathbf{x}^t_{\setminus a}$. With this, we say that $\boldsymbol{\theta} \in \mathbb{R}^{p-1}$ satisfies the system $(\mathcal{T})$ when for all $b = 1, \ldots, p-1$

$$\begin{aligned} 2\widetilde{\mathbf{X}}'_b(\widetilde{\mathbf{X}}_a - \widetilde{\mathbf{X}}_{\setminus a}\boldsymbol{\theta}) &= -\lambda \operatorname{sign}(\theta_b) \quad \text{if } \theta_b \neq 0 \\ |2\widetilde{\mathbf{X}}'_b(\widetilde{\mathbf{X}}_a - \widetilde{\mathbf{X}}_{\setminus a}\boldsymbol{\theta})| &\leq \lambda \quad \text{if } \theta_b = 0. \end{aligned} \tag{6.11}$$

Furthermore, $\boldsymbol{\theta} \in \widehat{\Theta}$ if and only if $\boldsymbol{\theta}$ satisfies the system $(\mathcal{T})$. The following result from [12] relates the two elements of $\widehat{\Theta}$.

Lemma 6.1 ([12])**.** *Let $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ be any two elements of $\widehat{\Theta}$. Then $\widetilde{\mathbf{X}}_{\setminus a}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) = 0$. Furthermore, all solutions have non-zero components in the same position.*

The above lemma guarantees that even though the problem (6.10) is not strongly convex, all the solutions will define the same neighborhood.

Recall that $S = S_a$ denotes the set of neighbors of the node a. Without loss of generality, we can write

$$\widehat{\mathbf{\Sigma}}^\tau = \begin{pmatrix} \widehat{\mathbf{\Sigma}}^\tau_{SS} & \widehat{\mathbf{\Sigma}}^\tau_{SS^C} \\ \widehat{\mathbf{\Sigma}}^\tau_{S^C S} & \widehat{\mathbf{\Sigma}}^\tau_{S^C S^C} \end{pmatrix}.$$

We will consider the following two events

$$\mathcal{E}_3 \;=\; \Big\{|(2\widehat{\boldsymbol{\Sigma}}^{\tau}_{SS})^{-1}[2\widetilde{\mathbf{X}}'_S\mathbf{E} - \lambda\,\mathrm{sign}(\boldsymbol{\theta}^{\tau}_S)]| < \theta_{\min}\Big\}$$

and

$$\mathcal{E}_4 = \Big\{|2\widehat{\boldsymbol{\Sigma}}^{\tau}_{S^CS}(\widehat{\boldsymbol{\Sigma}}^{\tau}_{SS})^{-1}[\widetilde{\mathbf{X}}'_S\mathbf{E} - \lambda\,\mathrm{sign}(\boldsymbol{\theta}^{\tau}_S)] - 2\widetilde{\mathbf{X}}'_{S^C}\mathbf{E}| < \lambda\Big\}, \tag{6.12}$$

where, in both events, inequalities hold element-wise and $\mathbf{E} \in \mathbb{R}^n$ is the noise term with elements $e^i = \sqrt{w_i^{\tau}}(\epsilon_a^i + (\boldsymbol{\theta}^i_{\setminus a} - \boldsymbol{\theta}^{\tau}_{\setminus a})'\mathbf{x}^i)$. Note that the noise term is not centered and includes the bias term. Using Lemma 13 in [105], the matrix $\widehat{\boldsymbol{\Sigma}}^{\tau}_{SS}$ is invertible and the events $\mathcal{E}_3$ and $\mathcal{E}_4$ are well defined.

We have an equivalent of proposition 6.1 for the neighborhood selection procedure.

Proposition 6.4. *Under the assumptions of Theorem 6.2, the event*

$$\Big\{\widehat{\boldsymbol{\theta}}^{\tau}_{\setminus a} \in \mathbb{R}^{p-1} \textit{ minimizer of } (6.10), \mathrm{sign}(\widehat{\theta}_{ab}) = \mathrm{sign}(\theta^{\tau}_{ab}) \textit{ for all } |\theta_{ab}| \notin (0, \theta_{\min})\Big\}$$

contains the event $\mathcal{E}_3 \bigcap \mathcal{E}_4$.

The theorem 6.2 will follow from Proposition 6.4, once we show that the event $\mathcal{E}_3 \bigcap \mathcal{E}_4$ occurs with high-probability. The proof of Proposition 6.4 is based on the analysis of the conditions given in (6.11) and, since it follows the same reasoning given in the proof of Proposition 6.1, the proof is omitted.

The following two lemmas establish that the events $\mathcal{E}_3$ and $\mathcal{E}_4$ occur with high probability under the assumptions of Theorem 6.2.

Lemma 6.2. *Under the assumptions of Theorem 6.2, we have that*

$$\mathbb{P}[\mathcal{E}_3] \geq 1 - C_1 \exp(-C_2 \frac{nh}{d^2 \log d})$$

with constants C_1 and C_2 depending only on M_K, M_Σ, M_θ and $\Lambda_{\max}$.

Proof. To prove the lemma, we will analyze the following three terms separately,

$$\begin{aligned} T_1 &= \lambda(2\widehat{\boldsymbol{\Sigma}}^{\tau}_{SS})^{-1}\,\mathrm{sign}(\boldsymbol{\theta}^{\tau}_{\setminus a}), \\ T_2 &= (2\widehat{\boldsymbol{\Sigma}}^{\tau}_{SS})^{-1}2\widetilde{\mathbf{X}}'_N\mathbf{E}^1, \text{ and} \\ T_3 &= (2\widehat{\boldsymbol{\Sigma}}^{\tau}_{SS})^{-1}2\widetilde{\mathbf{X}}'_N\mathbf{E}^2, \end{aligned}$$

where $\mathbf{E} = \mathbf{E}^1 + \mathbf{E}^2$, $\mathbf{E}^1 \in \mathbb{R}^n$ has elements $e^{t,1} = \sqrt{w_t^{\tau}}\epsilon_a^t$ and $\mathbf{E}^2 \in \mathbb{R}^n$ has elements $e^{t,2} = \sqrt{w_t^{\tau}}(\boldsymbol{\theta}^t_{\setminus a} - \boldsymbol{\theta}^{\tau}_{\setminus a})'\mathbf{x}^t$. Using the above defined terms and the triangle inequality, we need to show that $|T_1 + T_2 + T_3| \leq |T_1| + |T_2| + |T_3| < \theta_{\min}$.

Using Lemma 13 in [105], we have the following chain of inequalities

$$||T_1||_\infty \leq ||T_1||_2 \leq 2\lambda\Lambda_{\max}(\widehat{\boldsymbol{\Sigma}}^{-1}_{SS})_2||\,\mathrm{sign}(\boldsymbol{\theta}^{\tau}_{\setminus a})||_2 \leq C_1\lambda\sqrt{d}$$

with probability at least $1 - C_2\exp(-C_3\frac{nh}{d^2\log d})$ and C_1, C_2 and C_3 are some constants depending on M_K and $\Lambda_{\max}$.

Next, we turn to the analysis of T_2. Conditioning on $\mathbf{X}_N$ and using Lemma 13 in [105], we have that the components of T_2 are normally distributed with zero mean and variance bounded by $C_1(nh)^{-1}$, where C_1 depends on $M_K, \Lambda_{\max}$. Next, using Gaussian tail bounds, we have that

$$||T_2||_\infty \leq C_1\sqrt{\frac{\log d}{nh}}$$

with probability at least $1 - C_2\exp(-C_3\frac{nh}{d^2\log d})$, where C_1 is a constant depending on M_K, $\Lambda_{\max}$ and M_Σ.

For the term T_3, we have that

$$||T_3||_\infty \leq ||T_3||_2 \leq \Lambda_{\max}((\widehat{\mathbf{\Sigma}}^\tau_{SS})^{-1})||\mathbf{E}^2||_2 \leq 2\Lambda_{\max}||\mathbf{E}^2||_2$$

where the last inequality follows from an application of Lemma 13 in [105] with probability at least $1 - C_2\exp(-C_3\frac{nh}{d^2\log d})$. Furthermore, elements of $\mathbf{E}^2$ are normally distributed with zero mean and variance C_1hn^{-1}. Hence, we can conclude that the term T_3 is asymptotically dominated by T_2.

Combining all the terms, we have that $|T_1 + T_2 + T_3| \leq M_\theta\frac{\sqrt{d\log p}}{n^{1/3}} = \theta_{\min}$ with probability at least $1 - C_1\exp(-C_2\frac{nh}{d^2\log d})$ for constants C_1, C_2 and sufficiently large M_θ. □

Lemma 6.3. *Under the assumptions of Theorem 6.2, we have that*

$$\mathbb{P}[\mathcal{E}_4] \geq 1 - C_1\exp(-C_2\frac{nh}{d\log p})$$

with constants C_1 and C_2 depending only on $M_K, M_\Sigma, \Lambda_{\max}$ and γ.

Proof. Only a proof sketch is provided here. We analyze the event defined in (6.12) by splitting it into several terms. Observe that for $b \in S^C$, we can write

$$x^t_b = \mathbf{\Sigma}^\tau_{bS}(\mathbf{\Sigma}^\tau_{SS})^{-1}\mathbf{x}^t_S + [\mathbf{\Sigma}^t_{bS}(\mathbf{\Sigma}^t_{SS})^{-1} - \mathbf{\Sigma}^\tau_{bS}(\mathbf{\Sigma}^\tau_{SS})^{-1}]'\mathbf{x}^t_S + v^t_b$$

where $v^t_b \sim \mathcal{N}(0, (\sigma^t_b)^2)$ with $\sigma^t_b \leq 1$. Let us denote $\widetilde{\mathbf{V}}_b \in \mathbb{R}^n$ the vector with components $\widetilde{v}^t_b = \sqrt{w^\tau_t}v^t_b$. With this, we have the following decomposition of the components of the event $\mathcal{E}_4$. For all $b \in S^c$,

$$\begin{aligned}
w_{b,1} &= \mathbf{\Sigma}^\tau_{bS}(\mathbf{\Sigma}^\tau_{SS})^{-1}\lambda\,\mathrm{sign}(\boldsymbol{\theta}^\tau_S),\\
w_{b,2} &= \widetilde{\mathbf{V}}'_b\Big[(\widetilde{\mathbf{X}}_S(\widehat{\mathbf{\Sigma}}_{SS})^{-1}\lambda\,\mathrm{sign}(\boldsymbol{\theta}^\tau_S) + \Pi^\perp_{\widetilde{\mathbf{X}}_S}(\mathbf{E}^1)\Big],\\
w_{b,3} &= \widetilde{\mathbf{V}}'_b\Pi^\perp_{\widetilde{\mathbf{X}}_S}(\mathbf{E}^2),\ \text{and}\\
w_{b,4} &= \widetilde{\mathbf{F}}'_b\Big[(\widetilde{\mathbf{X}}_N(\widehat{\mathbf{\Sigma}}_{SS})^{-1}\lambda\,\mathrm{sign}(\boldsymbol{\theta}^\tau_N) + \Pi^\perp_{\widetilde{\mathbf{X}}_S}(\mathbf{E}^1 + \mathbf{E}^2)\Big],
\end{aligned}$$

where $\Pi^\perp_{\widetilde{\mathbf{X}}_S}$ is the projection operator defined as $\mathbf{I}_p - \widetilde{\mathbf{X}}_S(\widetilde{\mathbf{X}}'_S\widetilde{\mathbf{X}}_S)^{-1}\widetilde{\mathbf{X}}'_S$, $\mathbf{E}^1$ and $\mathbf{E}^2$ are defined in the proof of Lemma 6.2 and we have introduced $\widetilde{\mathbf{F}}_b \in \mathbb{R}^n$ as the vector with components

$$\widetilde{f}^t_b = \sqrt{w^\tau_t}[\mathbf{\Sigma}^t_{bS}(\mathbf{\Sigma}^t_{SS})^{-1} - \mathbf{\Sigma}^\tau_{bS}(\mathbf{\Sigma}^\tau_{SS})^{-1}]'\mathbf{x}^t_S.$$

The lemma will follow using the triangle inequality if we show that

$$\max_{b \in N^C} |w_{b,1}| + |w_{b,2}| + |w_{b,3}| + |w_{b,4}| \leq \lambda.$$

Under the assumptions of the lemma, it holds that $\max_{b \in N^C} |w_{b,1}| < (1-\gamma)\lambda$.

Next, we deal with the term $w_{b,2}$. We observe that conditioning on $\mathbf{X}_S$, we have that $w_{b,2}$ is normally distributed with variance that can be bounded combining results of Lemma 13 in [105] with the proof of Lemma 4 in [190]. Next, we use the Gaussian tail bound to conclude that $\max_{b \in N^C} |w_{b,2}| < \gamma\lambda/2$ with probability at least $1 - \exp(-C_2 nh(d \log p)^{-1})$.

An upper bound on the term $w_{b,3}$ is obtained as follows $w_{b,3} \leq ||\widetilde{\mathbf{V}}_b||_2 ||\Pi^{\perp}_{\widetilde{\mathbf{X}}_S}(\mathbf{E}^2)||_2$ and then observing that the term is asymptotically dominated by the term $w_{b,2}$. Using similar reasoning, we also have that $w_{b,4}$ is asymptotically smaller than $w_{b,2}$.

Combining all the upper bounds, we obtain the desired result. □

Now, Theorem 6.2 follows from Propositions 6.4, Lemma 6.2 and Lemma 6.3 and an application of the union bound.

6.4 Discussion

In this chapter, we focus on consistent estimation of the graph structure in high-dimensional time-varying multivariate Gaussian distributions, as introduced in [206]. The non-parametric estimate of the sample covariance matrix used together with the ℓ_1 penalized log-likelihood estimation produces a good estimate of the concentration matrix. Our contribution is the derivation of the sufficient conditions under which the estimate consistently recovers the graph structure.

This work complements the earlier work on value consistent estimation of time-varying Gaussian graphical models in [206] in that the main focus here is the consistent structure recovery of the graph associated with the probability distribution at a fixed time point. Obtaining an estimator that consistently recovers the structure is a harder problem than obtaining an estimator that is only consistent in, say, Frobenius norm. However, the price for the correct model identification comes in much more strict assumptions on the underlying model. Note that we needed to assume the "irrepresentable-like" condition on the Fisher information matrix (Assumption **F**), which is not needed in the work of [206]. In some problems, where we want to learn about the nature of the process that generates the data, estimating the structure of the graph associated with the distribution gives more insight into the nature than the values of the concentration matrix. This is especially true in cases where the estimated graph is sparse and easily interpretable by domain experts.

Motivated by many real world problems coming from diverse areas such as biology and finance, we extend the work of [152] which facilitates estimation under the assumption that the underlying distribution does not change. We assume that the distribution changes smoothly, an assumption that is more valid, but could still be unrealistic in real life. In the next chapter, we consider estimation of abrupt changes in the distribution and the graph structure.

Furthermore, we extend the neighborhood selection procedure as introduced in [135] to the time-varying Gaussian graphical models. This is done in a straightforward way using ideas from

the literature on the varying-coefficient models, where a kernel smoother is used to estimate the model parameters that change over time in an unspecified way. We have shown that the neighborhood selection procedure is a good alternative to the penalized log-likelihood estimation procedure, as it requires less strict assumptions on the model. In particular, the assumption **F** can be relaxed to $\widetilde{\boldsymbol{F}}$. We believe that our work provides important insights into the problem of estimating structure of dynamic networks.

Chapter 7

Time Varying Gaussian Graphical Models With Jumps

In this chapter, we consider the scenario in which the model evolves in a piece-wise constant fashion. We propose a procedure that estimates the structure of a graphical model by minimizing the temporally smoothed L1 penalized regression, which allows jointly estimating the partition boundaries of the model and the coefficient of the sparse precision matrix on each block of the partition. A highly scalable proximal gradient method is proposed to solve the resultant convex optimization problem; and the conditions for sparsistent estimation and the convergence rate of both the partition boundaries and the network structure are established for the first time for such estimators.

7.1 Introduction

In this chapter, we consider an estimation problem under a particular dynamic context, where the model evolves piecewise constantly, i.e., staying structurally invariant during unknown segments of time, and then jump to a different structure.

Approximately piecewise constantly evolving networks can be found underlying many natural dynamic systems of intellectual and practical interest. For example, in a biological developmental system such as the fruit fly, the entire life cycle of the fly consists of 4 discrete developmental stages, namely, embryo, larva, pupa, and adult; across the stages, one expect to see dramatical rewiring of the regulatory network to realize very different regulation functions due to different developmental needs, whereas within each stage, the change of the network topology are expected to be relatively more mild as revealed by the smoother trajectories of the gene expression activities, because a largely stable regulatory machinery is employed to control stage-specific developmental processes. Such phenomena are not uncommon in social systems. For example, in an underlying social network between the senators, even it is not visible to outsiders, we would imagine the network structure being more stable between the elections but more volatile when the campaigns start. Although it is legitimate to use a completely unconstrained time-evolving network model to describe or analysis such systems, an approximately piecewise constantly evolving network model is better at capturing the different amount of network dy-

namics during different phases of a entire life cycle, and detecting boundaries between different phases when desirable.

Let $\{\mathbf{x}_i\}_{i\in[n]} \in \mathbb{R}^p$ be a sequence of n independent observations from some p-dimensional multivariate Gaussian distributions, not necessarily the same for every observation. Let $\{\mathcal{B}^j\}_{j\in[B]}$ be a disjoint partitioning of the set $[n]$ where each block of the partition consists of consecutive elements, that is, $\mathcal{B}^j \cap \mathcal{B}^{j'} = \emptyset$ for $j \neq j'$ and $\bigcup_j \mathcal{B}^j = [n]$ and $\mathcal{B}^j = [T_{j-1} : T_j] := \{T_{j-1}, T_{j-1} + 1, \ldots, T_j - 1\}$. Let $\mathcal{T} := \{T_0 = 1 < T_1 < \cdots < T_B = n + 1\}$ denote the set of partition boundaries. We consider the following model

$$\mathbf{x}_i \sim \mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma}^j), \qquad i \in \mathcal{B}^j, \tag{7.1}$$

such that observations indexed by elements in $\mathcal{B}^j$ are p-dimensional realizations of a multivariate Gaussian distribution with zero mean and the covariance matrix $\mathbf{\Sigma}^j = (\sigma^j_{ab})_{a,b\in[p]}$, which suggest that it is only unique to segment j of the time series. Let $\mathbf{\Omega}^j := (\mathbf{\Sigma}^j)^{-1}$ denote the precision matrix with elements $(\omega^j_{ab})_{a,b\in[p]}$. With the number of partitions, B, and the boundaries of partitions, $\mathcal{T}$, unknown, we study the problem of estimating both the partition set $\{\mathcal{B}^j\}$ and the non-zero elements of the precision matrices $\{\mathbf{\Omega}^j\}_{j\in[B]}$ from the sample $\{\mathbf{x}_i\}_{i\in[n]}$. Note that in this work we study a particular case, where the coefficients of the model are piece-wise constant functions of time.

If the partitions $\{\mathcal{B}^j\}_j$ were known, the problem would be trivially reduced to the setting analyzed in the previous work. Dealing with the unknown partitions, together with the structure estimation of the model, calls for new methods. We propose and analyze a method based on *time-coupled neighborhood selection*, where the model estimates are forced to stay similar across time using a fusion-type total variation penalty and the sparsity of each neighborhood is obtained through the ℓ_1 penalty. Details of the approach are given in §7.2.

The model in (7.1) is related to the varying-coefficient models (for example, [96]) with the coefficients being piece-wise constant functions. Varying coefficient regression models with piece-wise constant coefficients are also known as segmented multivariate regression models [121] or linear models with structural changes [15]. The structural changes are commonly determined through hypothesis testing and a separate linear model is fit to each of the estimated segments. In our work, we use the penalized model selection approach to jointly estimate the partition boundaries and the model parameters.

The work presented in this chapter is very different from the one of [206] and §6, since under our assumptions the network changes abruptly rather than smoothly. The work of [2] is most similar to our setting, where they also use a fused-type penalty combined with an ℓ_1 penalty to estimate the structure of the varying Ising model. Here, in addition to focusing on GGMs, there is an additional subtle, but important, difference to [2]. In this chapter, we use a modification of the fusion penalty (formally described in §7.2) which allows us to characterize the model selection consistency of our estimates and the convergence properties of the estimated partition boundaries, which is not available in the earlier work.

7.2 Graph estimation via Temporal-Difference Lasso

In this section, we introduce our time-varying covariance selection procedure, which is based on the time-coupled neighborhood selection using the fused-type penalty. We call the proposed procedure Temporal-Difference Lasso (*TD-Lasso*).

We build on the neighbourhood selection procedure to estimate the changing graph structure in model (7.1). We use S_a^j to denote the neighborhood of the node a on the block $\mathcal{B}^j$ and N_a^j to denote nodes not in the neighborhood of the node a on the j-th block, $N_a^j = V \setminus S_a^j$. Consider the following estimation procedure

$$\widehat{\boldsymbol{\beta}}^a = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p-1 \times n}}{\operatorname{argmin}} \ \mathcal{L}(\boldsymbol{\beta}) + \operatorname{pen}_{\lambda_1, \lambda_2}(\boldsymbol{\beta}) \tag{7.2}$$

where the loss is defined for $\boldsymbol{\beta} = (\beta_{b,i})_{b \in [p-1], i \in [n]}$ as

$$\mathcal{L}(\boldsymbol{\beta}) := \sum_{i \in [n]} \left(x_{i,a} - \sum_{b \in \setminus a} x_{i,b} \beta_{b,i} \right)^2 \tag{7.3}$$

and the penalty is defined as

$$\operatorname{pen}_{\lambda_1, \lambda_2}(\boldsymbol{\beta}) := 2\lambda_1 \sum_{i=2}^{n} ||\boldsymbol{\beta}_{\cdot,i} - \boldsymbol{\beta}_{\cdot,i-1}||_2 + 2\lambda_2 \sum_{i=1}^{n} \sum_{b \in \setminus a} |\beta_{b,i}|. \tag{7.4}$$

The penalty term is constructed from two terms. The first term ensures that the solution is going to be piecewise constant for some partition of $[n]$ (possibly a trivial one). The first term can be seen as a sparsity inducing term in the temporal domain, since it penalizes the difference between the coefficients $\boldsymbol{\beta}_{\cdot,i}$ and $\boldsymbol{\beta}_{\cdot,i+1}$ at successive time-points. The second term results in estimates that have many zero coefficients within each block of the partition. The estimated set of partition boundaries

$$\widehat{\mathcal{T}} = \{\widehat{T}_0 = 1\} \cup \{\widehat{T}_j \in [2:n] \ : \ \widehat{\boldsymbol{\beta}}^a_{\cdot,\widehat{T}_j} \neq \widehat{\boldsymbol{\beta}}^a_{\cdot,\widehat{T}_j - 1}\} \cup \{\widehat{T}_{\widehat{B}} = n+1\}$$

contains indices of points at which a change is estimated, with $\widehat{B}$ being an estimate of the number of blocks B. The estimated number of the block $\widehat{B}$ is controlled through the user defined penalty parameter λ_1, while the sparsity of the neighborhood is controlled through the penalty parameter λ_2.

Based on the estimated set of partition boundaries $\widehat{\mathcal{T}}$, we can define the neighborhood estimate of the node a for each estimated block. Let $\widehat{\boldsymbol{\theta}}^{a,j} = \widehat{\boldsymbol{\beta}}^a_{\cdot,i}, \forall i \in [\widehat{T}_{j-1} : \widehat{T}_j]$ be the estimated coefficient vector for the block $\widehat{\mathcal{B}}^j = [\widehat{T}_{j-1} : \widehat{T}_j]$. Using the estimated vector $\widehat{\boldsymbol{\theta}}^{a,j}$, we define the neighborhood estimate of the node a for the block $\widehat{\mathcal{B}}^j$ as

$$\widehat{S}_a^j := S(\widehat{\boldsymbol{\theta}}^{a,j}) := \{b \in \setminus a \ : \ \widehat{\theta}^{a,j}_b \neq 0\}.$$

Solving (7.2) for each node $a \in V$ gives us a neighborhood estimate for each node. Combining the neighborhood estimates we can obtain an estimate of the graph structure for each point $i \in [n]$.

The choice of the penalty term is motivated by the work on penalization using total variation [131, 148], which results in a piece-wise constant approximation of an unknown regression function. The fusion-penalty has also been applied in the context of multivariate linear regression [177], where the coefficients that are spatially close, are also biased to have similar values. As a result, nearby coefficients are fused to the same estimated value. Instead of penalizing the ℓ_1 norm on the difference between coefficients, we use the ℓ_2 norm in order to enforce that all the changes occur at the same point.

The objective (7.2) estimates the neighborhood of one node in a graph for all time-points. After solving the objective (7.2) for all nodes $a \in V$, we need to combine them to obtain the graph structure. We will use the following procedure to combine $\{\widehat{\boldsymbol{\beta}}^a\}_{a \in V}$,

$$\widehat{E}_i = \{(a, b) \ : \ \max(|\beta^a_{b,i}|, |\beta^b_{a,i}|) > 0\}, \qquad i \in [n].$$

That is, an edge between nodes a and b is included in the graph if at least one of the nodes a or b is included in the neighborhood of the other node. We use the $\max$ operator to combine different neighborhoods as we believe that for the purpose of network exploration it is more important to occasionally include spurious edges than to omit relevant ones. For further discussion on the differences between the min and the max combination, we refer an interested reader to [19].

7.2.1 Numerical procedure

Finding a minimizer $\widehat{\boldsymbol{\beta}}^a$ of (7.2) can be a computationally challenging task for an off-the-shelf convex optimization procedure. We propose to use an accelerated gradient method with a smoothing technique [142], which converges in $\mathcal{O}(1/\epsilon)$ iterations where ϵ is the desired accuracy.

We start by defining a smooth approximation of the fused penalty term. Let $\mathbf{H} \in \mathbb{R}^{n \times n-1}$ be a matrix with elements

$$H_{ij} = \begin{cases} -1 & \text{if } i = j \\ 1 & \text{if } i = j + 1 \\ 0 & \text{otherwise.} \end{cases}$$

With the matrix $\mathbf{H}$ we can rewrite the fused penalty term as $2\lambda_1 \sum_{i=1}^{n-1} ||(\boldsymbol{\beta}\mathbf{H})_{\cdot,i}||_2$ and using the fact that the ℓ_2 norm is self dual (e.g., see [26]) we have the following representation

$$2\lambda_1 \sum_{i=2}^{n} ||\boldsymbol{\beta}_{\cdot,i} - \boldsymbol{\beta}_{\cdot,i-1}||_2 = \max_{\mathbf{U} \in \mathcal{Q}} \langle\langle \mathbf{U}, 2\lambda_1 \boldsymbol{\beta}\mathbf{H} \rangle\rangle$$

where $\mathcal{Q} := \{\mathbf{U} \in \mathbb{R}^{p-1 \times n-1} \ : \ ||\mathbf{U}_{\cdot,i}||_2 \leq 1, \ \forall i \in [n-1]\}$. The following function is defined as a smooth approximation to the fused penalty,

$$\Psi_\mu(\boldsymbol{\beta}) := \max_{\mathbf{U} \in \mathcal{Q}} \langle\langle \mathbf{U}, 2\lambda_1 \boldsymbol{\beta}\mathbf{H} \rangle\rangle - \mu ||\mathbf{U}||_F^2 \tag{7.5}$$

where $\mu > 0$ is the smoothness parameter. It is easy to see that

$$\Psi_\mu(\boldsymbol{\beta}) \leq \Psi_0(\boldsymbol{\beta}) \leq \Psi_\mu(\boldsymbol{\beta}) + \mu(n-1).$$

Setting the smoothness parameter to $\mu = \frac{\epsilon}{2(n-1)}$, the correct rate of convergence is ensured. Let $\mathbf{U}_\mu(\boldsymbol{\beta})$ be the optimal solution of the maximization problem in (7.5), which can be obtained analytically as

$$\mathbf{U}_\mu(\boldsymbol{\beta}) = \Pi_{\mathcal{Q}}\left(\frac{\lambda\boldsymbol{\beta}\mathbf{H}}{\mu}\right)$$

where $\Pi_{\mathcal{Q}}(\cdot)$ is the projection operator onto the set $\mathcal{Q}$. From Theorem 1 in [142], we have that $\Psi_\mu(\boldsymbol{\beta})$ is continuously differentiable and convex, with the gradient

$$\nabla\Psi_\mu(\boldsymbol{\beta}) = 2\lambda_1\mathbf{U}_\mu(\boldsymbol{\beta})\mathbf{H}'$$

that is Lipschitz continuous.

With the above defined smooth approximation, we focus on minimizing the following objective

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{p-1\times n}} F(\boldsymbol{\beta}) := \min_{\boldsymbol{\beta}\in\mathbb{R}^{p-1\times n}} \mathcal{L}(\boldsymbol{\beta}) + \Psi_\mu(\boldsymbol{\beta}) + 2\lambda_2||\boldsymbol{\beta}||_1.$$

Following [11] (see also [141]), we define the following quadratic approximation of $F(\boldsymbol{\beta})$ at a point $\boldsymbol{\beta}_0$

$$\begin{aligned} Q_L(\boldsymbol{\beta},\boldsymbol{\beta}_0) := \mathcal{L}(\boldsymbol{\beta}_0) + \Psi_\mu(\boldsymbol{\beta}_0) + \langle\langle\boldsymbol{\beta}-\boldsymbol{\beta}_0, \nabla\mathcal{L}(\boldsymbol{\beta}_0) + \nabla\Psi(\boldsymbol{\beta}_0)\rangle\rangle \\ + \frac{L}{2}||\boldsymbol{\beta}-\boldsymbol{\beta}_0||_F^2 + 2\lambda_2||\boldsymbol{\beta}||_1 \end{aligned}$$

where $L > 0$ is the parameter chosen as an upper bounds for the Lipschitz constant of $\nabla\mathcal{L}+\nabla\Psi$. Let $p_L(\boldsymbol{\beta}_0)$ be a minimizer of $Q_L(\boldsymbol{\beta},\boldsymbol{\beta}_0)$. Ignoring constant terms, $p_L(\boldsymbol{\beta}_0)$ can be obtained as

$$p_L(\boldsymbol{\beta}_0) = \operatorname*{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^{p-1\times n}} \frac{1}{2}\left\|\boldsymbol{\beta} - \left(\boldsymbol{\beta}_0 - \frac{1}{L}\big(\nabla\mathcal{L}+\nabla\Psi\big)(\boldsymbol{\beta}_0)\right)\right\|_F^2 + \frac{2\lambda_2}{L}\|\boldsymbol{\beta}\|_1.$$

It is clear that $p_L(\boldsymbol{\beta}_0)$ is the unique minimizer, which can be obtained in a closed form, as a result of the soft-thresholding,

$$p_L(\boldsymbol{\beta}_0) = T\left(\boldsymbol{\beta}_0 - \frac{1}{L}\big(\nabla\mathcal{L}+\nabla\Psi\big)(\boldsymbol{\beta}_0), \frac{2\lambda_2}{L}\right) \tag{7.6}$$

where $T(x,\lambda) = \operatorname{sign}(x)\max(0,|x|-\lambda)$ is the soft-thresholding operator that is applied element-wise.

In practice, an upper bound on the Lipschitz constant of $\nabla\mathcal{L}+\nabla\Psi$ can be expensive to compute, so the parameter L is going to be determined iteratively. Combining all of the above, we arrive at Algorithm 2. In the algorithm, $\boldsymbol{\beta}_0$ is set to zero or, if the optimization problem is solved for a sequence of tuning parameters, it can be set to the solution $\widehat{\boldsymbol{\beta}}$ obtained for the previous set of tuning parameters. The parameter γ is a constant used to increase the estimate of the Lipschitz constant L and we set it to $\gamma = 1.5$ in our experiments, while $L = 1$ initially. Compared to the gradient descent method (which can be obtain by iterating $\boldsymbol{\beta}_{k+1} = p_L(\boldsymbol{\beta}_k)$), the accelerated gradient method updates two sequences $\{\boldsymbol{\beta}_k\}$ and $\{\mathbf{z}_k\}$ recursively. Instead of performing the gradient step from the latest approximate solution $\boldsymbol{\beta}_k$, the gradient step is performed from the search point $\mathbf{z}_k$ that is obtained as a linear combination of the last two approximate solutions

$\boldsymbol{\beta}_{k-1}$ and $\boldsymbol{\beta}_k$. Since the condition $F(p_L(\mathbf{z}_k)) \leq Q_L(p_L(\mathbf{z}_k), \mathbf{z}_k)$ is satisfied in every iteration, we have the algorithm converges in $\mathcal{O}(1/\epsilon)$ iterations following [11]. As the convergence criterion, we stop iterating once the relative change in the objective value is below some threshold value (e.g., we use 10^{-4}).

7.2.2 Tuning parameter selection

The penalty parameters λ_1 and λ_2 control the complexity of the estimated model. In this work, we propose to use the BIC score to select the tuning parameters. Define the BIC score for each node $a \in V$ as

$$\mathrm{BIC}_a(\lambda_1, \lambda_2) := \log \frac{\mathcal{L}(\widehat{\boldsymbol{\beta}}^a)}{n} + \frac{\log n}{n} \sum_{j \in [\widehat{B}]} |S(\widehat{\boldsymbol{\theta}}^{a,j})|$$

where $\mathcal{L}(\cdot)$ is defined in (7.3) and $\widehat{\boldsymbol{\beta}}^a = \widehat{\boldsymbol{\beta}}^a(\lambda_1, \lambda_2)$ is a solution of (7.2). The penalty parameters can now be chosen as

$$\{\widehat{\lambda}_1, \widehat{\lambda}_2\} = \operatorname*{argmin}_{\lambda_1, \lambda_2} \sum_{a \in V} \mathrm{BIC}_a(\lambda_1, \lambda_2).$$

We will use the above formula to select the tuning parameters in our simulations, where we are going to search for the best choice of parameters over a grid.

7.3 Theoretical results

This section is going to address the statistical properties of the estimation procedure presented in Section 7.2. The properties are addressed in an asymptotic framework by letting the sample size

Input: $\mathbf{X} \in \mathbb{R}^{n \times p}$, $\boldsymbol{\beta}_0 \in \mathbb{R}^{p-1 \times n}$, $\gamma > 1$, $L > 0$, $\mu = \frac{\epsilon}{2(n-1)}$
Output: $\widehat{\boldsymbol{\beta}}^a$
Initialize $k := 1$, $\alpha_k := 1$, $\mathbf{z}_k := \boldsymbol{\beta}_0$
repeat
  while $F(p_L(\mathbf{z}_k)) > Q_L(p_L(\mathbf{z}_k), \mathbf{z}_k)$ **do**
    $L := \gamma L$
  end
  $\boldsymbol{\beta}_k := p_L(\mathbf{z}_k)$ (using Eq. (7.6))
  $\alpha_{k+1} := \frac{1+\sqrt{1+4\alpha_k}}{2}$
  $\mathbf{z}_{k+1} := \boldsymbol{\beta}_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\boldsymbol{\beta}_k - \boldsymbol{\beta}_{k-1})$
until *convergence*
$\widehat{\boldsymbol{\beta}}^a := \boldsymbol{\beta}_k$

Algorithm 2: Accelerated Gradient Method for Equation (7.2)

n grow, while keeping the other parameters fixed. For the asymptotic framework to make sense, we assume that there exists a fixed unknown sequence of numbers $\{\tau_j\}$ that defines the partition boundaries as $T_j = \lfloor n\tau_j \rfloor$, where $\lfloor a \rfloor$ denotes the largest integer smaller that a. This assures that as the number of samples grow, the same fraction of samples falls into every partition. We call $\{\tau_j\}$ the boundary fractions.

We give sufficient conditions under which the sequence $\{\tau_j\}$ is consistently estimated. In particular, if the number of partition blocks is estimated correctly, then we show that $\max_{j\in[B]} |\widehat{T}_j - T_j| \leq n\delta_n$ with probability tending to 1, where $\{\delta_n\}_n$ is a non-increasing sequence of positive numbers that tends to zero. If the number of partition segments is over estimated, then we show that for a distance defined for two sets A and B as

$$h(A, B) := \sup_{b\in B} \inf_{a\in A} |a - b|, \tag{7.7}$$

we have $h(\widehat{\mathcal{T}}, \mathcal{T}) \leq n\delta_n$ with probability tending to 1. With the boundary segments consistently estimated, we further show that under suitable conditions for each node $a \in V$ the correct neighborhood is selected on all estimated block partitions that are sufficiently large.

The proof technique employed in this section is quite involved, so we briefly describe the steps used. Our analysis is based on careful inspection of the optimality conditions that a solution $\widehat{\boldsymbol{\beta}}^a$ of the optimization problem (7.2) need to satisfy. The optimality conditions for $\widehat{\boldsymbol{\beta}}^a$ to be a solution of (7.2) are given in §7.3.2. Using the optimality conditions, we establish the rate of convergence for the partition boundaries. This is done by proof by contradiction. Suppose that there is a solution with the partition boundary $\widehat{\mathcal{T}}$ that satisfies $h(\widehat{\mathcal{T}}, \mathcal{T}) \geq n\delta_n$. Then we show that, with high-probability, all such solutions will not satisfy the KKT conditions and therefore cannot be optimal. This shows that all the solutions to the optimization problem (7.2) result in partition boundaries that are "close" to the true partition boundaries, with high-probability. Once it is established that $\widehat{\mathcal{T}}$ and $\mathcal{T}$ satisfy $h(\widehat{\mathcal{T}}, \mathcal{T}) \leq n\delta_n$, we can further show that the neighborhood estimates are consistently estimated, under the assumption that the estimated blocks of the partition have enough samples. This part of the analysis follows the commonly used strategy to prove that the Lasso is sparsistent (e.g., see [12, 135, 190]), however important modifications are required due to the fact that position of the partition boundaries are being estimated.

Our analysis is going to focus on one node $a \in V$ and its neighborhood. However, using the union bound over all nodes in V, we will be able to carry over conclusions to the whole graph. To simplify our notation, when it is clear from the context, we will omit the superscript a and write $\widehat{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{\theta}}$ and S, etc., to denote $\widehat{\boldsymbol{\beta}}^a$, $\widehat{\boldsymbol{\theta}}^a$ and S_a, etc.

7.3.1 Assumptions

Before presenting our theoretical results, we give some definitions and assumptions that are going to be used in this section. Let $\Delta_{\min} := \min_{j\in[B]} |T_j - T_{j-1}|$ denote the minimum length between change points, $\xi_{\min} := \min_{a\in V} \min_{j\in[B-1]} ||\boldsymbol{\theta}^{a,j+1} - \boldsymbol{\theta}^{a,j}||_2$ denote the minimum jump size and $\theta_{\min} = \min_{a\in V} \min_{j\in[B]} \min_{b\in S^j} |\theta_b^{a,j}|$ the minimum coefficient size. Throughout the section, we assume that the following holds.

A1 There exist two constants $\phi_{\min} > 0$ and $\phi_{\max} < \infty$ such that

$$\phi_{\min} = \min \{\Lambda_{\min}(\boldsymbol{\Sigma}^j) \ : \ j \in [B], a \in V\}$$

and

$$\phi_{\max} = \max \{\Lambda_{\max}(\mathbf{\Sigma}^j) \ : \ j \in [B], a \in V\}.$$

A2 Variables are scaled so that $\sigma^j_{aa} = 1$ for all $j \in [B]$ and all $a \in V$.

The assumption **A1** is commonly used to ensure that the model is identifiable. If the population covariance matrix is ill-conditioned, the question of the correct model identification if not well defined, as a neighborhood of a node may not be uniquely defined. The assumption **A2** is assumed for the simplicity of the presentation. The common variance can be obtained through scaling.

A3 There exists a constant $M > 0$ such that

$$\max_{a \in V} \max_{j,k \in [B]} \|\boldsymbol{\theta}^{a,k} - \boldsymbol{\theta}^{a,j}\|_2 \leq M.$$

The assumption **A3** states that the difference between coefficients on two different blocks, $||\boldsymbol{\theta}^{a,k} - \boldsymbol{\theta}^{a,j}||_2$, is bounded for all $j, k \in [B]$. This assumption is simply satisfied if the coefficients $\boldsymbol{\theta}^a$ were bounded in the ℓ_2 norm.

A4 There exist a constant $\alpha \in (0, 1]$, such that the following holds

$$\max_{j \in [B]} |||\mathbf{\Sigma}_{N^j_a S^j_a} (\mathbf{\Sigma}_{S^j_a S^j_a})^{-1}|||_\infty \leq 1 - \alpha, \qquad \forall a \in V.$$

The assumption **A4** states that the variables in the neighborhood of the node a, S^j_a, are not too correlated with the variables in the set N^j_a. This assumption is necessary and sufficient for correct identification of the relevant variables in the Lasso regression problems (e.g., see [181, 205]). Note that this condition is sufficient also in our case when the correct partition boundaries are not known.

A5 The minimum coefficient size $\theta_{\min}$ satisfies $\theta_{\min} = \Omega(\sqrt{\log(n)/n})$.

The lower bound on the minimum coefficient size $\theta_{\min}$ is necessary, since if a partial correlation coefficient is too close to zero the edge in the graph would not be detectable.

A6 The sequence of partition boundaries $\{T_j\}$ satisfy $T_j = \lfloor n\tau_j \rfloor$, where $\{\tau_j\}$ is a fixed, unknown sequence of the boundary fractions belonging to $[0, 1]$.

The assumption is needed for the asymptotic setting. As $n \to \infty$, there will be enough sample points in each of the blocks to estimate the neighborhood of nodes correctly.

7.3.2 Convergence of the partition boundaries

In this subsection we establish the rate of convergence of the boundary partitions for the estimator (7.2). We start by giving a lemma that characterizes solutions of the optimization problem given in (7.2). Note that the optimization problem in (7.2) is convex, however, there may be multiple solutions to it, since it is not strictly convex.

Lemma 7.1. *Let $x_{i,a} = \mathbf{x}'_{i,\setminus a}\boldsymbol{\theta}_a + \epsilon_i$. A matrix $\widehat{\boldsymbol{\beta}}$ is optimal for the optimization problem* (7.2) *if and only if there exist a collection of subgradient vectors $\{\widehat{\mathbf{z}}_i\}_{i\in[2:n]}$ and $\{\widehat{\mathbf{y}}_i\}_{i\in[n]}$, with $\widehat{\mathbf{z}}_i \in \partial||\widehat{\boldsymbol{\beta}}_{\cdot,i} - \widehat{\boldsymbol{\beta}}_{\cdot,i-1}||_2$ and $\widehat{\mathbf{y}}_i \in \partial||\widehat{\boldsymbol{\beta}}_{\cdot,i}||_1$, that satisfies*

$$\sum_{i=k}^{n} \mathbf{x}_{i,\setminus a}\langle \mathbf{x}_{i,\setminus a}, \widehat{\boldsymbol{\beta}}_{\cdot,i} - \boldsymbol{\beta}_{\cdot,i}\rangle - \sum_{i=k}^{n} \mathbf{x}_{i,\setminus a}\epsilon_i + \lambda_1\widehat{\mathbf{z}}_k + \lambda_2 \sum_{i=k}^{n} \widehat{\mathbf{y}}_i = 0 \tag{7.8}$$

for all $k \in [n]$ and $\widehat{\mathbf{z}}_1 = \widehat{\mathbf{z}}_{n+1} = \mathbf{0}$.

The following theorem provides the convergence rate of the estimated boundaries of $\widehat{\mathcal{T}}$, under the assumption that the correct number of blocks is known.

Theorem 7.1. *Let $\{\mathbf{x}_i\}_{i\in[n]}$ be a sequence of observation according to the model in* (7.1)*. Assume that* **A1-A3** *and* **A5-A6** *hold. Suppose that the penalty parameters λ_1 and λ_2 satisfy*

$$\lambda_1 \asymp \lambda_2 = \mathcal{O}(\sqrt{\log(n)/n}).$$

Let $\{\widehat{\boldsymbol{\beta}}_{\cdot,i}\}_{i\in[n]}$ be any solution of (7.2) *and let $\widehat{\mathcal{T}}$ be the associated estimate of the block partition. Let $\{\delta_n\}_{n\geq 1}$ be a non-increasing positive sequence that converges to zero as $n \to \infty$ and satisfies $\Delta_{\min} \geq n\delta_n$ for all $n \geq 1$. Furthermore, suppose that $(n\delta_n\xi_{\min})^{-1}\lambda_1 \to 0$, $\xi_{\min}^{-1}\sqrt{p}\lambda_2 \to 0$ and $(\xi_{\min}\sqrt{n\delta_n})^{-1}\sqrt{p\log n} \to 0$, then if $|\widehat{\mathcal{T}}| = B + 1$ the following holds*

$$\mathbb{P}[\max_{j\in[B]} |T_j - \widehat{T}_j| \leq n\delta_n] \xrightarrow{n\to\infty} 1.$$

The proof builds on techniques developed in [94] and is presented in §7.7.

Suppose that $\delta_n = (\log n)^\gamma/n$ for some $\gamma > 1$ and $\xi_{\min} = \Omega(\sqrt{\log n/(\log n)^\gamma})$, the conditions of Theorem 7.1 are satisfied, and we have that the sequence of boundary fractions $\{\tau_j\}$ is consistently estimated. Since the boundary fractions are consistently estimated, we will see below that the estimated neighborhood $S(\widehat{\boldsymbol{\theta}}^j)$ on the block $\widehat{\mathcal{B}}^j$ consistently recovers the true neighborhood S^j.

Unfortunately, the correct bound on the number of block B may not be known. However, a conservative upper bound $B_{\max}$ on the number of blocks B may be known. Suppose that the sequence of observation is over segmented, with the number of estimated blocks bounded by $B_{\max}$. Then the following proposition gives an upper bound on $h(\widehat{\mathcal{T}}, \mathcal{T})$ where $h(\cdot,\cdot)$ is defined in (7.7).

Proposition 7.1. *Let $\{\mathbf{x}_i\}_{i\in[n]}$ be a sequence of observation according to the model in* (7.1)*. Assume that the conditions of Theorem 7.1 are satisfied. Let $\widehat{\boldsymbol{\beta}}$ be a solution of* (7.2) *and $\widehat{\mathcal{T}}$ the corresponding set of partition boundaries, with $\widehat{B}$ blocks. If the number of blocks satisfy $B \leq \widehat{B} \leq B_{\max}$, then*

$$\mathbb{P}[h(\widehat{\mathcal{T}}, \mathcal{T}) \leq n\delta_n] \xrightarrow{n\to\infty} 1.$$

The proof of the proposition follows the same ideas of Theorem 7.1 and its sketch is given in the appendix.

The above proposition assures us that even if the number of blocks is overestimated, there will be a partition boundary close to every true unknown partition boundary. In many cases it is reasonable to assume that a practitioner would have an idea about the number of blocks that she

Figure 7.1: The figure illustrates where we expect to estimate a neighborhood of a node consistently. The blue region corresponds to the overlap between the true block (bounded by gray lines) and the estimated block (bounded by black lines). If the blue region is much larger than the orange regions, the additional bias introduced from the samples from the orange region will not considerably affect the estimation of the neighborhood of a node on the blue region. However, we cannot hope to consistently estimate the neighborhood of a node on the orange region.

wishes to discover. In that way, our procedure can be used to explore and visualize the data. It is still an open question to pick the tuning parameters in a data dependent way so that the number of blocks are estimated consistently.

7.3.3 Correct neighborhood selection

In this section, we give a result on the consistency of the neighborhood estimation. We will show that whenever the estimated block $\widehat{\mathcal{B}}^j$ is large enough, say $|\widehat{\mathcal{B}}^j| \geq r_n$ where $\{r_n\}_{n\geq 1}$ is an increasing sequence of numbers that satisfy $(r_n\lambda_2)^{-1}\lambda_1 \to 0$ and $r_n\lambda_2^2 \to \infty$ as $n \to \infty$, we have that $S(\widehat{\boldsymbol{\theta}}^j) = S(\boldsymbol{\beta}^k)$, where $\boldsymbol{\beta}^k$ is the true parameter on the true block $\mathcal{B}^k$ that overlaps $\widehat{\mathcal{B}}^j$ the most. Figure 7.1 illustrates this idea. The blue region in the figure denotes the overlap between the true block and the estimated block of the partition. The orange region corresponds to the overlap of the estimated block with a different true block. If the blue region is considerably larger than the orange region, the bias coming from the sample from the orange region will not be strong enough to disable us from selecting the correct neighborhood. On the other hand, since the orange region is small, as seen from Theorem 7.1, there is little hope of estimating the neighborhood correctly on that portion of the sample.

Suppose that we know that there is a solution to the optimization problem (7.2) with the partition boundary $\widehat{\mathcal{T}}$. Then that solution is also a minimizer of the following objective

$$\min_{\boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^{\widehat{B}}} \sum_{j\in\widehat{B}} ||\mathbf{X}_a^{\widehat{\mathcal{B}}^j} - \mathbf{X}_{\setminus a}^{\widehat{\mathcal{B}}^j}\boldsymbol{\theta}^j||_2^2 + 2\lambda_1 \sum_{j=2}^{\widehat{B}} ||\boldsymbol{\theta}^j - \boldsymbol{\theta}^{j-1}||_2 + 2\lambda_2 \sum_{j=1}^{\widehat{B}} |\widehat{\mathcal{B}}^j|||\boldsymbol{\theta}^j||_1. \tag{7.9}$$

Note that the problem (7.9) does not give a practical way of solving (7.2), but will help us to reason about the solutions of (7.2). In particular, while there may be multiple solutions to the problem (7.2), under some conditions, we can characterize the sparsity pattern of any solution that has specified partition boundaries $\widehat{\mathcal{T}}$.

Lemma 7.2. *Let $\widehat{\boldsymbol{\beta}}$ be a solution to* (7.2)*, with $\widehat{\mathcal{T}}$ being an associated estimate of the partition boundaries. Suppose that the subgradient vectors satisfy $|\widehat{y}_{i,b}| < 1$ for all $b \notin S(\widehat{\boldsymbol{\beta}}_{\cdot,i})$, then any other solution $\widetilde{\boldsymbol{\beta}}$ with the partition boundaries $\widehat{\mathcal{T}}$ satisfy $\widetilde{\beta}_{b,i} = 0$ for all $b \notin S(\widehat{\boldsymbol{\beta}}_{\cdot,i})$.*

The above Lemma states sufficient conditions under which the sparsity pattern of a solution with the partition boundary $\widehat{\mathcal{T}}$ is unique. Note, however, that there may other solutions to (7.2) that have different partition boundaries.

Now, we are ready to state the following theorem, which establishes that the correct neighborhood is selected on every sufficiently large estimated block of the partition.

Theorem 7.2. *Let $\{\mathbf{x}_i\}_{i\in[n]}$ be a sequence of observation according to the model in* (7.1)*. Assume that the conditions of theorem 7.1 are satisfied. In addition, suppose that* **A4** *also holds. Then, if $|\widehat{\mathcal{T}}| = B + 1$, it holds that*

$$\mathbb{P}[S^k = S(\widehat{\boldsymbol{\theta}}^k)] \xrightarrow{n\to\infty} 1, \qquad \forall k \in [B].$$

Under the assumptions of theorem 7.1 each estimated block is of size $\mathcal{O}(n)$. As a result, there are enough samples in each block to consistently estimate the underlying neighborhood structure. Observe that the neighborhood is consistently estimated at each $i \in \widehat{\mathcal{B}}^j \cap \mathcal{B}^j$ for all $j \in [B]$ and the error is made only on the small fraction of samples, when $i \notin \widehat{\mathcal{B}}^j \cap \mathcal{B}^j$, which is of order $\mathcal{O}(n\delta_n)$.

Using proposition 7.1 in place of theorem 7.1, it can be similarly shown that, for a large fraction of samples, the neighborhood is consistently estimated even in the case of over-segmentation. In particular, whenever there is a sufficiently large estimated block, with $|\widehat{\mathcal{B}}^k \cap \mathcal{B}^j| = \mathcal{O}(r_n)$, it holds that $S(\widehat{\mathcal{B}}^k) = S^j$ with probability tending to one.

7.4 Alternative estimation procedures

In this section, we discuss some alternative estimation methods to the neighborhood selection detailed in §7.2. We start describing how to solve the objective (7.2) for different penalties than the one given in (7.4). In particular, we describe how to minimize the objective when the ℓ_2 is replaced with the ℓ_q ($q \in \{1, \infty\}$) norm in (7.4). Next, we describe how to solve the penalized maximum likelihood objective with the temporal difference penalty. We do not provide statistical guarantees for solutions of these objective functions.

7.4.1 Neighborhood selection with modified penalty

We consider the optimization problem given in (7.2) with the following penalty

$$\mathrm{pen}_{\lambda_1,\lambda_2}(\boldsymbol{\beta}) := 2\lambda_1 \sum_{i=2}^{n} ||\boldsymbol{\beta}_{\cdot,i} - \boldsymbol{\beta}_{\cdot,i-1}||_q + 2\lambda_2 \sum_{i=1}^{n} \sum_{b\in\setminus a} |\beta_{b,i}|, \qquad q \in \{1, \infty\}. \tag{7.10}$$

We call the penalty in (7.10) the TD$_q$ penalty. As in §7.2.1, we apply the smoothing procedure to the first term in (7.10). Using the dual norm representation, we have

$$2\lambda_1 \sum_{i=2}^{n} ||\boldsymbol{\beta}_{\cdot,i} - \boldsymbol{\beta}_{\cdot,i-1}||_q = \max_{\mathbf{U}\in\mathcal{Q}^q} \langle\langle \mathbf{U}, 2\lambda_1 \boldsymbol{\beta}\mathbf{H} \rangle\rangle$$

where

$$\mathcal{Q}^1 := \{\mathbf{U} \in \mathbb{R}^{p-1 \times n-1} \ : \ ||\mathbf{U}_{\cdot,i}||_\infty \leq 1, \ \forall i \in [n-1]\}$$

and

$$\mathcal{Q}^\infty := \{\mathbf{U} \in \mathbb{R}^{p-1 \times n-1} \ : \ ||\mathbf{U}_{\cdot,i}||_1 \leq 1, \ \forall i \in [n-1]\}.$$

Next, we define smooth approximation to the norm as

$$\Psi_\mu^q(\boldsymbol{\beta}) := \max_{\mathbf{U} \in \mathcal{Q}^q} \langle\langle \mathbf{U}, 2\lambda_1 \boldsymbol{\beta}\mathbf{H} \rangle\rangle - \mu ||\mathbf{U}||_F^2 \tag{7.11}$$

where $\mu > 0$ is the smoothness parameter. Let

$$\mathbf{U}_\mu^q(\boldsymbol{\beta}) = \Pi_{\mathcal{Q}^q}\left(\frac{\lambda \boldsymbol{\beta}\mathbf{H}}{\mu}\right)$$

be the optimal solution of the maximization problem in (7.11), where $\Pi_{\mathcal{Q}^q}(\cdot)$ is the projection operator onto the set $\mathcal{Q}^q$. We observe that the projection on the ℓ_∞ unit ball can be easily obtained, while a fast algorithm for projection on the ℓ_1 unit ball can be found in [20]. The gradient can now be obtained as

$$\nabla \Psi_\mu^q(\boldsymbol{\beta}) = 2\lambda_1 \mathbf{U}_\mu^q(\boldsymbol{\beta})\mathbf{H}',$$

and we can proceed as in § 7.2.1 to arrive at the update (7.6).

We have described how to optimize (7.2) with the TD$_q$ penalty for $q \in \{1, 2, \infty\}$. Other ℓ_q norms are not commonly used in practice. We also note that a different procedure for $q = 1$ can be found in [133].

7.4.2 Penalized maximum likelihood estimation

In §7.2, we have related the problem of estimating zero elements of a precision matrix to a penalized regression procedure. Now, we consider estimating a sparse precision matrix using a penalized maximum likelihood approach. That is, we consider the following optimization procedure

$$\min_{\{\boldsymbol{\Omega}_i \succ \mathbf{0}\}_{i \in [n]}} \sum_{i \in [n]} (\operatorname{tr} \boldsymbol{\Omega}_i \mathbf{x}_i \mathbf{x}_i' - \log |\boldsymbol{\Omega}_i|) + \operatorname{pen}_{\lambda_1, \lambda_2}(\{\boldsymbol{\Omega}_t\}_{t \in [n]}) \tag{7.12}$$

where

$$\operatorname{pen}_{\lambda_1, \lambda_2}(\{\boldsymbol{\Omega}_i\}_{i \in [n]}) := 2\lambda_1 \sum_{i=2}^{n} ||\boldsymbol{\Omega}_i - \boldsymbol{\Omega}_{i-1}||_F + 2\lambda_2 \sum_{i=1}^{n} |\boldsymbol{\Omega}_i|_1.$$

In order to optimize (7.12) using the smoothing technique described in §7.2.1, we need to show that the gradient of the log-likelihood is Lipschitz continuous. The following Lemma establishes the desired result.

Lemma 7.3. *The function $f(\mathbf{A}) = \operatorname{tr} \mathbf{S}\mathbf{A} - \log |\mathbf{A}|$ has Lipschitz continuous gradient on the set $\{\mathbf{A} \in \mathcal{S}^p \ : \ \Lambda_{\min}(\mathbf{A}) \geq \gamma\}$, with Lipschitz constant $L = \gamma^{-2}$.*

Following [19], we can show that a solution to the optimization problem (7.12), on each estimated block, is indeed positive definite matrix with smallest eigenvalue bounded away from zero. This allows us to use the Nesterov's smoothing technique to solve (7.12).

Penalized maximum likelihood approach for estimating sparse precision matrix was proposed by [195]. Here, we have modified the penalty to perform estimation under the model (7.1). Although the parameters of the precision matrix can be estimated consistently using the penalized maximum likelihood approach, a number of theoretical results have shown that the neighborhood selection procedure requires lest stringent assumptions in order to estimate the underlying network consistently [135, 152]. We observe this phenomena in our simulation studies as well.

7.5 Numerical studies

In this section, we present a small numerical study on simulated networks. In all of our simulations studies we set $p = 30$ and $B = 3$ with $|\mathcal{B}_1| = 80$, $|\mathcal{B}_2| = 130$ and $|\mathcal{B}_3| = 90$, so that in total we have $n = 300$ samples. We consider two types of random networks: a chain and a nearest neighbor network. We measure the performance of the estimation procedure outlined in §7.2 on the following metrics: average precision of estimated edges, average recall of estimated edges and average F_1 score which combines the precision and recall score. The precision, recall and F_1 score are respectively defined as

$$
\begin{aligned}
precision &= \frac{1}{n}\sum_{i\in[n]} \frac{\sum_{a\in[p]}\sum_{b=a+1}^{p} \mathbb{1}\{(a,b) \in \widehat{E}_i \ \wedge (a,b) \in E_i\}}{\sum_{a\in[p]}\sum_{b=a+1}^{p} \mathbb{1}\{(a,b) \in \widehat{E}_i\}} \\
recall &= \frac{1}{n}\sum_{i\in[n]} \frac{\sum_{a\in[p]}\sum_{b=a+1}^{p} \mathbb{1}\{(a,b) \in \widehat{E}_i \ \wedge (a,b) \in E_i\}}{\sum_{a\in[p]}\sum_{b=a+1}^{p} \mathbb{1}\{(a,b) \in E_i\}} \\
F_1 &= \frac{2 * precision * recall}{precision + recall}.
\end{aligned}
$$

Furthermore, we report results on estimating the partition boundaries using $n^{-1}h(\widehat{\mathcal{T}}, \mathcal{T})$, where $h(\widehat{\mathcal{T}}, \mathcal{T})$ is defined in (7.7). Results are averaged over 50 simulation runs. We compare the TD-Lasso algorithm introduced in §7.2.1 against an oracle algorithm which exactly knows the true partition boundaries. In this case, it is only needed to run the algorithm of [135] on each block of the partition independently. We use a BIC criterion to select the tuning parameter for this oracle procedure as described in [146]. Furthermore, we report results using neighborhood selection procedures introduced in §7.4, which are denoted TD_1-Lasso and TD_∞-Lasso, as well as the penalized maximum likelihood procedure, which is denoted as $\text{LL}_{\max}$. We choose the tuning parameters for the penalized maximum likelihood procedure using the BIC procedure.

Chain networks We follow the simulation in [67] to generate a chain network (see Figure 7.2). This network corresponds to a tridiagonal precision matrix (after an appropriate permutation of nodes). The network is generated as follows. First, we choose to generate a random permutation

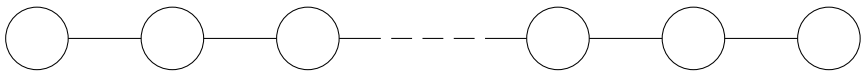


Figure 7.2: A chain graph

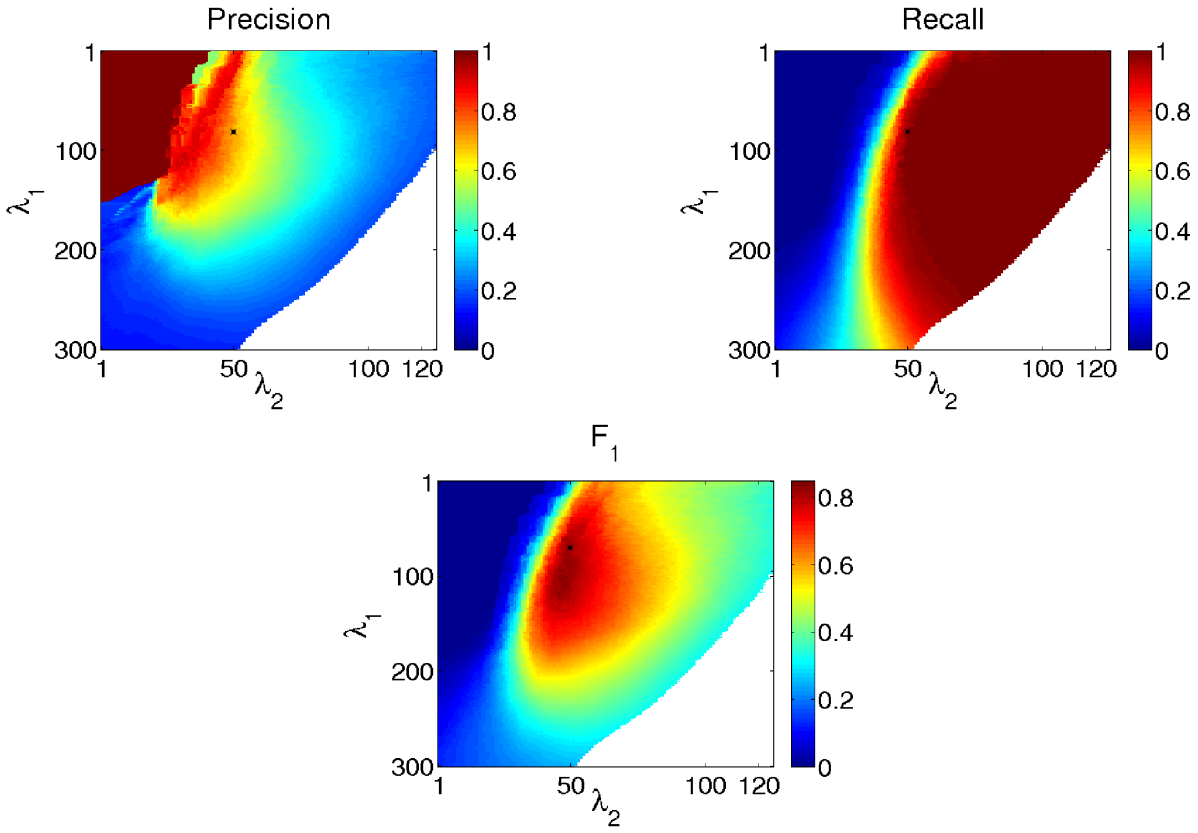


Figure 7.3: Plots of the precision, recall and F_1 scores as functions of the penalty parameters λ_1 and λ_2 for chain networks estimated using the TD-Lasso. The parameter λ_1 is obtained as $100 * 0.98^{50+i}$, where i indexes y-axis. The parameter λ_2 is computed as $285 * 0.98^{230+j}$, where j indexes x-axis. Black dot represents the selected tuning parameters. The white region of each plot corresponds to a region of the parameter space that we did not explore.

π of $[n]$. Next, the covariance matrix is generated as follows: the element at position (a, b) is chosen as $\sigma_{ab} = \exp(-|t_{\pi(a)} - t_{\pi(b)}|/2)$ where $t_1 < t_2 < \cdots < t_p$ and $t_i - t_{i-1} \sim \text{Unif}(0.5, 1)$ for $i = 2, \ldots, p$. This processes is repeated three times to obtain three different covariance matrices, from which we sample 80, 130 and 90 samples respectively.

For illustrative purposes, Figure 7.3 plots the precision, recall and F_1 score computed for different values of the penalty parameters λ_1 and λ_2. Table 7.1 shows the precision, recall and F_1 score for the parameters chosen using the BIC score described in 7.2.2, as well as the error in estimating the partition boundaries. The numbers in parentheses correspond to standard deviation. Due to the fact that there is some error in estimating the partition boundaries, we observe a decrease in performance compared to the oracle procedure that knows the correct position of the partition boundaries. Further, we observe that the neighborhood selection procedure estimate the graph structure more accurately than the maximum likelihood procedure. For TD$_1$-Lasso we do not report $n^{-1}h(\widehat{\mathcal{T}}, \mathcal{T})$, as the procedure does not estimate the partition boundaries.

Nearest neighbors networks We generate nearest neighbor networks following the procedure outlined in [119]. For each node, we draw a point uniformly at random on a unit square and com-

Table 7.1: Performance of different procedures when estimating chain networks

Method name	Precision	Recall	F_1 score	$n^{-1}h(\widehat{\mathcal{T}}, \mathcal{T})$
TD-Lasso	0.84 (0.04)	0.80 (0.04)	0.82 (0.04)	0.03 (0.01)
TD_1-Lasso	0.78 (0.05)	0.70 (0.03)	0.74 (0.04)	N/A
TD_∞-Lasso	0.83 (0.03)	0.80 (0.03)	0.81 (0.03)	0.03 (0.01)
$\text{LL}_{\max}$	0.72 (0.03)	0.65 (0.03)	0.68 (0.04)	0.06 (0.02)
Oracle procedure	0.97 (0.02)	0.89 (0.02)	0.93 (0.02)	0 (0)

pute the pairwise distances between nodes. Each node is then connected to 4 closest neighbors (see Figure 7.4). Since some of nodes will have more than 4 adjacent edges, we remove randomly edges from nodes that have degree larger than 4 until the maximum degree of a node in a network is 4. Each edge (a, b) in this network corresponds to a non-zero element in the precision matrix $\mathbf{\Omega}$, whose value is generated uniformly on $[-1, -0.5] \cup [0.5, 1]$. The diagonal elements of the precision matrix are set to a smallest positive number that makes the matrix positive definite. Next, we scale the corresponding covariance matrix $\mathbf{\Sigma} = \mathbf{\Omega}^{-1}$ to have diagonal elements equal to 1. This processes is repeated three times to obtain three different covariance matrices, from which we sample 80, 130 and 90 samples respectively.

For illustrative purposes, Figure 7.5 plots the precision, recall and F_1 score computed for different values of the penalty parameters λ_1 and λ_2. Table 7.2 shows the precision, recall, F_1 score and $n^{-1}h(\widehat{\mathcal{T}}, \mathcal{T})$ for the parameters chosen using the BIC score, together with their standard deviations. The results obtained for nearest neighbor networks are qualitatively similar to the results obtain for chain networks.

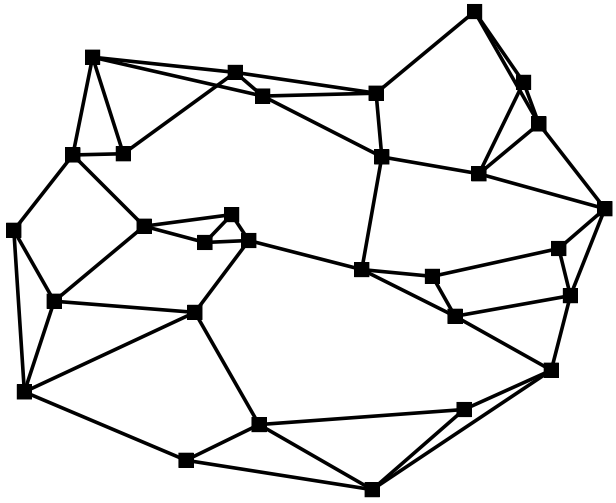


Figure 7.4: An instance of a random neighborhood graph with 30 nodes.

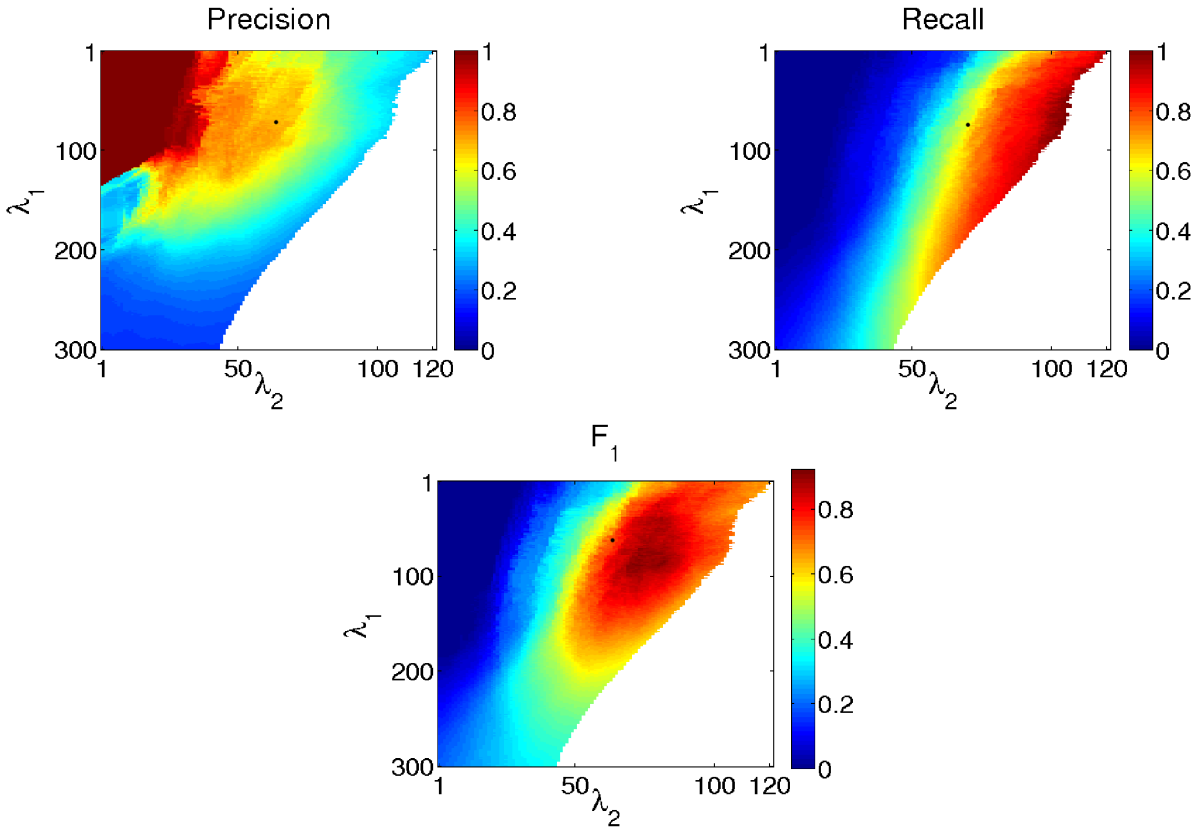


Figure 7.5: Plots of the precision, recall and F_1 scores as functions of the penalty parameters λ_1 and λ_2 for nearest neighbor networks estimated using the TD-Lasso. The parameter λ_1 is obtained as $100 * 0.98^{50+i}$, where i indexes y-axis. The parameter λ_2 is computed as $285 * 0.98^{230+j}$, where j indexes x-axis. Black dot represents the selected tuning parameters. The white region of each plot corresponds to a region of the parameter space that we did not explore.

Table 7.2: Performance of different procedure when estimating random nearest neighbor networks

Method name	Precision	Recall	F_1 score	$n^{-1}h(\widehat{\mathcal{T}}, \mathcal{T})$
TD-Lasso	0.79 (0.06)	0.76 (0.05)	0.77 (0.05)	0.04 (0.02)
TD_1-Lasso	0.70 (0.05)	0.68 (0.07)	0.69 (0.06)	N/A
TD_∞-Lasso	0.80 (0.06)	0.75 (0.06)	0.77 (0.06)	0.04 (0.02)
$\text{LL}_{\max}$	0.62 (0.08)	0.60 (0.06)	0.61 (0.06)	0.06 (0.02)
Oracle procedure	0.87 (0.05)	0.82 (0.05)	0.84 (0.04)	0 (0)

7.6 Discussion

We have addressed the problem of time-varying covariance selection when the underlying probability distribution changes abruptly at some unknown points in time. Using a penalized neighborhood selection approach with the fused-type penalty, we are able to consistently estimate times when the distribution changes and the network structure underlying the sample. The proof technique used to establish the convergence of the boundary fractions using the fused-type penalty is novel and constitutes an important contribution of the chapter. Furthermore, our procedure estimates the network structure consistently whenever there is a large overlap between the estimated blocks and the unknown true blocks of samples coming from the same distribution. The proof technique used to establish the consistency of the network structure builds on the proof for consistency of the neighborhood selection procedure, however, important modifications are necessary since the times of distribution changes are not known in advance. Applications of the proposed approach range from cognitive neuroscience, where the problem is to identify changing associations between different parts of a brain when presented with different stimuli, to system biology studies, where the task is to identify changing patterns of interactions between genes involved in different cellular processes. We conjecture that our estimation procedure is also valid in the high-dimensional setting when the number of variables p is much larger than the sample size n. We leave the investigations of the rate of convergence in the high-dimensional setting for a future work.

7.7 Technical Proofs

7.7.1 Proof of Lemma 7.1

For each $i \in [n]$, introduce a $(p-1)$-dimensional vector $\boldsymbol{\gamma}_i$ defined as

$$\boldsymbol{\gamma}_i = \begin{cases} \boldsymbol{\beta}_{\cdot,i} & \text{for } i = 1 \\ \boldsymbol{\beta}_{\cdot,i} - \boldsymbol{\beta}_{\cdot,i-1} & \text{otherwise} \end{cases}$$

and rewrite the objective (7.2) as

$$\begin{aligned} \{\widehat{\boldsymbol{\gamma}}^i\}_{i\in[n]} = \underset{\boldsymbol{\gamma}\in\mathbb{R}^{n\times p-1}}{\operatorname{argmin}} \sum_{i=1}^{n}\Bigg(x_{i,a} - \sum_{b\in\setminus a} x_{i,b}\sum_{j\leq i}\gamma_{j,b}\Bigg)^2 \\ + 2\lambda_1\sum_{i=2}^{n}||\boldsymbol{\gamma}_i||_2 + 2\lambda_2\sum_{i=1}^{n}\sum_{b\in\setminus a}\Bigg|\sum_{j\leq i}\gamma_{j,b}\Bigg|. \end{aligned} \tag{7.13}$$

A necessary and sufficient condition for $\{\widehat{\boldsymbol{\gamma}}_i\}_{i\in[n]}$ to be a solution of (7.13), is that for each $k \in [n]$ the $(p-1)$-dimensional zero vector, $\mathbf{0}$, belongs to the subdifferential of (7.13) with respect to $\boldsymbol{\gamma}_k$ evaluated at $\{\widehat{\boldsymbol{\gamma}}_i\}_{i\in[n]}$, that is,

$$\mathbf{0} = 2\sum_{i=k}^{n}(-\mathbf{x}_{i,\setminus a})\Bigg(x_{i,a} - \sum_{b\in\setminus a} x_{i,b}\widehat{\beta}^a_{b,i}\Bigg) + 2\lambda_1\widehat{\mathbf{z}}_k + 2\lambda_2\sum_{i=k}^{n}\widehat{\mathbf{y}}_i, \tag{7.14}$$

where $\widehat{\mathbf{z}}_k \in \partial||\cdot||_2(\widehat{\boldsymbol{\gamma}}_k)$, that is,

$$\widetilde{\mathbf{z}}_k = \begin{cases} \frac{\widetilde{\boldsymbol{\gamma}}_k}{||\widetilde{\boldsymbol{\gamma}}_k||_2} & \text{if } \widetilde{\boldsymbol{\gamma}}_k \neq 0 \\ \in \mathcal{B}_2(0,1) & \text{otherwise} \end{cases}$$

and for $k \leq i$, $\widehat{\mathbf{y}}_i \in \partial|\sum_{j\leq i}\widehat{\boldsymbol{\gamma}}_j|$, that is, $\mathbf{y}_i = \text{sign}(\sum_{j\leq i}\widehat{\boldsymbol{\gamma}}_j)$ with $\text{sign}(0) \in [-1,1]$. The Lemma now simply follows from (7.14).

7.7.2 Proof of Theorem 7.1

We build on the ideas presented in the proof of Proposition 5 in [94]. Using the union bound,

$$\mathbb{P}[\max_{j\in[B]}|T_j - \widehat{T}_j| > n\delta_n] \leq \sum_{j\in[B]}\mathbb{P}[|T_j - \widehat{T}_j| > n\delta_n]$$

and it is enough to show that $\mathbb{P}[|T_j - \widetilde{T}_j| > n\delta_n] \to 0$ for all $j \in [B]$. Define the event $A_{n,j}$ as

$$A_{n,j} := \left\{|T_j - \widehat{T}_j| > n\delta_n\right\}$$

and the event C_n as

$$C_n := \left\{\max_{j\in[B]}|\widehat{T}_j - T_j| < \frac{\Delta_{\min}}{2}\right\}.$$

We show that $\mathbb{P}[A_{n,j}] \to 0$ by showing that both $\mathbb{P}[A_{n,j}\cap C_n] \to 0$ and $\mathbb{P}[A_{n,j}\cap C_n^c] \to 0$ as $n \to \infty$. The idea here is that, in some sense, the event C_n is a good event on which the estimated boundary partitions and the true boundary partitions are not too far from each other. Considering the two cases will make the analysis simpler.

First, we show that $\mathbb{P}[A_{n,j}\cap C_n] \to 0$. Without loss of generality, we assume that $\widehat{T}_j < T_j$, since the other case follows using the same reasoning. Using (7.8) twice with $k = \widehat{T}_j$ and with $k = T_j$ and then applying the triangle inequality we have

$$2\lambda_1 \geq \left|\left|\sum_{i=\widehat{T}_j}^{T_j-1}\mathbf{x}_{i,\setminus a}\langle\mathbf{x}_{i,\setminus a}, \widehat{\boldsymbol{\beta}}_{\cdot,i} - \boldsymbol{\beta}_{\cdot,i}\rangle - \sum_{i=\widehat{T}_j}^{\widehat{T}_j-1}\mathbf{x}_{i,\setminus a}\epsilon_i + \lambda_2\sum_{i=\widehat{T}_j}^{T_j-1}\widehat{\mathbf{y}}_i\right|\right|_2. \tag{7.15}$$

Some algebra on the above display gives

$$\begin{aligned} 2\lambda_1 + (T_j - \widehat{T}_j)\sqrt{p}\lambda_2 \geq &\left|\left|\sum_{i=\widehat{T}_j}^{T_j-1}\mathbf{x}_{i,\setminus a}\langle\mathbf{x}_{i,\setminus a}, \boldsymbol{\theta}^j - \boldsymbol{\theta}^{j+1}\rangle\right|\right|_2 \\ &- \left|\left|\sum_{i=\widehat{T}_j}^{T_j-1}\mathbf{x}_{i,\setminus a}\langle\mathbf{x}_{i,\setminus a}, \boldsymbol{\theta}^{j+1} - \widehat{\boldsymbol{\theta}}^{j+1}\rangle\right|\right|_2 - \left|\left|\sum_{i=\widehat{T}_j}^{T_j-1}\mathbf{x}_{i,\setminus a}\epsilon_i\right|\right|_2 \\ =: &\,||R_1||_2 - ||R_2||_2 - ||R_3||_2. \end{aligned}$$

The above display occurs with probability one, so that the event $\{2\lambda_1 + (T_j - \widehat{T}_j)\sqrt{p}\lambda_2 \geq \frac{1}{3}||R_1||_2\} \cup \{||R_2||_2 \geq \frac{1}{3}||R_1||_2\} \cup \{||R_3||_2 \geq \frac{1}{3}||R_1||_2\}$ also occurs with probability one, which gives us the following bound

$$\begin{aligned}\mathbb{P}[A_{n,j} \cap C_n] &\leq \mathbb{P}[A_{n,j} \cap C_n \cap \{2\lambda_1 + (T_j - \widehat{T}_j)\sqrt{p}\lambda_2 \geq \frac{1}{3}||R_1||_2\}] \\ &+ \mathbb{P}[A_{n,j} \cap C_n \cap \{||R_2||_2 \geq \frac{1}{3}||R_1||_2\}] \\ &+ \mathbb{P}[A_{n,j} \cap C_n \cap \{||R_3||_2 \geq \frac{1}{3}||R_1||_2\}] \\ &=: \mathbb{P}[A_{n,j,1}] + \mathbb{P}[A_{n,j,2}] + \mathbb{P}[A_{n,j,3}].\end{aligned}$$

First, we focus on the event $A_{n,j,1}$. Using lemma 7.6, we can upper bound $\mathbb{P}[A_{n,j,1}]$ with

$$\mathbb{P}[2\lambda_1 + (T_j - \widehat{T}_j)\sqrt{p}\lambda_2 \geq \frac{\phi_{\min}}{27}(T_j - \widehat{T}_j)\xi_{\min}] + 2\exp(-n\delta_n/2 + 2\log n).$$

Since under the assumptions of the theorem $(n\delta_n\xi_{\min})^{-1}\lambda_1 \to 0$ and $\xi_{\min}^{-1}\sqrt{p}\lambda_2 \to 0$ as $n \to \infty$, we have that $\mathbb{P}[A_{n,j,1}] \to 0$ as $n \to \infty$.

Next, we show that the probability of the event $A_{n,j,2}$ converges to zero. Let $\overline{T}_j := \lfloor 2^{-1}(T_j + T_{j+1})\rfloor$. Observe that on the event C_n, $\widehat{T}_{j+1} > \overline{T}_j$ so that $\widehat{\boldsymbol{\beta}}_{\cdot,i} = \widehat{\boldsymbol{\theta}}^{j+1}$ for all $i \in [T_j, \overline{T}_j]$. Using (7.8) with $k = T_j$ and $k = \overline{T}_j$ we have that

$$2\lambda_1 + (\overline{T}_j - T_j)\sqrt{p}\lambda_2 \geq \left|\left|\sum_{i=T_j}^{\overline{T}_j - 1} \mathbf{x}_{i,\setminus a}\langle \mathbf{x}_{i,\setminus a}, \boldsymbol{\theta}^{j+1} - \widehat{\boldsymbol{\theta}}^{j+1}\rangle\right|\right|_2 - \left|\left|\sum_{i=T_j}^{\overline{T}_j - 1} \mathbf{x}_{i,\setminus a}\epsilon_i\right|\right|_2.$$

Using lemma 7.6 on the display above we have

$$||\boldsymbol{\theta}^{j+1} - \widehat{\boldsymbol{\theta}}^{j+1}||_2 \leq \frac{36\lambda_1 + 18(\overline{T}_j - T_j)\sqrt{p}\lambda_2 + 18||\sum_{i=T_j}^{\overline{T}_j - 1}\mathbf{x}_{i,\setminus a}\epsilon_i||_2}{(T_{j+1} - T_j)\phi_{\min}}, \tag{7.16}$$

which holds with probability at least $1 - 2\exp(-\Delta_{\min}/4 + 2\log n)$. We will use the above bound to deal with the event $\{||R_2||_2 \geq \frac{1}{3}||R_1||_2\}$. Using lemma 7.6, we have that $\phi_{\min}(T_j - \widehat{T}_j)\xi_{\min}/9 \leq ||R_1||_2$ and $||R_2||_2 \leq (T_j - \widehat{T}_j)9\phi_{\max}||\boldsymbol{\theta}^{j+1} - \widehat{\boldsymbol{\theta}}^{j+1}||_2$ with probability at least $1 - 4\exp(-n\delta_n/2 + 2\log n)$. Combining with (7.16), the probability $\mathbb{P}[A_{n,j,2}]$ is upper bounded by

$$\begin{aligned}&\mathbb{P}[c_1\phi_{\min}^2\phi_{\max}^{-1}\Delta_{\min}\xi_{\min} \leq \lambda_1] + \mathbb{P}[c_2\phi_{\min}^2\phi_{\max}^{-1}\xi_{\min} \leq \sqrt{p}\lambda_2] \\ &+ \mathbb{P}\left[c_3\phi_{\min}^2\phi_{\max}^{-1}\xi_{\min} \leq (\overline{T}_j - T_j)^{-1}\left|\left|\sum_{i=T_j}^{\overline{T}_j - 1}\mathbf{x}_{i,\setminus a}\epsilon_i\right|\right|_2\right] + c_4\exp(-n\delta_n/2 + 2\log n).\end{aligned}$$

Under the conditions of the theorem, the first term above converges to zero, since $\Delta_{\min} > n\delta_n$ and $(n\delta_n\xi_{\min})^{-1}\lambda_1 \to 0$. The second term also converges to zero, since $\xi_{\min}^{-1}\sqrt{p}\lambda_2 \to 0$. Using lemma 7.5, the third term converges to zero with the rate $\exp(-c_6\log n)$, since

$$(\xi_{\min}\sqrt{\Delta_{\min}})^{-1}\sqrt{p\log n} \to 0.$$

Combining all the bounds, we have that $\mathbb{P}[A_{n,j,2}] \to 0$ as $n \to \infty$.

Finally, we upper bound the probability of the event $A_{n,j,3}$. As before, $\phi_{\min}(T_j - \widehat{T}_j)\xi_{\min}/9 \leq ||R_1||_2$ with probability at least $1 - 2\exp(-n\delta_n/2 + 2\log n)$. This gives us an upper bound on $\mathbb{P}[A_{n,j,3}]$ as

$$\mathbb{P}\bigg[\frac{\phi_{\min}\xi_{\min}}{27} \leq \frac{||\sum_{i=\widehat{T}_j}^{T_j-1} \mathbf{x}_{i,\setminus a}\epsilon_i||_2}{T_j - \widehat{T}_j}\bigg] + 2\exp(-n\delta_n/2 + 2\log n),$$

which, using lemma 7.5, converges to zero as under the conditions of the theorem

$$(\xi_{\min}\sqrt{n\delta_n})^{-1}\sqrt{p\log n} \to 0.$$

Thus we have shown that $\mathbb{P}[A_{n,j,3}] \to 0$. Since the case when $\widehat{T}_j > T_j$ is shown similarly, we have proved that $\mathbb{P}[A_{n,j} \cap C_n] \to 0$ as $n \to \infty$.

We proceed to show that $\mathbb{P}[A_{n,j} \cap C_n^c] \to 0$ as $n \to \infty$. Recall that $C_n^c = \{\max_{j\in[B]} |\widehat{T}_j - T_j| \geq \Delta_{\min}/2\}$. Define the following events

$$\begin{aligned}
D_n^{(l)} &:= \Big\{\exists j \in [B],\ \widehat{T}_j \leq T_{j-1}\Big\} \cap C_n^c, \\
D_n^{(m)} &:= \Big\{\forall j \in [B],\ T_{j-1} < \widehat{T}_j < T_{j+1}\Big\} \cap C_n^c, \\
D_n^{(r)} &:= \Big\{\exists j \in [B],\ \widehat{T}_j \geq T_{j+1}\Big\} \cap C_n^c
\end{aligned}$$

and write $\mathbb{P}[A_{n,j} \cap C_n^c] = \mathbb{P}[A_{n,j} \cap D_n^{(l)}] + \mathbb{P}[A_{n,j} \cap D_n^{(m)}] + \mathbb{P}[A_{n,j} \cap D_n^{(r)}]$. First, consider the event $A_{n,j} \cap D_n^{(m)}$ under the assumption that $\widehat{T}_j \leq T_j$. Due to symmetry, the other case will follow in a similar way. Observe that

$$\begin{aligned}
&\mathbb{P}[A_{n,j} \cap D_n^{(m)}] \\
&\leq \mathbb{P}[A_{n,j} \cap \{(\widehat{T}_{j+1} - T_j) \geq \frac{\Delta_{\min}}{2}\} \cap D_n^{(m)}] \\
&\quad + \mathbb{P}[\{(T_{j+1} - \widehat{T}_{j+1}) \geq \frac{\Delta_{\min}}{2}\} \cap D_n^{(m)}] \\
&\leq \mathbb{P}[A_{n,j} \cap \{(\widehat{T}_{j+1} - T_j) \geq \frac{\Delta_{\min}}{2}\} \cap D_n^{(m)}] \\
&\quad + \sum_{k=j+1}^{B-1} \mathbb{P}[\{(T_k - \widehat{T}_k) \geq \frac{\Delta_{\min}}{2}\} \cap \{(\widehat{T}_{k+1} - T_k) \geq \frac{\Delta_{\min}}{2}\} \cap D_n^{(m)}].
\end{aligned} \tag{7.17}$$

We bound the first term in (7.17) and note that the other terms can be bounded in the same way. The following analysis is performed on the event $A_{n,j} \cap \{(\widehat{T}_{j+1} - T_j) \geq \Delta_{\min}/2\} \cap D_n^{(m)}$. Using (7.8) with $k = \widehat{T}_j$ and $k = T_j$, after some algebra (similar to the derivation of (7.15)) the following holds

$$||\boldsymbol{\theta}^j - \widehat{\boldsymbol{\theta}}^{j+1}||_2 \leq \frac{18\lambda_1 + 9(T_j - \widehat{T}_j)\sqrt{p}\lambda_2 + 9||\sum_{i=\widehat{T}_j}^{T_j-1} \mathbf{x}_{i,\setminus a}\epsilon_i||}{\phi_{\min}(T_j - \widehat{T}_j)},$$

with probability at least $1 - 2\exp(-n\delta_n/2 + 2\log n)$, where we have used lemma 7.6. Let $\overline{T}_j = \lfloor 2^{-1}(T_j + T_{j+1})\rfloor$. Using (7.8) with $k = \overline{T}_j$ and $k = T_j$ after some algebra (similar to the derivation of (7.16)) we obtain the following bound

$$||\boldsymbol{\theta}^j - \boldsymbol{\theta}^{j+1}||_2 \leq \frac{18\lambda_1 + 9(\overline{T}_j - T_j)\sqrt{p}\lambda_2 + 9||\sum_{i=T_j}^{\overline{T}_j - 1}\mathbf{x}_{i,\setminus a}\epsilon_i||_2}{\phi_{\min}(\overline{T}_j - T_j)} + 81\phi_{\max}\phi_{\min}^{-1}||\boldsymbol{\theta}^j - \widehat{\boldsymbol{\theta}}^{j+1}||_2,$$

which holds with probability at least $1 - c_1\exp(-n\delta_n/2 + 2\log n)$, where we have used lemma 7.6 twice. Combining the last two displays, we can upper bound the first term in (7.17) with

$$\mathbb{P}[\xi_{\min} n\delta_n \leq c_1\lambda_1] + \mathbb{P}[\xi_{\min} \leq c_2\sqrt{p}\lambda_2] + \mathbb{P}[\xi_{\min}\sqrt{n\delta_n} \leq c_3\sqrt{p\log n}] + c_4\exp(-c_5\log n),$$

where we have used lemma 7.5 to obtain the third term. Under the conditions of the theorem, all terms converge to zero. Reasoning similar about the other terms in (7.17), we can conclude that $\mathbb{P}[A_{n,j} \cap D_n^{(m)}] \to 0$ as $n \to \infty$.

Next, we bound the probability of the event $A_{n,j} \cap D_n^{(l)}$, which is upper bounded by

$$\mathbb{P}[D_n^{(l)}] \leq \sum_{j=1}^{B} 2^{j-1}\mathbb{P}[\max\{l \in [B] : \widehat{T}_l \leq T_{l-1}\} = j].$$

Observe that

$$\{\max\{l \in [B] : \widehat{T}_l \leq T_{l-1}\} = j\} \subseteq \bigcup_{l=j}^{B}\{T_j - \widehat{T}_j \geq \frac{\Delta_{\min}}{2}\} \cap \{\widehat{T}_{j+1} - T_j \geq \frac{\Delta_{\min}}{2}\}$$

so that we have

$$\mathbb{P}[D_n^{(l)}] \leq 2^{B-1}\sum_{j=1}^{B-1}\sum_{l>j}\mathbb{P}[\{T_l - \widehat{T}_l \geq \frac{\Delta_{\min}}{2}\} \cap \{\widehat{T}_{l+1} - T_l \geq \frac{\Delta_{\min}}{2}\}].$$

Using the same arguments as those used to bound terms in (7.17), we have that $\mathbb{P}[D_n^{(l)}] \to 0$ as $n \to \infty$ under the conditions of the theorem. Similarly, we can show that the term $\mathbb{P}[D_n^{(r)}] \to 0$ as $n \to \infty$. Thus, we have shown that $\mathbb{P}[A_{n,j} \cap C_n^c] \to 0$, which concludes the proof.

7.7.3 Proof of Lemma 7.2

Consider $\widehat{\mathcal{T}}$ fixed. The lemma is a simple consequence of the duality theory, which states that given the subdifferential $\widehat{\mathbf{y}}_i$ (which is constant for all $i \in \widehat{\mathcal{B}}^j$, $\widehat{\mathcal{B}}^j$ being an estimated block of the partition $\widehat{\mathcal{T}}$), all solutions $\{\check{\boldsymbol{\beta}}_{\cdot,i}\}_{i\in[n]}$ of (7.2) need to satisfy the complementary slackness condition $\sum_{b\in\setminus a}\widehat{y}_{i,b}\check{\beta}_{b,i} = ||\check{\boldsymbol{\beta}}_{\cdot,i}||_1$, which holds only if $\check{\beta}_{b,i} = 0$ for all $b \in \setminus a$ for which $|\widehat{y}_{i,b}| < 1$.

7.7.4 Proof of Theorem 7.2

Since the assumptions of theorem 7.1 are satisfied, we are going to work on the event

$$\mathcal{E} := \{\max_{j\in[B]} |\widehat{T}_j - T_j| \leq n\delta_n\}.$$

In this case, $|\widehat{\mathcal{B}}^k| = \mathcal{O}(n)$. For $i \in \widehat{\mathcal{B}}^k$, we write

$$x_{i,a} = \sum_{b\in S^j} x_{i,b}\theta_b^k + e_i + \epsilon_i$$

where $e_i = \sum_{b\in S} x_{i,b}(\beta_{b,i} - \theta_b^k)$ is the bias. Observe that $\forall i \in \widehat{\mathcal{B}}^k \cap \mathcal{B}^k$, the bias $e_i = 0$, while for $i \notin \widehat{\mathcal{B}}^k \cap \mathcal{B}^k$, the bias e_i is normally distributed with variance bounded by $M^2\phi_{\max}$ under the assumption **A1** and **A3**.

We proceed to show that $S(\widehat{\boldsymbol{\theta}}^k) \subset S^k$. Since $\widehat{\boldsymbol{\theta}}^k$ is an optimal solution of (7.2), it needs to satisfy

$$\begin{aligned}(\mathbf{X}_{\setminus a}^{\widehat{\mathcal{B}}^k})'\mathbf{X}_{\setminus a}^{\widehat{\mathcal{B}}^k}(\widehat{\boldsymbol{\theta}}^k - \boldsymbol{\theta}^k) - (\mathbf{X}_{\setminus a}^{\widehat{\mathcal{B}}^k})'(\mathbf{e}^{\widehat{\mathcal{B}}^k} + \boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}) \\ + \lambda_1(\widehat{\mathbf{z}}_{\widehat{T}_{k-1}} - \widehat{\mathbf{z}}_{\widehat{T}_k}) + \lambda_2|\widehat{\mathcal{B}}^k|\widehat{\mathbf{y}}_{\widehat{T}_{k-1}} = 0.\end{aligned} \tag{7.18}$$

Now, we will construct the vectors $\check{\boldsymbol{\theta}}^k, \check{\mathbf{z}}_{\widehat{T}_{k-1}}, \check{\mathbf{z}}_{\widehat{T}_k}$ and $\check{\mathbf{y}}_{\widehat{T}_{k-1}}$ that satisfy (7.18) and verify that the subdifferential vectors are dual feasible. Consider the following restricted optimization problem

$$\begin{aligned}\min_{\boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^{\widehat{B}};\ \boldsymbol{\theta}^k_{N^k}=\mathbf{0}} \sum_{j\in[\widehat{B}]} ||\mathbf{X}_a^{\widehat{\mathcal{B}}^j} - \mathbf{X}_{\setminus a}^{\widehat{\mathcal{B}}^j}\boldsymbol{\theta}^j||_2^2 \\ + 2\lambda_1\sum_{j=2}^{\widehat{B}} ||\boldsymbol{\theta}^j - \boldsymbol{\theta}^{j-1}||_2 + 2\lambda_2\sum_{j=1}^{\widehat{B}} |\widehat{\mathcal{B}}^j|||\boldsymbol{\theta}^j||_1,\end{aligned} \tag{7.19}$$

where the vector $\boldsymbol{\theta}^k_{N^k}$ is constrained to be $\mathbf{0}$. Let $\{\check{\boldsymbol{\theta}}^j\}_{j\in[\widehat{B}]}$ be a solution to the restricted optimization problem (7.19). Set the subgradient vectors as $\check{\mathbf{z}}_{\widehat{T}_{k-1}} \in \partial||\check{\boldsymbol{\theta}}^k - \check{\boldsymbol{\theta}}^{k-1}||$, $\check{\mathbf{z}}_{T_k} \in \partial||\check{\boldsymbol{\theta}}^{k+1} - \check{\boldsymbol{\theta}}^k||$ and $\check{\mathbf{y}}_{\widehat{T}_{k-1},S^k} = \text{sign}(\check{\boldsymbol{\theta}}^k_{S^k})$. Solve (7.18) for $\check{\mathbf{y}}_{\widehat{T}_{k-1},N^k}$. By construction, the vectors $\check{\boldsymbol{\theta}}^k, \check{\mathbf{z}}_{\widehat{T}_{k-1}}, \check{\mathbf{z}}_{\widehat{T}_k}$ and $\check{\mathbf{y}}_{\widehat{T}_{k-1}}$ satisfy (7.18). Furthermore, the vectors $\check{\mathbf{z}}_{\widehat{T}_{k-1}}$ and $\check{\mathbf{z}}_{\widehat{T}_k}$ are elements of the subdifferential, and hence dual feasible. To show that $\check{\boldsymbol{\theta}}^k$ is also a solution to (7.9), we need to show that $||\check{\mathbf{y}}_{\widehat{T}_{k-1},N^k}||_\infty \leq 1$, that is, that $\check{\mathbf{y}}^{\widehat{T}_{k-1}}$ is also dual feasible variable. Using lemma 7.2, if we show that $\check{\mathbf{y}}_{\widehat{T}_{k-1},N^k}$ is strict dual feasible, $||\check{\mathbf{y}}_{\widehat{T}_{k-1},N^k}||_\infty < 1$, then any other solution $\widehat{\check{\boldsymbol{\theta}}}^k$ to (7.9) will satisfy $\widehat{\check{\boldsymbol{\theta}}}^k_N = \mathbf{0}$.

From (7.18) we can obtain an explicit formula for $\check{\boldsymbol{\theta}}_{S^k}$

$$\begin{aligned}\check{\boldsymbol{\theta}}^k_{S^k} = \boldsymbol{\theta}^k_{S^k} + \left((\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k})'\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\right)^{-1}(\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}})'(\mathbf{e}^{\widehat{\mathcal{B}}^k} + \boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}) \\ - \left((\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k})'\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\right)^{-1}\left(\lambda_1(\check{\mathbf{z}}_{\widehat{T}_{k-1},S^k} - \check{\mathbf{z}}_{\widehat{T}_k,S^k}) + \lambda_2|\widehat{\mathcal{B}}^k|\check{\mathbf{y}}_{\widehat{T}_{k-1},S^k}\right).\end{aligned} \tag{7.20}$$

Recall that for large enough n we have that $|\widehat{\mathcal{B}}| > p$, so that the matrix $(\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k})'\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}$ is invertible with probability one. Plugging (7.20) into (7.18), we have that $||\check{\mathbf{y}}_{\widehat{T}_{k-1},N^k}||_\infty < 1$ if $\max_{b\in N^k}|Y_b| < 1$, where Y_b is defined to be

$$Y_b := \Big(\mathbf{X}_b^{\widehat{\mathcal{B}}^k}\Big)'\Bigg[\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\Big((\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k})'\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\Big)^{-1}\Big(\check{\mathbf{y}}_{\widehat{T}_{k-1},S^k} + \frac{\lambda_1(\widehat{\mathbf{z}}_{\widehat{T}_{k-1},S^k} - \widehat{\mathbf{z}}_{\widehat{T}_k,S^k})}{|\widehat{\mathcal{B}}^k|\lambda_2}\Big) + \mathbf{H}_{S^k}^{\widehat{\mathcal{B}}^k,\perp}\Big(\frac{\mathbf{e}^{\widehat{\mathcal{B}}^k} + \boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}}{|\widehat{\mathcal{B}}^k|\lambda_2}\Big)\Bigg] - \frac{\lambda_1(\check{z}_{\widehat{T}_{k-1},b} - \check{z}_{\widehat{T}_k,b})}{|\widehat{\mathcal{B}}^k|\lambda_2},$$

where $\mathbf{H}_{S^k}^{\widehat{\mathcal{B}}^k,\perp}$ is the projection matrix

$$\mathbf{H}_{S^k}^{\widehat{\mathcal{B}}^k,\perp} = \mathbf{I} - \mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\Big((\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k})'\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\Big)^{-1}\Big(\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\Big)'.$$

Let $\widetilde{\boldsymbol{\Sigma}}^k$ and $\widehat{\widetilde{\boldsymbol{\Sigma}}}^k$ be defined as

$$\widetilde{\boldsymbol{\Sigma}}^k = \frac{1}{|\widehat{\mathcal{B}}^k|}\sum_{i\in\widehat{\mathcal{B}}^k}\mathbb{E}[\mathbf{x}_{\setminus a}^i(\mathbf{x}_{\setminus a}^i)'] \quad \text{and} \quad \widehat{\widetilde{\boldsymbol{\Sigma}}}^k = \frac{1}{|\widehat{\mathcal{B}}^k|}\sum_{i\in\widehat{\mathcal{B}}^k}\mathbf{x}_{\setminus a}^i(\mathbf{x}_{\setminus a}^i)'.$$

For $i \in [n]$, we let $\mathcal{B}(i)$ index the block to which the sample i belongs to. Now, for any $b \in N^k$, we can write $x_b^i = \boldsymbol{\Sigma}_{bS^k}^{\mathcal{B}(i)}(\boldsymbol{\Sigma}_{S^kS^k}^{\mathcal{B}(i)})^{-1}\mathbf{x}_{S^k}^i + w_b^i$ where w_b^i is normally distributed with variance $\sigma_b^2 < 1$ and independent of $\mathbf{x}_{S^k}^i$. Let $\mathbf{F}_b \in \mathbb{R}^{|\widehat{\mathcal{B}}^k|}$ be the vector whose components are equal to $\boldsymbol{\Sigma}_{bS^k}^{\mathcal{B}(i)}(\boldsymbol{\Sigma}_{S^kS^k}^{\mathcal{B}(i)})^{-1}\mathbf{x}_{S^k}^i$, $i \in \widehat{\mathcal{B}}^k$, and $\mathbf{W}_b \in \mathbb{R}^{|\widehat{\mathcal{B}}^k|}$ be the vector with components equal to w_b^i. Using this notation, we write $Y_b = T_b^1 + T_b^2 + T_b^3 + T_b^4$ where

$$T_b^1 = \mathbf{F}_b'\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\Big((\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k})'\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\Big)^{-1}\Big(\check{\mathbf{y}}_{\widehat{T}_{k-1}} + \frac{\lambda_1(\check{\mathbf{z}}_{\widehat{T}_{k-1},S^k} - \check{\mathbf{z}}_{\widehat{T}_k,S^k})}{|\widehat{\mathcal{B}}^k|\lambda_2}\Big)$$

$$T_b^2 = \mathbf{F}_b'\mathbf{H}_{S^k}^{\widehat{\mathcal{B}}^k,\perp}\Big(\frac{\mathbf{e}^{\widehat{\mathcal{B}}^k} + \boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}}{|\widehat{\mathcal{B}}^k|\lambda_2}\Big)$$

$$T_b^3 = \Big(\widetilde{\mathbf{W}}_b\Big)'\Bigg[\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\Big((\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k})'\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k}\Big)^{-1}\Big(\check{\mathbf{y}}_{\widehat{T}_{k-1}} + \frac{\lambda_1(\check{\mathbf{z}}_{\widehat{T}_{k-1},S^k} - \check{\mathbf{z}}_{\widehat{T}_k,S^k})}{|\widehat{\mathcal{B}}^k|\lambda_2}\Big) + \mathbf{H}_{S^k}^{\widehat{\mathcal{B}}^k,\perp}\Big(\frac{\mathbf{e}^{\widehat{\mathcal{B}}^k} + \boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}}{|\widehat{\mathcal{B}}^k|\lambda_2}\Big)\Bigg], \text{ and}$$

$$T_b^4 = -\frac{\lambda_1(\check{z}_{\widehat{T}_{k-1},b} - \check{z}_{\widehat{T}_k,b})}{|\widehat{\mathcal{B}}^k|\lambda_2}.$$

We analyze each of the terms separately. Starting with the term T_b^1, after some algebra, we obtain that

$$
\begin{aligned}
&\mathbf{F}_b' \mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k} \left((\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k})' \mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k} \right)^{-1} \\
&\quad = \sum_{j \,:\, \widehat{\mathcal{B}}^k \cap \mathcal{B}^j \neq \emptyset} \frac{|\mathcal{B}^j \cap \widehat{\mathcal{B}}^k|}{|\widehat{\mathcal{B}}^k|} \mathbf{\Sigma}_{bS^k}^j (\mathbf{\Sigma}_{S^k S^k}^j)^{-1} (\widehat{\mathbf{\Sigma}}_{S^k S^k}^{\mathcal{B}^j \cap \widehat{\mathcal{B}}^k} - \mathbf{\Sigma}_{S^k S^k}^j) \left(\widehat{\widetilde{\mathbf{\Sigma}}}_{S^k S^k}^k \right)^{-1} \\
&\qquad + \widetilde{\mathbf{\Sigma}}_{bS^k}^k ((\widehat{\widetilde{\mathbf{\Sigma}}}_{S^k S^k}^k)^{-1} - (\widetilde{\mathbf{\Sigma}}_{S^k S^k}^k)^{-1}) \\
&\qquad + \widetilde{\mathbf{\Sigma}}_{bS^k}^k (\widetilde{\mathbf{\Sigma}}_{S^k S^k}^k)^{-1}.
\end{aligned} \tag{7.21}
$$

Recall that we are working on the event $\mathcal{E}$, so that

$$
\| \widetilde{\mathbf{\Sigma}}_{N^k S^k}^k (\widetilde{\mathbf{\Sigma}}_{S^k S^k}^k)^{-1} \|_\infty \xrightarrow{n \to \infty} \| \mathbf{\Sigma}_{N^k S^k}^k (\mathbf{\Sigma}_{S^k S^k}^k)^{-1} \|_\infty
$$

and

$$
(|\widehat{\mathcal{B}}^k| \lambda_2)^{-1} \lambda_1 (\check{\mathbf{z}}_{\widehat{T}_{k-1}, S^k} - \check{\mathbf{z}}_{\widehat{T}_k, S^k}) \xrightarrow{n \to \infty} 0
$$

element-wise. Using (7.25) we bound the first two terms in the equation above. We bound the first term by observing that for any j and any $b \in N^k$ and n sufficiently large

$$
\begin{aligned}
&\frac{|\mathcal{B}^j \cap \widehat{\mathcal{B}}^k|}{|\widehat{\mathcal{B}}^k|} ||\mathbf{\Sigma}_{bS^k}^j (\mathbf{\Sigma}_{S^k S^k}^j)^{-1} (\widehat{\mathbf{\Sigma}}_{S^k S^k}^{\mathcal{B}^j \cap \widehat{\mathcal{B}}^k} - \mathbf{\Sigma}_{S^k S^k}^j)||_\infty \\
&\qquad \leq \frac{|\mathcal{B}^j \cap \widehat{\mathcal{B}}^k|}{|\widehat{\mathcal{B}}^k|} ||\mathbf{\Sigma}_{bS^k}^j (\mathbf{\Sigma}_{S^k S^k}^j)^{-1}||_1 ||\widehat{\mathbf{\Sigma}}_{S^k S^k}^{\mathcal{B}^j \cap \widehat{\mathcal{B}}^k} - \mathbf{\Sigma}_{S^k S^k}^j||_\infty \\
&\qquad \leq C_1 \frac{|\mathcal{B}^j \cap \widehat{\mathcal{B}}^k|}{|\widehat{\mathcal{B}}^k|} ||\widehat{\mathbf{\Sigma}}_{S^k S^k}^{\mathcal{B}^j \cap \widehat{\mathcal{B}}^k} - \mathbf{\Sigma}_{S^k S^k}^j||_\infty \leq \epsilon_1
\end{aligned}
$$

with probability $1 - c_1 \exp(-c_2 \log n)$. Next, for any $b \in N^k$ we bound the second term as

$$
\begin{aligned}
&||\widetilde{\mathbf{\Sigma}}_{bS^k}^k ((\widehat{\widetilde{\mathbf{\Sigma}}}_{S^k S^k}^k)^{-1} - (\widetilde{\mathbf{\Sigma}}_{S^k S^k}^k)^{-1})||_1 \\
&\quad \leq C_2 ||(\widehat{\widetilde{\mathbf{\Sigma}}}_{S^k S^k}^k)^{-1} - (\widetilde{\mathbf{\Sigma}}_{S^k S^k}^k)^{-1})||_F \\
&\quad \leq C_2 ||\widetilde{\mathbf{\Sigma}}_{S^k S^k}^k||_F^2 ||\widehat{\widetilde{\mathbf{\Sigma}}}_{S^k S^k}^k - \widetilde{\mathbf{\Sigma}}_{S^k S^k}^k||_F + \mathcal{O}(||\widehat{\widetilde{\mathbf{\Sigma}}}_{S^k S^k}^k - \widetilde{\mathbf{\Sigma}}_{S^k S^k}^k||_F^2) \\
&\leq \epsilon_2
\end{aligned}
$$

with probability $1 - c_1 \exp(-c_2 \log n)$. Choosing ϵ_1, ϵ_2 sufficiently small and for n large enough, we have that $\max_b |T_b^1| \leq 1 - \alpha + o_p(1)$ under the assumption **A4**.

We proceed with the term T_b^2, which can be written as

$$
\begin{aligned}
T_b^2 = (|\widehat{\mathcal{B}}^k| \lambda_2)^{-1} &\left(\mathbf{\Sigma}_{bS^k}^k \left(\mathbf{\Sigma}_{S^k S^k}^k \right)^{-1} - \mathbf{F}_b' \mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k} \left((\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k})' \mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k} \right)^{-1} \right) \sum_{i \in \mathcal{B}^k \cap \widehat{\mathcal{B}}^k} \mathbf{x}_{S^k}^i \epsilon^i \\
+ (|\widehat{\mathcal{B}}^k| \lambda_2)^{-1} &\sum_{i \notin \mathcal{B}^k \cap \widehat{\mathcal{B}}^k} \left(\mathbf{\Sigma}_{bS^k}^{\mathcal{B}(i)} \left(\mathbf{\Sigma}_{S^k S^k}^{\mathcal{B}(i)} \right)^{-1} - \mathbf{F}_b' \mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k} \left((\mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k})' \mathbf{X}_{S^k}^{\widehat{\mathcal{B}}^k} \right)^{-1} \right) \mathbf{x}_{S^k}^i (e^i + \epsilon^i).
\end{aligned}
$$

Since we are working on the event $\mathcal{E}$ the second term in the above equation is dominated by the first term. Next, using (7.21) together with (7.25), we have that for all $b \in N^k$

$$||\mathbf{\Sigma}^k_{bS^k}\left(\mathbf{\Sigma}^k_{S^kS^k}\right)^{-1} - \mathbf{F}'_b\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k}\left((\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k})'\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k}\right)^{-1}||_2 = o_p(1).$$

Combining with Lemma 7.5, we have that under the assumptions of the theorem

$$\max_b |T^2_b| = o_p(1).$$

We deal with the term T^3_b by conditioning on $\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k}$ and $\boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}$, we have that $\mathbf{W}_b$ is independent of the terms in the squared bracket in T^3_b, since all $\check{\mathbf{z}}_{\widehat{T}_{k-1},S}$, $\check{\mathbf{z}}_{\widehat{T}_k,S}$ and $\widehat{\mathbf{y}}_{\widehat{T}_{k-1},S}$ are determined from the solution to the restricted optimization problem. To bound the second term, we observe that conditional on $\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k}$ and $\boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}$, the variance of T^3_b can be bounded as

$$\begin{aligned}\mathrm{Var}(T^3_b) &\leq ||\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k}\left((\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k})'\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k}\right)^{-1}\check{\eta}_{S^k} + \mathbf{H}^{\widehat{\mathcal{B}}^k,\perp}_{S^k}\Big(\frac{\mathbf{e}^{\widehat{\mathcal{B}}^k} + \boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}}{|\widehat{\mathcal{B}}^k|\lambda_2}\Big)||^2_2 \\ &\leq \check{\eta}'_{S^k}\left((\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k})'\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k}\right)^{-1}\check{\eta}_{S^k} + \Big\|\frac{\mathbf{e}^{\widehat{\mathcal{B}}^k} + \boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}}{|\widehat{\mathcal{B}}^k|\lambda_2}\Big\|^2_2,\end{aligned} \tag{7.22}$$

where

$$\check{\eta}_{S^k} = \Big(\check{\mathbf{y}}_{\widehat{T}_{k-1},S^k} + \frac{\lambda_1(\check{\mathbf{z}}_{\widehat{T}_{k-1},S^k} - \check{\mathbf{z}}_{\widehat{T}_k,S})}{|\widehat{\mathcal{B}}|\lambda_2}\Big).$$

Using lemma 7.6 and Young's inequality, the first term in (7.22) is upper bounded by

$$\frac{18}{|\widehat{\mathcal{B}}|\phi_{\min}}\left(s + \frac{2\lambda_1^2}{|\widehat{\mathcal{B}}|^2\lambda_2^2}\right)$$

with probability at least $1 - 2\exp(-|\widehat{\mathcal{B}}^k|/2 + 2\log n)$. Using lemma 7.4 we have that the second term is upper bounded by

$$\frac{(1+\delta')(1+M^2\phi_{\max})}{|\widehat{\mathcal{B}}|\lambda_2^2}$$

with probability at least $1 - \exp(-c_1|\widehat{\mathcal{B}}^k|\delta'^2 + 2\log n)$. Combining the two bounds, we have that $\mathrm{Var}(T^3_b) \leq c_1 s(|\widehat{\mathcal{B}}^k|)^{-1}$ with high probability, using the fact that $(|\widehat{\mathcal{B}}^k|\lambda_2)^{-1}\lambda_1 \to 0$ and $|\widehat{\mathcal{B}}^k|\lambda_2 \to \infty$ as $n \to \infty$. Using the bound on the variance of the term T^3_b and the Gaussian tail bound, we have that

$$\max_{b\in N} |T^3_b| = o_p(1).$$

Combining the results, we have that $\max_{b\in N^k}|Y_b| \leq 1 - \alpha + o_p(1)$. For a sufficiently large n, under the conditions of the theorem, we have shown that $\max_{b\in N}|Y_b| < 1$ which implies that $\mathbb{P}[S(\widehat{\boldsymbol{\theta}}^k) \subset S^k] \xrightarrow{n\to\infty} 1$.

Next, we proceed to show that $\mathbb{P}[S^k \subset S(\widehat{\boldsymbol{\theta}}^k)] \xrightarrow{n\to\infty} 1$. Observe that

$$\mathbb{P}[S^k \not\subset S(\widehat{\boldsymbol{\theta}}^k)] \leq \mathbb{P}[||\widehat{\boldsymbol{\theta}}^k_{S^k} - \boldsymbol{\theta}^k_{S^k}||_\infty \geq \theta_{\min}].$$

From (7.18) we have that $||\widehat{\boldsymbol{\theta}}^k_{S^k} - \boldsymbol{\theta}^k_{S^k}||_\infty$ is upper bounded by

$$\left|\left| \left(\frac{1}{|\widehat{\mathcal{B}}^k|} (\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k})' \mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k} \right)^{-1} \frac{1}{|\widehat{\mathcal{B}}^k|} (\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k})' (\widetilde{\mathbf{e}}^{\widehat{\mathcal{B}}^k} + \boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}) \right|\right|_\infty$$
$$+ \left|\left| \left((\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k})' \mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k} \right)^{-1} \left(\lambda_1 (\check{\mathbf{z}}_{\widehat{T}_{k-1}, S^k} - \check{\mathbf{z}}_{\widehat{T}_k, S^k}) - \lambda_2 |\widehat{\mathcal{B}}^{\widehat{\mathcal{B}}^k}| \check{\mathbf{y}}_{\widehat{T}_{k-1}, S^k} \right) \right|\right|_\infty .$$

Since $\widetilde{e}_i \neq 0$ only on $i \in \widehat{\mathcal{B}}^k \setminus \mathcal{B}^k$ and $n\delta_n / |\widehat{\mathcal{B}}^k| \to 0$, the term involving $\widetilde{\mathbf{e}}^{\widehat{\mathcal{B}}^k}$ is stochastically dominated by the term involving $\boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k}$ and can be ignored. Define the following terms

$$T_1 = \left(\frac{1}{|\widehat{\mathcal{B}}^k|} (\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k})' \mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k} \right)^{-1} \frac{1}{|\widehat{\mathcal{B}}^k|} (\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k})' \boldsymbol{\epsilon}^{\widehat{\mathcal{B}}^k},$$
$$T_2 = \left(\frac{1}{|\widehat{\mathcal{B}}^k|} (\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k})' \mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k} \right)^{-1} \frac{\lambda_1}{|\widehat{\mathcal{B}}^k| \lambda_2} (\check{\mathbf{z}}_{\widehat{T}_{k-1}, S^k} - \check{\mathbf{z}}_{\widehat{T}_k, S^k}),$$
$$T_3 = \left(\frac{1}{|\widehat{\mathcal{B}}^k|} (\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k})' \mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k} \right)^{-1} \check{\mathbf{y}}_{\widehat{T}_{k-1}, S^k}.$$

Conditioning on $\mathbf{X}^{\widehat{\mathcal{B}}^k}_{S^k}$, the term T_1 is a $|S^k|$ dimensional Gaussian with variance bounded by c_1/n with probability at least $1 - c_1 \exp(-c_2 \log n)$ using lemma 7.6. Combining with the Gaussian tail bound, the term $||T_1||_\infty$ can be upper bounded as

$$\mathbb{P}\left[||T_1||_\infty \geq c_1 \sqrt{\frac{\log s}{n}} \right] \leq c_2 \exp(-c_3 \log n).$$

Using lemma 7.6, we have that with probability greater than $1 - c_1 \exp(-c_2 \log n)$

$$||T_2||_\infty \leq ||T_2||_2 \leq c_3 \frac{\lambda_1}{|\widehat{\mathcal{B}}^k| \lambda_2} \to 0$$

under the conditions of theorem. Similarly $||T_3||_\infty \leq c_1 \sqrt{s}$, with probability greater than $1 - c_1 \exp(-c_2 \log n)$. Combining the terms, we have that

$$||\boldsymbol{\theta}^k - \widehat{\boldsymbol{\theta}}^k||_\infty \leq c_1 \sqrt{\frac{\log s}{n}} + c_2 \sqrt{s} \lambda_2$$

with probability at least $1 - c_3 \exp(-c_4 \log n)$. Since $\theta_{\min} = \Omega(\sqrt{\log(n)/n})$, we have shown that $S^k \subseteq S(\widehat{\boldsymbol{\theta}}^k)$. Combining with the first part, it follows that $S(\widehat{\boldsymbol{\theta}}^k) = S^k$ with probability tending to one.

7.7.5 Proof of Lemma 7.3

We have that $\nabla f(\mathbf{A}) = \mathbf{A}^{-1}$. Then

$$\begin{aligned} ||\nabla f(\mathbf{A}) - \nabla f(\mathbf{A}')||_F &= ||\mathbf{A}^{-1} - (\mathbf{A}')^{-1}||_F \\ &\leq \Lambda_{\max} \mathbf{A}^{-1} ||\mathbf{A} - \mathbf{A}'||_F \Lambda_{\max} \mathbf{A}^{-1} \\ &\leq \gamma^{-2} ||\mathbf{A} - \mathbf{A}'||_F. \end{aligned}$$

7.7.6 Proof of Proposition 7.1

The following proof follows main ideas already given in theorem 7.1. We provide only a sketch.

Given an upper bound on the number of partitions $B_{\max}$, we are going to perform the analysis on the event $\{\widehat{B} \leq B_{\max}\}$. Since

$$\mathbb{P}[h(\widehat{\mathcal{T}}, \mathcal{T}) \geq n\delta_n \;\big|\; \{\widehat{B} \leq B_{\max}\}] \leq \sum_{B'=B}^{B_{\max}} \mathbb{P}[h(\widehat{\mathcal{T}}, \mathcal{T}) \geq n\delta_n \;\big|\; \{|\widehat{\mathcal{T}}| = B'+1\}],$$

we are going to focus on $\mathbb{P}[h(\widehat{\mathcal{T}}, \mathcal{T}) \geq n\delta_n \;\big|\; \{|\widehat{\mathcal{T}}| = B'+1\}]$ for $B' > B$ (for $B' = B$ it follows from theorem 7.1 that $h(\widehat{\mathcal{T}}, \mathcal{T}) < n\delta_n$ with high probability). Let us define the following events

$$\begin{aligned}
\mathcal{E}_{j,1} &= \{\exists l \in [B'] \;:\; |\widehat{T}_l - T_j| \geq n\delta_n, |\widehat{T}_{l+1} - T_j| \geq n\delta_n \text{ and } \widehat{T}_l < T_j < \widehat{T}_{l+1}\} \\
\mathcal{E}_{j,2} &= \{\forall l \in [B'] \;:\; |\widehat{T}_l - T_j| \geq n\delta_n \text{ and } \widehat{T}_l < T_j\} \\
\mathcal{E}_{j,3} &= \{\forall l \in [B'] \;:\; |\widehat{T}_l - T_j| \geq n\delta_n \text{ and } \widehat{T}_l > T_j\}.
\end{aligned}$$

Using the above events, we have the following bound

$$\mathbb{P}[h(\widehat{\mathcal{T}}, \mathcal{T}) \geq n\delta_n \;\big|\; \{|\widehat{\mathcal{T}}| = B'+1\}] \leq \sum_{j \in [B]} \mathbb{P}[\mathcal{E}_{j,1}] + \mathbb{P}[\mathcal{E}_{j,2}] + \mathbb{P}[\mathcal{E}_{j,3}].$$

The probabilities of the above events can be bounded using the same reasoning as in the proof of theorem 7.1, by repeatedly using the KKT conditions given in (7.8). In particular, we can use the strategy used to bound the event $A_{n,j,2}$. Since the proof is technical and does not reveal any new insight, we omit the details.

7.7.7 Technical results

Lemma 7.4. *Let $\{\zeta^i\}_{i\in[n]}$ be a sequence of iid $\mathcal{N}(0,1)$ random variables. If $v_n \geq C \log n$, for some constant $C > 16$, then*

$$\mathbb{P}\Big[\bigcap_{\substack{1 \leq l < r \leq n \\ r-l > r_n}} \Big\{ \sum_{i=l}^{r} (\zeta^i)^2 \leq (1+C)(r-l+1) \Big\} \Big] \geq 1 - \exp(-c_1 \log n)$$

for some constant $c_1 > 0$.

Proof. For any $1 \leq l < r \leq n$, with $r - l > v_n$ we have

$$\begin{aligned}
\mathbb{P}\Big[\sum_{i=l}^{r} (\zeta^i)^2 \geq (1+C)(r-l+1) \Big] &\leq \exp(-C(r-l+1)/8) \\
&\leq \exp(-C \log n/8)
\end{aligned}$$

using (7.26). The lemma follows from an application of the union bound. □

Lemma 7.5. *Let $\{\mathbf{x}_i\}_{i\in[n]}$ be independent observations from* (7.1) *and let $\{\epsilon_i\}_{i\in[n]}$ be independent $\mathcal{N}(0,1)$. Assume that* **A1** *holds. If $v_n \geq C\log n$ for some constant $C > 16$, then*

$$\mathbb{P}\Big[\bigcap_{j\in[B]}\bigcap_{\substack{l,r\in\mathcal{B}^j\\ r-l>v_n}}\Big\{\frac{1}{r-l+1}\Big|\Big|\sum_{i=l}^{r}\mathbf{x}_i\epsilon_i\Big|\Big|_2 \leq \frac{\phi_{\max}^{1/2}\sqrt{1+C}}{\sqrt{r-l+1}}\sqrt{p(1+C\log n)}\Big\}\Big] \\ \geq 1 - c_1\exp(-c_2\log n),$$

for some constants $c_1, c_2 > 0$.

Proof. Let $\mathbf{\Sigma}^{1/2}$ denote the symmetric square root of the covariance matrix $\mathbf{\Sigma}_{SS}$ and let $\mathcal{B}(i)$ denote the block $\mathcal{B}^j$ of the true partition such that $i \in \mathcal{B}^j$. With this notation, we can write $\mathbf{x}_i = \left(\mathbf{\Sigma}^{\mathcal{B}(i)}\right)^{1/2}\mathbf{u}_i$ where $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. For any $l \leq r \in \mathcal{B}^j$ we have

$$\Big|\Big|\sum_{i=l}^{r}\mathbf{x}_i\epsilon_i\Big|\Big|_2 = \Big|\Big|\sum_{i=l}^{r}\left(\mathbf{\Sigma}^{j}\right)^{1/2}\mathbf{u}_i\epsilon_i\Big|\Big|_2 \leq \phi_{\max}^{1/2}\Big|\Big|\sum_{i=l}^{r}\mathbf{u}_i\epsilon_i\Big|\Big|_2.$$

Conditioning on $\{\epsilon_i\}_i$, for each $b \in [p]$, $\sum_{i=l}^{r}u_{i,b}\epsilon_i$ is a normal random variable with variance $\sum_{i=l}^{r}(\epsilon_i)^2$. Hence, $||\sum_{i=l}^{r}\mathbf{u}_i\epsilon_i||_2^2/(\sum_{i=l}^{r}(\epsilon_i)^2)$ conditioned on $\{\epsilon_i\}_i$ is distributed according to χ_p^2 and

$$\mathbb{P}\Big[\frac{1}{r-l+1}\Big|\Big|\sum_{i=l}^{r}\mathbf{x}_i\epsilon_i\Big|\Big|_2 \geq \frac{\phi_{\max}^{1/2}\sqrt{\sum_{i=l}^{r}(\epsilon_i)^2}}{r-l+1}\sqrt{p(1+C\log n)}\;\Big|\;\{\epsilon_i\}_{i=l}^{r}\Big] \\ \leq \mathbb{P}[\chi_p^2 \geq p(1+C\log n)] \leq \exp(-C\log n/8),$$

where the last inequality follows from (7.26). Using lemma 7.4, for all $l, r \in \mathcal{B}^j$ with $r - l > v_n$ the quantity $\sum_{i=l}^{r}(\epsilon_i)^2$ is bounded by $(1+C)(r-l+1)$ with probability at least $1-\exp(-c_1\log n)$, which gives us the following bound

$$\mathbb{P}\Big[\bigcap_{j\in[B]}\bigcap_{\substack{l,r\in\mathcal{B}^j\\ r-l>v_n}}\Big\{\frac{1}{r-l+1}\Big|\Big|\sum_{i=l}^{r}\mathbf{x}_i\epsilon_i\Big|\Big|_2 \leq \frac{\phi_{\max}^{1/2}\sqrt{1+C}}{\sqrt{r-l+1}}\sqrt{p(1+C\log n)}\Big\}\Big] \\ \geq 1 - c_1\exp(-c_2\log n).$$

□

Lemma 7.6. *Let $\{\mathbf{x}_i\}_{i\in[n]}$ be independent observations from* (7.1)*. Assume that* **A1** *holds. Then for any $v_n > p$,*

$$\mathbb{P}\left[\max_{\substack{1\leq l<r\leq n\\ r-l>v_n}}\Lambda_{\max}\left(\frac{1}{r-l+1}\sum_{i=l}^{r}\mathbf{x}_i\left(\mathbf{x}_i\right)'\right) \geq 9\phi_{\max}\right] \leq 2n^2\exp(-v_n/2)$$

and

$$\mathbb{P}\left[\min_{\substack{1\leq l<r\leq n\\ r-l>v_n}}\Lambda_{\min}\left(\frac{1}{r-l+1}\sum_{i=l}^{r}\mathbf{x}_i\left(\mathbf{x}_i\right)'\right) \leq \phi_{\min}/9\right] \leq 2n^2\exp(-v_n/2).$$

Proof. For any $1 \leq l < r \leq n$, with $r - l \geq v_n$ we have

$$\mathbb{P}\left[\Lambda_{\max}\left(\frac{1}{r-l+1}\sum_{i=l}^{r}\mathbf{x}_i\left(\mathbf{x}_i\right)'\right) \geq 9\phi_{\max}\right] \leq 2\exp(-(r-l+1)/2)$$
$$\leq 2\exp(-v_n/2)$$

using (7.23), convexity of $\Lambda_{\max}(\cdot)$ and **A1**. The lemma follows from an application of the union bound. The other inequality follows using a similar argument. □

7.7.8 A collection of known results

This section collects some known results that we have used in the chapter. We start by collecting some results on the eigenvalues of random matrices. Let $\mathbf{x} \overset{iid}{\sim} \mathcal{N}(0, \mathbf{\Sigma})$, $i \in [n]$, and $\widehat{\mathbf{\Sigma}} = n^{-1}\sum \mathbf{x}_i(\mathbf{x}_i)'$ be the empirical covariance matrix. Denote the elements of the covariance matrix $\mathbf{\Sigma}$ as $[\sigma_{ab}]$ and of the empirical covariance matrix $\widehat{\mathbf{\Sigma}}$ as $[\widehat{\sigma}_{ab}]$.

Using standard results on concentration of spectral norms and eigenvalues [54], [190] derives the following two crude bounds that can be very useful. Under the assumption that $p < n$,

$$\mathbb{P}[\Lambda_{\max}(\widehat{\mathbf{\Sigma}}) \geq 9\phi_{\max}] \leq 2\exp(-n/2) \tag{7.23}$$
$$\mathbb{P}[\Lambda_{\min}(\widehat{\mathbf{\Sigma}}) \leq \phi_{\min}/9] \leq 2\exp(-n/2). \tag{7.24}$$

From Lemma A.3. in [25] we have the following bound on the elements of the covariance matrix

$$\mathbb{P}[|\widehat{\sigma}_{ab} - \sigma_{ab}| \geq \epsilon] \leq c_1\exp(-c_2 n\epsilon^2), \qquad |\epsilon| \leq \epsilon_0 \tag{7.25}$$

where c_1 and c_2 are positive constants that depend only on $\Lambda_{\max}(\mathbf{\Sigma})$ and ϵ_0.

Next, we use the following tail bound for χ^2 distribution from [123], which holds for all $\epsilon > 0$,

$$\mathbb{P}[\chi^2_n > n + \epsilon] \leq \exp(-\frac{1}{8}\min(\epsilon, \frac{\epsilon^2}{n})). \tag{7.26}$$

Chapter 8

Conditional Estimation of Covariance Models

In the previous chapters, we discussed estimation of network structures as a function of time, however, in many applications, it is more natural to think of a network changing as a function of some other random variable. In this chapter, we focus on conditional estimation of network structures. We start by motivating the problem by few real world applications.

Consider the problem of gene network inference in systems biology, which is of increasing importance in drug development and disease treatment. A gene network is commonly represented as a fixed network, with edge weights denoting strength of associations between genes. Realistically, the strength of associations between genes can depend on many covariates such as blood pressure, sugar levels, and other body indicators; however, biologists have very little knowledge on how various factors affect strength of associations. Ignoring the influence of different factors leads to estimation procedures that overlook important subtleties of the regulatory networks. Consider another problem in quantitative finance, for which one wants to understand how different stocks are associated and how these associations vary with respect to external factors to help investors construct a diversified portfolio. The rule of *Diversification*, formalized by Modern Portfolio Theory [132], dictates that risk can be reduced by constructing a portfolio out of uncorrelated assets. However, it also assumes that the associations between assets are fixed (which is highly unrealistic) and a more robust approach to modeling assets would take into account how their associations change with respect to economic indicators, such as, gross domestic product (GDP), oil price or inflation rate. Unfortunately, there is very little domain knowledge on the exact relationship between economic indicators and associations between assets, which motivates the problem of *conditional covariance selection* we intend to investigate in this chapter.

8.1 Motivation

Let $\mathbf{X} \in \mathbb{R}^p$ denote a p-dimensional random vector representing genes or stock values, and $Z \in \mathbb{R}$ denote an index random variable representing some body factor or economic indicator of interest. Both of the above mentioned problems in biology and finance can be modeled as

inferring non-zero partial correlations between different components of the random vector $\mathbf{X}$ conditioned on a particular value of the index variable $Z = z$. We assume that the value of partial correlations change with z, however, the set of non-zero partial correlations is constant with respect to z. Let $\mathbf{\Sigma}(z) = \mathrm{Cov}(\mathbf{X}|Z = z)$ denote the *conditional* covariance of $\mathbf{X}$ given Z, which we assume to be positive definite, and let $\mathbf{\Omega}(z) = \mathbf{\Sigma}(z)^{-1}$ denote the conditional precision matrix. The structure of non-zero components of the matrix $\mathbf{\Omega}(z)$ tells us a lot about associations between different components of the vector $\mathbf{X}$, since the elements of $\mathbf{\Omega}(z)$ correspond to partial correlation coefficients. In this section we address the challenge of selecting non-zero components of $\mathbf{\Omega}(z)$ from noisy samples. Usually, very little is known about the relationship between the index variable Z and associations between components of the random variable $\mathbf{X}$; so, we develop a nonparametric method for estimating the non-zero elements of $\mathbf{\Omega}(z)$. Specifically, we develop a new method based on ℓ_1/ℓ_2 penalized kernel smoothing, that is able to estimate the functional relationship between the index Z and components of $\mathbf{\Omega}(z)$ with minimal assumptions on the distribution $(\mathbf{X}, Z)$ and only smoothness assumption on $z \mapsto \mathbf{\Omega}(z)$. In addition to developing an estimation procedure that works with minimal assumptions, we also focus on statistical properties of the estimator in the high-dimensional setting, where the number of dimensions p is comparable or even larger than the sample size. Ubiquity of high-dimensionality in many real world data forces us to carefully analyze statistical properties of the estimator, that would otherwise be apparent in a low-dimensional setting.

Our problem setting, as stated above, should be distinguished from the classical problem of covariance selection, introduced in the seminal paper by Dempster [47]. In the classical setting, the main goal is to select non-zero elements of the precision matrix; however, the precision matrix does not vary with respect to the index variables. As mentioned before, non-zero elements of the precision matrix correspond to partial correlation coefficients, which encode associations among sets of random variables.

There are only few references for work on nonparametric models for conditional covariance and precision matrices. [193] develop a kernel estimator of the conditional covariance matrix based on the local-likelihood approach. Since their approach does not perform estimation of non-zero elements in the precision matrix, it is suitable in low-dimensions. Other related work includes nonparametric estimation of the conditional variance function in longitudinal studies (see [65, 150] and references within).

In summary, here are the highlights of our this chapter. Our main contribution is a new nonparametric model for sparse conditional precision matrices, and the ℓ_1/ℓ_2 penalized kernel estimator for the proposed model. The estimation procedure was developed under minimal assumptions, with the focus on the high-dimensional setting, where the number of dimensions is potentially larger than the sample size. A modified Bayesian Information Criterion (BIC) is given that can be used to correctly identify the set of non-zero partial correlations. Finally, we demonstrate the performance of the algorithm on synthetic data and analyze the associations between the set of stocks in the S&P 500 as a function of oil price.

The work presented here is related, but different from estimation of time-varying networks. As we will see, the estimation procedure is based on the neighborhood selection described in §2 and is a slight modification of neighborhood estimation used to estimate time-varying networks. However, the difference comes from the fact that the variable we are conditioning is not fixed, but a random quantity.

8.2 The Model

Let $\mathbf{X} = (X_1, \ldots, X_p)^T \in \mathbb{R}^p$ be a p-dimensional random vector (representing gene expressions or stock values) and let random variable $Z \in [0, 1]$ be an associated univariate index (representing a body factor or an economy index). We will estimate associations between different components of $\mathbf{X}$ conditionally on Z. For simplicity of presentation, we assume that the index variable can be scaled into the interval $[0, 1]$ and, furthermore, we assume that it is a scalar variable. The kernel smoothing method, to be introduced, can be easily extended to multivariate Z. However, such an extension may only be practical in limited cases, due to the curse of dimensionality [125]. Throughout the chapter, we assume that $\mathbb{E}[\mathbf{X}|Z = z] = 0$ for all $z \in [0, 1]$. In practice, one can easily estimate the conditional mean of $\mathbf{X}$ given Z using local polynomial fitting [60] and subtract it from $\mathbf{X}$. We denote the conditional covariance matrix of $\mathbf{X}$ given Z as $\mathbf{\Sigma}(z) := \mathrm{Cov}(\mathbf{X}|Z = z) = (\sigma_{uv}(z))_{u,v\in[p]}$, where we use $[p]$ to denote the set $\{1, \ldots, p\}$. Assuming that $\mathbf{\Sigma}(z)$ is positive definite, for all $z \in [0, 1]$, the conditional precision matrix is given as $\mathbf{\Omega}(z) := \mathbf{\Sigma}(z)^{-1} = (\omega_{uv}(z))_{u,v\in[p]}$. Elements $(\omega_{uv}(z))_{u,v\in[p]}$ are smooth, but unknown functions of z.

With the notation introduced above, the problem of conditional covariance selection, e.g., recovering the strength of association between stocks as a function of oil price, or association between gene expressions as a function of blood pressure, can be formulated as estimating the non-zero elements in the conditional precision matrix $\mathbf{\Omega}(z)$. As mentioned before, association between different components of $\mathbf{X}$ can be expressed using the partial correlation coefficients, which are directly related to the elements of precision matrix as follows; the partial correlation $\rho_{uv}(z)$ between X_u and X_v $(u, v \in [p])$ given $Z = z$ can be computed as

$$\rho_{uv}(z) = -\frac{\omega_{uv}(z)}{\sqrt{\omega_{uu}(z)\omega_{vv}(z)}}.$$

The above equation confirms that the non-zero partial correlation coefficients can be selected by estimating non-zero elements of the precision matrix. Let $S := \{(u, v) : \int_{[0,1]} \omega_{uv}^2(z)dz > 0,\ u \neq v\}$ denote the set of non-zero partial correlation coefficients, which we assume to be constant with respect to z, i.e., we assume that the associations are fixed, but their strength can vary with respect to the index z. Furthermore, we assume that the number of non-zero partial correlation coefficients, $s := |S|$, is small. This is a reasonable assumption for many problems, e.g., in biological systems a gene usually interacts with only a handful of other genes. In the following paragraphs, we relate the partial correlation coefficients to a regression problem, and present a computationally efficient method for estimating non-zero elements of the precision matrix based on this insight. In particular, we extend the neighborhood selection procedure discussed in §2.

For each component X_u $(u \in [p])$ we set up a regression model, where X_u is the response variable, and all the other components are the covariates. Let $\mathbf{X}_{\setminus u} := \{X_v : v \neq u, v \in [p]\}$. Then we have the following regression model

$$X_u = \sum_{v\neq u} X_v b_{uv}(z) + \epsilon_u(z), \quad u \in [p],$$

with $\epsilon_u(z)$ being uncorrelated with $\mathbf{X}_{\setminus u}$ if and only if

$$b_{uv}(z) = -\frac{\omega_{uv}(z)}{\omega_{uu}(z)} = \rho_{uv}(z)\sqrt{\frac{\omega_{vv}(z)}{\omega_{uu}(z)}}.$$

We propose a locally weighted kernel estimator of the non-zero partial correlations. Let $\mathcal{D}^n = \{(\mathbf{x}^i, z^i)\}_{i\in[n]}$ be an independent sample of n realizations of $(\mathbf{X}, Z)$. For each $u \in [p]$, we define the loss function

$$\mathcal{L}_u(\mathbf{B}_u; \mathcal{D}^n) := \sum_{z\in\{z^j\}_{j\in[n]}} \sum_{i\in[n]} \big(x_u^i - \sum_{v\neq u} x_v^i b_{uv}(z)\big)^2 K_h(z - z^i) + 2\lambda \sum_{v\neq u} ||b_{uv}(\cdot)||_2 \tag{8.1}$$

where $\mathbf{B}_u = (\mathbf{b}_u(z^1), \ldots, \mathbf{b}_u(z^n))$, $\mathbf{b}_u(z^j) \in \mathbb{R}^{p-1}$, $K_h(z - z^i) = K(\frac{|z-z^i|}{h})$ is a symmetric density function with bounded support that defines local weights, h denotes the bandwidth, λ is the penalty parameter and $||b_{uv}(\cdot)||_2 := \sqrt{\sum_{z\in\{z^j\}_{j\in[n]}} b_{uv}(z)^2}$. Define $\widehat{\mathbf{B}}_u$ as a minimizer of the loss

$$\widehat{\mathbf{B}}_u := \underset{\mathbf{B}\in\mathbb{R}^{p-1\times n}}{\operatorname{argmin}} \mathcal{L}_u(\mathbf{B}; \mathcal{D}^n). \tag{8.2}$$

Combining $\{\widehat{\mathbf{B}}_u\}_{u\in[p]}$ gives an estimator

$$\widehat{S} := \{(u, v) : \max\{||\widehat{b}_{uv}(\cdot)||_2, ||\widehat{b}_{vu}(\cdot)||_2\} > 0\}$$

of the non-zero elements of the precision matrix.

In (8.1), the ℓ_1/ℓ_2 norm is used to penalize model parameters. This norm is commonly used in the Group Lasso [194]. In our case, since we assume the set of non-zero elements S, of the precision matrix, to be fixed with respect to z, the ℓ_2 norm is a natural way to shrink the whole group of coefficients $\{b_{uv}(z^i)\}_{i\in[n]}$ to zero. Note that the group consists of the same element, say (u, v), of the precision matrix for different values of z.

8.3 Optimization algorithm

In this section, we detail an efficient optimization algorithm that can be used to solve the problem given in (8.2). Given that the optimization problem is convex, a variety of techniques can be used to solve it. A particularly efficient optimization algorithm has been devised for ℓ_1/ℓ_2 penalized problems, that is based on the group-coordinate descent and is referred to as the active-shooting algorithm [72, 146]. A modification of the procedure, suitable for our objective, is outlined in Algorithm 3, which we now explain.

We point out that the group coordinate descent will converge to an optimum, since the loss function is smooth and the penalty term in (8.1) decomposes across different rows of the matrix $\mathbf{B}_u$ [72]. Now, we derive an update for row v, while keeping all other rows of $\mathbf{B}_u$ fixed. Let $\{\widetilde{b}_{uv}(z^j)\}_{j\in[n]}$ be a minimizer of

$$\mathcal{L}_u^v(\{b_{uv}(z^j)\}_{j\in[n]}; \mathcal{D}^n) := \sum_{z\in\{z^j\}_{j\in[n]}} \sum_{i\in[n]} \big(r_{uv}^i(z) - x_v^i b_{uv}(z)\big)^2 K_h(z - z^i) + 2\lambda\, ||b_{uv}(\cdot)||_2, \tag{8.3}$$

Input: Data $\mathcal{D}^n = \{\mathbf{x}^i, z^i\}_{i\in[n]}$, initial solution $\widetilde{\mathbf{B}}_u^{(0)}$
Output: Solution $\widehat{\mathbf{B}}_u$ to Eq. (8.2)

$\mathcal{A} := \{v \in [p] \setminus u : ||\widetilde{b}_{uv}^{(0)}(\cdot)||_2 > 0\}$, $t = 0$
repeat
  repeat iterate over $v \in \mathcal{A}$
    Compute $\{r_{uv}^i(z^j)\}_{i,j\in[n]}$ using (8.4)
    if *condition* (8.5) *is satisfied* **then**
      $\widetilde{b}_{uv}(\cdot) \leftarrow 0$
    else
      $\widetilde{b}_{uv}(\cdot) \leftarrow \operatorname{argmin} \mathcal{L}_u^v\big(b_{uv}(\cdot); \mathcal{D}^n\big)$
    end
  until *convergence on* $\mathcal{A}$
  forall the $v \in [p] \setminus u$ **do**
    if *condition* (8.5) *is satisfied* **then**
      $\widetilde{b}_{uv}(\cdot) \leftarrow 0$
    else
      $\widetilde{b}_{uv}(\cdot) \leftarrow \operatorname{argmin} \mathcal{L}_u^v\big(b_{uv}(\cdot); \mathcal{D}^n\big)$
    end
  end
  $\mathcal{A} := \{v \in [p] \setminus u : ||\widetilde{b}_{uv}(\cdot)||_2 > 0\}$
until $\mathcal{A}$ *did not change*
$\widehat{\mathbf{B}}_u \leftarrow \{\widetilde{b}_{uv}(\cdot)\}_{v\in[p]\setminus u}$

Algorithm 3: Procedure for solving Eq. (8.2)

where

$$r_{uv}^i(z) = x_u^i - \sum_{v'\neq u,v} x_{v'}^i \widetilde{b}_{uv'}(z) \tag{8.4}$$

and $\{\widetilde{b}_{uv'}(z)\}$ denotes the current solution for all the other variables. Solving (8.3) iteratively, by cycling through rows $v \in [p] \setminus u$, will lead to an optimal solution $\widehat{\mathbf{B}}_u$ of (8.2). By analyzing Karush-Kuhn-Tucker conditions of the optimization problem in Eq. (8.3), we can conclude that the necessary and sufficient condition for $\{\widetilde{b}_{uv}(z^j)\}_{j\in[n]} \equiv 0$ is

$$\frac{1}{\lambda^2} \sum_{z\in\{z^j\}_{j\in[n]}} \left(\sum_{i\in[n]} x_v^i r_{uv}^i(z) K_h(z - z^i) \right)^2 \leq 1. \tag{8.5}$$

Eq. (8.5) gives a fast way to explicitly check if the row v of a solution is identical to zero or not. If the condition in (8.5) is not satisfied, only then we need to find a minimizer of (8.3), which can be done by the gradient descent, since the objective is differentiable when $\{b_{uv}(z^j)\}_{j\in[n]} \not\equiv 0$.

In practice, one needs to find a solution to (8.2) for a large number of penalty parameters λ. Computing solutions across a large set of possible λ values can effectively be implemented using

the warm start technique [70]. In this technique, Eq. (8.2) is solved for a decreasing sequence of penalty parameters $\lambda_1 > \ldots > \lambda_N$ and the initial value $\widetilde{\mathbf{B}}_u^{(0)}$ provided to Algorithm 3 for λ_i is the final solution $\widehat{\mathbf{B}}_u$ for λ_{i-1}. This experimentally results in faster convergence and a more stable algorithm.

8.4 Theoretical properties

In this section, we give some theoretical properties of the estimation procedure given in §8.2. These results are given for completeness and are presented without proofs, which will be reported elsewhere. In particular, we provide conditions under which there exists a set $\widehat{S} = \widehat{S}(\lambda)$ of selected non-zero partial correlations, which consistently estimates S, the true set of non-zero partial correlations. Observe that $\widehat{S}$ depends on the penalty parameter λ, so it is of practical importance to correctly select the parameter λ for which $\widehat{S}$ consistently recovers S. We give conditions under which the modified BIC criterion is able to identify the correct penalty parameter λ. We start by giving general regularity conditions.

The following regularity conditions are standard in the literature [61, 183]: **(A1)** There is an $s > 2$ such that $\mathbb{E}[||\mathbf{X}||_2^{2s}] \leq \infty$; **(A2)** The density function $f(z)$ of the random variable Z is bounded away from 0 on $[0, 1]$ and has bounded second order derivative; **(A3)** The matrix $\mathbf{\Omega}(z)$ is positive definite for all $z \in [0, 1]$ and its elements $(\omega_{uv}(z))$ are functions that have bounded second derivatives; **(A4)** The function $\mathbb{E}[||X||_2^4 \mid Z = z]$ is bounded; **(A5)** The kernel $K(\cdot)$ is a symmetric density with compact support. In addition the standard regularity conditions, we need the following identifiability condition, which allows us to correctly identify the true model **(A6)** $\sup_{z\in[0,1]} \max_{u\neq v} |\omega_{uv}(z^i)| \leq \mathcal{O}(\frac{1}{d})$, where $d := \max_{u\in[p]} |\{v : (u, v) \in S\}|$

Theorem 8.1. *Assume that the regularity conditions **(A1)-(A6)** are satisfied. Furthermore, assume that* $\mathbb{E}[\exp(tX)|Z = z] \leq \exp(\sigma^2 t^2/2)$ *for all* $z \in [0, 1]$, $t \in \mathbb{R}$ *and some* $\sigma \in (0, \infty)$. *Let* $h = \mathcal{O}(n^{-1/5})$, $\lambda = \mathcal{O}(n^{7/10}\sqrt{\log p})$ *and* $n^{-9/5}\lambda \to 0$. *If* $\frac{n^{11/10}}{\sqrt{\log p}} \min_{u,v\in S} ||b_{uv}(\cdot)||_2 \to \infty$, *then* $\mathbb{P}[\widehat{S} = S] \to 1$.

Assuming that $\mathbf{X}$ is a subgaussian random variable in Theorem 8.1 is due to technical reasons. The assumption is needed to establish exponential inequalities for the probability that each solution $\widehat{\mathbf{B}}_u$ of Eq. (8.2) correctly identifies the set of non-zero rows of $\mathbf{B}_u$. Then consistency of $\widehat{S}$ can be established by applying the union bound over the events that estimators $\{\widehat{\mathbf{B}}_u\}_{u\in[p]}$ consistently estimate non-zero rows of $\{\mathbf{B}_u\}_{u\in[p]}$. For the last claim to be true when the dimension p is large, e.g., $p = \mathcal{O}(\exp(n^\alpha)), \alpha > 0$, we need a good tail behavior of the distribution of $\mathbf{X}$. The statement of the theorem still holds true, even if we do not establish exponential inequalities, but only for smaller dimensions. Another commonly used regularity condition on $\mathbf{X}$ is to assume that it is bounded with probability 1, which would again allow us to establish exponential inequalities needed in the proof. Finally, we need to assume that for $(u, v) \in S$, $||b_{uv}(\cdot)||_2$ does not decay to zero too quickly. Otherwise, the element of the precision matrix would be to hard to distinguish from 0.

Next, we show that the correct penalty parameter λ can be chosen using the modified BIC criterion of [31]. Denote $\widehat{\mathbf{B}}_{u,\lambda}$ as the solution of Eq. (8.2) obtained for the penalty parameter λ.

We define the residual sum of squares as

$$\mathrm{RSS}_u(\lambda) := n^{-2} \sum_z \sum_{i \in [n]} \big(x_u^i - \sum_{v \neq u} x_v^i \widehat{b}_{uv,\lambda}(z)\big)^2 K_h(z - z^i)$$

and the BIC-type criterion

$$\mathrm{BIC}_u(\lambda) = \log(\mathrm{RSS}_u(\lambda)) + \frac{\widehat{df}_{u,\lambda}(\log(nh) + 2\log p)}{nh},$$

where $\widehat{df}_{u,\lambda}$ denotes the number of non-zero rows of $\widehat{\mathbf{B}}_{u,\lambda}$. We used the modified version of the BIC criterion, since the ordinary BIC criterion tends to include many spurious variables when the complexity of the model space is large [31]. Now, λ is chosen by a minimization:

$$\widehat{\lambda} = \operatorname*{argmin}_{\lambda} \sum_{u \in [p]} \mathrm{BIC}_u(\lambda), \tag{8.6}$$

and the final estimator of the non-zero components of the precision matrix $\widehat{S} = \widehat{S}(\widehat{\lambda})$ is obtained by combining $\{\widehat{\mathbf{B}}_{u,\widehat{\lambda}}\}_{u \in [p]}$. We have the following theorem.

Theorem 8.2. *Assume that the conditions of Theorem 8.1 are satisfied. Then the tuning parameter $\widehat{\lambda}$ obtained by minimizing criterion* (8.6) *asymptotically identifies the correct model, i.e.,* $\mathbb{P}[\widehat{S}(\widehat{\lambda}) = S] \to 1$.

8.5 Simulation results

8.5.1 Toy example

We first consider a small toy example in order to demonstrate our algorithm's performance. We draw n samples, from the joint distribution of $(\mathbf{X}, Z)$ where the conditional distribution of $\mathbf{X}$ given $Z = z$ is a 5-dimensional multivariate Gaussian with mean 0 and precision matrix $\mathbf{\Omega}(z)$, and Z is uniformly distributed on $[0, 1]$. The set $S = \{(1,2), (3,4), (2,4), (1,5), (3,5)\}$ denotes the non-zero elements of $\mathbf{\Omega}(z)$. We set elements $\omega_{uv}(z) = \omega_{uv}(z) = f_{uv}(z)$ for all $(u, v) \in S$, where the functions $\{f_{uv}(z)\}$ are defined as follows: **(1)** $f_{1,2} \equiv 1$ (constant), **(2)** $f_{3,4} \equiv 1$ (constant), **(3)** $f_{2,4}(z) = 1$ if $z \leq .5$ and -1 for $z > .5$ (piecewise constant), **(4)** $f_{1,5}(z) = 2z - 1$ (linear), **(5)** $f_{3,5}(z) = sin(2\pi z)$ (sinusoid). The diagonal elements $\omega_{uu}(z)$ $(z \in [0, 1])$ are set to a constant number such that $\mathbf{\Omega}(z)$ is diagonally dominant, and hence positive definite.

We compared our method against the approach of [135] (referred to as `MB`), which assumes an invariant covariance matrix and ignores z, and against a simpler variant of our algorithm (called "kernel, ℓ_1 penalty"), which replaces the group ℓ_1/ℓ_2 penalty in (8.1) with the ℓ_1 penalty. Recall that the ℓ_1 penalty does not encourage the set of non-zero elements in the precision matrix to remain fixed for all $z \in [0, 1]$. Our algorithm, developed in §8.2 is referred to as "kernel, group penalty".

We average our results over 100 random trials. For each trial, $n = 300$ samples are randomly generated using the procedure described above. We counted the number of times each of the

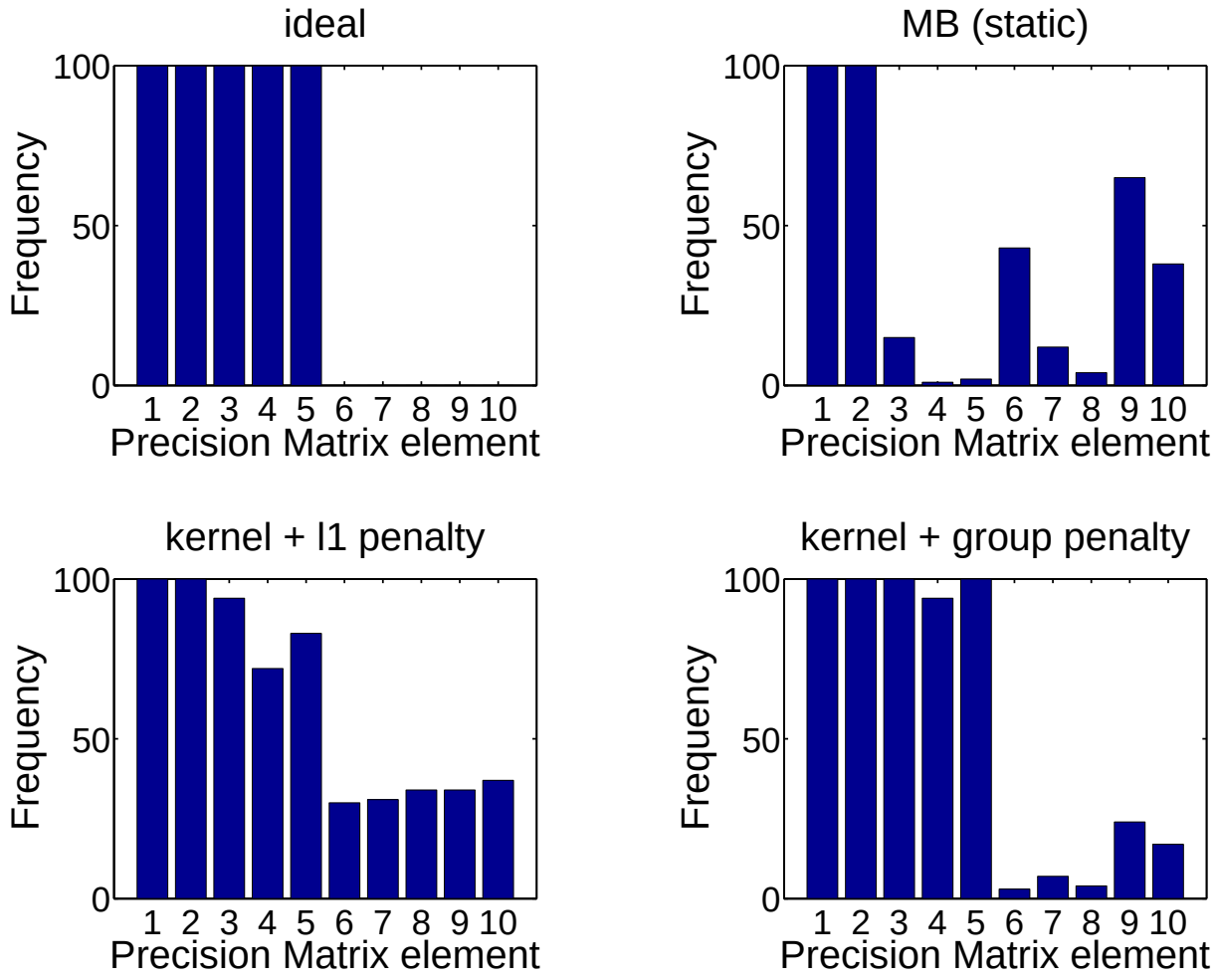


Figure 8.1: Toy example results. Each bar represents the number of times the corresponding precision matrix element was included in $\widehat{S}$. Performance of the ideal algorithm is shown in the top left part. Our algorithm gets close to this, and far outperforms both the other methods.

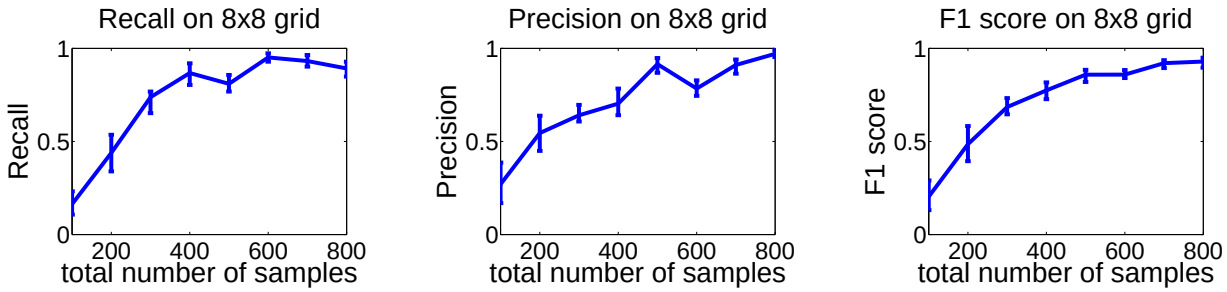


Figure 8.2: Simulation results for 8x8 grid. See §8.5.2 for details.

$\binom{5}{2} = 10$ possible off-diagonal elements of the precision matrix were selected as non-zeros. Figure 8.1 displays results as histograms. Bars 1-5 correspond to the true non-zero elements in S, as enumerated above, while bars 6-10 correspond to the elements that should be set to zero. Thus, in the ideal case, bars 1-5 should be estimated as non-zero for all 100 trials, while bars 6-10 should never be selected. As we can see, all algorithms select the constant elements $\omega_{12}(\cdot)$ (bar 1) and $\omega_{34}(\cdot)$ (bar 2). However, the `MB` approach fails to recover the three varying precision matrix elements and also recovers many false elements. Just using the kernel + ℓ_1 penalty, described above, performs better, but still selects many elements not in S. Our algorithm, on the other hand, selects all the elements in S almost all of the time, and also excludes the elements not in S the vast majority of the time. This higher precision is the result of our group penalty, and gives superior performance to just using an ℓ_1 penalty (assuming that the set of non-zero partial correlation coefficients is fixed with respect to z).

8.5.2 Large simulations

We next tested our algorithm on a larger problem where $\mathbf{X} \in \mathbb{R}^{64}$. The components of $\mathbf{X}$ were arranged into an 8x8 grid, so that only adjacent components in the grid have has non-zero partial correlation. For all adjacent (u, v), $\omega_{uv}(z) = sin(2\pi z + c_{uv})$, where $c_{uv} \sim \text{Unif}([0, 1])$ is a random offset. We measure how well the algorithm recovers the true set of non-zero precision matrix elements. Both `MB` and "kernel + ℓ_1" perform much worse than our estimator, so we do not display their performance. Performance of the "kernel + group penalty" estimator is shown in Figure 8.2. Even though the problem is significantly harder, after 800 samples our algorithm achieves an F1 score above 0.9.

8.6 Analyzing the stock market

We next apply our method to analyzing relationships among stocks in the S&P 500. Such an analysis would be useful to an economist studying the effect of various indicators on the market, or an investor who is seeking to minimize his risk by constructing a diverse portfolio according to Modern Portfolio Theory [132]. Rather than assume static associations among stocks we believe it is more realistic to model them as a function of an economic indicator, such as oil price. We acquired closing stock prices from all stocks in the S&P 500 [1] and oil prices [2] for all the days that the market was open from Jan 1, 2003 through Dec 31, 2005. This gave us 750 samples of 469 stocks (we only considered stocks that remained in the S&P 500 during the entire time period). Instead of considering the raw prices, which often are a reflection of other factors, such as number of shares, we used the logarithm of the ratio of the price at time t to the price at time $t - 1$ and subtracted the mean value and divided by the standard deviation for each stock.

Our data consists of pairs $\{\mathbf{x}^i, z^i\}$, the vector of standardized stock prices and the oil price, respectively, obtained over a period of time. We analyze the data to recover the strength of associations between different stocks as a function of the oil price. Our belief is that each stock is associated with a small number of other stocks and that the set of associations is fixed over a

[1] Can be obtained at `http://www.finance.yahoo.com`.
[2] Can be obtained at `http://tonto.eia.doe.gov/`.

time-period of interest, although the strengths may change. We believe this is justified since we are looking for long-term trends among stocks and want to ignore transient effects. Figure 8.3 illustrates the estimated network, where an edge between two nodes correspond to a non-zero element in the precision matrix. Note that the presented network is not a representation of an undirected probabilistic graphical model.

Clusters of related stocks are circled in Figure 8.3, and these largely confirm our intuition. Here are some of the stocks in a few of the clusters: **(1)** *Technology/semiconductors* - Hewlett Packard, Intel, Teradyne, Analog Devices etc.; **(2)** *Oil/drilling/energy* - Diamond Offshore Drilling, Baker Hughes, Halliburton, etc.; **(3)** *Manufacturing* - Alcoa, PPG Industries (coating products), International Paper Co. etc.; **(4)** *Financial* - American Express, Wells Fargo, Franklin Resources etc. It is also interesting that there exist coherent subgroups inside these clusters. For example, the "Retail stores" sector could be further divided into companies that specialize in clothes, like Gap and Limited, and those that are more general purpose department stores, like Wal-Mart and Target.

Another point of interest are two hubs (enlarged and highlighted in green in Figure 8.3), that connect a set of diverse stocks that do not easily categorize into an industrial sector. They correspond to JPMorgan Chase and Citigroup (two prominent financial institutions). It possible that these stocks are good indicators of the status of the market or have certain investment portfolios that contribute to their central positions in the network.

Finally, we explore the evolving nature of our edge weights as a function of oil price to demonstrate the advantages over simply assuming static partial correlations. Recall that the edge weights vary with oil price and are proportional to the estimated partial correlation coefficients. Consider the two stocks Analog Devices (ADI), which makes signal processing solutions, and NVIDIA (NVDA), which makes graphics processing units. Ignoring the effect of the oil price, both of these companies are highly related since they belong to the semiconductor sector. However, if one analyzes the edge weights as a function of oil price, as shown in Figure 8.4 (a) and (b), both behave quite differently. This changing relationship is reflected by the varying strength of the edge weight between NVIDIA and Analog Devices (shown in Figure 8.4 (c)). Note that when oil prices are low, the edge weight is high since Analog Devices and NVIDIA are both rising as a function of oil price. However, as oil prices increase, Analog Devices stabilizes while NVIDIA is more erratic (although it is mostly rising), so the edge weight sharply decreases. Thus, if an investor is aiming for diversification to reduce risk, he/she may be wary of investing in both of these stocks together when oil prices are low since they are highly associated, but might consider it if oil prices are high and the stocks are less associated.

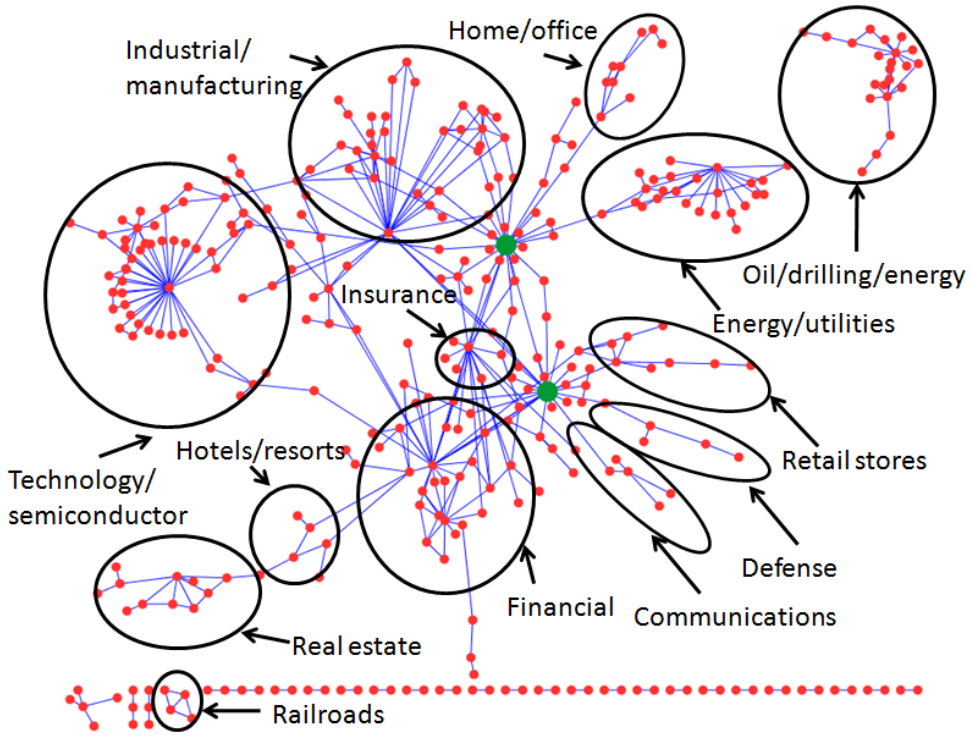


Figure 8.3: Overall stock market network that was recovered by the algorithm. Edges in the graph correspond to non-zero elements in the precision matrix. As one can see, the recovered network contains many clusters of related stocks. The green (and enlarged) hubs are described in the text.

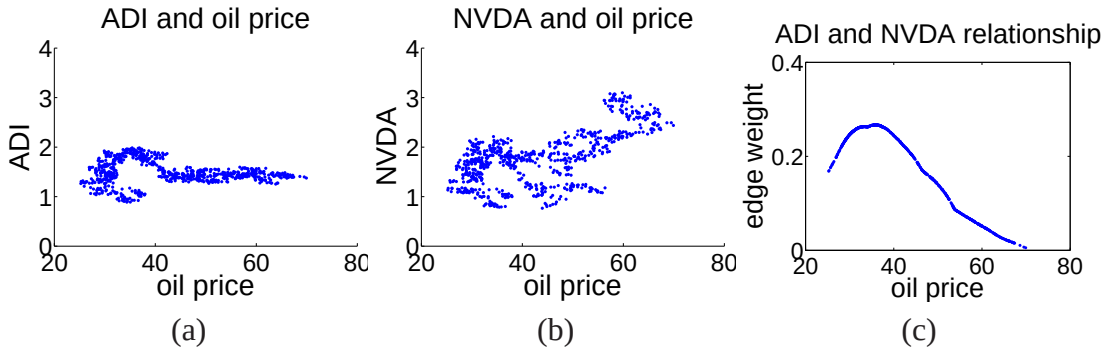


Figure 8.4: This figure demonstrates how the changing edge weight between Analog Devices and NVIDIA ((c)) corroborates with the fact that Analog Devices and NVIDIA behave quite differently as a function of oil price ((a) and (b)). In (a) and (b), the y-axis is the ratio of the stock price to its price on January 1, 2003.

Chapter 9

Estimation From Data with Missing Values

In this chapter, we study a simple two step procedure for estimating sparse precision matrices from data with missing values, which is tractable in high-dimensions and does not require imputation of the missing values. We provide rates of convergence for this estimator in the spectral norm, Frobenius norm and element-wise ℓ_∞ norm. Simulation studies show that this estimator compares favorably with the EM algorithm. Our results have important practical consequences as they show that standard tools for estimating sparse precision matrices can be used when data contains missing values, without resorting to the iterative EM algorithm that can be slow to converge in practice for large problems. Furthermore, the tools developed here could be extended to estimation of time-varying networks in previous chapters.

9.1 Introduction

Covariance matrices and their inverses, precision matrices, arise in a number of applications including principal component analysis, classification by linear and quadratic discriminant analysis, and the identification of conditional independence assumptions in the context of Gaussian graphical models. As a result, obtaining good estimators of covariance and precision matrices under various contexts is of essential importance in statistics and machine learning research. In §2 we provide an overview of methods for learning GGMs from fully observed data.

In practice, we often have to analyze data that contains missing values [129]. Missing values may occur due to a number of reasons, for example, faulty machinery that collects data, subjects not being available in subsequent experiments (longitudinal studies), limits from experimental design, etc. When missing values are present, they are usually imputed to obtain a complete data set on which standard methods can be applied. However, methods that directly perform statistical inference, without imputing missing values, are preferred. A systematic approach to missing values problem is based on likelihoods of observed values. However, with an arbitrary pattern of missing values, no explicit maximization of the likelihood is possible even for the mean values and covariance matrices [129]. Expectation maximization algorithms, which are iterative methods, are commonly used in cases where explicit maximization of the likelihood is not possible; however, providing theoretical guarantees for such procedures is difficult. This approach was employed in [164] to estimate sparse inverse covariance matrices, which we will

review in the following section. In recent work, [122] deals with the estimation of covariance matrices from data with missing values under the assumption that the true covariance matrix is approximately low rank. [124] recently studied high-dimensional regression problems when data contains missing values. Casting the estimation of a precision matrix as a sequence of regression problems, they obtain an estimator of the precision matrix without maximizing partially observed likelihood function using an EM algorithm.

In this chapter, we present a simple, principled method that directly estimates a large dimensional precision matrix from data with missing values. We form an unbiased estimator of the covariance matrix from available data, which is then plugged into the penalized maximum likelihood objective for a multivariate Normal distribution to obtain a sparse estimator of the precision matrix. Even though the initial estimator of the covariance matrix is not necessarily positive-definite, we can show that the final estimator of the precision matrix is positive definite. Furthermore, unlike the EM algorithm, which is only guaranteed to converge to a local maximum, we prove consistency and convergence rate for our estimator in the Frobenius norm, spectral norm and ℓ_∞ norm. Our results have important practical consequences as they allow practitioners to use existing tools for penalized covariance selection (see, for example, [71]), which are very efficient in high-dimensions for data sets with missing values without changing the algorithm or resorting to the iterative EM algorithm.

Throughout the chapter we assume that the missing values are missing at random in the sense of [155]. Let $\mathbf{X} = (x_{ij}) \in \mathbb{R}^{n \times p}$ be a matrix of observations with samples organized into rows, and let $\mathbf{R} = (r_{ij}) \in \mathbb{R}^{n \times p}$ be a matrix of indicators of observed values, that is, $r_{ij} = 1$ if the value x_{ij} was observed and $r_{ij} = 0$ otherwise. We assume that the data is missing completely at random (MCAR), which means that $\mathbb{P}[\mathbf{R}|\mathbf{X}, \varphi] = \mathbb{P}[\mathbf{R}|\varphi]$ for all $\mathbf{X}$ and φ, where φ denotes unknown parameters. The MCAR assumption implies that the missingness does not depend on the observed values, e.g., in a distributed environment, each sensor may fail independently from other sensors. This assumption is relaxed in the experimental section where we test the robustness of our procedure when the missing data mechanism departs from the MCAR assumption. Another more realistic assumption is called missing at random (MAR), which assumes $\mathbb{P}[\mathbf{R}|\mathbf{X}, \varphi] = P[\mathbf{R}|\mathbf{X}_{\text{obs}}, \varphi]$ for all $\mathbf{X}_{\text{mis}}$ and φ, where $\mathbf{X}_{\text{obs}}$ denotes the observed components of $\mathbf{X}$ and $\mathbf{X}_{\text{mis}}$ denotes the missing components. The MAR assumes that the distribution of $\mathbf{R}$ depends on the observed values of $\mathbf{X}$, but not on the missing values, e.g., cholesterol level may be measured only if patient has high blood pressure. Finally, the missing data mechanism is called not-missing at random (NMAR) if the distribution of $\mathbf{R}$ depends on the non-observed values of $\mathbf{X}$. Estimation under NMAR is a hard problem, as one needs to make assumptions on the model for missing values. The method presented in this chapter can, in theory, be extended to handle the MAR case.

9.2 Problem setup and the EM algorithm

Let $\{\mathbf{x}_i\}_{i=1}^n$ be an *i.i.d.* sample from the multivariate Normal distribution with the mean $\boldsymbol{\mu} \in \mathbb{R}^p$ and the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$. Let $\mathbf{R} \in \mathbb{R}^{n \times p}$ be a matrix of missing values indicators with $r_{ij} = 1$ if x_{ij} is observed and 0 otherwise. The goal is to estimate the sparse precision matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ from the data with missing values.

Estimating covariance matrices from data with missing values is quite an old problem. See, for example, [1, 7, 87, 90, 167]. However, literature on high-dimensional estimation of covariance matrices from incomplete data is missing. Recently [164] proposed to use an EM algorithm, called MissGlasso, to estimate sparse precision matrices, which we review below.

As discussed in §2 the sparse precision matrix can be estimated by solving the following ℓ_1-norm penalized maximization problem

$$\widehat{\mathbf{\Omega}} = \arg\max_{\mathbf{\Omega} \succeq 0} \{\log |\mathbf{\Omega}| - \operatorname{tr} \mathbf{\Omega}\widehat{\mathbf{S}} - \lambda||\mathbf{\Omega}^-||_1\}, \tag{9.1}$$

where $\widehat{\mathbf{S}}$ is the empirical covariance matrix, $\mathbf{\Omega}^- := \mathbf{\Omega} - \operatorname{diag}(\mathbf{\Omega})$ and $||\mathbf{A}||_1 = \sum_{ij} |A_{ij}|$.

When the data are fully observed, [195] arrived at the optimization procedure in (9.1) from the penalized maximum likelihood approach, with $\widehat{\mathbf{S}} = n^{-1} \sum_{i=1}^n (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})'$. In the case when data contains missing values, the log-likelihood of observed data takes the following form

$$\ell(\boldsymbol{\mu}, \mathbf{\Omega}; \{\mathbf{x}_{i,\text{obs}}\}_i) = \\ -\frac{1}{2} \sum_{i=1}^n \Big(\log |(\mathbf{\Omega}^{-1})_{i,\text{obs}}| + (\mathbf{x}_{i,\text{obs}} - \boldsymbol{\mu}_{i,\text{obs}})'((\mathbf{\Omega}^{-1})_{i,\text{obs}})^{-1}(\mathbf{x}_{i,\text{obs}} - \boldsymbol{\mu}_{i,\text{obs}}) \Big),$$

where for a sample point $\mathbf{x}_i$ we write $\mathbf{x}_i = (\mathbf{x}_{i,\text{obs}}, \mathbf{x}_{i,\text{mis}})$ to denote observed and missing components, and $\boldsymbol{\mu}_{i,\text{obs}}$ and $\mathbf{\Omega}_{i,\text{obs}}$ are the mean and precision matrix of the observed components of $\mathbf{x}_i$. MissGlasso is an EM algorithm that finds a local maximum $(\widehat{\boldsymbol{\mu}}, \widehat{\mathbf{\Omega}})$ of the ℓ_1 penalized observed log-likelihood. In the E-step, MissGLasso imputes the missing values by conditional means of the distribution. That is, imputation is done by $\widehat{\mathbf{x}}_{i,\text{mis}} = \widehat{\boldsymbol{\mu}}_{\text{mis}} - (\widehat{\mathbf{\Omega}}_{\text{mis,mis}})^{-1}\widehat{\mathbf{\Omega}}_{\text{mis,obs}}(\mathbf{x}_{i,\text{obs}} - \mu_{\text{obs}})$, where $\widehat{\boldsymbol{\mu}}$ and $\widehat{\mathbf{\Omega}}$ are the current estimates of the parameters. In the M-step, the optimization problem (9.1) is solved using the GLasso on data with imputed missing values. The procedure iterates between the E-step and the M-step until convergence to a local optimum of the penalized observed log-likelihood. We will denote $\widehat{\mathbf{\Omega}}^{\text{EM}}$, the final estimator of the precision matrix obtained by the EM algorithm. As the objective is non-convex, it is difficult to establish theoretical guarantees on the solution produced by the EM. Next, we present our estimator.

9.3 Plug-in estimator and related procedures

In this section, we propose a simple procedure based on the plug-in estimator of the covariance matrix from available data that can be used together with existing procedures for estimating precision matrices from fully observed data. Specifically, we will use the penalized likelihood approach, which was introduced in the previous section in (9.1). From (9.1) it is obvious that we only need a sample estimate of the covariance matrix, which is plugged into a convex program that produces an estimate of the precision matrix.

We form a sample covariance matrix from the available samples containing missing values as follows. Let $\widehat{\mathbf{S}} = [\widehat{\sigma}_{ab}]_{ab}$ be the sample covariance matrix with elements

$$\widehat{\sigma}_{ab} = \frac{\sum_{i=1}^n r_{ia} r_{ib} (x_{ia} - \widehat{\mu}_a)(x_{ib} - \widehat{\mu}_b)}{\sum_{i=1}^n r_{ia} r_{ib}} \tag{9.2}$$

where $\widehat{\mu} = (\widehat{\mu}_a)$ is the sample mean defined as $\widehat{\mu}_a = (\sum_{i=1}^n r_{ia})^{-1} \sum_{i=1}^n r_{ia} x_{ia}$. Observe that the missing values in $\mathbf{X}$ are taken into account naturally and that the mean and covariance elements are estimated only from the observed sample. Under the MCAR assumption, it is simple to show that $\widehat{\mathbf{S}}$ is an unbiased estimator of $\mathbf{\Sigma}$, that is, $\mathbb{E}[\widehat{\mathbf{S}}] = \mathbf{\Sigma}$.

Our estimator is formed by plugging $\widehat{\mathbf{S}}$ into the objective in (9.1), which we will denote as $\widehat{\mathbf{\Omega}}^{\mathrm{mGLasso}}$. Note that $\widehat{\mathbf{S}}$ is not necessarily a positive definite matrix, however, the minimization problem in (9.1) is still convex and the resulting estimator $\widehat{\mathbf{\Omega}}^{\mathrm{mGLasso}}$ will be positive definite and unique. In the next section, we leverage the analysis of [152] to establish a number of good statistical properties of the estimator $\widehat{\mathbf{\Omega}}^{\mathrm{mGLasso}}$.

9.3.1 Selecting tuning parameters

The procedure described in the previous section requires selection of the tuning parameters λ, which controls the sparsity of the solution and balances it to the fit to data. A common approach is to form a grid of candidate values for the tuning parameter λ and choose one that minimizes a modified BIC criterion

$$\mathrm{BIC}(\lambda) = -2\ell(\widehat{\boldsymbol{\mu}}, \widehat{\mathbf{\Omega}}; \{\mathbf{x}_{i,\mathrm{obs}}\}_i) + \log(n) \sum_{a \leq b} \mathbb{1}\{\widehat{\omega}_{ab} \neq 0\}.$$

Here $(\widehat{\boldsymbol{\mu}}, \widehat{\mathbf{\Omega}})$ are estimates obtained using the tuning parameter λ and $\ell(\widehat{\boldsymbol{\mu}}, \widehat{\mathbf{\Omega}}; \{\mathbf{x}_{i,\mathrm{obs}}\}_i)$ is the observed log-likelihood. [195] proposed to use $\sum_{a \leq b} \mathbb{1}\{\widehat{\omega}_{ab} \neq 0\}$ to measure the degrees of freedom.

Performing cross-validation is another possibility for finding the optimal parameter λ. In the V-fold cross-validation, samples are divided into V disjoint folds, say $\mathcal{D}_v$ for $v = 1, \ldots, V$, and the score is computed as

$$\mathrm{CV}(\lambda) = \sum_{v=1}^{V} \sum_{i \in \mathcal{D}_v} \log |(\widehat{\mathbf{\Omega}}_v^{-1})_{i,\mathrm{obs}}| + (\mathbf{x}_{i,\mathrm{obs}} - (\widehat{\boldsymbol{\mu}}_v)_{i,\mathrm{obs}})'((\widehat{\mathbf{\Omega}}_v^{-1})_{i,\mathrm{obs}})^{-1}(\mathbf{x}_{i,\mathrm{obs}} - (\widehat{\boldsymbol{\mu}}_v)_{i,\mathrm{obs}}),$$

where $(\widehat{\boldsymbol{\mu}}_v, \widehat{\mathbf{\Omega}}_v)$ denote estimates obtained from the sample $\{\mathbf{x}_i\}_{i=1}^n \backslash \mathcal{D}_v$. The optimal tuning parameter $\widehat{\lambda}$ is the one that minimizes $\mathrm{CV}(\lambda)$. The final estimates $(\widehat{\boldsymbol{\mu}}, \widehat{\mathbf{\Omega}})$ are constructed using the optimization procedure with the tuning parameter $\widehat{\lambda}$ on all the data.

9.3.2 Related procedures

[122] and [124] have recently proposed procedures for estimating approximately low-rank covariance matrices and sparse precision matrices, respectively, from high-dimensional data with missing values. In both works, a sample covariance estimator is formed, which is then plugged into an optimization procedure. The sample covariance estimator they consider, assuming $(r_{ia})_{ia} \overset{iid}{\sim} \mathrm{Bern}(\gamma)$ with $\gamma \in (0, 1]$ known, is defined as

$$\widetilde{\mathbf{\Sigma}} = (\gamma^{-1} - \gamma^{-2})\mathrm{diag}(\check{\mathbf{\Sigma}}) + \gamma^{-2}\check{\mathbf{\Sigma}}$$

where $\check{\mathbf{\Sigma}} = [\check{\sigma}_{ab}]_{ab}$ and $\check{\sigma}_{ab} = n^{-1}\sum_{i=1}^{n} r_{ia}r_{ib}x_{ia}x_{ib}$. The estimator $\widetilde{\mathbf{\Sigma}}$ is an unbiased estimator of the covariance matrix, however, it requires knowledge of the parameter γ.

Procedure of [122] is focused on estimating a covariance matrix under the assumption that the true covariance matrix is approximately low rank and hence is not comparable to our procedure. [124] used a projected gradient descent method to obtain a solution to a high-dimensional regression problem when data contains missing values. A sparse precision matrix can be obtained by maximizing an ℓ_1 penalized pseudo-likelihood, which reduces to a sequence of regression problems. We note that the estimator $\widehat{\mathbf{\Omega}}^{\mathrm{mGLasso}}$ can be obtained using any convex program solver that can solve (9.1), while the results of [124] rely on the usage of projected gradient descent.

9.4 Theoretical results

In this section, we provide theoretical analysis of the estimates $\widehat{\mathbf{\Omega}}^{\mathrm{mGLasso}}$, which we denote $\widehat{\mathbf{\Omega}}$ throughout the section for notational simplicity, under the MCAR assumption. We start by analyzing the sample covariance matrix $\widehat{\mathbf{S}}$ in (9.2). We will assume that each element of the missing values indicator matrix $\mathbf{R}$ is independently distributed as $r_{ia} \sim \mathrm{Bern}(\gamma)$, $i = 1, \ldots, n,\ a = 1, \ldots, p$. Furthermore, we assume that a distribution of $\mathbf{X}$ has sub-Gaussian tails, that is, there exists a constant $\sigma \in (0, \infty)$ such that

$$\mathbb{E}[\exp(t(X_{ia} - \mu_a))] \leq \exp(\sigma^2 t^2), \text{ for all } t \in \mathbb{R}.$$

A multivariate Gaussian distribution satisfies this condition. We define the function $f(n, \gamma, \delta)$, which will be useful for characterizing probabilistic deviation of different quantities, as

$$f(n, \gamma, \delta) = (n\gamma^2 - \sqrt{2n\gamma^2 \log(2/\delta)})^{-1} \log(8/\delta).$$

Our first result characterizes the deviation of the sample covariance matrix from the true covariance matrix.

Lemma 9.1. *Assume that $X_a/\sqrt{\Sigma_{aa}}$ is sub-Gaussian with parameter σ^2. Fix $\delta > 0$ and assume that n is big enough so that $f(n, \gamma, \delta) \leq 1/2$. Then for any fixed $(a, b) \in \{1, \ldots, p\}^2$, $a \neq b$, with probability at least $1 - \delta$, we have that $|\widehat{\sigma}_{ab} - \sigma_{ab}| \leq C_\sigma \sqrt{f(n, \gamma, \delta)}$ where $C_\sigma = 16\sqrt{2}(1 + 4\sigma^2) \max_a \sigma_{aa}$.*

Similarly, we can obtain that for any diagonal elements of $\widehat{\mathbf{S}}$ the statement $|\widehat{\sigma}_{aa} - \sigma_{aa}| \leq C_\sigma \sqrt{f(n, \sqrt{\gamma}, \delta)}$ holds with probability $1 - \delta$.

We use Lemma 9.1 to prove properties of the estimate $\widehat{\mathbf{\Omega}}^{\mathrm{mGLasso}}$. We start by introducing some additional notation and assumptions. Following [152], we introduce the *irrepresentable condition*:

$$\|\mathbf{\Gamma}_{S^C S}(\mathbf{\Gamma}_{SS})^{-1}\|_\infty \leq 1 - \alpha, \quad \alpha \in (0, 1] \tag{9.3}$$

where $\mathbf{\Gamma} = \mathbf{\Omega} \otimes \mathbf{\Omega}$, $S := \{(a, b) : \omega_{ab} \neq 0\}$ is support of $\mathbf{\Omega}$ and $S^C := \{(a, b) : \omega_{ab} = 0\}$, and $\|\cdot\|_\infty$ is the ℓ_∞/ℓ_∞-operator norm. Furthermore, we define $K_{\mathbf{\Sigma}} := \|\Sigma\|_\infty$ and $K_{\mathbf{\Gamma}} := \|(\mathbf{\Gamma}_{SS})^{-1}\|_\infty$. The maximum number of non-zero elements in a row of $\mathbf{\Omega}$ is denoted $d := \max_{a=1,\ldots,p} |\{b : \omega_{ab} \neq 0\}|$. The rate of convergence will depend on these quantities.

Theorem 9.1. *Suppose that the distribution of* $\mathbf{X}$ *satisfies the irrepresentable condition in* (9.3) *with parameter* $\alpha \in (0, 1]$ *and that the missing values indicator matrix* $\mathbf{R}$ *has i.i.d.* $\mathrm{Bern}(\gamma)$ *elements, that is, the data is missing completely at random with probability* $1 - \gamma$*. Furthermore, assume that the conditions of Lemma 9.1 hold. Let* $\widehat{\mathbf{\Omega}}$ *be the unique solution for the regularization parameter* $\lambda = \frac{8}{\alpha} C_\sigma \sqrt{f(n, \gamma, p^{-\tau})}$ *with some* $\tau > 2$ *and* $C_\sigma = 16\sqrt{2}(1 + 4\sigma^2) \max_a \sigma_{aa}$*. If the sample size satisfies*

$$n > \frac{2(C_1^2(1 + 8\alpha^{-1})^2 d^2 + C_1(1 + 8\alpha^{-1})d) \log 8p^\tau}{\gamma^2}$$

where $C_1 = 6C_\sigma \max\{K_{\mathbf{\Sigma}} K_{\mathbf{\Gamma}}, K_{\mathbf{\Sigma}}^3 K_{\mathbf{\Gamma}}^2\}$ *then with probability at least* $1 - p^{2-\tau}$

$$\max_{a,b} |\widehat{\omega}_{ab} - \omega_{ab}| \leq 2(1 + 8\alpha^{-1}) K_{\mathbf{\Gamma}} C_\sigma \sqrt{f(n, \gamma, p^{-\tau})},$$

where $\widehat{\mathbf{\Omega}} = [\widehat{\omega}_{ab}]_{ab}$ *and* $\mathbf{\Omega} = [\omega_{ab}]_{ab}$*.*

The result follows from application of Theorem 1 in [152] to the tail bound in Lemma 9.1 and some algebra. A simple consequence of Theorem 9.1 is that the support $\widehat{S}$ of $\widehat{\mathbf{\Omega}}$ consistently estimates the support S of $\mathbf{\Omega}$ if all the elements of $\mathbf{\Omega}$ are large enough in absolute values.

Corollary 9.1. *Under the same assumptions as in Theorem 9.1, we have that* $\mathbb{P}[\widehat{S} = S] \geq 1 - p^{2-\tau}$ *if* $\min_{ab} |\omega_{ab}| \geq 2(1 + 8\alpha^{-1}) K_{\mathbf{\Gamma}} C_\sigma \sqrt{f(n, \gamma, p^{-\tau})}$*.*

Proof follows by straightforward algebra from Theorem 9.1. Using the element-wise ℓ_∞ bound on deviation of $\widehat{\mathbf{\Omega}}$ from $\mathbf{\Omega}$ established in Theorem 9.1, we can simply establish the bounds on the convergence in the Frobenius and spectral norms.

Corollary 9.2. *Under the same assumptions as in Theorem 9.1, we have that with probability at least* $1 - p^{2-\tau}$*,*

$$||\widehat{\mathbf{\Omega}} - \mathbf{\Omega}||_F \leq K \sqrt{|S| f(n, \gamma, p^{-\tau})}, \text{ and}$$
$$|||\widehat{\mathbf{\Omega}} - \mathbf{\Omega}|||_2 \leq K \min\{\sqrt{|S|}, d\} \sqrt{f(n, \gamma, p^{-\tau})}$$

where $K = 2(1 + 8\alpha^{-1}) K_{\mathbf{\Gamma}} C_\sigma$*.*

Proof follows by straightforward algebra from Theorem 9.1. We can compare the established results for $\widehat{\mathbf{\Omega}}$ under the MCAR assumption to results of [152] for the fully observed case. We observe that the sample size increases by a factor of $\mathcal{O}(\gamma^{-2})$, while the rate of convergence in the element-wise ℓ_∞ norm is slower by a factor of $\mathcal{O}(\gamma^{-1})$. The parameter γ which controls the rate of missing data is commonly considered a constant, however, it is clear from Theorem 9.1 that we could let $\gamma \to 0$ slowly as a function of n and p, while maintaining the convergence properties of the procedure.

9.5 Simulation Analysis

In this section, we perform a set of simulation studies to illustrate finite sample performance of our procedure. First, we show that the scalings predicted by the theory are sharp. Next, we compare our procedure to the EM algorithm, MissGLasso [164] and the projected gradient method [124], PGLasso. Furthermore, we can explore robustness of our method when the data generating process departs from the one assumed in Section 9.4.

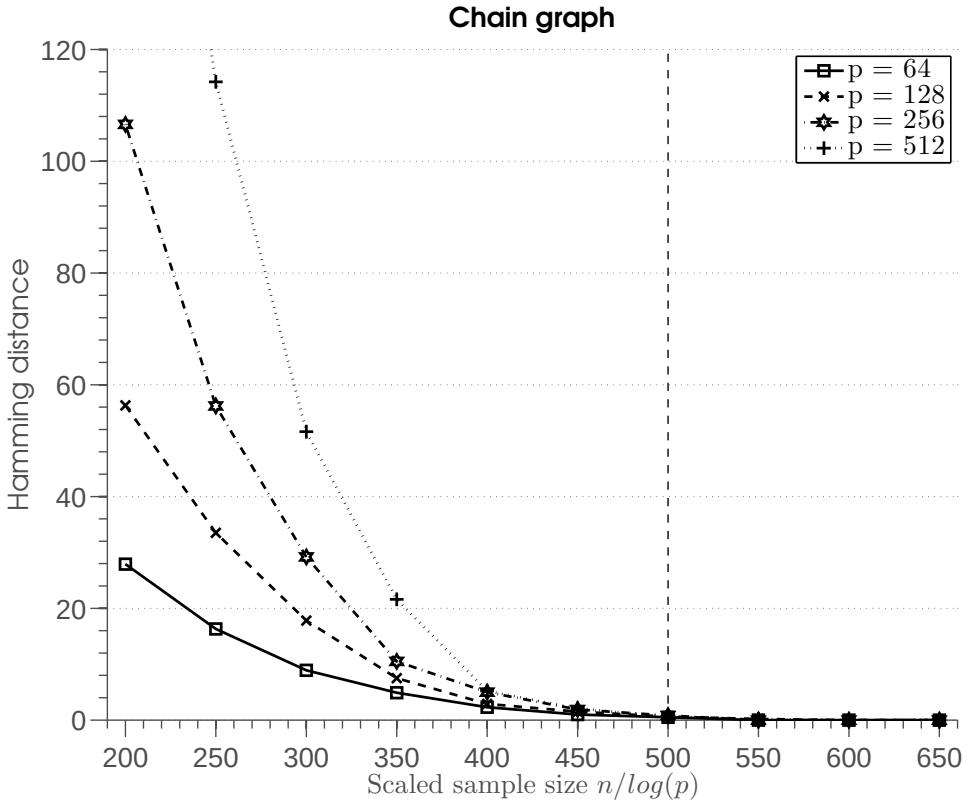


Figure 9.1: Hamming distance between the support of $\widehat{\mathbf{\Omega}}$ and $\mathbf{\Omega}$ averaged over 100 runs. Vertical line marks a threshold at which the graph structure is consistently estimated.

9.5.1 Verifying theoretical scalings

Theoretical results given in Section 9.4 predict behavior of the error when estimating the precision matrix. In particular, Corollary 9.1 suggests that we need $\mathcal{O}(d^2 \log(p))$ samples to estimate the graph structure consistently and Corollary 9.2 states that the error in the operator norm decreases as $\mathcal{O}(d\sqrt{\log(p)/n})$. Therefore, if we plot the error curves against appropriately rescaled sample size, we expect them to align for different problem sizes. To verify this, we create a chain-structured Gaussian graphical model (following [124]), so that $d = 2$ and the precision matrix $\mathbf{\Omega}$ is created as follows. Each diagonal element is set to 1, and all the entries corresponding to the chain are set equal to 0.1. The precision matrix is rescaled so that $\|\mathbf{\Omega}\|_2 = 1$ and $\gamma = 0.8$.

Figure 9.1 shows the hamming distance between the support of $\widehat{\mathbf{\Omega}}$ and $\mathbf{\Omega}$ plotted against the rescaled sample size. Vertical line marks a threshold in scaled sample size after which the pattern of non-zero element of the precision matrix is consistently recovered. Figure 9.2 shows that the error curves align when the sample size is rescaled, as predicted by the theory.

9.5.2 Data missing completely at random

Our first simulation explores the MCAR assumption. We use models from [164]:

Model 1: $\sigma_{ab} = 0.7^{|a-b|}$, so that the elements of the covariance matrix decay exponentially.

			Recall			Precision		
			MissGLasso	mGLasso	PGLasso	MissGLasso	mGLasso	PGLasso
Model 1	p=100	0%	NA	1.000(0.000)	1.000(0.000)	NA	0.973(0.045)	0.991(0.015)
		10%	1.000(0.000)	1.000(0.000)	0.998(0.008)	0.608(0.068)	0.915(0.059)	0.998(0.010)
		20%	0.999(0.004)	1.000(0.003)	0.967(0.006)	0.636(0.081)	0.897(0.073)	0.999(0.003)
		30%	0.977(0.062)	0.989(0.003)	0.759(0.140)	0.642(0.064)	0.836(0.057)	0.998(0.009)
	p=200	0%	NA	1.000(0.000)	0.891(0.005)	NA	0.950(0.046)	0.999(0.004)
		10%	0.860(0.022)	0.950(0.006)	0.782(0.024)	0.858(0.043)	0.803(0.046)	0.984(0.027)
		20%	0.833(0.053)	0.930(0.001)	0.556(0.006)	0.763(0.048)	0.734(0.062)	0.952(0.091)
		30%	0.794(0.138)	0.923(0.003)	0.553(0.009)	0.729(0.059)	0.731(0.060)	0.941(0.052)
	p=500	0%	NA	1.000(0.001)	0.889(0.015)	NA	0.912(0.022)	0.995(0.003)
		10%	0.931(0.011)	0.933(0.031)	0.855(0.023)	0.834(0.029)	0.862(0.044)	0.966(0.010)
		20%	0.852(0.064)	0.920(0.024)	0.767(0.026)	0.811(0.037)	0.841(0.037)	0.965(0.025)
		30%	0.808(0.045)	0.887(0.028)	0.526(0.031)	0.739(0.043)	0.781(0.030)	0.963(0.033)
Model 2	p=100	0%	NA	0.330(0.008)	0.403(0.006)	NA	0.420(0.012)	0.297(0.012)
		10%	0.278(0.019)	0.280(0.011)	0.380(0.007)	0.342(0.012)	0.375(0.010)	0.319(0.008)
		20%	0.240(0.022)	0.253(0.018)	0.259(0.012)	0.339(0.028)	0.372(0.027)	0.320(0.026)
		30%	0.231(0.031)	0.241(0.027)	0.174(0.030)	0.267(0.033)	0.281(0.037)	0.331(0.042)
	p=200	0%	NA	0.281(0.011)	0.410(0.013)	NA	0.570(0.012)	0.270(0.021)
		10%	0.331(0.011)	0.261(0.010)	0.361(0.011)	0.354(0.013)	0.471(0.015)	0.257(0.018)
		20%	0.261(0.012)	0.243(0.015)	0.283(0.013)	0.274(0.018)	0.354(0.021)	0.313(0.021)
		30%	0.218(0.017)	0.232(0.017)	0.208(0.017)	0.281(0.019)	0.267(0.031)	0.453(0.059)
	p=500	0%	NA	0.309(0.006)	0.302(0.012)	NA	0.510(0.007)	0.540(0.018)
		10%	0.305(0.007)	0.307(0.005)	0.357(0.009)	0.461(0.008)	0.462(0.010)	0.224(0.012)
		20%	0.297(0.010)	0.315(0.027)	0.243(0.015)	0.272(0.026)	0.223(0.048)	0.383(0.019)
		30%	0.238(0.025)	0.242(0.023)	0.203(0.028)	0.267(0.031)	0.259(0.033)	0.396(0.021)
Model 3	p=100	0%	NA	0.943(0.002)	0.971(0.015)	NA	0.532(0.017)	0.251(0.051)
		10%	0.857(0.010)	0.857(0.003)	0.994(0.005)	0.857(0.009)	0.882(0.004)	0.200(0.006)
		20%	0.829(0.017)	0.857(0.012)	0.886(0.035)	0.691(0.022)	0.588(0.015)	0.307(0.059)
		30%	0.771(0.053)	0.829(0.033)	0.595(0.096)	0.780(0.050)	0.671(0.050)	0.797(0.053)
	p=200	0%	NA	0.783(0.005)	1.000(0.003)	NA	0.921(0.002)	0.245(0.023)
		10%	0.747(0.005)	0.733(0.006)	0.998(0.007)	0.887(0.009)	0.921(0.004)	0.233(0.030)
		20%	0.667(0.009)	0.747(0.030)	0.931(0.014)	0.909(0.015)	0.737(0.031)	0.311(0.023)
		30%	0.480(0.037)	0.600(0.052)	0.801(0.045)	0.837(0.059)	0.804(0.033)	0.412(0.035)
	p=500	0%	NA	0.744(0.005)	0.998(0.002)	NA	0.844(0.003)	0.191(0.019)
		10%	0.627(0.006)	0.718(0.006)	0.994(0.003)	0.893(0.003)	0.835(0.005)	0.180(0.020)
		20%	0.601(0.010)	0.699(0.031)	0.923(0.029)	0.887(0.034)	0.789(0.037)	0.259(0.054)
		30%	0.511(0.039)	0.614(0.038)	0.851(0.041)	0.800(0.043)	0.755(0.027)	0.355(0.047)

Table 9.1: Average (standard deviation) recall and precision under the MCAR assumption.

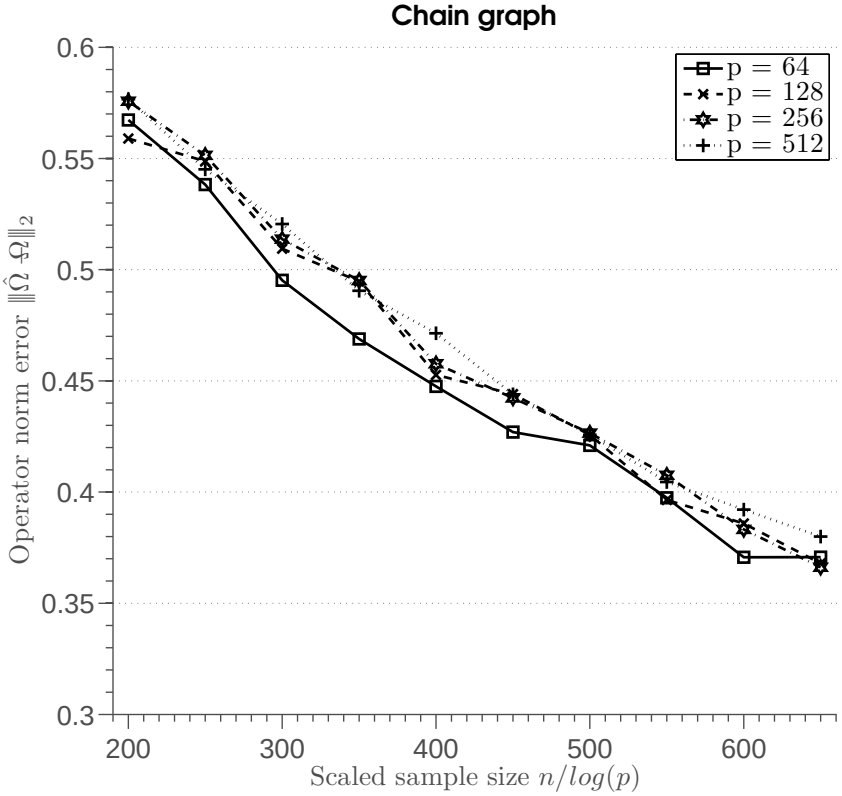


Figure 9.2: Operator norm error averaged over 100 runs. We observe that the error curve align when plotted against the rescaled sample size.

Model 2:

$$\sigma_{ab} = \mathbb{1}_{\{a=b\}} + 0.4\,\mathbb{1}_{\{|a-b|=1\}} + 0.2\,\mathbb{1}_{\{|a-b|=2\}} + 0.2\,\mathbb{1}_{\{|a-b|=3\}} + 0.1\,\mathbb{1}_{\{|a-b|=4\}},$$

where the symbol $\mathbb{1}$ represents the indicator function which is 1 if $a = b$ and 0 otherwise.

Model 3: $\mathbf{\Omega} = \mathbf{B} + \delta\mathbf{I}$, where each off-diagonal entry of $\mathbf{B}$ is generated independently and equals 0.5 with probability $\alpha = 0.1$ or 0 with probability $1 - \alpha$. Diagonal entries of $\mathbf{B}$ are zero, and δ is chosen so that the condition number of $\mathbf{\Omega}$ is p.

We report convergence results in the operator norm. We also report precision and recall for the performance on recovering the sparsity structure of $\mathbf{\Omega}$, where $\text{precision} = \frac{|\widehat{S} \cap S|}{|\widehat{S}|}$ and $\text{recall} = \frac{|\widehat{S} \cap S|}{|S|}$. As described in Section 9.3.1, the tuning parameter λ is selected by minimizing the BIC criterion. We observed that using the tuning parameters that minimize the cross-validation loss result in complex estimates with many falsely selected edges (results not reported).

We set the sample size and number of dimensions $(n, p) = (100, 100), (150, 200), (200, 500)$ for each model and report results averaged over 50 independent runs for each setting. For each generated data set, we remove completely at random 10%, 20% and 30% entries. Results on recall and precision for different degrees of missingness are reported in Table 9.1, while the operator norm convergence results are reported in Table 9.2. From the simulations, we observe that mGLasso performs better than the EM algorithm on the task of recovering the sparsity pattern of the precision matrix. PGLasso does well on Model 1, but does not perform so well

under Model 2 and 3. Model 2 is a difficult one for recovering non-zero patterns, as the true precision matrix contains many small non-zero elements. The EM algorithm performs better than mGLasso and PGLasso measured by $\|\widehat{\mathbf{\Omega}} - \mathbf{\Omega}\|_2$, with mGLasso doing better than PGLasso. However, on average the EM algorithm requires 20 iterations for convergence, which makes mGLasso about 20 times faster on average.

9.5.3 Data missing at random

In the previous section, we have simulated data with missing values completely at random, under which consistency of the estimator $\widehat{\mathbf{\Omega}}^{\text{mGLasso}}$ given in Section 3 can be proven. When the missing values are produced at random (MAR), the EM algorithm described is still valid, however, the estimator $\widehat{\mathbf{\Omega}}^{\text{mGLasso}}$ is not. [128] provided a statistical test for checking whether missing values are missing completely at random, however, no such tests exist for high-dimensional data. In this section, we will observe how robust our estimator is when the data generating mechanism departs from the MCAR assumption. When the missing data mechanism is NMAR, then neither the EM algorithm, nor the procedures described Section 3 are valid.

We will use the model considered in [164] in Section 4.1.2. The model is a Gaussian with $p = 30$, $n = 100$ and the covariance matrix is block-diagonal, $\mathbf{\Sigma} = \text{diag}(\mathbf{B}, \mathbf{B}, \ldots, \mathbf{B})$ with $\mathbf{B} \in \mathbb{R}^{3\times 3}$, $b_{ab} = 0.7^{|a-b|}$. Missing values are created using the following three mechanisms:

1. For all $j = 1, \ldots, \lfloor p/3 \rfloor$ and $i = 1, \ldots, n$: $x_{i,3*j}$ is missing if $r_{i,j} = 0$ where $r_{i,j} \overset{iid}{\sim} \text{Bern}(\pi)$.
2. For all $j = 1, \ldots, \lfloor p/3 \rfloor$ and $i = 1, \ldots, n$: $x_{i,3*j}$ is missing if $x_{i,3*j-2} < T$.
3. For all $j = 1, \ldots, \lfloor p/3 \rfloor$ and $i = 1, \ldots, n$: $x_{i,3*j}$ is missing if $x_{i,3*j} < T$.

The threshold value T determines the percentage of missing values. We consider three settings: 1) $\pi = 0.25$ and $T = \Phi^{-1}(0.25)$, 2) $\pi = 0.5$ and $T = \Phi^{-1}(0.5)$. and 3) $\pi = 0.75$ and $T = \Phi^{-1}(0.75)$ where $\Phi(\cdot)$ is the standard Normal cumulative distribution function. The first missing data mechanism is MCAR as the missing values do not depend on the observed values. The second missing data mechanism is MAR as the missing value indicators depend on the observed values of other variables. Finally, the third missing data mechanism is NMAR as the missing data indicators depend on the unobserved values.

Results of the simulation, averaged over 50 independent runs, are summarized in Table 9.3 and Table 9.4. We first observe that when the missing values are not missing at random, performance of all procedures degrades. Furthermore, the EM algorithm performs better than the other two methods when the data is generated under MAR. This is expected, since our proposed procedure is not valid under this assumption. Note, however, that mGLasso performs better than PGLasso under this simulation scenario.

9.6 Discussion and extensions

We have proposed a simple estimator for the precision matrix in high-dimensions from data with missing values. The estimator is based on a convex program that can be solved efficiently. In particular, from our simulation studies, we observed that the algorithm runs on average 20 times

		MissGLasso	mGLasso	PGLasso
			Model 1	
p=100	0%	NA	2.10(0.01)	4.35(0.01)
	10%	2.25(0.01)	2.31(0.01)	4.69(0.01)
	20%	2.35(0.04)	2.42(0.03)	4.78(0.04)
	30%	2.69(0.05)	2.85(0.04)	4.82(0.06)
p=200	0%	NA	2.26(0.01)	4.49(0.01)
	10%	2.32(0.01)	2.73(0.01)	4.76(0.02)
	20%	2.51(0.01)	2.88(0.01)	4.86(0.02)
	30%	2.96(0.02)	3.04(0.01)	4.98(0.05)
p=500	0%	NA	3.59(0.03)	4.94(0.03)
	10%	3.71(0.02)	3.85(0.02)	5.25(0.04)
	20%	3.99(0.03)	3.99(0.02)	5.32(0.04)
	30%	4.11(0.05)	4.77(0.04)	5.76(0.05)
			Model 2	
p=100	0%	NA	1.25(0.01)	1.63(0.01)
	10%	1.32(0.01)	1.66(0.01)	1.75(0.01)
	20%	1.59(0.01)	1.75(0.01)	1.88(0.02)
	30%	1.66(0.02)	1.86(0.01)	1.99(0.02)
p=200	0%	NA	1.31(0.01)	1.69(0.01)
	10%	1.41(0.01)	1.71(0.01)	1.71(0.01)
	20%	1.61(0.01)	1.79(0.02)	1.99(0.01)
	30%	1.69(0.01)	1.87(0.01)	2.08(0.01)
p=500	0%	NA	1.44(0.01)	1.73(0.01)
	10%	1.49(0.01)	1.74(0.01)	1.84(0.02)
	20%	1.66(0.01)	1.81(0.02)	2.05(0.03)
	30%	1.72(0.02)	1.95(0.02)	2.22(0.04)
			Model 3	
p=100	0%	NA	1.12(0.01)	1.35(0.01)
	10%	1.16(0.01)	1.32(0.01)	1.42(0.02)
	20%	1.20(0.01)	1.64(0.02)	1.70(0.03)
	30%	1.49(0.05)	1.67(0.03)	1.83(0.03)
p=200	0%	NA	1.35(0.01)	1.59(0.01)
	10%	1.43(0.01)	1.62(0.01)	1.83(0.01)
	20%	1.46(0.03)	1.71(0.02)	1.87(0.01)
	30%	1.52(0.03)	1.82(0.01)	1.93(0.03)
p=500	0%	NA	1.42(0.01)	1.64(0.02)
	10%	1.47(0.01)	1.69(0.02)	1.86(0.01)
	20%	1.55(0.02)	1.73(0.04)	1.92(0.03)
	30%	1.59(0.02)	1.87(0.03)	2.01(0.03)

Table 9.2: Average (standard deviation) distance in the operator norm $|||\mathbf{\Omega} - \widehat{\mathbf{\Omega}}|||_2$ under the MCAR assumption.

		MissGLasso	mGLasso	PGLasso
$\pi = 0.25$	MCAR	2.88(0.02)	3.16(0.01)	3.72(0.01)
	MAR	3.24(0.01)	3.92(0.03)	4.15(0.05)
	NMAR	5.78(0.05)	6.57(0.08)	7.64(0.10)
$\pi = 0.5$	MCAR	2.97(0.03)	3.28(0.02)	3.77(0.02)
	MAR	3.41(0.05)	4.16(0.06)	4.58(0.04)
	NMAR	6.15(0.07)	6.61(0.10)	8.12(0.12)
$\pi = 0.75$	MCAR	3.17(0.02)	3.31(0.03)	3.99(0.03)
	MAR	3.59(0.05)	4.47(0.04)	4.87(0.05)
	NMAR	6.87(0.11)	7.04(0.13)	8.76(0.15)

Table 9.3: Average (standard deviation) distance in the operator norm $\|\|\boldsymbol{\Omega} - \widehat{\boldsymbol{\Omega}}\|\|_2$ when missing values mechanism is MCAR, MAR and NMAR. The fraction of the observed data is controlled by π.

		Recall			Precision		
		MissGLasso	mGLasso	PGLasso	MissGLasso	mGLasso	PGLasso
$\pi = 0.25$	MCAR	0.900(0.003)	0.950(0.005)	1.000(0.000)	0.900(0.002)	0.861(0.006)	0.333(0.030)
	MAR	0.512(0.026)	0.815(0.070)	0.501(0.067)	0.995(0.006)	0.471(0.052)	0.634(0.025)
	NMAR	0.500(0.015)	0.443(0.052)	0.465(0.112)	0.698(0.086)	0.188(0.021)	0.213(0.091)
$\pi = 0.5$	MCAR	0.800(0.005)	0.900(0.003)	1.000(0.000)	0.889(0.008)	0.774(0.068)	0.263(0.050)
	MAR	0.650(0.034)	0.900(0.005)	0.551(0.061)	0.921(0.021)	0.393(0.089)	0.453(0.072)
	NMAR	0.531(0.042)	0.613(0.477)	0.463(0.073)	0.684(0.092)	0.370(0.285)	0.315(0.109)
$\pi = 0.75$	MCAR	0.626(0.062)	0.635(0.220)	0.775(0.081)	0.924(0.053)	0.891(0.063)	0.221(0.039)
	MAR	0.619(0.014)	0.611(0.132)	0.431(0.075)	0.879(0.061)	0.555(0.074)	0.399(0.044)
	NMAR	0.491(0.046)	0.557(0.115)	0.411(0.076)	0.688(0.059)	0.464(0.067)	0.368(0.071)

Table 9.4: Average (standard deviation) recall and precision when missing values mechanism is MCAR, MAR and NMAR.

faster than the EM algorithm. Furthermore, the estimator does not require imputation of the missing values and can be found using existing numerical procedures. As such, we believe that it represents a viable alternative to the iterative EM algorithm.

From the analysis in Section 9.4, it is clear that other procedures for estimating precision matrices from fully observed data, such as the Clime estimator [37], could be easily extended to handle data with missing values. Theoretical properties of those procedures would be established using the tail bounds on the sample covariance matrix given in Lemma 9.1.

There are two directions in which this work should be extended. First, the MCAR assumption is very strong and it is hard to check whether it holds in practice. However, we have observed in our simulation studies that under the MAR assumption, which is a more realistic assumption than MCAR, performance of the estimators does not degrade dramatically when estimating the support of the precision matrix. However, estimated parameters are quite far from the true parameters. This could be improved by using a weighted estimator for the sample covariance matrix (see, for example, [156]). Second, it is important to establish sharp lower bounds for the estimation problem from data with missing values, which should reflect dependence on the proportion of observed entries γ (see [122]).

Chapter 10

Estimation of Networks From Multi-attribute Data

The existing methods for estimating structure of undirected graphical models focus on data where each node represents a scalar random variable, even though, in many real world problems, nodes are representing multivariate variables, such as images, text or multi-view feature vectors. In this chapter, we study a principled framework for estimating structure of undirected graphical models from multivariate (or multi-attribute) nodal data. The structure of a graph is estimated through estimation of non-zero partial canonical correlation between nodes, which under the Gaussian model is equivalent to estimating conditional independencies between random vectors represented by the nodes. We develop a method that efficiently minimize the penalized Gaussian likelihood. Extensive simulation studies demonstrate the effectiveness of the method under various conditions. We provide illustrative applications to uncovering gene regulatory networks from gene and protein profiles, and uncovering brain connectivity graph from functional magnetic resonance imaging data. Finally, we provide sufficient conditions which guarantee consistent graph recovery.

10.1 Motivation

Undirected Gaussian graphical models are commonly used to represent and explore conditional independencies between variables in a complex system. As we discuss in §2, these conditional dependencies are represented by a network, where an edge connects two conditionally dependent random variables. Current approaches to estimating structure of an undirected graphical model focus on cases where nodes represent scalar variables, however, in many modern problems, we are interested in studying a network where nodes represent vector variables or multi-attribute objects. For example, when modeling a social network, a node may correspond to a person for which a vector of attributes is available, such as personal information, demographics, interests, and other features. In the current literature on social graph estimation based on Markov random fields it is commonly assumed that a node represents a scalar variable, such as a binary vote (see for example [19, 112]). As another example, consider modeling gene regulatory networks. A node in a graphical model corresponds to a gene and the graph structure is estimated from gene

expression levels (see for example [146]). However, due to advances of modern data acquisition technologies, researchers are able to measure the activities of a single gene in a high-dimensional space, such as an image of the spatial distribution of the gene expression, or a multi-view snapshot of the gene activity such as mRNA and protein abundances. Therefore, there is a need for methods that estimate the structure of an undirected graphical model from multi-attribute data.

In this chapter, we present new methodology for estimating the structure of undirected graphical models where nodes correspond to vectors, that is, multi-attribute objects. We consider the following setting. Let $\mathbf{X} = (\mathbf{X}_1', ..., \mathbf{X}_p')'$ where $\mathbf{X}_1 \in \mathbb{R}^{k_1}, \ldots, \mathbf{X}_p \in \mathbb{R}^{k_p}$ are random vectors that jointly follow a multivariate Gaussian distribution with mean $\boldsymbol{\mu} = (\boldsymbol{\mu}_1', \ldots, \boldsymbol{\mu}_p')'$ and covariance matrix $\boldsymbol{\Sigma}^*$, which is partitioned as

$$\boldsymbol{\Sigma}^* = \begin{pmatrix} \boldsymbol{\Sigma}^*_{11} & \cdots & \boldsymbol{\Sigma}^*_{1p} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}^*_{p1} & \cdots & \boldsymbol{\Sigma}^*_{pp} \end{pmatrix}, \tag{10.1}$$

with $\boldsymbol{\Sigma}^*_{ij} = \mathrm{Cov}(\mathbf{X}_i, \mathbf{X}_j)$. Without loss of generality, we assume $\boldsymbol{\mu} = \mathbf{0}$. Let $G = (V, E)$ be a graph with the vertex set $V = \{1, \ldots, p\}$ and the set of edges $E \subseteq V \times V$ that encodes the conditional independence relationships among $(\mathbf{X}_a)_{a \in V}$. That is, each node $a \in V$ of the graph G corresponds to the random vector $\mathbf{X}_a$ and there is no edge between nodes a and b in the graph if and only if $\mathbf{X}_a$ is conditionally independent of $\mathbf{X}_b$ given all the vectors corresponding to the remaining nodes, $\mathbf{X}_{\neg ab} = \{\mathbf{X}_c \ : \ c \in V \setminus \{a, b\}\}$. Such a graph is also known as a Markov network (of Markov graph), which we shall emphasize in this chapter to compare with an alternative graph over V known as the association network, which is based on pairwise marginal independence. Conditional independence can be read from the inverse of the covariance matrix, as the block corresponding to $\mathbf{X}_a$ and $\mathbf{X}_b$ will be equal to zero. Let $\mathcal{D}_n = \{\mathbf{x}_i\}_{i=1}^n$ be a sample of n independent and identically distributed vectors drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. For a vector $\mathbf{x}_i$, we denote $x_{i,a} \in \mathbb{R}^{k_a}$ the component corresponding to the node $a \in V$. Our goal is to estimate the structure of the graph G from the sample $\mathcal{D}_n$. Note that we allow for different nodes to have different number of attributes, which is useful in many applications, e.g., when a node represents a gene pathway in a regulatory network.

Methods discussed in §2 cannot be extended to handle multi-attribute data in an obvious way. For example, if the number of attributes is the same for each node, one may naively estimate one graph per attribute, however, it is not clear how to combine such graphs into a summary graph with a clear statistical interpretation. The situation becomes even more difficult when nodes correspond to objects that have different number of attributes.

In a related work, [114] use canonical correlation to estimate association networks from multi-attribute data, however, such networks have different interpretation to undirected graphical models. In particular, association networks are known to confound the direct interactions with indirect ones as they only represent marginal associations, where as undirected graphical models represent conditional independence assumptions that are better suited for separating direct interactions from indirect confounders. Our work is related to the literature on simultaneous estimation of multiple Gaussian graphical models under a multi-task setting [30, 45, 77, 88, 179]. However, the model given in (10.1) is different from models considered in various multi-task settings and the optimization algorithms developed in the multi-task literature do not extend to

handle the optimization problem given in our setting.

Unlike the standard procedures for estimating the structure of Gaussian graphical models (e.g., neighborhood selection [135] or glasso [71]), which infer the partial correlations between pairs of nodes, our proposed method estimates the graph structure based on the partial canonical correlation, which can naturally incorporate complex nodal observations. Under that the Gaussian model in (10.1), the estimated graph structure has the same probabilistic independence interpretations as the Gaussian graphical model over univariate nodes. The main contributions of the chapter are the following. First, we introduce a new framework for learning structure of undirected graphical models from multi-attribute data. Second, we develop an efficient algorithm that estimates the structure of a graph from the observed data. Third, we provide extensive simulation studies that demonstrate effectiveness of our method and illustrate how the framework can be used to uncover gene regulatory networks from gene and protein profiles, and to uncover brain connectivity graph from functional magnetic resonance imaging data. Finally, we provide theoretical results, which give sufficient conditions for consistent structure recovery.

10.2 Methodology

In this section, we propose to estimate the graph by estimating non-zero partial canonical correlation between the nodes. This leads to a penalized maximum likelihood objective, for which we develop an efficient optimization procedure.

10.2.1 Preliminaries

Let $\mathbf{X}_a$ and $\mathbf{X}_b$ be two multivariate random vectors. Canonical correlation is defined between $\mathbf{X}_a$ and $\mathbf{X}_b$ as

$$\rho_c(\mathbf{X}_a, \mathbf{X}_b) = \max_{\mathbf{u}\in\mathbb{R}^{k_a}, \mathbf{v}\in\mathbb{R}^{k_b}} \mathrm{Corr}(\mathbf{u}'\mathbf{X}_a, \mathbf{v}'\mathbf{X}_b).$$

That is, computing canonical correlation between $\mathbf{X}_a$ and $\mathbf{X}_b$ is equivalent to maximizing the correlation between two linear combinations $\mathbf{u}'\mathbf{X}_a$ and $\mathbf{v}'\mathbf{X}_b$ with respect to vectors $\mathbf{u}$ and $\mathbf{v}$. Canonical correlation can be used to measure association strength between two nodes with multi-attribute observations. For example, in [114], a graph is estimated from multi-attribute nodal observations by elementwise thresholding the canonical correlation matrix between nodes, but such a graph estimator may confound the direct interactions with indirect ones.

We exploit the partial canonical correlation to estimate a graph from multi-attribute nodal observations. A graph is going to be formed by connecting nodes with non-zero partial canonical correlation. Let $\widehat{\mathbf{A}} = \mathrm{argmin}\ \mathbb{E}\left(||\mathbf{X}_a - \mathbf{A}\mathbf{X}_{\neg ab}||_2^2\right)$ and $\widehat{\mathbf{B}} = \mathrm{argmin}\ \mathbb{E}\left(||\mathbf{X}_b - \mathbf{B}\mathbf{X}_{\neg ab}||_2^2\right)$, then the partial canonical correlation between $\mathbf{X}_a$ and $\mathbf{X}_b$ is defined as

$$\rho_c(\mathbf{X}_a, \mathbf{X}_b; \mathbf{X}_{\neg ab}) = \max_{\mathbf{u}\in\mathbb{R}^{k_a}, \mathbf{v}\in\mathbb{R}^{k_b}} \mathrm{Corr}\{\mathbf{u}'(\mathbf{X}_a - \widehat{\mathbf{A}}\mathbf{X}_{\neg ab}), \mathbf{v}'(\mathbf{X}_b - \widehat{\mathbf{B}}\mathbf{X}_{\neg ab})\},$$

that is, the partial canonical correlation between $\mathbf{X}_a$ and $\mathbf{X}_b$ is equal to the canonical correlation between the residual vectors of $\mathbf{X}_a$ and $\mathbf{X}_b$ after the effect of $\mathbf{X}_{\neg ab}$ is removed [21].

Let $\mathbf{\Omega}^*$ denote the precision matrix under the model in (10.1). Using standard results for the multivariate Gaussian distribution (see also Equation (7) in [21]), a straightforward calculation (given in §10.8.3) shows that

$$\rho_c(\mathbf{X}_a, \mathbf{X}_b; \mathbf{X}_{\neg ab}) \neq 0 \quad \text{if and only if} \quad \max_{\mathbf{u}\in\mathbb{R}^{k_a}, \mathbf{v}\in\mathbb{R}^{k_b}} \mathbf{u}'\mathbf{\Omega}^*_{ab}\mathbf{v} \neq 0. \tag{10.2}$$

This implies that estimating whether the partial canonical correlation is zero or not can be done by estimating whether a block of the precision matrix is zero or not. Furthermore, under the model in (10.1), vectors $\mathbf{X}_a$ and $\mathbf{X}_b$ are conditionally independent given $\mathbf{X}_{\neg ab}$ if and only if partial canonical correlation is zero. A graph built on this type of inter-nodal relationship is known as a Markov graph, as it captures both local and global Markov properties over all arbitrary subsets of nodes in the graph, even though the graph is built based on pairwise conditional independence properties. In §10.2.2, we use the above observations to design an algorithm that estimates the non-zero partial canonical correlation between nodes from data $\mathcal{D}_n$ using the penalized maximum likelihood estimation of the precision matrix.

Based on the relationship given in (10.2), we can motivate an alternative method for estimating the non-zero partial canonical correlation. Let $\overline{a} = \{b \ : \ b \in V \setminus \{a\}\}$ denote the set of all nodes minus the node a. Then

$$\mathbb{E}\left(\mathbf{X}_a \mid \mathbf{X}_{\overline{a}} = \mathbf{x}_{\overline{a}}\right) = \mathbf{\Sigma}^*_{a,\overline{a}}\mathbf{\Sigma}^{*,-1}_{\overline{a},\overline{a}}\mathbf{x}_{\overline{a}}.$$

Since $\mathbf{\Omega}^*_{a,\overline{a}} = -(\mathbf{\Sigma}^*_{aa} - \mathbf{\Sigma}^*_{a,\overline{a}}\mathbf{\Sigma}^{*,-1}_{\overline{a},\overline{a}}\mathbf{\Sigma}^*_{\overline{a},a})^{-1}\mathbf{\Sigma}^*_{a,\overline{a}}\mathbf{\Sigma}^{*,-1}_{\overline{a},\overline{a}}$, we observe that a zero block $\mathbf{\Omega}_{ab}$ can be identified from the regression coefficients when each component of $\mathbf{X}_a$ is regressed on $\mathbf{X}_{\overline{a}}$. We do not build an estimation procedure around this observation, however, we note that this relationship shows how one would develop a regression based analogue of the work presented in [114].

10.2.2 Penalized Log-Likelihood Optimization

Based on the data $\mathcal{D}_n$, we propose to minimize the penalized negative Gaussian log-likelihood under the model in (10.1),

$$\min_{\mathbf{\Omega}\succ\mathbf{0}} \left\{ \operatorname{tr} \mathbf{S}\mathbf{\Omega} - \log|\mathbf{\Omega}| + \lambda \sum_{a,b} ||\mathbf{\Omega}_{ab}||_F \right\} \tag{10.3}$$

where $\mathbf{S} = n^{-1}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i'$ is the sample covariance matrix, $||\mathbf{\Omega}_{ab}||_F)$ denotes the Frobenius norm of $\mathbf{\Omega}_{ab}$ and λ is a user defined parameter that controls the sparsity of the solution $\widehat{\mathbf{\Omega}}$. The Frobenius norm penalty encourages blocks of the precision matrix to be equal to zero, similar to the way that the ℓ_2 penalty is used in the group Lasso [194]. Here we assume that the same number of samples is available per attribute. However, the same method can be used in cases when some samples are obtained on a subset of attributes. Indeed, we can simply estimate each element of the matrix $\mathbf{S}$ from available samples, treating non-measured attributes as missing completely at random (for more details see [107] and §9).

The dual problem to (10.3) is

$$\max_{\mathbf{\Sigma}} \sum_{j\in V} k_j + \log|\mathbf{\Sigma}| \qquad \text{subject to} \qquad \max_{a,b} \ ||\mathbf{S}_{ab} - \mathbf{\Sigma}_{ab}||_F \leq \lambda,$$

where $\mathbf{\Sigma}$ is the dual variable to $\mathbf{\Omega}$ and $|\mathbf{\Sigma}|$ denotes the determinant of $\mathbf{\Sigma}$. Note that the primal problem gives us an estimate of the precision matrix, while the dual problem estimates the covariance matrix. The proposed optimization procedure, described below, will simultaneously estimate the precision matrix and covariance matrix, without explicitly performing an expensive matrix inversion.

We propose to optimize the objective function in (10.3) using an inexact block coordinate descent procedure, inspired by [138]. The block coordinate descent is an iterative procedure that operates on a block of rows and columns while keeping the other rows and columns fixed. We write

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{\Omega}_{aa} & \mathbf{\Omega}_{a,\overline{a}} \\ \mathbf{\Omega}_{\overline{a},a} & \mathbf{\Omega}_{\overline{a},\overline{a}} \end{pmatrix}, \quad \mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_{aa} & \mathbf{\Sigma}_{a,\overline{a}} \\ \mathbf{\Sigma}_{\overline{a},a} & \mathbf{\Sigma}_{\overline{a},\overline{a}} \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} \mathbf{S}_{aa} & \mathbf{S}_{a,\overline{a}} \\ \mathbf{S}_{\overline{a},a} & \mathbf{S}_{\overline{a},\overline{a}} \end{pmatrix}$$

and suppose that $(\widetilde{\mathbf{\Omega}}, \widetilde{\mathbf{\Sigma}})$ are the current estimates of the precision matrix and covariance matrix. With the above block partition, we have $\log|\mathbf{\Omega}| = \log(\mathbf{\Omega}_{\overline{a},\overline{a}}) + \log(\mathbf{\Omega}_{aa} - \mathbf{\Omega}_{a,\overline{a}}(\mathbf{\Omega}_{\overline{a},\overline{a}})^{-1}\mathbf{\Omega}_{\overline{a},a})$. In the next iteration, $\widehat{\mathbf{\Omega}}$ is of the form

$$\widehat{\mathbf{\Omega}} = \widetilde{\mathbf{\Omega}} + \begin{pmatrix} \mathbf{\Delta}_{aa} & \mathbf{\Delta}_{a,\overline{a}} \\ \mathbf{\Delta}_{\overline{a},a} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \widehat{\mathbf{\Omega}}_{aa} & \widehat{\mathbf{\Omega}}_{a,\overline{a}} \\ \widehat{\mathbf{\Omega}}_{\overline{a},a} & \widetilde{\mathbf{\Omega}}_{\overline{a},\overline{a}} \end{pmatrix}$$

and is obtained by minimizing

$$\operatorname{tr}\mathbf{S}_{aa}\mathbf{\Omega}_{aa} + 2\operatorname{tr}\mathbf{S}_{a,\overline{a}}\mathbf{\Omega}_{\overline{a},a} - \log|\mathbf{\Omega}_{aa} - \mathbf{\Omega}_{a,\overline{a}}(\widetilde{\mathbf{\Omega}}_{\overline{a},\overline{a}})^{-1}\mathbf{\Omega}_{\overline{a},a}| + \lambda||\mathbf{\Omega}_{aa}||_F + 2\lambda\sum_{b\neq a}||\mathbf{\Omega}_{ab}||_F. \quad (10.4)$$

Exact minimization over the variables $\mathbf{\Omega}_{aa}$ and $\mathbf{\Omega}_{a,\overline{a}}$ at each iteration of the block coordinate descent procedure can be computationally expensive. Therefore, we propose to update $\mathbf{\Omega}_{aa}$ and $\mathbf{\Omega}_{a,\overline{a}}$ using one generalized gradient step update (see [11]) in each iteration. Note that the objective function in (10.4) is a sum of a smooth convex function and a non-smooth convex penalty so that the gradient descent method cannot be directly applied. Given a step size t, generalized gradient descent optimizes a quadratic approximation of the objective at the current iterate $\widetilde{\mathbf{\Omega}}$, which results in the following two updates

$$\widehat{\mathbf{\Omega}}_{aa} = \underset{\mathbf{\Omega}_{aa}}{\operatorname{argmin}}\Big\{\operatorname{tr}(\mathbf{S}_{aa} - \widetilde{\mathbf{\Sigma}}_{aa})\mathbf{\Omega}_{aa} + \frac{1}{2t}||\mathbf{\Omega}_{aa} - \widetilde{\mathbf{\Omega}}_{aa}||_F^2 + \lambda||\mathbf{\Omega}_{aa}||_F\Big\}, \quad \text{and} \quad (10.5)$$

$$\widehat{\mathbf{\Omega}}_{ab} = \underset{\mathbf{\Omega}_{ab}}{\operatorname{argmin}}\Big\{\operatorname{tr}(\mathbf{S}_{ab} - \widetilde{\mathbf{\Sigma}}_{ab})\mathbf{\Omega}_{ba} + \frac{1}{2t}||\mathbf{\Omega}_{ab} - \widetilde{\mathbf{\Omega}}_{ab}||_F^2 + \lambda||\mathbf{\Omega}_{ab}||_F\Big\}, \quad \forall b \in \overline{a}. \quad (10.6)$$

Solutions to (10.5) and (10.6) can be computed in a closed form as

$$\widehat{\mathbf{\Omega}}_{aa} = (1 - t\lambda/||\widetilde{\mathbf{\Omega}}_{aa} + t(\widetilde{\mathbf{\Sigma}}_{aa} - \mathbf{S}_{aa})||_F)_+(\widetilde{\mathbf{\Omega}}_{aa} + t(\widetilde{\mathbf{\Sigma}}_{aa} - \mathbf{S}_{aa})), \quad \text{and} \quad (10.7)$$

$$\widehat{\mathbf{\Omega}}_{ab} = (1 - t\lambda/||\widetilde{\mathbf{\Omega}}_{ab} + t(\widetilde{\mathbf{\Sigma}}_{ab} - \mathbf{S}_{ab})||_F)_+(\widetilde{\mathbf{\Omega}}_{ab} + t(\widetilde{\mathbf{\Sigma}}_{ab} - \mathbf{S}_{ab})), \quad \forall b \in \overline{a}, \quad (10.8)$$

where $(x)_+ = \max(0, x)$. If the resulting estimator $\widehat{\mathbf{\Omega}}$ is not positive definite or the update does not decrease the objective, we halve the step size t and find a new update. Once the update of the

precision matrix $\widehat{\mathbf{\Omega}}$ is obtained, we update the covariance matrix $\widehat{\mathbf{\Sigma}}$. This update can be found efficiently, without inverting the whole $\widehat{\mathbf{\Omega}}$ matrix, using the matrix inversion lemma as follows

$$\begin{aligned}
\widehat{\mathbf{\Sigma}}_{\overline{a},\overline{a}} &= (\widetilde{\mathbf{\Omega}}_{\overline{a},\overline{a}})^{-1} + (\widetilde{\mathbf{\Omega}}_{\overline{a},\overline{a}})^{-1}\widehat{\mathbf{\Omega}}_{\overline{a},a}(\widehat{\mathbf{\Omega}}_{aa} - \widehat{\mathbf{\Omega}}_{a,\overline{a}}(\widetilde{\mathbf{\Omega}}_{\overline{a},\overline{a}})^{-1}\widehat{\mathbf{\Omega}}_{\overline{a},a})^{-1}\widehat{\mathbf{\Omega}}_{a,\overline{a}}(\widetilde{\mathbf{\Omega}}_{\overline{a},\overline{a}})^{-1}, \\
\widehat{\mathbf{\Sigma}}_{a,\overline{a}} &= -\widehat{\mathbf{\Omega}}_{aa}\widehat{\mathbf{\Omega}}_{a,\overline{a}}\widehat{\mathbf{\Sigma}}_{\overline{a},\overline{a}}, \\
\widehat{\mathbf{\Sigma}}_{aa} &= (\widehat{\mathbf{\Omega}}_{aa} - \widehat{\mathbf{\Omega}}_{a,\overline{a}}(\widetilde{\mathbf{\Omega}}_{\overline{a},\overline{a}})^{-1}\widehat{\mathbf{\Omega}}_{\overline{a},a})^{-1},
\end{aligned} \tag{10.9}$$

with $(\widetilde{\mathbf{\Omega}}_{\overline{a},\overline{a}})^{-1} = \widetilde{\mathbf{\Sigma}}_{\overline{a},\overline{a}} - \widetilde{\mathbf{\Sigma}}_{\overline{a},a}\widetilde{\mathbf{\Sigma}}_{aa}^{-1}\widetilde{\mathbf{\Sigma}}_{a,\overline{a}}$. Combining all three steps we get the following algorithm:

1. Set the initial estimator $\widetilde{\mathbf{\Omega}} = \mathrm{diag}(\mathbf{S})$ and $\widetilde{\mathbf{\Sigma}} = \widetilde{\mathbf{\Omega}}^{-1}$. Set the step size $t = 1$.
2. For each $a \in V$ perform the following:

   Update $\widehat{\mathbf{\Omega}}$ using (10.7) and (10.8).
   If $\widehat{\mathbf{\Omega}}$ is not positive definite, set $t \leftarrow t/2$ and repeat the update.
   Update $\widehat{\mathbf{\Sigma}}$ using (10.9).
3. Repeat Step 2 until the duality gap

$$\left|\mathrm{tr}(\mathbf{S}\widehat{\mathbf{\Omega}}) - \log|\widehat{\mathbf{\Omega}}| + \lambda\sum_{a,b}||\widehat{\mathbf{\Omega}}_{ab}||_F - \sum_{j\in V}k_j - \log|\mathbf{\Sigma}|\right| \leq \epsilon,$$

   where ϵ is a prefixed precision parameter (for example, $\epsilon = 10^{-3}$).

Finally, we form a graph $\widehat{G} = (V, \widehat{E})$ by connecting nodes with $||\widehat{\mathbf{\Omega}}_{ab}||_F \neq 0$.

Step 2 of the estimation algorithm updates portions of the precision and covariance matrices corresponding to one node at a time. We observe that the computational complexity of updating the precision matrix is $\mathcal{O}(pk^2)$. Updating the covariance matrix requires computing $(\widetilde{\mathbf{\Omega}}_{\overline{a},\overline{a}})^{-1}$, which can be efficiently done in $\mathcal{O}(p^2k^2 + pk^2 + k^3) = \mathcal{O}(p^2k^2)$ operations, assuming that $k \ll p$. With this, the covariance matrix can be updated in $\mathcal{O}(p^2k^2)$ operations. Therefore the total cost of updating the covariance and precision matrices is $\mathcal{O}(p^2k^2)$ operations. Since step 2 needs to be performed for each node $a \in V$, the total complexity is $\mathcal{O}(p^3k^2)$. Let T denote the total number of times step 2 is executed. This leads to the overall complexity of the algorithm as $\mathcal{O}(Tp^3k^2)$. In practice, we observe that $T \approx 10$ to 20 for sparse graphs. Furthermore, when the whole solution path is computed, we can use warm starts to further speed up computation, leading to $T < 5$ for each λ.

Convergence of the above described procedure to the unique minimum of the objective function in (10.3) does not follow from the standard results on the block coordinate descent algorithm [169] for two reasons. First, the minimization problem in (10.4) is not solved exactly at each iteration, since we only update $\mathbf{\Omega}_{aa}$ and $\mathbf{\Omega}_{a,\overline{a}}$ using one generalized gradient step update in each iteration. Second, the blocks of variables, over which the optimization is done at each iteration, are not completely separable between iterations due to the symmetry of the problem. The proof of the following convergence result is given in §10.8.

Lemma 10.1. *For every value of $\lambda > 0$, the above described algorithm produces a sequence of estimates $\left\{\widetilde{\mathbf{\Omega}}^{(t)}\right\}_{t\geq 1}$ of the precision matrix that monotonically decrease the objective values given in* (10.3)*. Every element of this sequence is positive definite and the sequence converges to the unique minimizer $\widehat{\mathbf{\Omega}}$ of* (10.3)*.*

10.2.3 Efficient Identification of Connected Components

When the target graph $\widehat{G}$ is composed of smaller, disconnected components, the solution to the problem in (10.3) is block diagonal (possibly after permuting the node indices) and can be obtained by solving smaller optimization problems. That is, the minimizer $\widehat{\mathbf{\Omega}}$ can be obtained by solving (10.3) for each connected component independently, resulting in massive computational gains. We give necessary and sufficient condition for the solution $\widehat{\mathbf{\Omega}}$ of (10.3) to be block-diagonal, which can be easily checked by inspecting the empirical covariance matrix $\mathbf{S}$.

Our first result follows immediately from the Karush-Kuhn-Tucker conditions for the optimization problem (10.3) and states that if $\widehat{\mathbf{\Omega}}$ is block-diagonal, then it can be obtained by solving a sequence of smaller optimization problems.

Lemma 10.2. *If the solution to* (10.3) *takes the form* $\widehat{\mathbf{\Omega}} = \mathrm{diag}(\widehat{\mathbf{\Omega}}_1, \widehat{\mathbf{\Omega}}_2, \ldots, \widehat{\mathbf{\Omega}}_l)$*, that is,* $\widehat{\mathbf{\Omega}}$ *is a block diagonal matrix with the diagonal blocks* $\widehat{\mathbf{\Omega}}_1, \ldots, \widehat{\mathbf{\Omega}}_l$*, then it can be obtained by solving*

$$\min_{\mathbf{\Omega}_{l'} \succ \mathbf{0}} \left\{ \mathrm{tr}\, \mathbf{S}_{l'} \mathbf{\Omega}_{l'} - \log |\mathbf{\Omega}_{l'}| + \lambda \sum_{a,b} ||\mathbf{\Omega}_{ab}||_F \right\}$$

separately for each $l' = 1, \ldots, l$*, where* $\mathbf{S}_{l'}$ *are submatrices of* $\mathbf{S}$ *corresponding to* $\mathbf{\Omega}_{l'}$*.*

Next, we describe how to identify diagonal blocks of $\widehat{\mathbf{\Omega}}$. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_l\}$ be a partition of the set V and assume that the nodes of the graph are ordered in a way that if $a \in P_j$, $b \in P_{j'}$, $j < j'$, then $a < b$. The following lemma states that the blocks of $\widehat{\Omega}$ can be obtained from the blocks of a thresholded sample covariance matrix.

Lemma 10.3. *A necessary and sufficient conditions for* $\widehat{\mathbf{\Omega}}$ *to be block diagonal with blocks* $P_1, P_2, \ldots, P_l$ *is that* $||\mathbf{S}_{ab}||_F \leq \lambda$ *for all* $a \in P_j$, $b \in P_{j'}$, $j \neq j'$*.*

Blocks $P_1, P_2, \ldots, P_l$ can be identified by forming a $p \times p$ matrix $\mathbf{Q}$ with elements $q_{ab} = \mathbb{I}\{||\mathbf{S}_{ab}||_F > \lambda\}$ and computing connected components of the graph with adjacency matrix $\mathbf{Q}$. The lemma states also that given two penalty parameters $\lambda_1 < \lambda_2$, the set of unconnected nodes with penalty parameter λ_1 is a subset of unconnected nodes with penalty parameter λ_2. The simple check above allows us to estimate graphs on datasets with large number of nodes, if we are interested in graphs with small number of edges. However, this is often the case when the graphs are used for exploration and interpretation of complex systems. Lemma 10.3 is related to existing results established for speeding-up computation when learning single and multiple Gaussian graphical models [45, 139, 189]. Each condition is different, since the methods optimize different objective functions.

10.3 Consistent Graph Identification

In this section, we provide theoretical analysis of the estimator described in §10.2.2. In particular, we provide sufficient conditions for consistent graph recovery. For simplicity of presentation, we assume that $k_a = k$, for all $a \in V$, that is, we assume that the same number of attributes is observed for each node. For each $a = 1, \ldots, kp$, we assume that $(\sigma^*_{aa})^{-1/2}\mathbf{X}_a$ is sub-Gaussian with parameter γ, where σ^*_{aa} is the ath diagonal element of $\mathbf{\Sigma}^*$. Recall that Z is a sub-Gaussian random variable if there exists a constant $\sigma \in (0, \infty)$ such that

$$\mathbb{E}\left(\exp(tZ)\right) \leq \exp(\sigma^2 t^2), \text{ for all } t \in \mathbb{R}.$$

Our assumptions involve the Hessian of the function $f(\mathbf{A}) = \operatorname{tr}\mathbf{S}\mathbf{A} - \log|\mathbf{A}|$ evaluated at the true $\mathbf{\Omega}^*$, $\mathcal{H} = \mathcal{H}(\mathbf{\Omega}^*) = (\mathbf{\Omega}^*)^{-1} \otimes (\mathbf{\Omega}^*)^{-1} \in \mathbb{R}^{(pk)^2 \times (pk)^2}$, with $\otimes$ denoting the Kronecker product, and the true covariance matrix $\mathbf{\Sigma}^*$. The Hessian and the covariance matrix can be thought of as block matrices with blocks of size $k^2 \times k^2$ and $k \times k$, respectively. We will make use of the operator $\mathcal{C}(\cdot)$ that operates on these block matrices and outputs a smaller matrix with elements that equal to the Frobenius norm of the original blocks. For example, $\mathcal{C}(\mathbf{\Sigma}^*) \in \mathbb{R}^{p \times p}$ with elements $\mathcal{C}(\mathbf{\Sigma}^*)_{ab} = ||\mathbf{\Sigma}^*_{ab}||_F$. Let $\mathcal{T} = \{(a,b) : ||\mathbf{\Omega}_{ab}||_F \neq 0\}$ and $\mathcal{N} = \{(a,b) : ||\mathbf{\Omega}_{ab}||_F = 0\}$. With this notation introduced, we assume that the following irrepresentable condition holds. There exists a constant $\alpha \in [0,1)$ such that

$$\|\mathcal{C}\left(\mathcal{H}_{\mathcal{N}\mathcal{T}}(\mathcal{H}_{\mathcal{T}\mathcal{T}})^{-1}\right)\|_\infty \leq 1 - \alpha,$$

where $\|A\|_\infty = \max_i \sum_j |A_{ij}|$. We will also need the following quantities to specify the results $\kappa_{\mathbf{\Sigma}^*} = \|\mathcal{C}(\mathbf{\Sigma}^*)\|_\infty$ and $\kappa_{\mathcal{H}} = \|\mathcal{C}(\mathcal{H}^{-1}_{\mathcal{T}\mathcal{T}})\|_\infty$. These conditions extend the conditions specified in [152] needed for estimating graphs from single attribute observations.

We have the following result that provides sufficient conditions for the exact recovery of the graph.

Proposition 10.1. *Let $\tau > 2$. We set the penalty parameter λ in* (10.3) *as*

$$\lambda = 8k\alpha^{-1}\left(128(1+4\gamma^2)^2(\max_a(\sigma^*_{aa})^2)n^{-1}(2\log(2k) + \tau\log(p))\right)^{1/2}.$$

If $n > C_1 s^2 k^2 (1+8\alpha^{-1})^2(\tau\log p + \log 4 + 2\log k)$, where s is the maximal degree of nodes in G, $C_1 = (48\sqrt{2}(1+4\gamma^2)(\max_a \sigma^*_{aa})\max(\kappa_{\mathbf{\Sigma}^*}\kappa_{\mathcal{H}}, \kappa^3_{\mathbf{\Sigma}^*}\kappa^2_{\mathcal{H}}))^2$ and

$$\min_{(a,b)\in\mathcal{T}, a\neq b} ||\mathbf{\Omega}_{ab}||_F > 16\sqrt{2}(1+4\gamma^2)(\max_a \sigma^*_{aa})(1+8\alpha^{-1})\kappa_{\mathcal{H}} k\left(\frac{\tau\log p + \log 4 + 2\log k}{n}\right)^{1/2},$$

then $\mathbb{P}\left(\widehat{G} = G\right) \geq 1 - p^{2-\tau}$.

The proof of Proposition 10.1 is given in §10.8. We extend the proof of [152] to accommodate the Frobenius norm penalty on blocks of the precision matrix. This proposition specifies the sufficient sample size and a lower bound on the Frobenius norm of the off-diagonal blocks needed for recovery of the unknown graph. Under these conditions and correctly specified tuning parameter λ, the solution to the optimization problem in (10.3) correctly recovers the graph with high probability. In practice, one needs to choose the tuning parameter in a data dependent way. For example, using the Bayesian information criterion. Even though our theoretical analysis obtains the same rate of convergence as that of [152], our method has a significantly improved finite-sample performance (More details will be provided in §10.5.). It remains an open question whether the sample size requirement can be improved as in the case of group Lasso (see, for example, [123]). The analysis of [123] relies heavily on the special structure of the least squares regression. Hence, their method does not carry over to the more complicated objective function as in (10.3).

10.4 Interpreting Edges

We propose a post-processing step that will allow us to quantify the strength of links identified by the method proposed in §10.2.2, as well as identify important attributes that contribute to the existence of links.

For any two nodes a and b for which $\mathbf{\Omega}_{ab} \neq 0$, we define $\mathcal{N}(a,b) = \{c \in V \setminus \{a,b\} \ : \ \mathbf{\Omega}_{ac} \neq 0 \text{ or } \mathbf{\Omega}_{bc} \neq 0\}$, which is the Markov blanket for the set of nodes $\{\mathbf{X}_a, \mathbf{X}_b\}$. Note that the conditional distribution of $(\mathbf{X}_a', \mathbf{X}_b')'$ given $\mathbf{X}_{\neg ab}$ is equal to the conditional distribution of $(\mathbf{X}_a', \mathbf{X}_b')'$ given $\mathbf{X}_{\mathcal{N}(a,b)}$. Now,

$$\begin{aligned}\rho_c(\mathbf{X}_a, \mathbf{X}_b; \mathbf{X}_{\neg ab}) &= \rho_c(\mathbf{X}_a, \mathbf{X}_b; \mathbf{X}_{\mathcal{N}(a,b)}) \\ &= \max_{\mathbf{w}_a \in \mathbb{R}^{k_a}, \mathbf{w}_b \in \mathbb{R}^{k_b}} \mathrm{Corr}(\mathbf{u}'(\mathbf{X}_a - \widetilde{\mathbf{A}}\mathbf{X}_{\mathcal{N}(a,b)}), \mathbf{v}'(\mathbf{X}_b - \widetilde{\mathbf{B}}\mathbf{X}_{\mathcal{N}(a,b)})),\end{aligned}$$

where $\widetilde{\mathbf{A}} = \mathrm{argmin}\ \mathbb{E}\left(||\mathbf{X}_a - \mathbf{A}\mathbf{X}_{\mathcal{N}(a,b)}||_2^2\right)$ and $\widetilde{\mathbf{B}} = \mathrm{argmin}\ \mathbb{E}\left(||\mathbf{X}_b - \mathbf{B}\mathbf{X}_{\mathcal{N}(a,b)}||_2^2\right)$. Let $\overline{\mathbf{\Sigma}}(a,b) = \mathrm{Var}(\mathbf{X}_a, \mathbf{X}_b \mid \mathbf{X}_{\mathcal{N}(a,b)})$. Now we can express the partial canonical correlation as

$$\rho_c(\mathbf{X}_a, \mathbf{X}_b; \mathbf{X}_{\mathcal{N}(a,b)}) = \max_{\mathbf{w}_a \in \mathbb{R}^{k_a}, \mathbf{w}_b \in \mathbb{R}^{k_a}} \frac{\mathbf{w}_a' \overline{\mathbf{\Sigma}}_{ab} \mathbf{w}_b}{\left(\mathbf{w}_a' \overline{\mathbf{\Sigma}}_{aa} \mathbf{w}_a\right)^{1/2} \left(\mathbf{w}_b' \overline{\mathbf{\Sigma}}_{bb} \mathbf{w}_b\right)^{1/2}}$$

where

$$\overline{\mathbf{\Sigma}}(a,b) = \begin{pmatrix} \overline{\mathbf{\Sigma}}_{aa} & \overline{\mathbf{\Sigma}}_{ab} \\ \overline{\mathbf{\Sigma}}_{ba} & \overline{\mathbf{\Sigma}}_{bb} \end{pmatrix}.$$

The weight vectors $\mathbf{w}_a$ and $\mathbf{w}_b$ can be easily found by solving the system of eigenvalue equations

$$\begin{cases} \overline{\mathbf{\Sigma}}_{aa}^{-1} \overline{\mathbf{\Sigma}}_{ab} \overline{\mathbf{\Sigma}}_{bb}^{-1} \overline{\mathbf{\Sigma}}_{ba} \mathbf{w}_a = \phi^2 \mathbf{w}_a \\ \overline{\mathbf{\Sigma}}_{bb}^{-1} \overline{\mathbf{\Sigma}}_{ba} \overline{\mathbf{\Sigma}}_{aa}^{-1} \overline{\mathbf{\Sigma}}_{ab} \mathbf{w}_b = \phi^2 \mathbf{w}_b \end{cases} \tag{10.10}$$

with $\mathbf{w}_a$ and $\mathbf{w}_b$ being the vectors that correspond to the maximum eigenvalue ϕ^2. Furthermore, we have $\rho_c(\mathbf{X}_a, \mathbf{X}_b; \mathbf{X}_{\mathcal{N}(a,b)}) = \phi$. Following [114], the weights $\mathbf{w}_a$, $\mathbf{w}_b$ can be used to access the relative contribution of each attribute to the edge between the nodes a and b. In particular, the weight $(w_{a,i})^2$ characterizes the relative contribution of the ith attribute of node a to $\rho_c(\mathbf{X}_a, \mathbf{X}_b; \mathbf{X}_{\mathcal{N}(a,b)})$.

Given an estimate $\widehat{\mathcal{N}}(a,b) = \{c \in V \setminus \{a,b\} \ : \ \widehat{\mathbf{\Omega}}_{ac} \neq 0 \text{ or } \widehat{\mathbf{\Omega}}_{bc} \neq 0\}$ of the Markov blanket $\mathcal{N}(a,b)$, we form the residual vectors

$$\mathbf{r}_{i,a} = \mathbf{x}_{i,a} - \check{\mathbf{A}}\mathbf{x}_{i,\widehat{\mathcal{N}}(a,b)}, \qquad \mathbf{r}_{i,b} = \mathbf{x}_{i,b} - \check{\mathbf{B}}\mathbf{x}_{i,\widehat{\mathcal{N}}(a,b)},$$

where $\check{\mathbf{A}}$ and $\check{\mathbf{B}}$ are the least square estimators of $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$. Given the residuals, we form $\check{\mathbf{\Sigma}}(a,b)$, the empirical version of the matrix $\overline{\mathbf{\Sigma}}(a,b)$, by setting

$$\check{\mathbf{\Sigma}}_{aa} = \mathrm{Corr}\left(\{\mathbf{r}_{i,a}\}_{i\in[n]}\right), \quad \check{\mathbf{\Sigma}}_{bb} = \mathrm{Corr}\left(\{\mathbf{r}_{i,b}\}_{i\in[n]}\right), \quad \check{\mathbf{\Sigma}}_{ab} = \mathrm{Corr}\left(\{\mathbf{r}_{i,a}\}_{i\in[n]}, \{\mathbf{r}_{i,a}\}_{i\in[n]}\right).$$

Now, solving the eigenvalue system in (10.10) will give us estimates of the vectors $\mathbf{w}_a$, $\mathbf{w}_b$ and the partial canonical correlation.

Note that we have described a way to interpret the elements of the off-diagonal blocks in the estimated precision matrix. The elements of the diagonal blocks, which correspond to coefficients between attributes of the same node, can still be interpreted by their relationship to the partial correlation coefficients.

10.5 Simulation Studies

In this section, we perform a set of simulation studies to illustrate finite sample performance of our method. We demonstrate that the scalings of (n, p, s) predicted by the theory are sharp. Furthermore, we compare against three other methods: 1) a method that uses the glasso first to estimate one graph over each of the k individual attributes and then creates an edge in the resulting graph if an edge appears in at least one of the single attribute graphs, 2) the method of [77] and 3) the method of [45]. We have also tried applying the glasso to estimate the precision matrix for the model in (10.1) and then post-processing it, so that an edge appears in the resulting graph if the corresponding block of the estimated precision matrix is non-zero. The result of this method is worse compared to the first baseline, so we do not report it here.

All the methods above require setting one or two tuning parameters that control the sparsity of the estimated graph. We select these tuning parameters by minimizing the Bayesian information criterion, which balances the goodness of fit of the model and its complexity, over a grid of parameter values. For our multi-attribute method, the Bayesian information criterion takes the following form

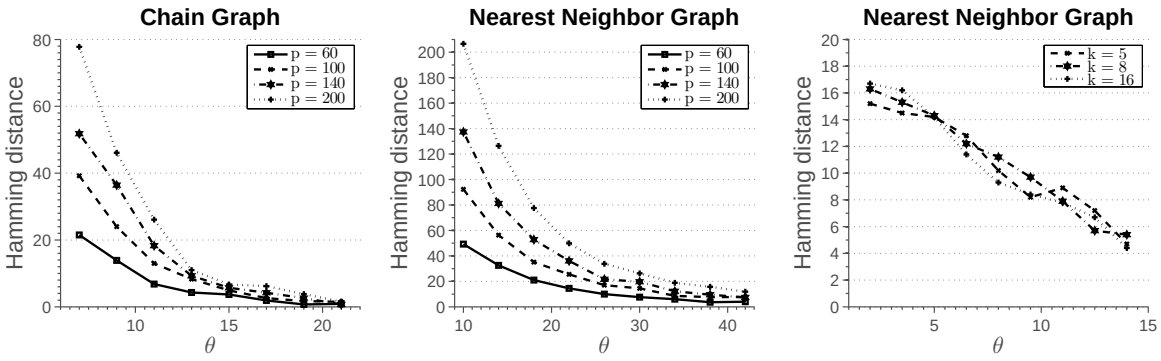
$$\mathrm{BIC}(\lambda) = \mathrm{tr}(\mathbf{S}\widehat{\mathbf{\Omega}}) - \log|\widehat{\mathbf{\Omega}}| + \sum_{a<b} \mathbb{I}\{\widehat{\mathbf{\Omega}}_{ab} \neq \mathbf{0}\} k_a k_b \log(n).$$

Other methods for selecting tuning parameters are possible, like minimization of cross-validation or Akaike information criterion. However, these methods tend to select models that are too dense.

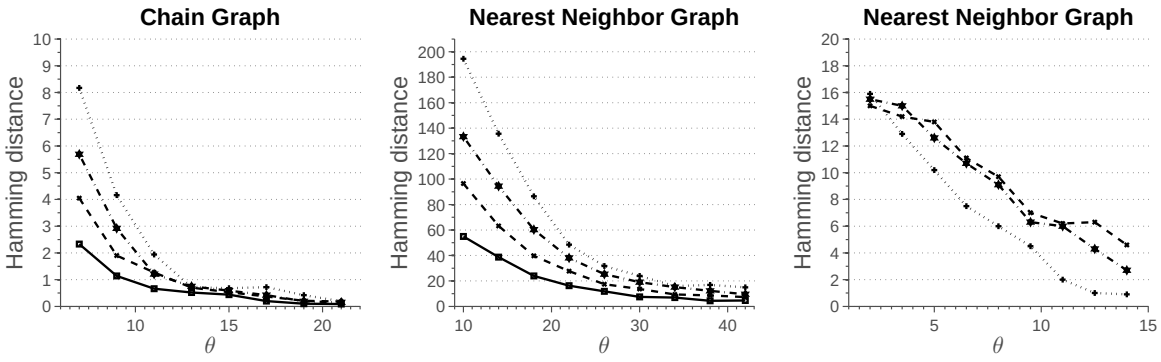
Theoretical results given in §10.3 characterize the sample size needed for consistent recovery of the underlying graph. In particular, Proposition 10.1 suggests that we need $n = \theta s^2 k^2 \log(pk)$ samples to estimate the graph structure consistently, for some $\theta > 0$. Therefore, if we plot the hamming distance between the true and recovered graph against θ, we expect the curves to reach zero distance for different problem sizes at a same point. We verify this on randomly generated chain and nearest-neighbors graphs.

We generate data as follows. A random graph with p nodes is created by first partitioning nodes into $p/20$ connected components, each with 20 nodes, and then forming a random graph over these 20 nodes. A chain graph is formed by permuting the nodes and connecting them in succession, while a nearest-neighbor graph is constructed following the procedure outlined in [119]. That is, for each node, we draw a point uniformly at random on a unit square and compute the pairwise distances between nodes. Each node is then connected to $s = 4$ closest neighbors. Since some of nodes will have more than 4 adjacent edges, we randomly remove edges from nodes that have degree larger than 4 until the maximum degree of a node in a network is 4. Once the graph is created, we construct a precision matrix, with non-zero blocks corresponding to edges in the graph. Elements of diagonal blocks are set as $0.5^{|a-b|}$, $0 \leq a, b \leq k$, while off-diagonal blocks have elements with the same value, 0.2 for chain graphs and $0.3/k$ for nearest-neighbor networks. Finally, we add ρI to the precision matrix, so that its minimum eigenvalue is equal to 0.5. Note that $s = 2$ for the chain graph and $s = 4$ for the nearest-neighbor graph. Simulation results are averaged over 100 replicates.

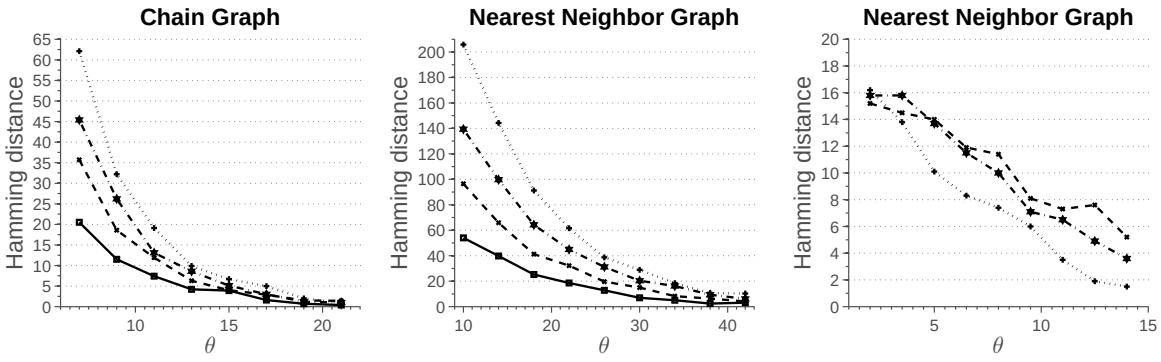
Figure 10.1 shows simulation results. Each row in the figure reports results for one method, while each column in the figure represents a different simulation setting. For the first two



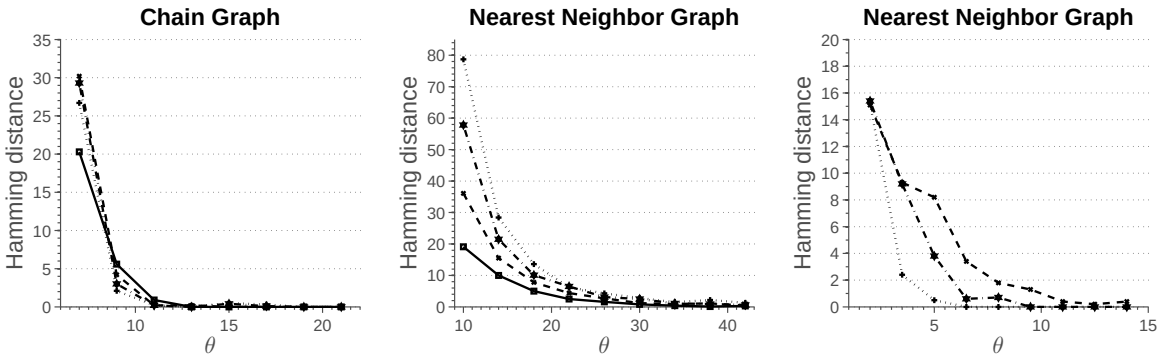
(a) glasso procedure



(b) Procedure of [45]

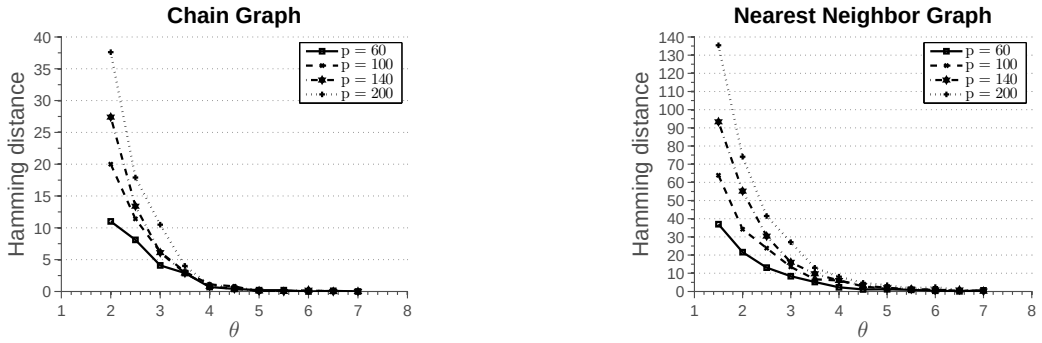


(c) Procedure of [77]

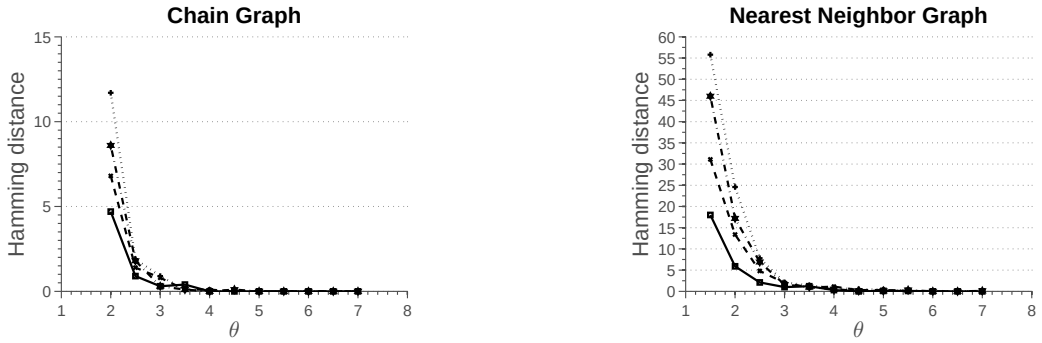


(d) Multi-attribute procedure

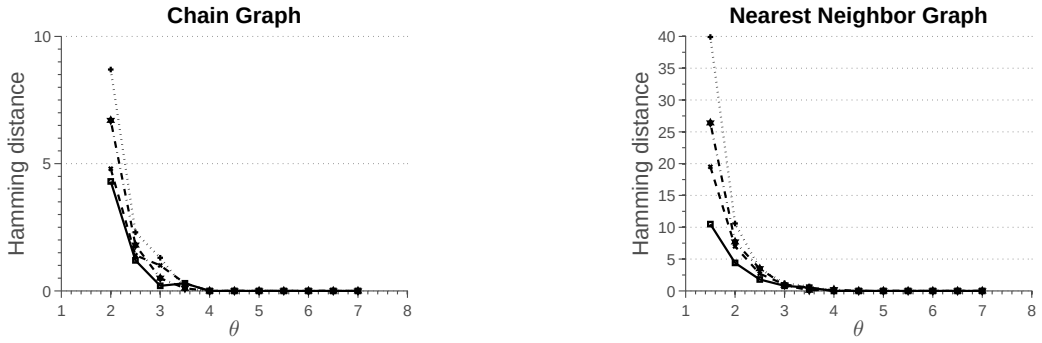
Figure 10.1: Average hamming distance plotted against the rescaled sample size. Off-diagonal blocks are full matrices.



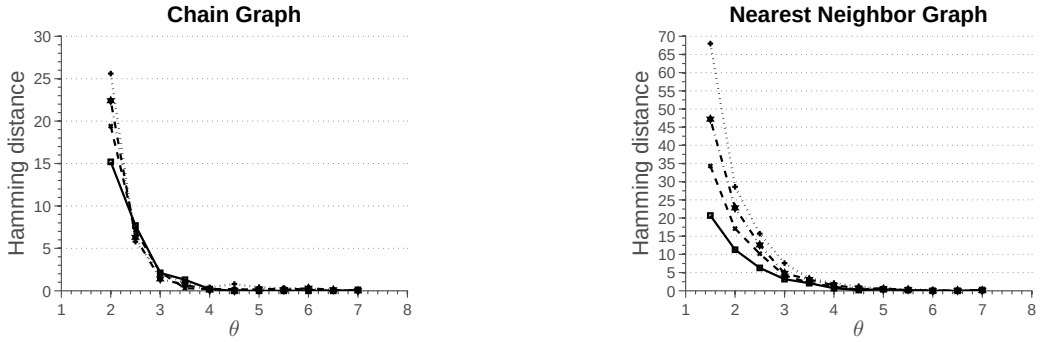
(a) glasso procedure



(b) Procedure of [45]

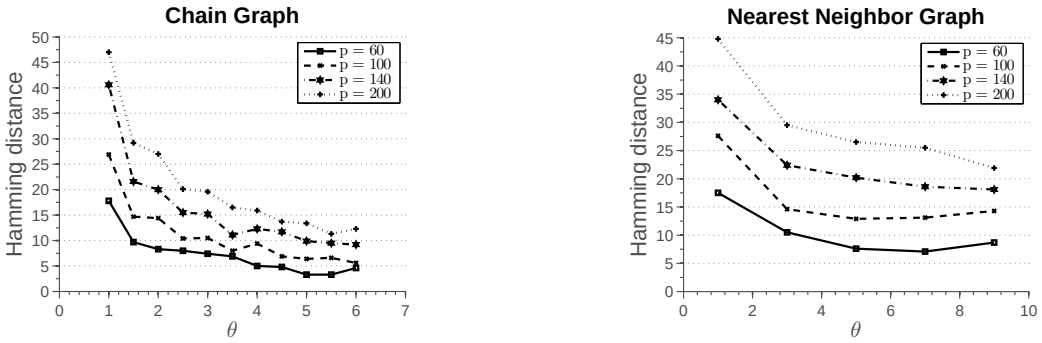


(c) Procedure of [77]

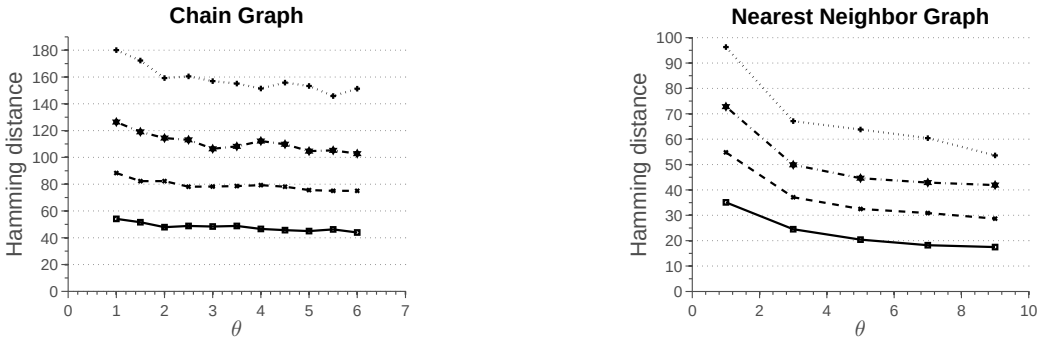


(d) Multi-attribute procedure

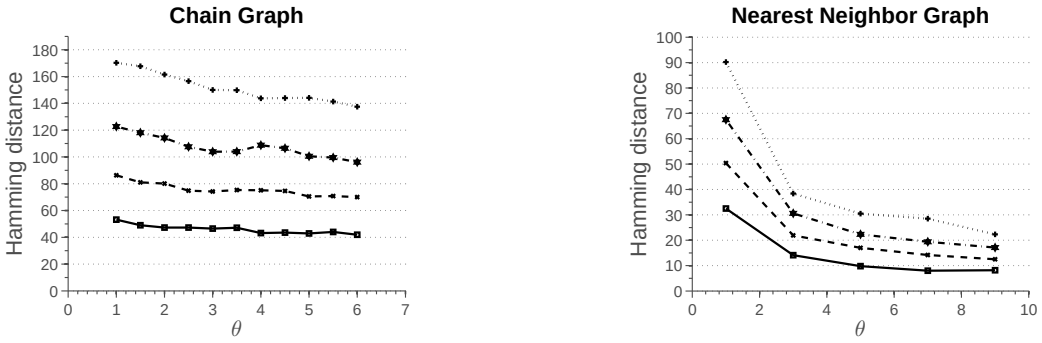
Figure 10.2: Average hamming distance plotted against the rescaled sample size. Blocks $\mathbf{\Omega}_{ab}$ of the precision matrix $\mathbf{\Omega}$ are diagonal matrices.



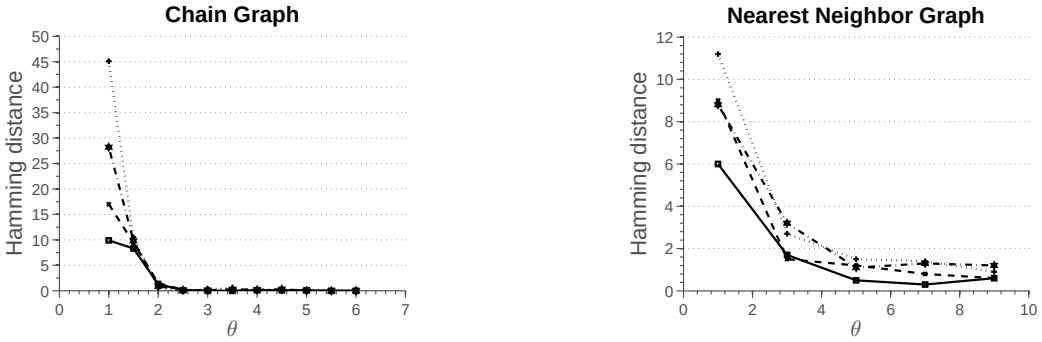
(a) glasso procedure



(b) Procedure of [45]

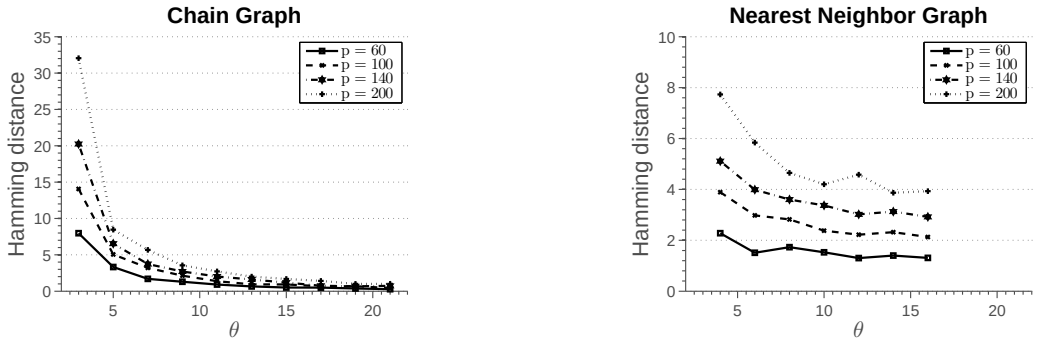


(c) Procedure of [77]

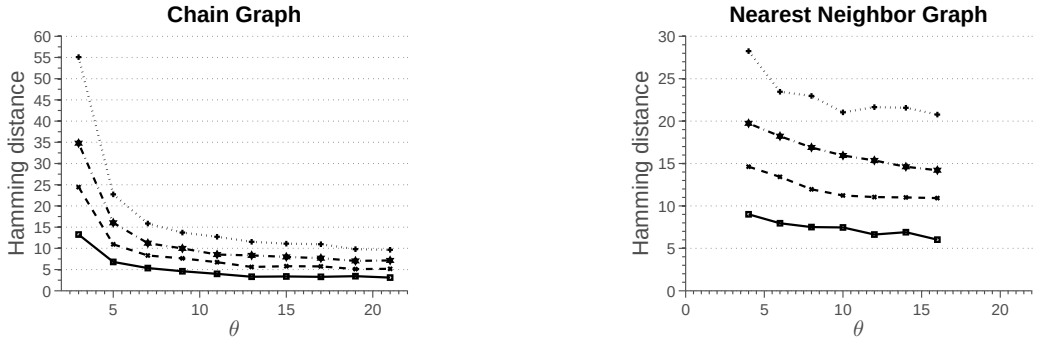


(d) Multi-attribute procedure

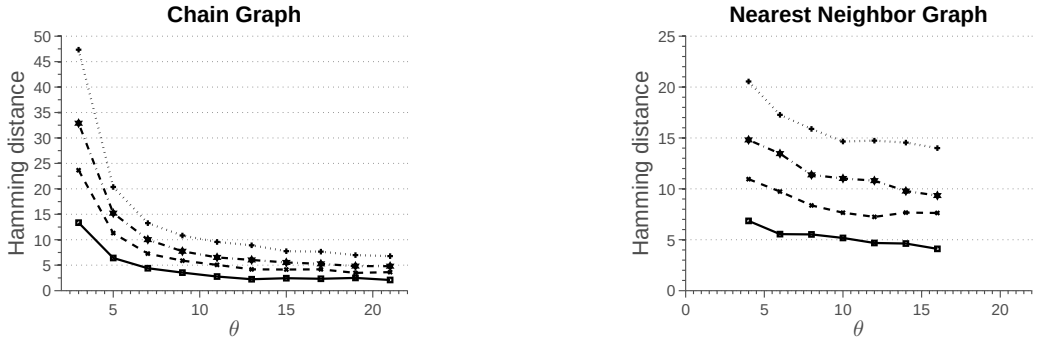
Figure 10.3: Average hamming distance plotted against the rescaled sample size. Off-diagonal blocks $\mathbf{\Omega}_{ab}$ of the precision matrix $\mathbf{\Omega}$ have zeros as diagonal elements.



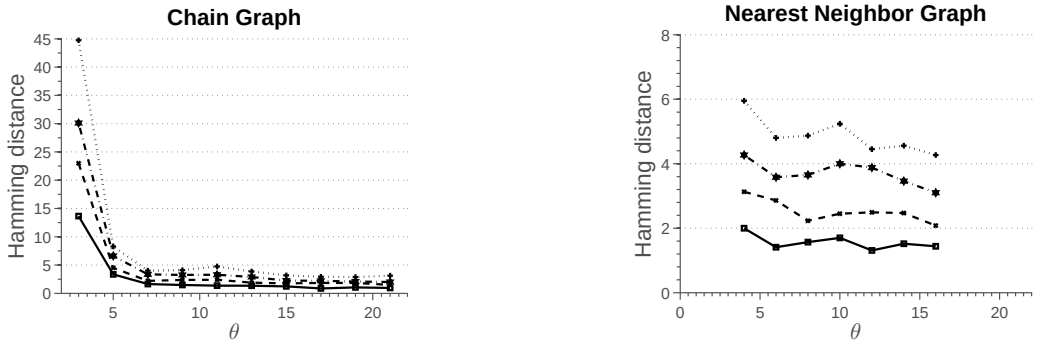
(a) glasso procedure



(b) Procedure of [45]



(c) Procedure of [77]



(d) Multi-attribute procedure

Figure 10.4: Average hamming distance plotted against the rescaled sample size. Off-diagonal blocks $\mathbf{\Omega}_{ab}$ of the precision matrix $\mathbf{\Omega}$ have elements uniformly sampled from $[-0.3, -0.1] \bigcup [0.1, 0.3]$.

columns, we set $k = 3$ and vary the total number of nodes in the graph. The third simulation setting sets the total number of nodes $p = 20$ and changes the number of attributes k. In the case of the chain graph, we observe that for small sample sizes the method of [45] outperforms all the other methods. We note that the multi-attribute method is estimating many more parameters, which require large sample size in order to achieve high accuracy. However, as the sample size increases, we observe that multi-attribute method starts to outperform the other methods. In particular, for the sample size indexed by $\theta = 13$ all the graph are correctly recovered, while other methods fail to recover the graph consistently at the same sample size. In the case of nearest-neighbor graph, none of the methods recover the graph well for small sample sizes. However, for moderate sample sizes, multi-attribute method outperforms the other methods. Furthermore, as the sample size increases none of the other methods recover the graph exactly. This suggests that the conditions for consistent graph recovery may be weaker in the multi-attribute setting.

10.5.1 Alternative Structure of Off-diagonal Blocks

In this section, we investigate performance of different estimation procedures under different assumptions on the elements of the off-diagonal blocks of the precision matrix.

First, we investigate a situation where the multi-attribute method does not perform as well as the methods that estimate multiple graphical models. One such situation arises when different attributes are conditionally independent. To simulate this situation, we use the data generating approach as before, however, we make each block $\mathbf{\Omega}_{ab}$ of the precision matrix Ω a diagonal matrix. Figure 10.2 summarizes results of the simulation. We see that the methods of [45] and [77] perform better, since they are estimating much fewer parameters than the multi-attribute method. glasso does not utilize any structural information underlying the estimation problem and requires larger sample size to correctly estimate the graph than other methods.

A completely different situation arises when the edges between nodes can be inferred only based on inter-attribute data, that is, when a graph based on any individual attribute is empty. To generate data under this situation, we follow the procedure as before, but with the diagonal elements of the off-diagonal blocks $\mathbf{\Omega}_{ab}$ set to zero. Figure 10.3 summarizes results of the simulation. In this setting, we clearly see the advantage of the multi-attribute method, compared to other three methods. Furthermore, we can see that glasso does better than multi-graph methods of [45] and [77]. The reason is that glasso can identify edges based on inter-attribute relationships among nodes, while multi-graph methods rely only on intra-attribute relationships. This simulation illustrates an extreme scenario where inter-attribute relationships are important for identifying edges.

So far, off-diagonal blocks of the precision matrix were constructed to have constant values. Now, we use the same data generating procedure, but generate off-diagonal blocks of a precision matrix in a different way. Each element of the off-diagonal block Ω_{ab} is generated independently and uniformly from the set $[-0.3, -0.1] \bigcup [0.1, 0.3]$. The results of the simulation are given in Figure 10.4. Again, qualitatively, the results are similar to those given in Figure 10.1, except that in this setting more samples are needed to recover the graph correctly.

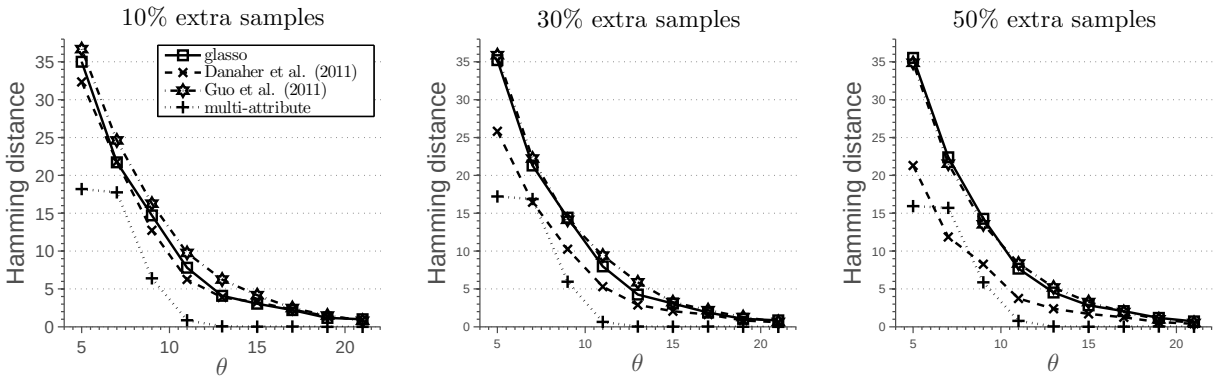


Figure 10.5: Average hamming distance plotted against the rescaled sample size. Results are averaged over 100 independent runs. Additional samples available for the first attribute.

10.5.2 Different Number of Samples per Attribute

In this section, we show how to deal with a case when different number of samples is available per attribute. As noted in §10.2.2, we can treat non-measured attributes as missing completely at random (see [107] for more details).

Let $R = (r_{il})_{i\in\{1,\ldots,n\},l\in\{1,\ldots,pk\}} \in \mathbb{R}^{n\times pk}$ be an indicator matrix, which denotes for each sample point $\mathbf{x}_i$ the components that are observed. Then we can form an estimate of the sample covariance matrix $S = (\sigma_{lk}) \in \mathbb{R}^{pk\times pk}$ as

$$\sigma_{lk} = \frac{\sum_{i=1}^{n} r_{i,l} r_{i,k} x_{i,l} x_{i,k}}{\sum_{i=1}^{n} r_{i,l} r_{i,k}}.$$

This estimate is plugged into the objective in (10.3).

We generate a chain graph with $p = 60$ nodes, construct a precision matrix associated with the graph and $k = 3$ attributes, and generate $n = \theta s^2 k^2 \log(pk)$ samples, $\theta > 0$. Next, we generate additional 10%, 30% and 50% samples from the same model, but record only the values for the first attribute. Results of the simulation are given in Figure 10.5. Qualitatively, the results are similar to those presented in Figure 10.1.

10.6 Illustrative Applications to Real Data

In this section, we illustrate how to apply our method to data arising in studies of biological regulatory networks and Alzheimer's disease.

10.6.1 Analysis of a Gene/Protein Regulatory Network

We provide illustrative, exploratory analysis of data from the well-known NCI-60 database, which contains different molecular profiles on a panel of 60 diverse human cancer cell lines. Data set consists of protein profiles (normalized reverse-phase lysate arrays for 92 antibodies) and gene profiles (normalized RNA microarray intensities from Human Genome U95 Affymetrix chip-set for > 9000 genes). We focus our analysis on a subset of 91 genes/proteins for which

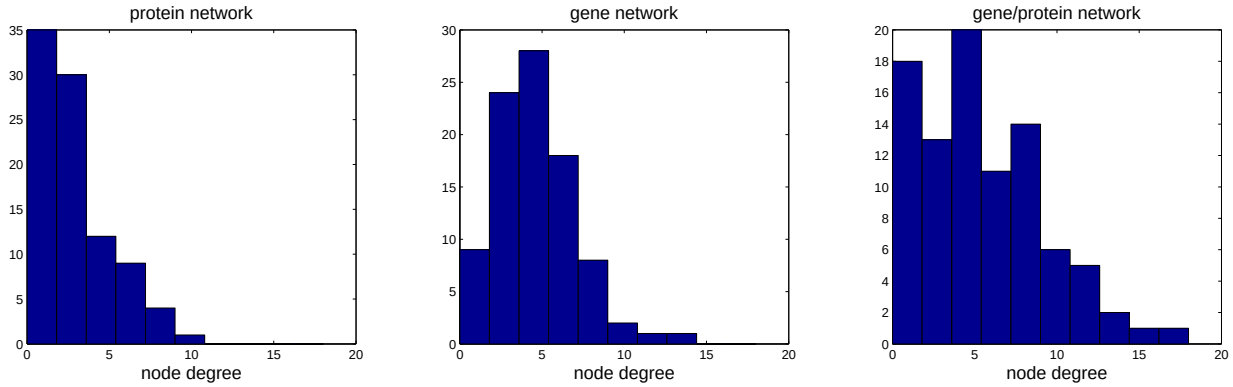


Figure 10.6: Node degree distributions for protein, gene and gene/protein networks.

both types of profiles are available. These profiles are available across the same set of 60 cancer cells. More detailed description of the data set can be found in [114].

We inferred three types of networks: a network based on protein measurements alone, a network based on gene expression profiles and a single gene/protein network. For protein and gene networks we use the `glasso`, while for the gene/protein network, we use our procedure outlined in §10.2.2. We use the stability selection [136] procedure to estimate stable networks. In particular, we first select the penalty parameter λ using cross-validation, which over-selects the number of edges in a network. Next, we use the selected λ to estimate 100 networks based on random subsamples containing 80% of the data-points. Final network is composed of stable edges that appear in at least 95 of the estimated networks. Table 10.1 provides a few summary statistics for the estimated networks. Furthermore, protein and gene/protein networks share 96 edges, while gene and gene/protein networks share 104 edges. Gene and protein network share only 17 edges. Finally, 66 edges are unique to gene/protein network. Figure 10.6 shows node degree distributions for the three networks. We observe that the estimated networks are much sparser than the association networks in [114], as expected due to marginal correlations between a number of nodes. The differences in networks require a closer biological inspection by a domain scientist.

We proceed with a further exploratory analysis of the gene/protein network. We investigate the contribution of two nodal attributes to the existence of an edges between the nodes. Following [114], we use a simple heuristic based on the weight vectors to classify the nodes and edges into three classes. For an edge between the nodes a and b, we take one weight vector, say $\mathbf{w}_a$, and normalize it to have unit norm. Denote w_p the component corresponding to the protein attribute.

Table 10.1: Summary statistics for protein, gene, and gene/protein networks ($p = 91$).

	protein network	gene network	gene/protein network
Number of edges	122	214	249
Density	0.03	0.05	0.06
Largest connected component	62	89	82
Avg Node Degree	2.68	4.70	5.47
Avg Clustering Coefficient	0.0008	0.001	0.003

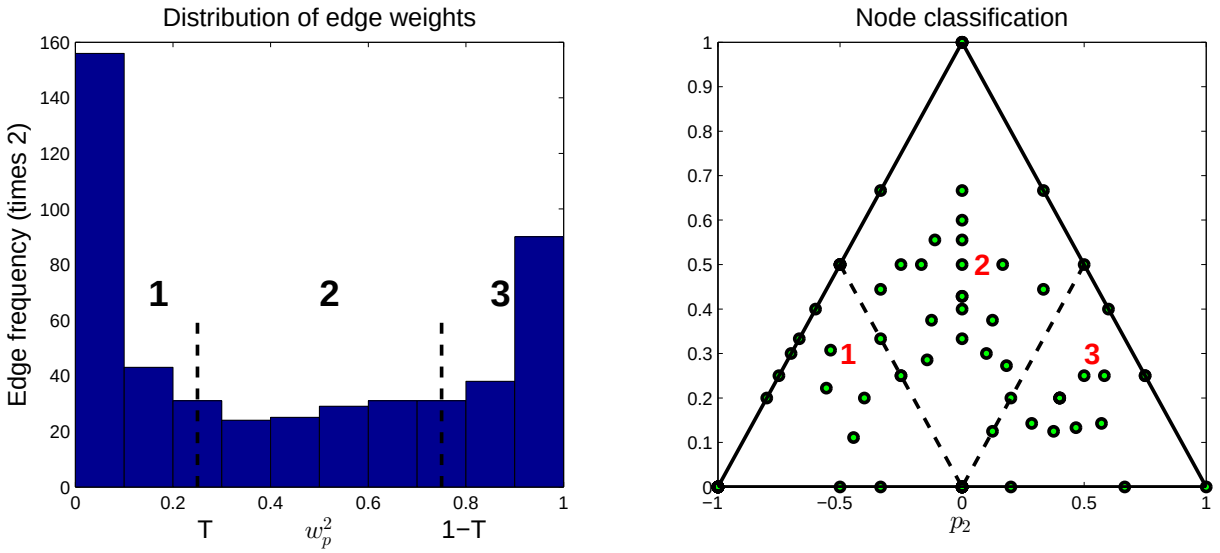


Figure 10.7: Edge and node classification based on w_p^2.

Left plot in Figure 10.7 shows the values of w_p^2 over all edges. The edges can be classified into three classes based on the value of w_p^2. Given a threshold T, the edges for which $w_p^2 \in (0, T)$ are classified as gene-influenced, the edges for which $w_p^2 \in (1-T, 1)$ are classified as protein influenced, while the remainder of the edges are classified as mixed type. In the left plot of Figure 10.7, the threshold is set as $T = 0.25$. Similar classification can be performed for nodes after computing the proportion of incident edges. Let p_1, p_2 and p_3 denote proportions of gene, protein and mixed edges, respectively, incident with a node. These proportions are represented in a simplex in the right subplot of Figure 10.7. Nodes with mostly gene edges are located in the lower left corner, while the nodes with mostly protein edges are located in the lower right corner. Mixed nodes are located in the center and towards the top corner of the simplex. Further biological enrichment analysis is possible (see [114]), however, we do not pursue this here.

10.6.2 Uncovering Functional Brain Network

We apply our method to the Positron Emission Tomography dataset, which contains 259 subjects, of whom 72 are healthy, 132 have mild cognitive Impairment and 55 are diagnosed as Alzheimer's & Dementia. Note that mild cognitive impairment is a transition stage from normal aging to Alzheimer's & Dementia. The data can be obtained from *http://adni.loni.ucla.edu/*. The preprocessing is done in the same way as in [91].

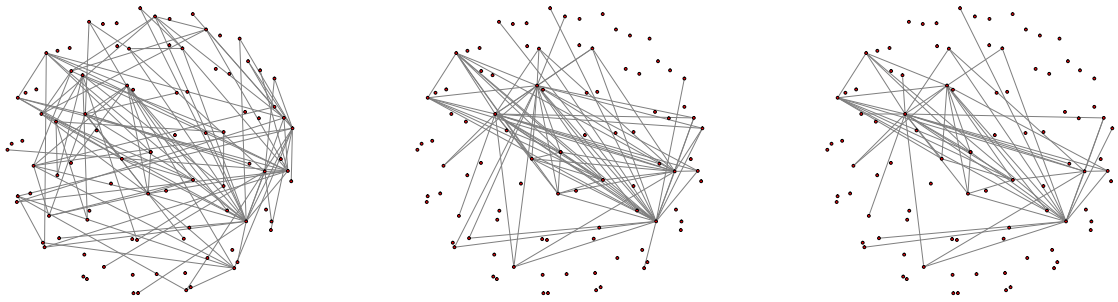
Each Positron Emission Tomography image contains $91 \times 109 \times 91 = 902,629$ voxels. The effective brain region contains $180,502$ voxels, which are partitioned into 95 regions, ignoring the regions with fewer than 500 voxels. The largest region contains $5,014$ voxels and the smallest region contains 665 voxels. Our preprocessing stage extracts 948 representative voxels from these regions using the K-median clustering algorithm. The parameter K is chosen differently for each region, proportionally to the initial number of voxels in that region. More specifically, for each category of subjects we have an $n \times (d_1 + \ldots + d_{95})$ matrix, where n is the number of subjects and $d_1 + \ldots + d_{95} = 902,629$ is the number of voxels. Next we set $K_i = \lceil d_i / \sum_j d_j \rceil$,

the number of representative voxels in region i, $i = 1, \ldots, 95$. The representative voxels are identified by running the K-median clustering algorithm on a sub-matrix of size $n \times d_i$ with $K = K_i$.

We inferred three networks, one for each subtype of subjects using the procedure outlined in §10.2.2. Note that for different nodes we have different number of attributes, which correspond to medians found by the clustering algorithm. We use the stability selection [136] approach to estimate stable networks. The stability selection procedure is combined with our estimation procedure as follows. We first select the penalty parameter λ in (10.3) using cross-validation, which overselects the number of edges in a network. Next, we create 100 subsampled data sets, each of which contain 80% of the data points, and estimate one network for each dataset using the selected λ. The final network is composed of stable edges that appear in at least 95 of the estimated networks.

We visualize the estimated networks in Figure 10.8. Table 10.2 provides a few summary statistics for the estimated networks. [108] contains names of different regions, as well as the adjacency matrices for networks. From the summary statistics, we can observe that in normal subjects there are many more connections between different regions of the brain. Loss of connectivity in Alzheimer's & Dementia has been widely reported in the literature [8, 78, 93, 186].

Learning functional brain connectivity is potentially valuable for early identification of signs of Alzheimer's disease. [91] approach this problem using exploratory data analysis. The framework of Gaussian graphical models is used to explore functional brain connectivity. Here we point out that our approach can be used for the same exploratory task, without the need to reduce the information in the whole brain to one number. For example, from our estimates, we observe the loss of connectivity in the cerebellum region of patients with Alzheimer's disease, which has been reported previously in [162]. As another example, we note increased connectivity between the frontal lobe and other regions in the patients, which was linked to compensation for the lost connections in other regions [82, 197].



(a) Healthy subjects (b) Mild Cognitive Impairment (c) Alzheimer's & Dementia

Figure 10.8: Brain connectivity networks

Table 10.2: Summary statistics for protein, gene, and gene/protein networks ($p = 91$)

	Healthy subjects	Mild Cognitive Impairment	Alzheimer's & Dementia
Number of edges	116	84	59
Density	0.030	0.020	0.014
Largest connected component	48	27	25
Avg Node Degree	2.40	1.73	1.2
Avg Clustering Coefficient	0.001	0.0023	0.0007

10.7 Discussion

In this chapter, we have proposed a solution to the problem of learning networks from multivariate nodal attributes, which arises in a variety of domains. Our method is based on simultaneously estimating non-zero partial canonical correlations between nodes in a network. When all the attributes across all the nodes follow joint multivariate Normal distribution, our procedure is equivalent to estimating conditional independencies between nodes, which is revealed by relating the blocks of the precision matrix to partial canonical correlation. Although a penalized likelihood framework is adopted for estimation of the non-zero blocks of the precision matrix, other approaches like neighborhood pursuit or greedy pursuit can also be developed. Thorough numerical evaluations and theoretical analysis of these methods is an interesting direction for future work.

10.8 Technical Proofs

10.8.1 Proof of Lemma 10.1

We start the proof by giving to technical results needed later. The following lemma states that the minimizer of (10.3) is unique and has bounded minimum and maximum eigenvalues, denoted as $\Lambda_{\min}$ and $\Lambda_{\max}$.

Lemma 10.4. *For every value of $\lambda > 0$, the optimization problem in* (10.3) *has a unique minimizer $\widehat{\mathbf{\Omega}}$, which satisfies $\Lambda_{\min}(\widehat{\mathbf{\Omega}}) \geq (\Lambda_{\max}(\mathbf{S}) + \lambda p)^{-1} > 0$ and $\Lambda_{\max}(\widehat{\mathbf{\Omega}}) \leq \lambda^{-1} \sum_{j \in V} k_j$.*

Proof. The optimization objective given in (10.3) can be written in the equivalent constrained form as

$$\min_{\mathbf{\Omega} \succ \mathbf{0}} \operatorname{tr} \mathbf{S}\mathbf{\Omega} - \log |\mathbf{\Omega}| \quad \text{subject to} \quad \sum_{a,b} ||\mathbf{\Omega}_{ab}||_F \leq C(\lambda).$$

The procedure involves minimizing a continuous objective over a compact set, and so by Weierstrass theorem, the minimum is always achieved. Furthermore, the objective is strongly convex and therefore the minimum is unique.

The solution $\widehat{\mathbf{\Omega}}$ to the optimization problem (10.3) satisfies

$$\mathbf{S} - \widehat{\mathbf{\Omega}}^{-1} + \lambda \mathbf{Z} = \mathbf{0} \tag{10.11}$$

where $\mathbf{Z} \in \partial \sum_{a,b} ||\widehat{\mathbf{\Omega}}_{ab}||_F$ is the element of the sub-differential and satisfies $||\mathbf{Z}_{ab}||_F \leq 1$ for all $(a,b) \in V^2$. Therefore,

$$\Lambda_{\max}(\widehat{\mathbf{\Omega}}^{-1}) \leq \Lambda_{\max}(\mathbf{S}) + \lambda \Lambda_{\max}(\mathbf{Z}) \leq \Lambda_{\max}(\mathbf{S}) + \lambda p.$$

Next, we prove an upper bound on $\Lambda_{\max}(\widehat{\mathbf{\Omega}})$. At optimum, the primal-dual gap is zero, which gives that

$$\sum_{a,b} ||\widehat{\Omega}_{ab}||_F \leq \lambda^{-1}(\sum_{j \in V} k_j - \operatorname{tr} \mathbf{S}\widehat{\mathbf{\Omega}}) \leq \lambda^{-1} \sum_{j \in V} k_j,$$

as $\mathbf{S} \succeq \mathbf{0}$ and $\widehat{\mathbf{\Omega}} \succ \mathbf{0}$. Since $\Lambda_{\max}(\widehat{\mathbf{\Omega}}) \leq \sum_{a,b} ||\widehat{\mathbf{\Omega}}_{ab}||_F$, the proof is done. □

The next results states that the objective function has a Lipschitz continuous gradient, which will be used to show that the generalized gradient descent can be used to find $\widehat{\mathbf{\Omega}}$.

Lemma 10.5. *The function $f(\mathbf{A}) = \operatorname{tr} \mathbf{S}\mathbf{A} - \log |\mathbf{A}|$ has a Lipschitz continuous gradient on the set $\{\mathbf{A} \in \mathcal{S}^p \ : \ \Lambda_{\min}(\mathbf{A}) \geq \gamma\}$, with the Lipschitz constant $L = \gamma^{-2}$.*

Proof. We have that $\nabla f(\mathbf{A}) = \mathbf{S} - \mathbf{A}^{-1}$. Then

$$\begin{aligned} ||\nabla f(\mathbf{A}) - \nabla f(\mathbf{A}')||_F &= ||\mathbf{A}^{-1} - (\mathbf{A}')^{-1}||_F \\ &\leq \Lambda_{\max}\mathbf{A}^{-1} ||\mathbf{A} - \mathbf{A}'||_F \Lambda_{\max}\mathbf{A}^{-1} \\ &\leq \gamma^{-2} ||\mathbf{A} - \mathbf{A}'||_F, \end{aligned}$$

which completes the proof. □

Now, we provide the proof of Lemma 10.1.

By construction, the sequence of estimates $(\widetilde{\mathbf{\Omega}}^{(t)})_{t \geq 1}$ decrease the objective value and are positive definite.

To prove the convergence, we first introduce some additional notation. Let $f(\mathbf{\Omega}) = \operatorname{tr} \mathbf{S}\mathbf{\Omega} - \log |\mathbf{\Omega}|$ and $F(\mathbf{\Omega}) = f(\mathbf{\Omega}) + \sum_{ab} ||\mathbf{\Omega}_{ab}||_F$. For any $L > 0$, let

$$Q_L(\mathbf{\Omega}; \overline{\mathbf{\Omega}}) := f(\overline{\mathbf{\Omega}}) + \operatorname{tr}[(\mathbf{\Omega} - \overline{\mathbf{\Omega}}) \nabla f(\overline{\mathbf{\Omega}})] + \frac{L}{2} ||\mathbf{\Omega} - \overline{\mathbf{\Omega}}||_F^2 + \sum_{ab} ||\mathbf{\Omega}_{ab}||_F$$

be a quadratic approximation of $F(\mathbf{\Omega})$ at a given point $\overline{\mathbf{\Omega}}$, which has a unique minimizer

$$p_L(\overline{\mathbf{\Omega}}) := \arg\min_{\mathbf{\Omega}} Q_L(\mathbf{\Omega}; \overline{\mathbf{\Omega}}).$$

From Lemma 2.3. in [11], we have that

$$F(\overline{\mathbf{\Omega}}) - F(p_L(\overline{\mathbf{\Omega}})) \geq \frac{L}{2} ||p_L(\overline{\mathbf{\Omega}}) - \overline{\mathbf{\Omega}}||_F^2 \tag{10.12}$$

if $F(p_L(\overline{\mathbf{\Omega}})) \leq Q_L(p_L(\overline{\mathbf{\Omega}}); \overline{\mathbf{\Omega}})$. Note that $F(p_L(\overline{\mathbf{\Omega}})) \leq Q_L(p_L(\overline{\mathbf{\Omega}}); \overline{\mathbf{\Omega}})$ always holds if L is as large as the Lipschitz constant of ∇F.

Let $\widetilde{\mathbf{\Omega}}^{(t-1)}$ and $\widetilde{\mathbf{\Omega}}^{(t)}$ denote two successive iterates obtained by the procedure. Without loss of generality, we can assume that $\widetilde{\mathbf{\Omega}}^{(t)}$ is obtained by updating the rows/columns corresponding to the node a. From (10.12), it follows that

$$\frac{2}{L_k}(F(\widetilde{\mathbf{\Omega}}^{(t-1)}) - F(\widetilde{\mathbf{\Omega}}^{(t)})) \geq ||\widetilde{\mathbf{\Omega}}_{aa}^{(t-1)} - \widetilde{\mathbf{\Omega}}_{aa}^{(t)}||_F + 2\sum_{b\neq a} ||\widetilde{\mathbf{\Omega}}_{ab}^{(t-1)} - \widetilde{\mathbf{\Omega}}_{ab}^{(t)}||_F \tag{10.13}$$

where L_k is a current estimate of the Lipschitz constant. Recall that in our procedure the scalar t serves as a local approximation of $1/L$. Since eigenvalues of $\widehat{\mathbf{\Omega}}$ are bounded according to Lemma 10.4, we can conclude that the eigenvalues of $\widetilde{\mathbf{\Omega}}^{(t-1)}$ are bounded as well. Therefore the current Lipschitz constant is bounded away from zero, using Lemma 10.5. Combining the results, we observe that the right hand side of (10.13) converges to zero as $t \to \infty$, since the optimization procedure produces iterates that decrease the objective value. This shows that $||\widetilde{\mathbf{\Omega}}_{aa}^{(t-1)} - \widetilde{\mathbf{\Omega}}_{aa}^{(t)}||_F + 2\sum_{b\neq a} ||\widetilde{\mathbf{\Omega}}_{ab}^{(t-1)} - \widetilde{\mathbf{\Omega}}_{ab}^{(t)}||_F$ converges to zero, for any $a \in V$. Since ($\widetilde{\mathbf{\Omega}}^{(t)}$ is a bounded sequence, it has a limit point, which we denote $\widehat{\mathbf{\Omega}}$. It is easy to see, from the stationary conditions for the optimization problem given in (10.4), that the limit point $\widehat{\mathbf{\Omega}}$ also satisfies the global KKT conditions to the optimization problem in (10.3).

10.8.2 Proof of Lemma 10.3

Suppose that the solution $\widehat{\mathbf{\Omega}}$ to (10.3) is block diagonal with blocks $P_1, P_2, \ldots, P_l$. For two nodes a, b in different blocks, we have that $(\widehat{\mathbf{\Omega}})_{ab}^{-1} = 0$ as the inverse of the block diagonal matrix is block diagonal. From the KKT conditions, it follows that $||\mathbf{S}_{ab}||_F \leq \lambda$.

Now suppose that $||\mathbf{S}_{ab}||_F \leq \lambda$ for all $a \in P_j, b \in P_{j'}, j \neq j'$. For every $l' = 1, \ldots, l$ construct

$$\widetilde{\mathbf{\Omega}}_{l'} = \arg\min_{\mathbf{\Omega}_{l'} \succ \mathbf{0}} \ \mathrm{tr}\, \mathbf{S}_{l'}\mathbf{\Omega}_{l'} - \log|\mathbf{\Omega}_{l'}| + \lambda \sum_{a,b} ||\mathbf{\Omega}_{ab}||_F.$$

Then $\widehat{\mathbf{\Omega}} = \mathrm{diag}(\widehat{\mathbf{\Omega}}_1, \widehat{\mathbf{\Omega}}_2, \ldots, \widehat{\mathbf{\Omega}}_l)$ is the solution of (10.3) as it satisfies the KKT conditions.

10.8.3 Proof of Equation 10.2

First, we note that

$$\mathrm{Var}\left((\mathbf{X}_a', \mathbf{X}_b')' \mid \mathbf{X}_{\overline{ab}}\right) = \mathbf{\Sigma}_{ab,ab} - \mathbf{\Sigma}_{ab,\overline{ab}}\mathbf{\Sigma}_{\overline{ab},\overline{ab}}^{-1}\mathbf{\Sigma}_{\overline{ab},ab}$$

is the conditional covariance matrix of $(\mathbf{X}_a', \mathbf{X}_b')'$ given the remaining nodes $\mathbf{X}_{\overline{ab}}$ (see Proposition C.5 in [130]). Define $\overline{\mathbf{\Sigma}} = \mathbf{\Sigma}_{ab,ab} - \mathbf{\Sigma}_{ab,\overline{ab}}\mathbf{\Sigma}_{\overline{ab},\overline{ab}}^{-1}\mathbf{\Sigma}_{\overline{ab},ab}$. Partial canonical correlation between $\mathbf{X}_a$ and $\mathbf{X}_b$ is equal to zero if and only if $\overline{\mathbf{\Sigma}}_{ab} = \mathbf{0}$. On the other hand, the matrix inversion lemma gives that $\mathbf{\Omega}_{ab,ab} = \overline{\mathbf{\Sigma}}^{-1}$. Now, $\mathbf{\Omega}_{ab} = \mathbf{0}$ if and only if $\overline{\mathbf{\Sigma}}_{ab} = 0$. This shows the equivalence relationship in (10.2).

10.8.4 Proof of Proposition 10.1

We provide sufficient conditions for consistent network estimation. Proposition 10.1 given in §10.3 is then a simple consequence. To provide sufficient conditions, we extend the work of [152] to our setting, where we observe multiple attributes for each node. In particular, we extend their Theorem 1.

For simplicity of presentation, we assume that $k_a = k$, for all $a \in V$, that is, we assume that the same number of attributes is observed for each node. Our assumptions involve the Hessian of the function $f(\mathbf{A}) = \operatorname{tr} \mathbf{SA} - \log |\mathbf{A}|$ evaluated at the true $\mathbf{\Omega}^*$,

$$\mathcal{H} = \mathcal{H}(\mathbf{\Omega}^*) = (\mathbf{\Omega}^*)^{-1} \otimes (\mathbf{\Omega}^*)^{-1} \in \mathbb{R}^{(pk)^2 \times (pk)^2},$$

and the true covariance matrix $\mathbf{\Sigma}^*$. The Hessian and the covariance matrix can be thought of block matrices with blocks of size $k^2 \times k^2$ and $k \times k$, respectively. We will make use of the operator $\mathcal{C}(\cdot)$ that operates on these block matrices and outputs a smaller matrix with elements that equal to the Frobenius norm of the original blocks,

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1p} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2p} \\ \vdots & & \ddots & \vdots \\ \mathbf{A}_{p1} & \cdots & & \mathbf{A}_{pp} \end{pmatrix} \xrightarrow{\mathcal{C}(\cdot)} \begin{pmatrix} ||\mathbf{A}_{11}||_F & ||\mathbf{A}_{12}||_F & \cdots & ||\mathbf{A}_{1p}||_F \\ ||\mathbf{A}_{21}||_F & ||\mathbf{A}_{22}||_F & \cdots & ||\mathbf{A}_{2p}||_F \\ \vdots & & \ddots & \vdots \\ ||\mathbf{A}_{p1}||_F & \cdots & & ||\mathbf{A}_{pp}||_F \end{pmatrix}$$

In particular, $\mathcal{C}(\mathbf{\Sigma}^*) \in \mathbb{R}^{p \times p}$ and $\mathcal{C}(\mathcal{H}) \in \mathbb{R}^{p^2 \times p^2}$.

We denote the index set of the non-zero blocks of the precision matrix as

$$\mathcal{T} := \{(a,b) \in V \times V \ : \ ||\mathbf{\Omega}^*_{ab}||_2 \neq 0\} \cup \{(a,a) \ : \ a \in V\}$$

and let $\mathcal{N}$ denote its complement in $V \times V$, that is,

$$\mathcal{N} = \{(a,b) \ : \ ||\mathbf{\Omega}_{ab}||_F = 0\}.$$

As mentioned earlier, we need to make an assumption on the Hessian matrix, which takes the standard irrepresentable-like form. There exists a constant $\alpha \in [0,1)$ such that

$$\|\mathcal{C}\left(\mathcal{H}_{\mathcal{N}\mathcal{T}}(\mathcal{H}_{\mathcal{T}\mathcal{T}})^{-1}\right)\|_\infty \leq 1 - \alpha. \tag{10.14}$$

These condition extends the irrepresentable condition given in [152], which was needed for estimation of networks from single attribute observations. It is worth noting, that the condition given in (10.14) can be much weaker than the irrepresentable condition of [152] applied directly to the full Hessian matrix. This can be observed in simulations done in §10.5, where a chain network is not consistently estimated even with a large number of samples.

We will also need the following two quantities to specify the results

$$\kappa_{\mathbf{\Sigma}^*} = \|\mathcal{C}(\mathbf{\Sigma}^*)\|_\infty$$

and

$$\kappa_{\mathcal{H}} = \|\mathcal{C}(\mathcal{H}^{-1}_{\mathcal{T}\mathcal{T}})\|_\infty.$$

Finally, the results are going to depend on the tail bounds for the elements of the matrix $\mathcal{C}(\mathbf{S}-\mathbf{\Sigma}^*)$. We will assume that there is a constant $v_* \in (0,\infty]$ and a function $f : \mathbb{N} \times (0,\infty) \mapsto (0,\infty)$ such that for any $(a,b) \in V \times V$

$$\mathbb{P}\left(\mathcal{C}(\mathbf{S}-\mathbf{\Sigma}^*)_{ab} \geq \delta\right) \leq \frac{1}{f(n,\delta)} \qquad \delta \in (0, v_*^{-1}]. \tag{10.15}$$

The function $f(n,\delta)$ will be monotonically increasing in both n and δ. Therefore, we define the following two inverse functions

$$\overline{n}_f(\delta; r) = \arg\max\{n \ : \ f(n,\delta) \leq r\}$$

and

$$\overline{\delta}_f(r; n) = \arg\max\{\delta \ : \ f(n,\delta) \leq r\}$$

for $r \in [1,\infty)$.

With the notation introduced, we have the following result.

Theorem 10.1. *Assume that the irrepresentable condition in* (10.14) *is satisfied and that there exists a constant* $v_* \in (0,\infty]$ *and a function* $f(n,\delta)$ *so that* (10.15) *is satisfied for any* $(a,b) \in V \times V$. *Let*

$$\lambda = \frac{8}{\alpha}\overline{\delta}_f(n, p^\tau)$$

for some $\tau > 2$. *If*

$$n > \overline{n}_f\left(\frac{1}{\max(v_*, 6(1+8\alpha^{-1})s\max(\kappa_{\mathbf{\Sigma}^*}\kappa_{\mathcal{H}}, \kappa^3_{\mathbf{\Sigma}^*}\kappa^2_{\mathcal{H}}))}, p^\tau\right)$$

then

$$||\mathcal{C}(\widehat{\mathbf{\Omega}} - \mathbf{\Omega})||_\infty \leq 2(1+8\alpha^{-1})\kappa_{\mathcal{H}}\overline{\delta}_f(n, p^\tau)$$

with probability at least $1 - p^{2-\tau}$.

Theorem 10.1 is of the same form as Theorem 1 in [152], but the ℓ_∞ element-wise convergence is established for $\mathcal{C}(\widehat{\mathbf{\Omega}} - \mathbf{\Omega})$, which will guarantee successful recovery of non-zero partial canonical correlations if the blocks of the true precision matrix are sufficiently large.

Theorem 10.1 is proven as Theorem 1 in [152]. We provide technical results in Lemma 10.6, Lemma 10.7 and Lemma 10.8, which can be used to substitute results of Lemma 4, Lemma 5 and Lemma 6 in [152] under our setting. The rest of the arguments then go through. Below we provide some more details.

First, let $\mathcal{Z} : \mathbb{R}^{pk\times pk} \mapsto \mathbb{R}^{pk\times pk}$ be the mapping defined as

$$\mathcal{Z}(\mathbf{A})_{ab} = \begin{cases} \frac{\mathbf{A}_{ab}}{||\mathbf{A}_{ab}||_F} & \text{if } ||\mathbf{A}_{ab}||_F \neq 0, \\ \mathbf{Z} \text{ with } ||\mathbf{Z}||_F \leq 1 & \text{if } ||\mathbf{A}_{ab}||_F = 0, \end{cases}$$

Next, define the function

$$G(\mathbf{\Omega}) = \operatorname{tr}\mathbf{\Omega}\mathbf{S} - \log|\mathbf{\Omega}| + \lambda||\mathcal{C}(\mathbf{\Omega})||_1, \quad \forall \mathbf{\Omega} \succ 0$$

and the following system of equations

$$\begin{cases} \mathbf{S}_{ab} - (\mathbf{\Omega}^{-1})_{ab} &= -\lambda \mathcal{Z}(\mathbf{\Omega})_{ab}, & \text{if } \mathbf{\Omega}_{ab} \neq 0 \\ ||\mathbf{S}_{ab} - (\mathbf{\Omega}^{-1})_{ab}||_F &\leq \lambda, & \text{if } \mathbf{\Omega}_{ab} = 0. \end{cases} \tag{10.16}$$

It is known that $\mathbf{\Omega} \in \mathbb{R}^{\widetilde{p} \times \widetilde{p}}$ is the minimizer of optimization problem in (10.3) if and only if it satisfies the system of equations given in (10.16). We have already shown in Lemma 10.4 that the minimizer is unique.

Let $\widetilde{\mathbf{\Omega}}$ be the solution to the following constrained optimization problem

$$\min_{\mathbf{\Omega} \succ \mathbf{0}} \ \text{tr}\, \mathbf{S}\mathbf{\Omega} - \log |\mathbf{\Omega}| + \lambda ||\mathcal{C}(\mathbf{\Omega})||_1 \text{ subject to } \mathcal{C}(\mathbf{\Omega})_{ab} = 0, \ \forall (a,b) \in \mathcal{N}.$$

Observe that one cannot find $\widetilde{\mathbf{\Omega}}$ in practice, as it depends on the unknown set $\mathcal{N}$. However, it is a useful construction in the proof. We will prove that $\widetilde{\mathbf{\Omega}}$ is solution to the optimization problem given in (10.3), that is, we will show that $\widetilde{\mathbf{\Omega}}$ satisfies the system of equations (10.16).

Using the first-order Taylor expansion we have that

$$\widetilde{\mathbf{\Omega}}^{-1} = (\mathbf{\Omega}^*)^{-1} - (\mathbf{\Omega}^*)^{-1} \mathbf{\Delta} (\mathbf{\Omega}^*)^{-1} + R(\mathbf{\Delta}), \tag{10.17}$$

where $\mathbf{\Delta} = \mathbf{\Omega} - \mathbf{\Omega}^*$ and $R(\mathbf{\Delta})$ denotes the remainder term. With this, we state and prove Lemma 10.6, Lemma 10.7 and Lemma 10.8. They can be combined as in [152] to complete the proof of Theorem 10.1.

Lemma 10.6. *Assume that*

$$\max_{ab} ||\mathbf{\Delta}_{ab}||_F \leq \frac{\alpha \lambda}{8} \quad \textit{and} \quad \max_{ab} ||\mathbf{\Sigma}^*_{ab} - \mathbf{S}_{ab}||_F \leq \frac{\alpha \lambda}{8}. \tag{10.18}$$

Then $\widetilde{\mathbf{\Omega}}$ is the solution to the optimization problem in (10.3)*.*

Proof. We use $\mathbf{R}$ to denote $\mathbf{R}(\Delta)$. Recall that $\Delta_{\mathcal{N}} = 0$ by construction. Using (10.17) we can rewrite (10.16) as

$$\mathcal{H}_{ab,\mathcal{T}} \overline{\mathbf{\Delta}}_{\mathcal{T}} - \overline{\mathbf{R}}_{ab} + \overline{\mathbf{S}}_{ab} - \overline{\mathbf{\Sigma}}^*_{ab} + \lambda \overline{\mathcal{Z}}(\widetilde{\mathbf{\Omega}})_{ab} = 0 \qquad \text{if } (a,b) \in \mathcal{T} \tag{10.19}$$

$$||\mathcal{H}_{ab,\mathcal{T}} \overline{\mathbf{\Delta}}_{\mathcal{T}} - \overline{\mathbf{R}}_{ab} + \overline{\mathbf{S}}_{ab} - \overline{\mathbf{\Sigma}}^*_{ab}||_2 \leq \lambda \qquad \text{if } (a,b) \in \mathcal{N}. \tag{10.20}$$

By construction, the solution $\widetilde{\mathbf{\Omega}}$ satisfy (10.19). Under the assumptions, we show that (10.20) is also satisfied with inequality.

From (10.19), we can solve for $\Delta_{\mathcal{T}}$,

$$\Delta_{\mathcal{T}} = \mathcal{H}^{-1}_{\mathcal{T},\mathcal{T}} [\overline{\mathbf{R}}_{\mathcal{T}} - \overline{\mathbf{\Sigma}}_{\mathcal{T}} + \overline{\mathbf{S}}_{\mathcal{T}} - \lambda \overline{\mathcal{Z}}(\widetilde{\mathbf{\Omega}})_{\mathcal{T}}].$$

Then

$$\begin{aligned} &||\mathcal{H}_{ab,\mathcal{T}} \mathcal{H}^{-1}_{\mathcal{T},\mathcal{T}} [\overline{\mathbf{R}}_{\mathcal{T}} - \overline{\mathbf{\Sigma}}_{\mathcal{T}} + \overline{\mathbf{S}}_{\mathcal{T}} - \lambda \overline{\mathcal{Z}}(\widetilde{\mathbf{\Omega}})_{\mathcal{T}}] - \overline{\mathbf{R}}_{ab} + \overline{\mathbf{S}}_{ab} - \overline{\mathbf{\Sigma}}^*_{ab}||_2 \\ &\quad \leq \lambda ||\mathcal{H}_{ab,\mathcal{T}} \mathcal{H}^{-1}_{\mathcal{T},\mathcal{T}} \overline{\mathcal{Z}}(\widetilde{\mathbf{\Omega}})_{\mathcal{T}}||_2 + ||\mathcal{H}_{ab,\mathcal{T}} \mathcal{H}^{-1}_{\mathcal{T},\mathcal{T}} [\overline{\mathbf{R}}_{\mathcal{T}} - \overline{\mathbf{\Sigma}}_{\mathcal{T}} + \overline{\mathbf{S}}_{\mathcal{T}}]||_2 + ||\overline{\mathbf{R}}_{ab} + \overline{\mathbf{S}}_{ab} - \overline{\mathbf{\Sigma}}^*_{ab}||_2 \\ &\quad \leq \lambda(1-\alpha) + (2-\alpha)\frac{\alpha\lambda}{4} \\ &\quad < \lambda \end{aligned}$$

using assumption on $\mathcal{H}$ in (10.14) and (10.18). This shows that $\widetilde{\mathbf{\Omega}}$ satisfies (10.16). □

Lemma 10.7. *Assume that*

$$||\mathcal{C}(\mathbf{\Delta})||_\infty \leq \frac{1}{3\kappa_{\mathbf{\Sigma}^*} s}.$$

Then

$$||\mathcal{C}(\mathbf{R}(\mathbf{\Delta}))||_\infty \leq \frac{3s}{2}\kappa^3_{\mathbf{\Sigma}^*}||\mathcal{C}(\mathbf{\Delta})||^2_\infty.$$

Proof. Remainder term can be written as

$$\mathbf{R}(\mathbf{\Delta}) = (\mathbf{\Omega}^* + \mathbf{\Delta})^{-1} - (\mathbf{\Omega}^*)^{-1} + (\mathbf{\Omega}^*)^{-1}\mathbf{\Delta}(\mathbf{\Omega}^*)^{-1}.$$

Using (10.21), we have that

$$\begin{aligned}
|||\mathcal{C}((\mathbf{\Omega}^*)^{-1}\mathbf{\Delta})|||_\infty &\leq |||\mathcal{C}((\mathbf{\Omega}^*)^{-1})|||_\infty |||\mathcal{C}(\mathbf{\Delta})|||_\infty \\
&\leq s|||\mathcal{C}((\mathbf{\Omega}^*)^{-1})|||_\infty ||\mathcal{C}(\mathbf{\Delta})||_\infty \\
&\leq \frac{1}{3}
\end{aligned}$$

which gives us the following expansion

$$(\mathbf{\Omega}^* + \mathbf{\Delta})^{-1} = (\mathbf{\Omega}^*)^{-1} - (\mathbf{\Omega}^*)^{-1}\mathbf{\Delta}(\mathbf{\Omega}^*)^{-1} + (\mathbf{\Omega}^*)^{-1}\mathbf{\Delta}(\mathbf{\Omega}^*)^{-1}\mathbf{\Delta}\mathbf{J}(\mathbf{\Omega}^*)^{-1},$$

with $\mathbf{J} = \sum_{k\geq 0}(-1)^k((\mathbf{\Omega}^*)^{-1}\mathbf{\Delta})^k$. Using (10.22) and (10.21), we have that

$$\begin{aligned}
||\mathcal{C}(\mathbf{R})||_\infty &\leq ||\mathcal{C}((\mathbf{\Omega}^*)^{-1}\mathbf{\Delta})||_\infty |||\mathcal{C}((\mathbf{\Omega}^*)^{-1}\mathbf{\Delta}\mathbf{J}(\mathbf{\Omega}^*)^{-1})'|||_\infty \\
&\leq |||\mathcal{C}((\mathbf{\Omega}^*)^{-1})|||^3_\infty ||\mathcal{C}(\mathbf{\Delta})||_\infty |||\mathcal{C}(\mathbf{J}')|||_\infty |||\mathcal{C}(\mathbf{\Delta})|||_\infty \\
&\leq s|||\mathcal{C}((\mathbf{\Omega}^*)^{-1})|||^3_\infty ||\mathcal{C}(\mathbf{\Delta})||^2_\infty |||\mathcal{C}(\mathbf{J}')|||_\infty.
\end{aligned}$$

Next, we have that

$$\begin{aligned}
|||\mathcal{C}(\mathbf{J}')|||_\infty &\leq \sum_{k>0} |||\mathcal{C}(\mathbf{\Delta}(\mathbf{\Omega}^*)^{-1})|||^k_\infty \\
&\leq \frac{1}{1 - |||\mathcal{C}(\mathbf{\Delta}(\mathbf{\Omega}^*)^{-1})|||_\infty} \\
&\leq \frac{3}{2},
\end{aligned}$$

which gives us

$$||\mathcal{C}(\mathbf{R})||_\infty \leq \frac{3s}{2}\kappa^3_{\mathbf{\Sigma}^*}||\mathcal{C}(\mathbf{\Delta})||^2_\infty$$

as claimed. □

Lemma 10.8. *Assume that*

$$r := 2\kappa_{\mathcal{H}}(||\mathcal{C}(\mathbf{S} - \mathbf{\Sigma}^*)||_\infty + \lambda) \leq \min\left(\frac{1}{3\kappa_{\mathbf{\Sigma}^*} s}, \frac{1}{3\kappa_{\mathcal{H}}\kappa^3_{\mathbf{\Sigma}^*} s}\right).$$

Then

$$||\mathcal{C}(\mathbf{\Delta})||_\infty \leq r.$$

Proof. The proof follows the proof of Lemma 6 in [152]. Define the ball

$$\mathcal{B}(r) := \{\mathbf{A} \ : \ \mathcal{C}(\mathbf{A})_{ab} \leq r, \forall (a,b) \in \mathcal{T}\},$$

the gradient mapping

$$G(\mathbf{\Omega}_{\mathcal{T}}) = -(\mathbf{\Omega}^{-1})_{\mathcal{T}} + \mathbf{S}_{\mathcal{T}} + \lambda \mathcal{Z}(\mathbf{\Omega})_{\mathcal{T}}$$

and

$$F(\overline{\mathbf{\Delta}}_{\mathcal{T}}) = -\mathcal{H}^{-1}_{\mathcal{T}\mathcal{T}} \overline{G}(\mathbf{\Omega}^*_{\mathcal{T}} + \mathbf{\Delta}_{\mathcal{T}}) + \overline{\mathbf{\Delta}}_{\mathcal{T}}.$$

We need to show that $F(\mathcal{B}(r)) \subseteq \mathcal{B}(r)$, which implies that $||\mathcal{C}(\mathbf{\Delta}_{\mathcal{T}})||_\infty \leq r$.

Under the assumptions of the lemma, for any $\mathbf{\Delta}_S \in \mathcal{B}(r)$, we have the following decomposition

$$F(\overline{\mathbf{\Delta}}_{\mathcal{T}}) = \mathcal{H}^{-1}_{\mathcal{T}\mathcal{T}} \overline{\mathbf{R}}(\mathbf{\Delta})_{\mathcal{T}} + \mathcal{H}^{-1}_{\mathcal{T}\mathcal{T}} (\overline{\mathbf{S}}_{\mathcal{T}} - \overline{\mathbf{\Sigma}}^*_{\mathcal{T}} + \lambda \overline{\mathcal{Z}}(\mathbf{\Omega}^* + \mathbf{\Delta})_{\mathcal{T}}).$$

Using Lemma 10.7, the first term can be bounded as

$$\begin{aligned}
||\mathcal{C}(\mathcal{H}^{-1}_{\mathcal{T}\mathcal{T}} \overline{\mathbf{R}}(\mathbf{\Delta})_{\mathcal{T}})||_\infty &\leq |\!|\!|\mathcal{C}(\mathcal{H}^{-1}_{\mathcal{T}\mathcal{T}})|\!|\!|_\infty ||\mathcal{C}(\mathbf{R}(\mathbf{\Delta})||_\infty \\
&\leq \frac{3s}{2} \kappa_{\mathcal{H}} \kappa^3_{\mathbf{\Sigma}^*} ||\mathcal{C}(\mathbf{\Delta})||^2_\infty \\
&\leq \frac{3s}{2} \kappa_{\mathcal{H}} \kappa^3_{\mathbf{\Sigma}^*} r^2 \\
&\leq r/2
\end{aligned}$$

where the last inequality follows under the assumptions. Similarly

$$\begin{aligned}
&||\mathcal{C}(\mathcal{H}^{-1}_{\mathcal{T}\mathcal{T}} (\overline{\mathbf{S}}_{\mathcal{T}} - \overline{\mathbf{\Sigma}}^*_{\mathcal{T}} + \lambda \overline{\mathcal{Z}}(\mathbf{\Omega}^* + \mathbf{\Delta})_{\mathcal{T}})||_\infty \\
&\quad \leq |\!|\!|\mathcal{C}(\mathcal{H}^{-1}_{\mathcal{T}\mathcal{T}})|\!|\!|_\infty (||\mathcal{C}(\mathbf{S} - \mathbf{\Sigma}^*)||_\infty + \lambda ||\mathcal{C}(\mathcal{Z}(\mathbf{\Omega}^* + \mathbf{\Delta}))||_\infty) \\
&\quad \leq \kappa_{\mathcal{H}} (||\mathcal{C}(\mathbf{S} - \overline{\mathbf{\Sigma}}^*)||_\infty + \lambda) \\
&\quad \leq r/2.
\end{aligned}$$

This shows that $F(\mathcal{B}(r)) \subseteq \mathcal{B}(r)$. □

The following result is a corollary of Theorem 10.1, which shows that the graph structure can be estimated consistently under some assumptions.

Corollary 10.1. *Assume that the conditions of Theorem 10.1 are satisfied. Furthermore, suppose that*

$$\min_{(a,b) \in \mathcal{T}, \ a \neq b} ||\mathbf{\Omega}||_F > 2(1 + 8\alpha^{-1}) \kappa_{\mathcal{H}} \overline{\delta}_f(n, p^\tau)$$

then Algorithm 1 estimates a graph $\widehat{G}$ which satisfies

$$\mathbb{P}\left(\widehat{G} \neq G\right) \geq 1 - p^{2-\tau}.$$

Next, we specialize the result of Theorem 10.1 to a case where $\mathbf{X}$ has sub-Gaussian tails. That is, the random vector $\mathbf{X} = (X_1, \ldots, X_{pk})'$ is zero-mean with covariance $\mathbf{\Sigma}^*$. Each $(\sigma^*_{aa})^{-1/2} X_a$ is sub-Gaussian with parameter γ.

Proposition 10.2. *Set the penalty parameter in* λ *in* (10.3) *as*

$$\lambda = 8k\alpha^{-1}\left(128(1+4\gamma^2)^2(\max_a(\sigma^*_{aa})^2)n^{-1}(2\log(2k)+\tau\log(p))\right)^{1/2}.$$

If

$$n > C_1 s^2 k^2 (1+8\alpha^{-1})^2(\tau\log p + \log 4 + 2\log k)$$

where $C_1 = (48\sqrt{2}(1+4\gamma^2)(\max_a \sigma^*_{aa})\max(\kappa_{\mathbf{\Sigma}^*}\kappa_{\mathcal{H}}, \kappa^3_{\mathbf{\Sigma}^*}\kappa^2_{\mathcal{H}}))^2$ *then*

$$||\mathcal{C}(\widehat{\mathbf{\Omega}} - \mathbf{\Omega})||_\infty \leq 16\sqrt{2}(1+4\gamma^2)\max_i \sigma^*_{ii}(1+8\alpha^{-1})\kappa_{\mathcal{H}} k\left(\frac{\tau\log p + \log 4 + 2\log k}{n}\right)^{1/2}$$

with probability $1 - p^{2-\tau}$.

The proof simply follows by observing that, for any (a,b),

$$\begin{aligned}
\mathbb{P}\left(\mathcal{C}(\mathbf{S}-\mathbf{\Sigma}^*)_{ab} > \delta\right) &\leq \mathbb{P}\left(\max_{(c,d)\in(a,b)}(\sigma_{cd}-\sigma^*_{cd})^2 > \delta^2/k^2\right) \\
&\leq k^2\mathbb{P}\left(|\sigma_{cd}-\sigma^*_{cd}| > \delta/k\right) \\
&\leq 4k^2\exp\left(-\frac{n\delta^2}{c_* k^2}\right)
\end{aligned}$$

for all $\delta \in (0, 8(1+4\gamma^2)(\max_a \sigma^*_{aa}))$ with $c_* = 128(1+4\gamma^2)^2(\max_a(\sigma^*_{aa})^2)$. Therefore,

$$\begin{aligned}
f(n,\delta) &= \frac{1}{4k^2}\exp(c_*\frac{n\delta^2}{k^2}) \\
\overline{n}_f(\delta; r) &= \frac{k^2\log(4k^2 r)}{c_*\delta^2} \\
\overline{\delta}_f(r; n) &= \left(\frac{k^2\log(4k^2 r)}{c_* n}\right)^{1/2}.
\end{aligned}$$

Theorem 10.1 and some simple algebra complete the proof.

Proposition 10.1 is a simple conseqeuence of Proposition 10.2.

10.8.5 Some Results on Norms of Block Matrices

Let $\mathcal{T}$ be a partition of V. Throughout this section, we assume that matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{p\times p}$ and a vector $\mathbf{b} \in \mathbb{R}^p$ are partitioned into blocks according to $\mathcal{T}$.

Lemma 10.9.

$$\max_{a\in\mathcal{T}} ||\mathbf{A}_{a\cdot}\mathbf{b}||_2 \leq \max_{a\in\mathcal{T}} \sum_{b\in\mathcal{T}} ||\mathbf{A}_{ab}||_F \max_{c\in\mathcal{T}} ||\mathbf{b}_c||_2.$$

Proof. For any $a \in \mathcal{T}$,

$$\begin{aligned}
||\mathbf{A}_{a\cdot}\mathbf{b}||_2 &\leq \sum_{b\in\mathcal{T}} ||\mathbf{A}_{ab}\mathbf{b}_b||_2 \\
&= \sum_{b\in\mathcal{T}} \left(\sum_{i\in a} (\mathbf{A}_{ib}\mathbf{b}_b)^2 \right)^{1/2} \\
&\leq \sum_{b\in\mathcal{T}} \left(\sum_{i\in a} ||\mathbf{A}_{ib}||_2^2 ||\mathbf{b}_b||_2^2 \right)^{1/2} \\
&\leq \sum_{b\in\mathcal{T}} \left(\sum_{i\in a} ||\mathbf{A}_{ib}||_2^2 \right)^{1/2} \max_{c\in\mathcal{T}} ||\mathbf{b}_c||_2 \\
&= \sum_{b\in\mathcal{T}} ||\mathbf{A}_{ab}||_F \max_{c\in\mathcal{T}} ||\mathbf{b}_c||_2.
\end{aligned}$$

□

Lemma 10.10.

$$\|\mathcal{C}(\mathbf{AB})\|_\infty \leq \|\mathcal{C}(\mathbf{B})\|_\infty \|\mathcal{C}(\mathbf{A})\|_\infty. \tag{10.21}$$

Proof. Let $\mathbf{C} = \mathbf{AB}$ and let $\mathcal{T}$ be a partition of V.

$$\begin{aligned}
\|\mathcal{C}(\mathbf{AB})\|_\infty &= \max_{a\in\mathcal{T}} \sum_{b\in\mathcal{T}} ||\mathbf{C}_{ab}||_F \\
&\leq \max_{a\in\mathcal{T}} \sum_b \sum_c ||\mathbf{A}_{ac}||_F ||\mathbf{B}_{cb}||_F \\
&\leq \{\max_{a\in\mathcal{T}} \sum_c ||\mathbf{A}_{ac}||_F\} \{\max_{c\in\mathcal{T}} \sum_b ||\mathbf{B}_{cb}||_F\} \\
&= \|\mathcal{C}(\mathbf{A})\|_\infty \|\mathcal{C}(\mathbf{B})\|_\infty.
\end{aligned}$$

□

Lemma 10.11.

$$||\mathcal{C}(\mathbf{AB})||_\infty \leq ||\mathcal{C}(\mathbf{A})||_\infty \|\mathcal{C}(\mathbf{B})'\|_\infty. \tag{10.22}$$

Proof. For a fixed a and b,

$$\begin{aligned}
\mathcal{C}(\mathbf{AB})_{ab} &= ||\sum_c \mathbf{A}_{ac}\mathbf{B}_{cb}||_F \\
&\leq \sum_c ||\mathbf{A}_{ac}||_F ||\mathbf{B}_c b||_F \\
&\leq \max_c ||\mathbf{A}_{ac}|| \sum_c ||\mathbf{B}_{cb}||_F.
\end{aligned}$$

Maximizing over a and b gives the result. □

Part II

Feature Selection in Multi-task Learning

Chapter 11

Multi-task learning

It has been empirically observed, on various data sets ranging from cognitive neuroscience Liu et al. (2009) to genome-wide association mapping studies Kim et al. (2009), that considering related estimation tasks jointly, improves estimation performance. Because of this, joint estimation from related tasks or multi-task learning has received much attention in the machine learning and statistics community.

In this part of the thesis, we focus on a particular form of multi-task learning, in which the problem is to estimate the coefficients of several multiple regressions

$$\mathbf{y}_j = \mathbf{X}_j \boldsymbol{\beta}_j + \boldsymbol{\epsilon}_j, \qquad j \in [k] \tag{11.1}$$

where $\mathbf{X}_j \in \mathbb{R}^{n \times p}$ is the design matrix, $\mathbf{y}_j \in \mathbb{R}^n$ is the vector of observations, $\boldsymbol{\epsilon}_j \in \mathbb{R}^n$ is the noise vector and $\boldsymbol{\beta}_j \in \mathbb{R}^p$ is the unknown vector of regression coefficients for the j-th task, with $[k] = \{1, \ldots, k\}$.

Under the model in (11.1), we focus on variable selection under the assumption that the same variables are relevant for different regression problems. We sharply characterize the performance of different penalization schemes on the problem of selecting the relevant variables. Casting the problem of variable selection in the context of the Normal means, we are able to sharply characterize the sparsity patterns under which the Lasso procedure performs better than the group Lasso. Similarly, our results characterize how the group Lasso can perform better when each non-zero row is dense.

Next, we focus on efficient algorithms for screening relevant variables under the multi-task regression model. In particular, we analyze forward regression and marginal regression, which are extremely efficient in ultra-high dimensions. Common tool for variable selection in multi-task regression problems is the penalized least squares procedure, where the penalty biases solution to have many zero coefficients. Though efficient algorithms for these objectives exist, they still do not scale to million of input variables. Therefore, screening procedures are extremely useful for initial reduction of the dimensionality.

11.1 Related Work

Multi-task learning has been an active research area for more than a decade [14, 32, 170]. For an estimation procedure to benefit from multiple tasks, there need to be some connections between

the tasks. One common assumption is that tasks share the feature structure. Along this direction, researchers have proposed to select relevant variables that are predictive for all tasks [116, 120, 123, 143, 144, 172, 203, 204] or to learn transformation of the original variables so that in the transformed space only few features are relevant [3, 144].

The model given in (11.1) has been used in many different domains ranging from multivariate regression [104, 123, 143, 144] and sparse approximation [174] to neural science [120], multi-task learning [3, 123] and biological network estimation [147]. A number of authors have provided theoretical understanding of the estimation in the model using convex programing. [144] propose to minimize the penalized least squares objective with a mixed $(2, 1)$-norm on the coefficients as the penalty term. The authors focus on consistent estimation of the support set S, albeit under the assumption that the number of tasks k is fixed. [143] use the mixed $(\infty, 1)$-norm to penalize the coefficients and focus on the exact recovery of the non-zero pattern of the regression coefficients, rather than the support set S. For a rather limited case of $k = 2$, the authors show that when the regression do not share a common support, it may be harmful to consider the regression problems jointly using the mixed $(\infty, 1)$-norm penalty. In [123], the focus is shifted from the consistent selection to benefits of the joint estimation for the prediction accuracy and consistent estimation. The authors showed the benefits of the joint estimation, when there is a small set of variables common to all outputs and the number of outputs is large.

The Orthogonal Matching Pursuit (OMP) has been analyzed before in the literature (see, for example, [23, 117, 182, 207]). [182] showed that the OMP has the sure screening property in a linear regression with a single output. The exact variable selection property of the OMP is analyzed in [207] and [117]. The exact variable selection requires much stronger assumptions on the design, such as the irrepresentable condition, that are hard to satisfy in the ultra-high dimensional setting. In §13, we focus on the sure screening property, which can be shown to hold under much weaker assumptions.

Marginal regression, also known as correlation learning, marginal learning and sure screening, is one computationally superior alternative to the Lasso. This is a very old and simple procedure, which has recently gained popularity due to its desirable properties in high-dimensional setting [62, 66, 68, 83, 184]. Motivated by successful applications to variable selection in single task problems, we study properties of the marginal regression in a multitask setting in §14.

Chapter 12

Multi-Normal Means Model

Despite many previous investigations, the theory of variable selection in multi-task regression models prior to our work [110] was far from settled. A simple clear picture of when sharing between tasks actually improves performance did not emerge. In particular, to the best of our knowledge, there has been no previous work that sharply characterizes the performance of different penalization schemes on the problem of selecting the relevant variables in the multi-task setting.

In this chapter we study multi-task learning in the context of the *many Normal means model*. This is a simplified model that is often useful for studying the theoretical properties of statistical procedures. The use of the many Normal means model is fairly common in statistics but appears to be less common in machine learning. Our results provide a sharp characterization of the sparsity patterns under which the Lasso procedure performs better than the group Lasso. Similarly, our results characterize how the group Lasso (with the mixed $(2,1)$ norm) can perform better when each non-zero row is dense.

12.1 Introduction

We consider the problem of estimating a sparse signal in the presence of noise. It has been empirically observed, on various data sets ranging from cognitive neuroscience [120] to genome-wide association mapping studies [116], that considering related estimation tasks jointly can improve estimation performance. Because of this, joint estimation from related tasks or *multi-task learning* has received much attention in the machine learning and statistics community (see, for example, [3, 116, 120, 123, 123, 143, 144, 172, 203, 204] and references therein). However, the theory behind multi-task learning is not yet settled.

An example of multi-task learning is the problem of estimating the coefficients of several multiple regressions

$$\mathbf{y}_j = \mathbf{X}_j \boldsymbol{\beta}_j + \boldsymbol{\epsilon}_j, \qquad j \in [k] \tag{12.1}$$

where $\mathbf{X}_j \in \mathbb{R}^{n \times p}$ is the design matrix, $\mathbf{y}_j \in \mathbb{R}^n$ is the vector of observations, $\boldsymbol{\epsilon}_j \in \mathbb{R}^n$ is the noise vector and $\boldsymbol{\beta}_j \in \mathbb{R}^p$ is the unknown vector of regression coefficients for the j-th task, with $[k] = \{1, \ldots, k\}$.

When the number of variables p is much larger than the sample size n, it is commonly assumed that the regression coefficients are jointly sparse, that is, there exists a small subset $S \subset [p]$ of the regression coefficients, with $s := |S| \ll n$, that are non-zero for all or most of the tasks.

The model in (12.1) under the joint sparsity assumption was analyzed in, for example, [144], [123], [143], [123] and [104]. [144] propose to minimize the penalized least squares objective with a mixed $(2, 1)$-norm on the coefficients as the penalty term. The authors focus on consistent estimation of the support set S, albeit under the assumption that the number of tasks k is fixed. [143] use the mixed $(\infty, 1)$-norm to penalize the coefficients and focus on the exact recovery of the non-zero pattern of the regression coefficients, rather than the support set S. For a rather limited case of $k = 2$, the authors show that when the regression do not share a common support, it may be harmful to consider the regression problems jointly using the mixed $(\infty, 1)$-norm penalty. [104] address the feature selection properties of simultaneous greedy forward selection. However, it is not clear what the benefits are compared to the ordinary forward selection done on each task separately. In [123] and [123], the focus is shifted from the consistent selection to benefits of the joint estimation for the prediction accuracy and consistent estimation. The number of tasks k is allowed to increase with the sample size. However, it is assumed that all tasks share the same features; that is, a relevant coefficient is non-zero for all tasks.

Despite these previous investigations, the theory is far from settled. A simple clear picture of when sharing between tasks actually improves performance has not emerged. In particular, to the best of our knowledge, there has been no previous work that sharply characterizes the performance of different penalization schemes on the problem of selecting the relevant variables in the multi-task setting.

In this chapter we study multi-task learning in the context of the *many Normal means model*. This is a simplified model that is often useful for studying the theoretical properties of statistical procedures. The use of the many Normal means model is fairly common in statistics but appears to be less common in machine learning. Our results provide a sharp characterization of the sparsity patterns under which the Lasso procedure performs better than the group Lasso. Similarly, our results characterize how the group Lasso (with the mixed $(2, 1)$ norm) can perform better when each non-zero row is dense.

12.1.1 The Normal Means Model

The simplest Normal means model has the form

$$Y_i = \mu_i + \sigma\epsilon_i, \quad i = 1, \ldots, p \tag{12.2}$$

where $\mu_1, \ldots, \mu_p$ are unknown parameters and $\epsilon_1, \ldots, \epsilon_p$ are independent, identically distributed Normal random variables with mean 0 and variance 1. There are a variety of results [24, 140] showing that many learning problems can be converted into a Normal means problem. This implies that results obtained in the Normal means setting can be transferred to many other settings. As a simple example, consider the nonparametric regression model $Z_i = m(i/n) + \delta_i$ where m is a smooth function on $[0, 1]$ and $\delta_i \sim N(0, 1)$. Let $\phi_1, \phi_2, \ldots,$ be an orthonormal basis on [0,1] and write $m(x) = \sum_{j=1}^{\infty} \mu_j \phi_j(x)$ where $\mu_j = \int_0^1 m(x)\phi_j(x)dx$. To estimate the regression function m we need only estimate $\mu_1, \mu_2, \ldots,$. Let $Y_j = n^{-1} \sum_{i=1}^{n} Z_i \, \phi_j(i/n)$. Then $Y_j \approx N(\mu_j, \sigma^2)$

where $\sigma^2 = 1/n$. This has the form of (12.2) with $\sigma = 1/\sqrt{n}$. Hence this regression problem can be converted into a Normal means model.

However, the most important aspect of the Normal means model is that it allows a clean setting for studying complex problems. We consider the following Normal means model. Let

$$Y_{ij} \sim \begin{cases} (1-\epsilon)\mathcal{N}(0,\sigma^2) + \epsilon\mathcal{N}(\mu_{ij},\sigma^2) & j \in [k], \quad i \in S \\ N(0,\sigma^2) & j \in [k], \quad i \in S^c \end{cases} \tag{12.3}$$

where $(\mu_{ij})_{i,j}$ are unknown real numbers, $\sigma = \sigma_0/\sqrt{n}$ is the variance with $\sigma_0 > 0$ known, $(Y_{ij})_{i,j}$ are random observations, $\epsilon \in [0,1]$ is the parameter that controls the sparsity of features across tasks and $S \subset [p]$ is the set of relevant features. Let $s = |S|$ denote the number of relevant features. Denote the matrix $M \in \mathbb{R}^{p\times k}$ of means

	Tasks			
	1	2	$\dots$	k
1	μ_{11}	μ_{12}	$\dots$	μ_{1k}
2	μ_{21}	μ_{22}	$\dots$	μ_{2k}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
p	μ_{p1}	μ_{p2}	$\dots$	μ_{pk}

and let $\boldsymbol{\theta}_i = (\mu_{ij})_{j\in[k]}$ denote the i-th row of the matrix M. The set $S^c = [p] \setminus S$ indexes the zero rows of the matrix M and the associated observations are distributed according to the Normal distribution with zero mean and variance σ^2. The rows indexed by S are non-zero and the corresponding observation are coming from a mixture of two Normal distributions. The parameter ϵ determines the proportion of observations coming from a Normal distribution with non-zero mean. The reader should regard each column as one vector of parameters that we want to estimate. The question is whether sharing across columns improves the estimation performance.

It is known from the work on the Lasso that in regression problems, the design matrix needs to satisfy certain conditions in order for the Lasso to correctly identify the support S [see 181, for an extensive discussion on the different conditions]. These regularity conditions are essentially unavoidable. However, the Normal means model (12.3) allows us to analyze the estimation procedure in (12.4) and focus on the scaling of the important parameters $(n, k, p, s, \epsilon, \mu_{\min})$ for the success of the support recovery. Using the model (12.3) and the estimation procedure in (12.4), we are able to identify regimes in which estimating the support is more efficient using the ordinary Lasso than with the multi-task Lasso and vice versa. Our results suggest that the multi-task Lasso does not outperform the ordinary Lasso when the features are not considerably shared across tasks; thus, practitioners should be careful when applying the multi-task Lasso without knowledge of the task structure.

An alternative representation of the model is

$$Y_{ij} = \begin{cases} \mathcal{N}(\xi_{ij}\mu_{ij},\sigma^2) & j \in [k], \quad i \in S \\ N(0,\sigma^2) & j \in [k], \quad i \in S^c \end{cases}$$

where ξ_{ij} is a Bernoulli random variable with success probability ϵ. Throughout the chapter, we will set $\epsilon = k^{-\beta}$ for some parameter $\beta \in [0,1)$; $\beta < 1/2$ corresponds to dense rows and $\beta > 1/2$ corresponds to sparse rows. Let $\mu_{\min}$ denote the following quantity $\mu_{\min} = \min |\mu_{ij}|$.

Under the model (12.3), we analyze penalized least squares procedures of the form

$$\widehat{\boldsymbol{\mu}} = \underset{\boldsymbol{\mu} \in \mathbb{R}^{p \times k}}{\operatorname{argmin}} \ \frac{1}{2}||\mathbf{Y} - \boldsymbol{\mu}||_F^2 + \operatorname{pen}(\boldsymbol{\mu}) \tag{12.4}$$

where $||A||_F = \sum_{jk} A_{jk}^2$ is the Frobenious norm, $\operatorname{pen}(\cdot)$ is a penalty function and $\boldsymbol{\mu}$ is a $p \times k$ matrix of means. We consider the following penalties:

1. the ℓ_1 penalty
$$\operatorname{pen}(\boldsymbol{\mu}) = \lambda \sum_{i \in [p]} \sum_{j \in [k]} |\mu_{ij}|,$$
which corresponds to the Lasso procedure applied on each task independently, and denote the resulting estimate as $\widehat{\boldsymbol{\mu}}^{\ell_1}$
2. the mixed $(2,1)$-norm penalty
$$\operatorname{pen}(\boldsymbol{\mu}) = \lambda \sum_{i \in [p]} ||\boldsymbol{\theta}_i||_2,$$
which corresponds to the multi-task Lasso formulation in [144] and [123], and denote the resulting estimate as $\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_2}$
3. the mixed $(\infty,1)$-norm penalty
$$\operatorname{pen}(\boldsymbol{\mu}) = \lambda \sum_{i \in [p]} ||\boldsymbol{\theta}_i||_\infty,$$
which correspond to the multi-task Lasso formulation in [143], and denote the resulting estimate as $\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_\infty}$.

For any solution $\widehat{\boldsymbol{\mu}}$ of (12.4), let $S(\widehat{\boldsymbol{\mu}})$ denote the set of estimated non-zero rows

$$S(\widehat{\boldsymbol{\mu}}) = \{i \in [p] \ : \ ||\widehat{\boldsymbol{\theta}}_i||_2 \neq 0\}.$$

We establish sufficient conditions under which $\mathbb{P}[S(\widehat{\boldsymbol{\mu}}) \neq S] \leq \alpha$ for different methods. These results are complemented with necessary conditions for the recovery of the support set S.

We focus our attention on the three penalties outlined above. There is a large literature on the penalized least squares estimation using concave penalties as introduced in [64]. These penalization methods have better theoretical properties in the presence of the design matrix, especially when the design matrix is far from satisfying the irrepresentable condition [205]. In the Normal means model, due to the lack of the design matrix, there is no advantage to concave penalties in terms of variable selection.

12.1.2 Overview of the Main Results

The main contributions of the chapter can be summarized as follows.

1. We establish a lower bound on the parameter $\mu_{\min}$ as a function of the parameters (n, k, p, s, β). Our result can be interpreted as follows: for any estimation procedure there exists a model given by (12.3) with non-zero elements equal to $\mu_{\min}$ such that the estimation procedure will make an error when identifying the set S with probability bounded away from zero.

2. We establish the sufficient conditions on the signal strength $\mu_{\min}$ for the Lasso and both variants of the group Lasso under which these procedures can correctly identify the set of non-zero rows S.

By comparing the lower bounds with the sufficient conditions, we are able to identify regimes in which each procedure is optimal for the problem of identifying the set of non-zero rows S. Furthermore, we point out that the usage of the popular group Lasso with the mixed $(\infty, 1)$ norm can be disastrous when features are not perfectly shared among tasks. This is further demonstrated through an empirical study.

12.2 Lower Bound on the Support Recovery

In this section, we derive a lower bound for the problem of identifying the correct variables. In particular, we derive conditions on $(n, k, p, s, \epsilon, \mu_{\min})$ under which any method is going to make an error when estimating the correct variables. Intuitively, if $\mu_{\min}$ is very small, a non-zero row may be hard to distinguish from a zero row. Similarly, if ϵ is very small, many elements in a row will be zero and, again, as a result it may be difficult to identify a non-zero row. Before, we give the main result of the section, we introduce the class of models that are going to be considered.

Let

$$\mathcal{F}[\mu] := \{\boldsymbol{\theta} \in \mathbb{R}^k \ : \ \min_j |\theta_j| \geq \mu\}$$

denote the set of feasible non-zero rows. For each $j \in \{0, 1, \ldots, k\}$, let $\mathcal{M}(j, k)$ be the class of all the subsets of $\{1, \ldots, k\}$ of cardinality j. Let

$$\mathbb{M}[\mu, s] = \bigcup_{\omega \in \mathcal{M}(s,p)} \left\{(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p)' \in \mathbb{R}^{p \times k} \ : \ \boldsymbol{\theta}_i \in \mathcal{F}[\mu] \text{ if } i \in \omega, \ \boldsymbol{\theta}_i = \mathbf{0} \text{ if } i \notin \omega\right\} \tag{12.5}$$

be the class of all feasible matrix means. For a matrix $M \in \mathbb{M}[\mu, s]$, let $\mathbb{P}_M$ denote the joint law of $\{Y_{ij}\}_{i \in [p], j \in [k]}$. Since $\mathbb{P}_M$ is a product measure, we can write $\mathbb{P}_M = \otimes_{i \in [p]} \mathbb{P}_{\boldsymbol{\theta}_i}$. For a non-zero row $\boldsymbol{\theta}_i$, we set

$$\mathbb{P}_{\boldsymbol{\theta}_i}(A) = \int \mathcal{N}(A; \widehat{\boldsymbol{\theta}}, \sigma^2 \mathbf{I}_k) d\nu(\widehat{\boldsymbol{\theta}}), \qquad A \in \mathcal{B}(\mathbb{R}^k),$$

where ν is the distribution of the random variable $\sum_{j \in [k]} \mu_{ij} \xi_j e_j$ with $\xi_j \sim \text{Bernoulli}(k^{-\beta})$ and $\{e_j\}_{j \in [k]}$ denoting the canonical basis of $\mathbb{R}^k$. For a zero row $\boldsymbol{\theta}_i = \mathbf{0}$, we set

$$\mathbb{P}_{\mathbf{0}}(A) = \mathcal{N}(A; \mathbf{0}, \sigma^2 \mathbf{I}_k), \qquad A \in \mathcal{B}(\mathbb{R}^k).$$

With this notation, we have the following result.

Theorem 12.1. *Let*

$$\mu_{\min}^2 = \mu_{\min}^2(n, k, p, s, \epsilon, \beta) = \ln\left(1 + u + \sqrt{2u + u^2}\right)\sigma^2$$

where

$$u = \frac{\ln\left(1 + \frac{\alpha^2(p-s+1)}{2}\right)}{2k^{1-2\beta}}.$$

If $\alpha \in (0, \frac{1}{2})$ *and* $k^{-\beta}u < 1$*, then for all* $\mu \leq \mu_{\min}$*,*

$$\inf_{\widehat{\mu}} \sup_{M \in \mathbb{M}[\mu, s]} \mathbb{P}_M[S(\widehat{\mu}) \neq S(M)] \geq \frac{1}{2}(1 - \alpha)$$

where $\mathbb{M}[\mu, s]$ *is given by* (12.5)*.*

The result can be interpreted in words in the following way: whatever the estimation procedure $\widehat{\mu}$, there exists some matrix $M \in \mathbb{M}[\mu_{\min}, s]$ such that the probability of incorrectly identifying the support $S(M)$ is bounded away from zero. In the next section, we will see that some estimation procedures achieve the lower bound given in Theorem 12.1.

12.3 Upper Bounds on the Support Recovery

In this section, we present sufficient conditions on $(n, p, k, \epsilon, \mu_{\min})$ for different estimation procedures, so that

$$\mathbb{P}[S(\widehat{\boldsymbol{\mu}}) \neq S] \leq \alpha.$$

Let $\alpha', \delta' > 0$ be two parameters such that $\alpha' + \delta' = \alpha$. The parameter α' controls the probability of making a type one error

$$\mathbb{P}[\exists i \in [p] \ : \ i \in S(\widehat{\boldsymbol{\mu}}) \text{ and } i \notin S] \leq \alpha',$$

that is, the parameter α' upper bounds the probability that there is a zero row of the matrix M that is estimated as a non-zero row. Likewise, the parameter δ' controls the probability of making a type two error

$$\mathbb{P}[\exists i \in [p] \ : \ i \notin S(\widehat{\boldsymbol{\mu}}) \text{ and } i \in S] \leq \delta',$$

that is, the parameter δ' upper bounds the probability that there is a non-zero row of the matrix M that is estimated as a zero row.

The control of the type one and type two errors is established through the tuning parameter λ. It can be seen that if the parameter λ is chosen such that, for all $i \in S$, it holds that $\mathbb{P}[i \notin S(\widehat{\boldsymbol{\mu}})] \leq \delta'/s$ and, for all $i \in S^c$, it hold that $\mathbb{P}[i \in S(\widehat{\boldsymbol{\mu}})] \leq \alpha'/(p - s)$, then using the union bound we have that $\mathbb{P}[S(\widehat{\boldsymbol{\mu}}) \neq S] \leq \alpha$. In the following subsections, we will use the outlined strategy to choose λ for different estimation procedures.

12.3.1 Upper Bounds for the Lasso

Recall that the Lasso estimator is given as

$$\widehat{\boldsymbol{\mu}}^{\ell_1} = \underset{\boldsymbol{\mu} \in \mathbb{R}^{p \times k}}{\operatorname{argmin}} \ \frac{1}{2}||\mathbf{Y} - \boldsymbol{\mu}||_F^2 + \lambda||\boldsymbol{\mu}||_1.$$

It is easy to see that the solution of the above estimation problem is given as the following soft-thresholding operation

$$\widehat{\mu}_{ij}^{\ell_1} = \left(1 - \frac{\lambda}{|Y_{ij}|}\right)_+ Y_{ij}, \tag{12.6}$$

where $(x)_+ := \max(0, x)$. From (12.6), it is obvious that $i \in S(\widehat{\boldsymbol{\mu}}^{\ell_1})$ if and only if the maximum statistic, defined as

$$M_k(i) = \max_j |Y_{ij}|,$$

satisfies $M_k(i) \geq \lambda$. Therefore it is crucial to find the critical value of the parameter λ such that

$$\left\{ \begin{array}{rcll} \mathbb{P}[M_k(i) < \lambda] & < & \delta'/s & i \in S \\ \mathbb{P}[M_k(i) \geq \lambda] & < & \alpha'/(p-s) & i \in S^c. \end{array} \right.$$

We start by controlling the type one error. For $i \in S^c$ it holds that

$$\mathbb{P}[M_k(i) \geq \lambda] \leq k\mathbb{P}[|\mathcal{N}(0, \sigma^2)| \geq \lambda] \leq \frac{2k\sigma}{\sqrt{2\pi}\lambda} \exp\big(-\frac{\lambda^2}{2\sigma^2}\big) \tag{12.7}$$

using a standard tail bound for the Normal distribution. Setting the right hand side to $\alpha'/(p-s)$ in the above display, we obtain that λ can be set as

$$\lambda = \sigma\sqrt{2\ln\frac{2k(p-s)}{\sqrt{2\pi}\alpha'}} \tag{12.8}$$

and (12.7) holds as soon as $2\ln\frac{2k(p-s)}{\sqrt{2\pi}\alpha'} \geq 1$. Next, we deal with the type two error. Let

$$\pi_k = \mathbb{P}[|(1-\epsilon)\mathcal{N}(0, \sigma^2) + \epsilon\mathcal{N}(\mu_{\min}, \sigma^2)| > \lambda]. \tag{12.9}$$

Then for $i \in S$, $\mathbb{P}[M_k(i) < \lambda] \leq \mathbb{P}[\mathrm{Bin}(k, \pi_k) = 0]$, where $\mathrm{Bin}(k, \pi_k)$ denotes the binomial random variable with parameters (k, π_k). Control of the type two error is going to be established through careful analysis of π_k for various regimes of problem parameters.

Theorem 12.2. *Let λ be defined by* (12.8)*. Suppose $\mu_{\min}$ satisfies one of the following two cases:*

(i) $\mu_{\min} = \sigma\sqrt{2r\ln k}$ *where*

$$r > \left(\sqrt{1 + C_{k,p,s}} - \sqrt{1-\beta}\right)^2$$

with

$$C_{k,p,s} = \frac{\ln\frac{2(p-s)}{\sqrt{2\pi}\alpha'}}{\ln k}$$

and $\lim_{n\to\infty} C_{k,p,s} \in [0, \infty)$*;*

(ii) $\mu_{\min} \geq \lambda$ *when*

$$\lim_{n\to\infty} \frac{\ln k}{\ln(p-s)} = 0$$

and $k^{1-\beta}/2 \geq \ln(s/\delta')$*.*

Then

$$\mathbb{P}[S(\widehat{\boldsymbol{\mu}}^{\ell_1}) \neq S] \leq \alpha.$$

The proof is given in Section 12.6.2. The two different cases describe two different regimes characterized by the ratio of $\ln k$ and $\ln(p-s)$.

Now we can compare the lower bound on $\mu_{\min}^2$ from Theorem 12.1 and the upper bound from Theorem 12.2. Without loss of generality we assume that $\sigma = 1$. We have that when $\beta < 1/2$ the lower bound is of the order $\mathcal{O}\left(\ln\left(k^{\beta-1/2}\ln(p-s)\right)\right)$ and the upper bound is of the order $\ln(k(p-s))$. Ignoring the logarithmic terms in p and s, we have that the lower bound is of the order $\widetilde{\mathcal{O}}(k^{\beta-1/2})$ and the upper bound is of the order $\widetilde{\mathcal{O}}(\ln k)$, which implies that the Lasso does not achieve the lower bound when the non-zero rows are dense. When the non-zero rows are sparse, $\beta > 1/2$, we have that both the lower and upper bound are of the order $\widetilde{\mathcal{O}}(\ln k)$ (ignoring the terms depending on p and s).

12.3.2 Upper Bounds for the Group Lasso

Recall that the group Lasso estimator is given as

$$\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_2} = \underset{\boldsymbol{\mu}\in\mathbb{R}^{p\times k}}{\operatorname{argmin}} \ \frac{1}{2}||\mathbf{Y}-\boldsymbol{\mu}||_F^2 + \lambda\sum_{i\in[p]}||\boldsymbol{\theta}_i||_2,$$

where $\boldsymbol{\theta}_i = (\mu_{ij})_{j\in[k]}$. The group Lasso estimator can be obtained in a closed form as a result of the following thresholding operation [see, for example, 72]

$$\widehat{\boldsymbol{\theta}}_i^{\ell_1/\ell_2} = \left(1 - \frac{\lambda}{||Y_{i\cdot}||^2}\right)_+ Y_{i\cdot} \tag{12.10}$$

where $Y_{i\cdot}$ is the i^{th} row of the data. From (12.10), it is obvious that $i \in S(\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_2})$ if and only if the statistic defined as

$$S_k(i) = \sum_j Y_{ij}^2,$$

satisfies $S_k(i) \geq \lambda$. The choice of λ is crucial for the control of type one and type two errors. We use the following result, which directly follows from Theorem 2 in [22].

Lemma 12.1. *Let $\{Y_i = f_i + \sigma\xi_i\}_{i\in[n]}$ be a sequence of independent observations, where $f = \{f_i\}_{i\in[n]}$ is a sequence of numbers, $\xi_i \overset{iid}{\sim} \mathcal{N}(0,1)$ and σ is a known positive constant. Suppose that $t_{n,\alpha} \in \mathbb{R}$ satisfies $\mathbb{P}[\chi_n^2 > t_{n,\alpha}] \leq \alpha$. Let*

$$\phi_\alpha = I\{\sum_{i\in[n]} Y_i^2 \geq t_{n,\alpha}\sigma^2\}$$

be a test for $f = 0$ versus $f \neq 0$. Then the test ϕ_α satisfies

$$\mathbb{P}[\phi_\alpha = 1] \leq \alpha$$

when $f = 0$ and

$$\mathbb{P}[\phi_\alpha = 0] \leq \delta$$

for all f such that

$$||f||_2^2 \geq 2(\sqrt{5}+4)\sigma^2\ln\left(\frac{2e}{\alpha\delta}\right)\sqrt{n}.$$

Proof. This follows immediately from Theorem 2 in [22]. □

It follows directly from lemma 12.1 that setting

$$\lambda = t_{n,\alpha'/(p-s)}\sigma^2 \tag{12.11}$$

will control the probability of type one error at the desired level, that is,

$$\mathbb{P}[S_k(i) \geq \lambda] \leq \alpha'/(p-s), \qquad \forall i \in S^c.$$

The following theorem gives us the control of the type two error.

Theorem 12.3. *Let* $\lambda = t_{n,\alpha'/(p-s)}\sigma^2$. *Then*

$$\mathbb{P}[S(\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_2}) \neq S] \leq \alpha$$

if

$$\mu_{\min} \geq \sigma\sqrt{2(\sqrt{5}+4)}\sqrt{\frac{k^{-1/2+\beta}}{1-c}}\sqrt{\ln\frac{2e(2s-\delta')(p-s)}{\alpha'\delta'}}$$

where $c = \sqrt{2\ln(2s/\delta')/k^{1-\beta}}$.

The proof is given in Section 12.6.3.

Using Theorem 12.1 and Theorem 12.3 we can compare the lower bound on $\mu_{\min}^2$ and the upper bound. Without loss of generality we assume that $\sigma = 1$. When each non-zero row is dense, that is, when $\beta < 1/2$, we have that both lower and upper bounds are of the order $\widetilde{\mathcal{O}}(k^{\beta-1/2})$ (ignoring the logarithmic terms in p and s). This suggest that the group Lasso performs better than the Lasso for the case where there is a lot of feature sharing between different tasks. Recall from previous section that the Lasso in this setting does not have the optimal dependence on k. However, when $\beta > 1/2$, that is, in the sparse non-zero row regime, we see that the lower bound is of the order $\widetilde{\mathcal{O}}(\ln(k))$ whereas the upper bound is of the order $\widetilde{\mathcal{O}}(k^{\beta-1/2})$. This implies that the group Lasso does not have optimal dependence on k in the sparse non-zero row setting.

12.3.3 Upper Bounds for the Group Lasso with the Mixed $(\infty, 1)$ Norm

In this section, we analyze the group Lasso estimator with the mixed $(\infty, 1)$ norm, defined as

$$\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_\infty} = \underset{\boldsymbol{\mu}\in\mathbb{R}^{p\times k}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{Y}-\boldsymbol{\mu}||_F^2 + \lambda\sum_{i\in[p]}||\boldsymbol{\theta}_i||_\infty,$$

where $\boldsymbol{\theta}_i = (\mu_{ij})_{j\in[k]}$. The closed form solution for $\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_\infty}$ can be obtained [see 120], however, we are only going to use the following lemma.

Lemma 12.2. *[120]* $\widehat{\boldsymbol{\theta}}_i^{\ell_1/\ell_\infty} = 0$ *if and only if* $\sum_j |Y_{ij}| \leq \lambda$.

Proof. See the proof of Proposition 5 in [120]. □

Suppose that the penalty parameter λ is set as

$$\lambda = k\sigma\sqrt{2\ln\frac{k(p-s)}{\alpha'}}. \tag{12.12}$$

It follows immediately using a tail bound for the Normal distribution that

$$\mathbb{P}[\sum_j |Y_{ij}| \geq \lambda] \leq k \max_j \mathbb{P}[|Y_{ij}| \geq \lambda/k] \leq \alpha'/(p-s), \qquad \forall i \in S^c,$$

which implies that the probability of the type one error is controlled at the desired level.

Theorem 12.4. *Let the penalty parameter λ be defined by* (12.12)*. Then*

$$\mathbb{P}[S(\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_\infty}) \neq S] \leq \alpha$$

if

$$\mu_{\min} \geq \frac{1+\tau}{1-c}k^{-1+\beta}\lambda$$

where $c = \sqrt{2\ln(2s/\delta')/k^{1-\beta}}$ *and* $\tau = \sigma\sqrt{2k\ln\frac{2s-\delta'}{\delta'}}/\lambda$.

The proof is given in Section 12.6.4.

Comparing upper bounds for the Lasso and the group Lasso with the mixed $(2,1)$ norm with the result of Theorem 12.4, we can see that both the Lasso and the group Lasso have better dependence on k than the group Lasso with the mixed $(\infty,1)$ norm. The difference becomes more pronounced as β increases. This suggest that we should be very cautious when using the group Lasso with the mixed $(\infty,1)$ norm, since as soon as the tasks do not share exactly the same features, the other two procedures have much better performance on identifying the set of non-zero rows.

12.4 Simulation Results

We conduct a small-scale empirical study of the performance of the Lasso and the group Lasso (both with the mixed $(2,1)$ norm and with the mixed $(\infty,1)$ norm). Our empirical study shows that the theoretical findings of Section 12.3 describe sharply the behavior of procedures even for small sample studies. In particular, we demonstrate that as the minimum signal level $\mu_{\min}$ varies in the model (12.3), our theory sharply determines points at which probability of identifying non-zero rows of matrix M successfully transitions from 0 to 1 for different procedures.

The simulation procedure can be described as follows. Without loss of generality we let $S = [s]$ and draw the samples $\{Y_{ij}\}_{i\in[p],j\in[k]}$ according to the model in (12.3). The total number of rows p is varied in $\{128, 256, 512, 1024\}$ and the number of columns is set to $k = \lfloor p\log_2(p)\rfloor$. The sparsity of each non-zero row is controlled by changing the parameter β in $\{0, 0.25, 0.5, 0.75\}$ and setting $\epsilon = k^{-\beta}$. The number of non-zero rows is set to $s = \lfloor\log_2(p)\rfloor$, the sample size is set to $n = 0.1p$ and $\sigma_0 = 1$. The parameters α' and δ' are both set to 0.01. For each setting of the parameters, we report our results averaged over 1000 simulation runs.

Simulations with other choices of parameters n, s and k have been tried out, but the results were qualitatively similar and, hence, we do not report them here.

The regularization parameter λ is chosen according to Equations (12.8), (12.11) and (12.12), which assume that the noise level σ_0 is known. In practice, estimating the standard deviation of the noise in high-dimensions is a hard problem and practitioners often use cross-validation as a data-driven way to choose the penalty parameter. For recent work on data-driven tuning of the penalty parameters, we refer the reader to [6].

12.4.1 Lasso

We investigate the performance on the Lasso for the purpose of estimating the set of non-zero rows, S. Figure 12.1 plots the probability of success as a function of the signal strength. On the same figure we plot the probability of success for the group Lasso with both $(2, 1)$ and $(\infty, 1)$-mixed norms. Using theorem 12.2, we set

$$\mu_{\text{lasso}} = \sqrt{2(r + 0.001) \ln k} \tag{12.13}$$

where r is defined in theorem 12.2. Next, we generate data according to (12.3) with all elements $\{\mu_{ij}\}$ set to $\mu = \rho\mu_{\text{lasso}}$, where $\rho \in [0.05, 2]$. The penalty parameter λ is chosen as in (12.8). Figure 12.1 plots probability of success as a function of the parameter ρ, which controls the signal strength. This probability transitions very sharply from 0 to 1. A rectangle on a horizontal line represents points at which the probability $\mathbb{P}[\widehat{S} = S]$ is between 0.05 and 0.95. From each subfigure in Figure 12.1, we can observe that the probability of success for the Lasso transitions from 0 to 1 for the same value of the parameter ρ for different values of p, which indicates that, except for constants, our theory correctly characterizes the scaling of $\mu_{\min}$. In addition, we can see that the Lasso outperforms the group Lasso (with $(2, 1)$-mixed norm) when each non-zero row is very sparse (the parameter β is close to one).

12.4.2 Group Lasso

Next, we focus on the empirical performance of the group Lasso with the mixed $(2, 1)$ norm. Figure 12.2 plots the probability of success as a function of the signal strength. Using theorem 12.3, we set

$$\mu_{\text{group}} = \sigma\sqrt{2(\sqrt{5} + 4)}\sqrt{\frac{k^{-1/2+\beta}}{1 - c}}\sqrt{\ln \frac{(2s - \delta')(p - s)}{\alpha'\delta'}} \tag{12.14}$$

where c is defined in theorem 12.3. Next, we generate data according to (12.3) with all elements $\{\mu_{ij}\}$ set to $\mu = \rho\mu_{\text{group}}$, where $\rho \in [0.05, 2]$. The penalty parameter λ is given by (12.11). Figure 12.2 plots probability of success as a function of the parameter ρ, which controls the signal strength. A rectangle on a horizontal line represents points at which the probability $\mathbb{P}[\widehat{S} = S]$ is between 0.05 and 0.95. From each subfigure in Figure 12.2, we can observe that the probability of success for the group Lasso transitions from 0 to 1 for the same value of the parameter ρ for different values of p, which indicated that, except for constants, our theory correctly characterizes

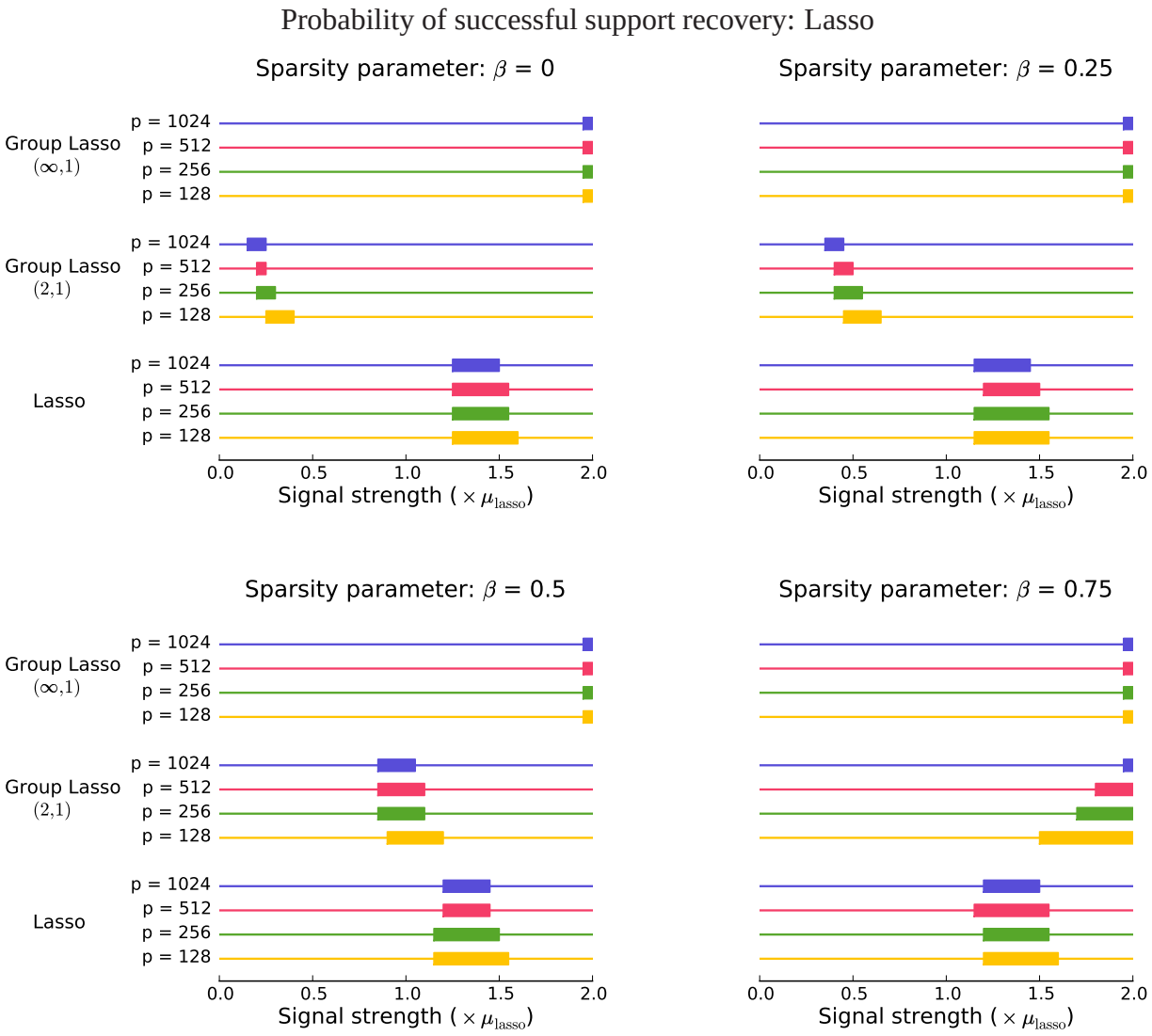


Figure 12.1: The probability of success for the Lasso for the problem of estimating S plotted against the signal strength, which is varied as a multiple of μ_{lasso} defined in (12.13). A rectangle on each horizontal line represents points at which the probability $\mathbb{P}[\widehat{S} = S]$ is between 0.05 and 0.95. To the left of the rectangle the probability is smaller than 0.05, while to the right the probability is larger than 0.95. Different subplots represent the probability of success as the sparsity parameter β changes.

the scaling of $\mu_{\min}$. We observe also that the group Lasso outperforms the Lasso when each non-zero row is not too sparse, that is, when there is a considerable overlap of features between different tasks.

12.4.3 Group Lasso with the Mixed $(\infty, 1)$ Norm

Next, we focus on the empirical performance of the group Lasso with the mixed $(\infty, 1)$ norm. Figure 12.3 plots the probability of success as a function of the signal strength. Using theorem 12.4, we set

$$\mu_{\text{infty}} = \frac{1+\tau}{1-c} k^{-1+\beta} \lambda \tag{12.15}$$

where τ and c are defined in theorem 12.4 and λ is given by (12.12). Next, we generate data according to (12.3) with all elements $\{\mu_{ij}\}$ set to $\mu = \rho \mu_{\text{infty}}$, where $\rho \in [0.05, 2]$. Figure 12.3 plots probability of success as a function of the parameter ρ, which controls the signal strength. A rectangle on a horizontal line represents points at which the probability $\mathbb{P}[\widehat{S} = S]$ is between 0.05 and 0.95. From each subfigure in Figure 12.3, we can observe that the probability of success for the group Lasso transitions from 0 to 1 for the same value of the parameter ρ for different values of p, which indicated that, except for constants, our theory correctly characterizes the scaling of $\mu_{\min}$. We also observe that the group Lasso with the mixed $(\infty, 1)$ norm never outperforms the Lasso or the group Lasso with the mixed $(2, 1)$ norm.

12.5 Discussion

We have studied the benefits of task sharing in sparse problems. Under many scenarios, the group lasso outperforms the lasso. The ℓ_1/ℓ_2 penalty seems to be a much better choice for the group lasso than the ℓ_1/ℓ_∞. However, as pointed out to us by Han Liu, for screening, where false discoveries are less important than accurate recovery, it is possible that the ℓ_1/ℓ_∞ penalty could be useful. From the results in Section 12.3, we can further conclude that the Lasso procedure performs better than the group Lasso when each non-zero row is sparse, while the group Lasso (with the mixed $(2, 1)$ norm) performs better when each non-zero row is dense. Since in many practical situations one does not know how much overlap there is between different tasks, it would be useful to combine the Lasso and the group Lasso in order to improve the performance. For example, one can take the union of the Lasso and the group Lasso estimate, $\widehat{S} = S(\widehat{\boldsymbol{\mu}}^{\ell_1}) \cup S(\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_2})$. The suggested approach has the advantage that one does not need to know in advance which estimation procedure to use. While such a combination can be justified in the Normal means problem as a way to increase the power to detect the non-zero rows, it is not clear whether the same approach can be justified in the multi-task regression model (12.1).

The analysis of the Normal means model in (12.3) provides insights into the theoretical results we could expect in the conventional multi-task learning given in (12.1). However, there is no direct way to translate our results into valid results for the model in (12.1); a separate analysis needs to be done in order to establish sharp theoretical results.

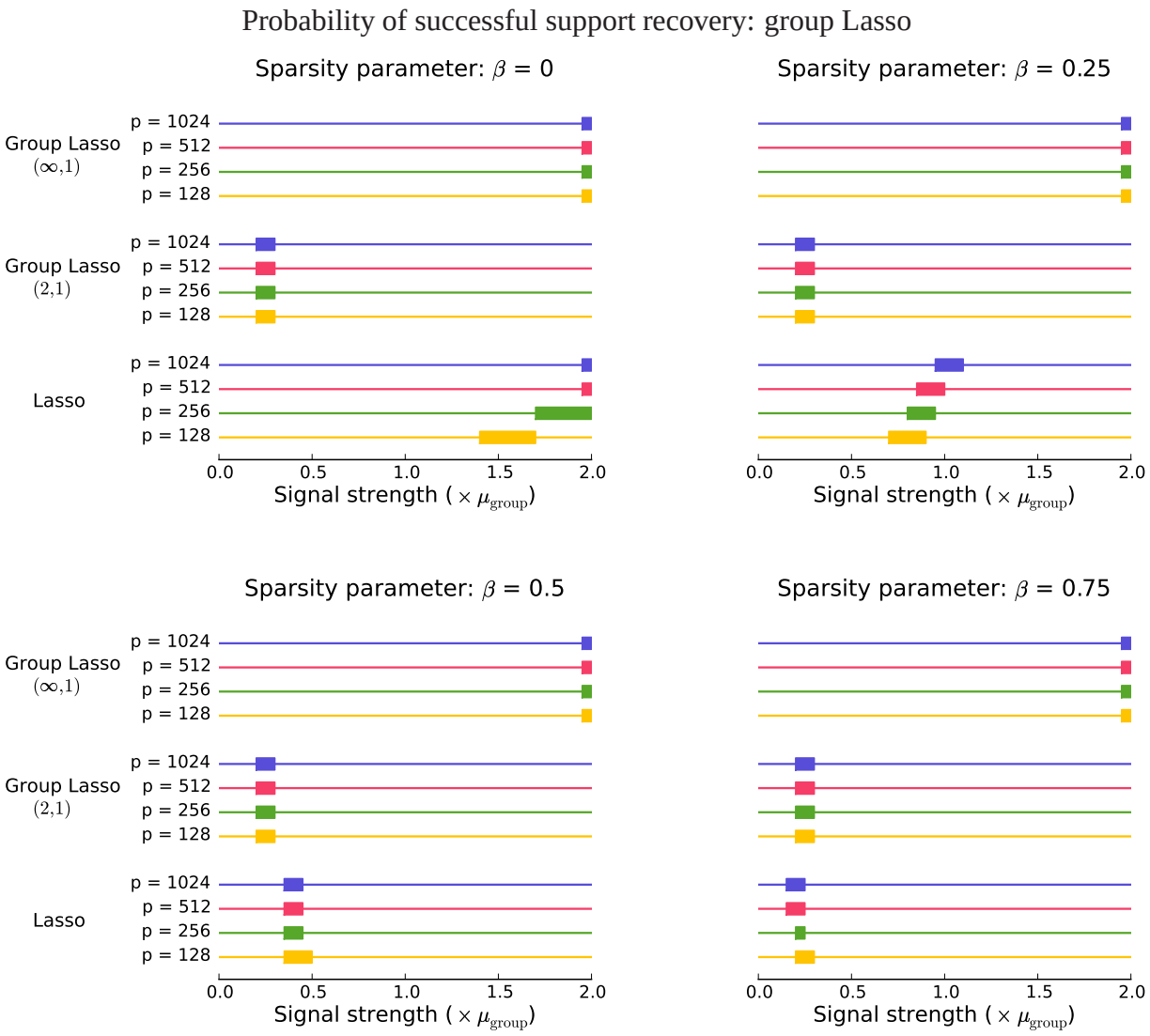


Figure 12.2: The probability of success for the group Lasso for the problem of estimating S plotted against the signal strength, which is varied as a multiple of μ_{group} defined in (12.14). A rectangle on each horizontal line represents points at which the probability $\mathbb{P}[\widehat{S} = S]$ is between 0.05 and 0.95. To the left of the rectangle the probability is smaller than 0.05, while to the right the probability is larger than 0.95. Different subplots represent the probability of success as the sparsity parameter β changes.

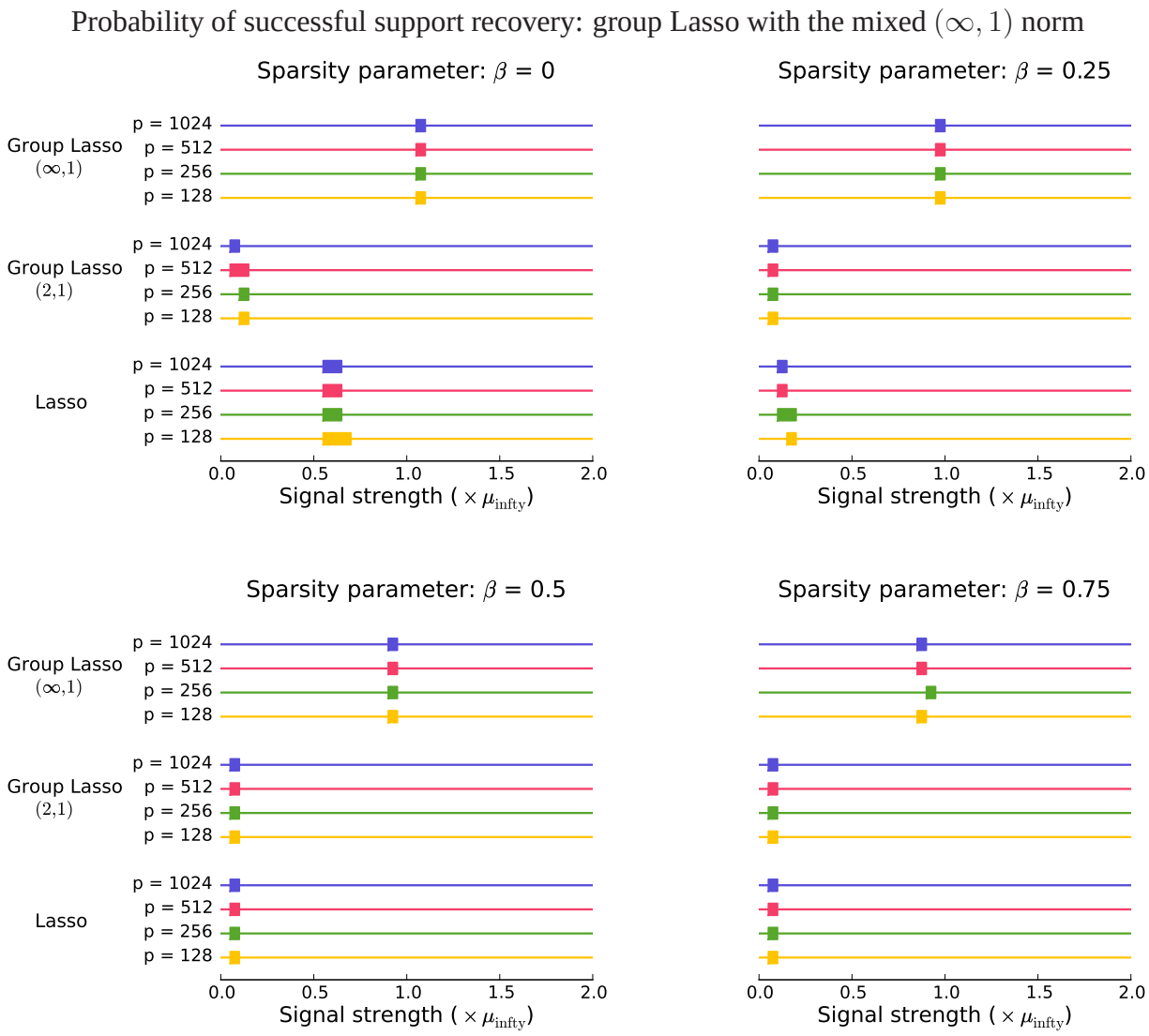


Figure 12.3: The probability of success for the group Lasso with mixed $(\infty, 1)$ norm for the problem of estimating S plotted against the signal strength, which is varied as a multiple of μ_{infty} defined in (12.15). A rectangle on each horizontal line represents points at which the probability $\mathbb{P}[\widehat{S} = S]$ is between 0.05 and 0.95. To the left of the rectangle the probability is smaller than 0.05, while to the right the probability is larger than 0.95. Different subplots represent the probability of success as the sparsity parameter β changes.

12.6 Technical Proofs

This section collects technical proofs of the results presented in the chapter. Throughout the section we use $c_1, c_2, \ldots$ to denote positive constants whose value may change from line to line.

12.6.1 Proof of Theorem 12.1

Without loss of generality, we may assume $\sigma = 1$. Let $\phi(u)$ be the density of $\mathcal{N}(0,1)$ and define $\mathbb{P}_{\mathbf{0}}$ and $\mathbb{P}_{\mathbf{1}}$ to be two probability measures on $\mathbb{R}^k$ with the densities with respect to the Lebesgue measure given as

$$f_0(a_1, \ldots, a_k) = \prod_{j \in [k]} \phi(a_j) \tag{12.16}$$

and

$$f_1(a_1, \ldots, a_k) = \mathbb{E}_Z \mathbb{E}_m \mathbb{E}_\xi \prod_{j \in m} \phi(a_j - \xi_j \mu_{\min}) \prod_{j \notin m} \phi(a_j) \tag{12.17}$$

where $Z \sim \mathrm{Bin}(k, k^{-\beta})$, m is a random variable uniformly distributed over $\mathcal{M}(Z,k)$ and $\{\xi_j\}_{j\in[k]}$ is a sequence of Rademacher random variables, independent of Z and m. A Rademacher random variable takes values ± 1 with probability $\frac{1}{2}$.

To simplify the discussion, suppose that $p - s + 1$ is divisible by 2. Let $T = (p - s + 1)/2$. Using $\mathbb{P}_{\mathbf{0}}$ and $\mathbb{P}_{\mathbf{1}}$, we construct the following three measures,

$$\widetilde{\mathbb{Q}} = \mathbb{P}_{\mathbf{1}}^{s-1} \otimes \mathbb{P}_{\mathbf{0}}^{p-s+1},$$

$$\mathbb{Q}_0 = \frac{1}{T} \sum_{\substack{j \in \{s, \ldots, p\} \\ j \text{ odd}}} \mathbb{P}_{\mathbf{1}}^{s-1} \otimes \mathbb{P}_{\mathbf{0}}^{j-s} \otimes \mathbb{P}_{\mathbf{1}} \otimes \mathbb{P}_{\mathbf{0}}^{p-j}$$

and

$$\mathbb{Q}_1 = \frac{1}{T} \sum_{\substack{j \in \{s, \ldots, p\} \\ j \text{ even}}} \mathbb{P}_{\mathbf{1}}^{s-1} \otimes \mathbb{P}_{\mathbf{0}}^{j-s} \otimes \mathbb{P}_{\mathbf{1}} \otimes \mathbb{P}_{\mathbf{0}}^{p-j}.$$

It holds that

$$\begin{aligned} \inf_{\widehat{\mu}} \sup_{M \in \mathbb{M}} \mathbb{P}_M[S(M) \neq S(\widehat{\mu})] &\geq \inf_{\Psi} \max \Big(\mathbb{Q}_0(\Psi = 1), \mathbb{Q}_1(\Psi = 0) \Big) \\ &\geq \frac{1}{2} - \frac{1}{2} ||\mathbb{Q}_0 - \mathbb{Q}_1||_1, \end{aligned}$$

where the infimum is taken over all tests Ψ taking values in $\{0,1\}$ and $||\cdot||_1$ is the total variation distance between probability measures. For a readable introduction on lower bounds on the minimax probability of error, see Section 2 in [171]. In particular, our approach is related to the one described in Section 2.7.4. We proceed by upper bounding the total variation distance

between $\mathbb{Q}_0$ and $\mathbb{Q}_1$. Let $g = d\mathbb{P}_{\mathbf{1}}/d\mathbb{P}_{\mathbf{0}}$ and let $u_i \in \mathbb{R}^k$ for each $i \in [p]$, then

$$\begin{aligned}
&\frac{d\mathbb{Q}_0}{d\widetilde{\mathbb{Q}}}(u_1, \ldots, u_p) \\
&\quad = \frac{1}{T} \sum_{\substack{j \in \{s,\ldots,p\} \\ j \text{ even}}} \prod_{i \in \{1,\ldots,s-1\}} \frac{d\mathbb{P}_{\mathbf{1}}}{d\mathbb{P}_{\mathbf{1}}}(u_i) \prod_{i \in \{s,\ldots,j-1\}} \frac{d\mathbb{P}_{\mathbf{0}}}{d\mathbb{P}_{\mathbf{0}}}(u_i) \frac{d\mathbb{P}_{\mathbf{1}}}{d\mathbb{P}_{\mathbf{0}}}(u_j) \prod_{i \in \{j+1,\ldots,p\}} \frac{d\mathbb{P}_{\mathbf{0}}}{d\mathbb{P}_{\mathbf{0}}}(u_i) \\
&\quad = \frac{1}{T} \sum_{\substack{j \in \{s,\ldots,p\} \\ j \text{ even}}} g(u_j)
\end{aligned}$$

and, similarly, we can compute $d\mathbb{Q}_1/d\widetilde{\mathbb{Q}}$. The following holds

$$\begin{aligned}
&\|\mathbb{Q}_0 - \mathbb{Q}_1\|_1^2 \\
&\quad = \left(\int \Big| \frac{1}{T} \Big(\sum_{\substack{j \in \{s,\ldots,p\} \\ j \text{ even}}} g(u_j) - \sum_{\substack{j \in \{s,\ldots,p\} \\ j \text{ odd}}} g(u_j) \Big) \Big| \prod_{i \in \{s,\ldots,p\}} d\mathbb{P}_{\mathbf{0}}(u_i) \right)^2 \\
&\quad \leq \frac{1}{T^2} \int \Big(\sum_{\substack{j \in \{s,\ldots,p\} \\ j \text{ even}}} g(u_j) - \sum_{\substack{j \in \{s,\ldots,p\} \\ j \text{ odd}}} g(u_j) \Big)^2 \prod_{i \in \{s,\ldots,p\}} d\mathbb{P}_{\mathbf{0}}(u_i) \\
&\quad = \frac{2}{T} \big(\mathbb{P}_{\mathbf{0}}(g^2) - 1 \big),
\end{aligned} \tag{12.18}$$

where the last equality follows by observing that

$$\int \sum_{\substack{j \in \{s,\ldots,p\} \\ j \text{ even}}} \sum_{\substack{j' \in \{s,\ldots,p\} \\ j' \text{ even}}} g(u_j) g(u_{j'}) \prod_{\substack{i \in \{s,\ldots,p\} \\ i \text{ even}}} d\mathbb{P}_{\mathbf{0}}(u_i) = T\, \mathbb{P}_{\mathbf{0}}(g^2) + T^2 - T$$

and

$$\int \sum_{\substack{j \in \{s,\ldots,p\} \\ j \text{ even}}} \sum_{\substack{j' \in \{s,\ldots,p\} \\ j' \text{ odd}}} g(u_j) g(u_{j'}) \prod_{i \in \{s,\ldots,p\}} d\mathbb{P}_{\mathbf{0}}(u_i) = T^2.$$

Next, we proceed to upper bound $\mathbb{P}_{\mathbf{0}}(g^2)$, using some ideas presented in the proof of Theorem 1 in [22]. Recall definitions of f_0 and f_1 in (12.16) and (12.17) respectively. Then $g = d\mathbb{P}_{\mathbf{1}}/d\mathbb{P}_{\mathbf{0}} = f_1/f_0$ and we have

$$\begin{aligned}
g(a_1, \ldots, a_k) &= \mathbb{E}_Z \mathbb{E}_m \mathbb{E}_\xi \left[\exp\Big(-\frac{Z\mu_{\min}^2}{2} + \mu_{\min} \sum_{j \in m} \xi_j a_j \Big) \right] \\
&= \mathbb{E}_Z \left[\exp\Big(-\frac{Z\mu_{\min}^2}{2} \Big)\, \mathbb{E}_m \Big[\prod_{j \in m} \cosh(\mu_{\min} a_j) \Big] \right].
\end{aligned}$$

Furthermore, let $Z' \sim \mathrm{Bin}(k, k^{-\beta})$ be independent of Z and m' uniformly distributed over

$\mathcal{M}(Z', k)$. The following holds

$$\begin{aligned}
&\mathbb{P}_{\mathbf{0}}(g^2) \\
&= \mathbb{P}_{\mathbf{0}}\Big(\mathbb{E}_{Z',Z}\Big[\exp\Big(-\frac{(Z+Z')\mu_{\min}^2}{2}\Big)\,\mathbb{E}_{m,m'}\prod_{j\in m}\cosh(\mu_{\min}a_j)\prod_{j\in m'}\cosh(\mu_{\min}a_j)\Big]\Big) \\
&= \mathbb{E}_{Z',Z}\Big[\exp\Big(-\frac{(Z+Z')\mu_{\min}^2}{2}\Big) \\
&\qquad\qquad \mathbb{E}_{m,m'}\Big[\prod_{j\in m\cap m'}\int\cosh^2(\mu_{\min}a_j)\phi(a_j)da_j \\
&\qquad\qquad\qquad \prod_{j\in m\triangle m'}\int\cosh(\mu_{\min}a_j)\phi(a_j)da_j\Big]\Big],
\end{aligned}$$

where we use $m\triangle m'$ to denote $(m\cup m')\setminus(m\cap m')$. By direct calculation, we have that

$$\int\cosh^2(\mu_{\min}a_j)\phi(a_j)da_j = \exp(\mu_{\min}^2)\cosh(\mu_{\min}^2)$$

and

$$\int\cosh(\mu_{\min}a_j)\phi(a_j)da_j = \exp(\mu_{\min}^2/2).$$

Since $\frac{1}{2}|m\triangle m'| + |m\cap m'| = (Z+Z')/2$, we have that

$$\begin{aligned}
\mathbb{P}_{\mathbf{0}}(g^2) &= \mathbb{E}_{Z,Z'}\Big[E_{m,m'}\Big[\big(\cosh(\mu_{\min}^2)\big)^{|m\cap m'|}\Big]\Big] \\
&= \mathbb{E}_{Z,Z'}\Big[\sum_{j=0}^{k}p_j\big(\cosh(\mu_{\min}^2)\big)^j\Big] \\
&= \mathbb{E}_{Z,Z'}\Big[\mathbb{E}_X\Big[\cosh(\mu_{\min}^2)^X\Big]\Big],
\end{aligned}$$

where

$$p_j = \begin{cases} 0 & \text{if } j < Z+Z'-k \text{ or } j > \min(Z,Z') \\ \frac{\binom{Z'}{j}\binom{k-Z'}{Z-j}}{\binom{k}{Z}} & \text{otherwise} \end{cases}$$

and $P[X=j] = p_j$. Therefore, X follows a hypergeometric distribution with parameters k, Z, Z'/k. [The first parameter denotes the total number of stones in an urn, the second parameter denotes the number of stones we are going to sample without replacement from the urn and the last parameter denotes the fraction of white stones in the urn.] Then following [9, p. 173; see also [22]], we know that X has the same distribution as the random variable $\mathbb{E}[\widetilde{X}|\mathcal{T}]$ where $\widetilde{X}$ is a binomial random variable with parameters Z and Z'/k, and $\mathcal{T}$ is a suitable σ-algebra. By convexity, it follows that

$$\begin{aligned}
\mathbb{P}_{\mathbf{0}}(g^2) &\le \mathbb{E}_{Z,Z'}\Big[\mathbb{E}_{\widetilde{X}}\Big[\cosh(\mu_{\min}^2)^{\widetilde{X}}\Big]\Big] \\
&= \mathbb{E}_{Z,Z'}\left[\exp\left(Z\ln\left(1+\frac{Z'}{k}\big(\cosh(\mu_{\min}^2)-1\big)\right)\right)\right] \\
&= \mathbb{E}_{Z'}\mathbb{E}_{Z}\left[\exp\left(Z\ln\left(1+\frac{Z'}{k}u\right)\right)\right]
\end{aligned}$$

where $\mu_{\min}^2 = \ln(1 + u + \sqrt{2u + u^2})$ with

$$u = \frac{\ln\left(1 + \frac{\alpha^2 T}{2}\right)}{2k^{1-2\beta}}.$$

Continuing with our calculations, we have that

$$\begin{aligned}
\mathbb{P}_{\mathbf{0}}(g^2) &= \mathbb{E}_{Z'} \exp\left(k \ln\left(1 + k^{-(1+\beta)} u Z'\right)\right) \\
&\leq \mathbb{E}_{Z'} \exp\left(k^{-\beta} u Z'\right) \\
&= \exp\left(k \ln\left(1 + k^{-\beta}\big(\exp(k^{-\beta}u) - 1\big)\right)\right) \\
&\leq \exp\left(k^{1-\beta}\big(\exp\left(k^{-\beta}u\right) - 1\big)\right) \\
&\leq \exp\left(2k^{1-2\beta}u\right) \\
&= 1 + \frac{\alpha^2 T}{2},
\end{aligned} \tag{12.19}$$

where the last inequality follows since $k^{-\beta}u < 1$ for all large p. Combining (12.19) with (12.18), we have that

$$\|\mathbb{Q}_0 - \mathbb{Q}_1\|_1 \leq \alpha,$$

which implies that

$$\inf_{\widehat{\mu}} \sup_{M \in \mathbb{M}} \mathbb{P}_M[S(M) \neq S(\widehat{\mu})] \geq \frac{1}{2} - \frac{1}{2}\alpha.$$

12.6.2 Proof of Theorem 12.2

Without loss of generality, we can assume that $\sigma = 1$ and rescale the final result. For λ given in (12.8), it holds that $\mathbb{P}[|\mathcal{N}(0,1) \geq \lambda] = o(1)$. For the probability defined in (12.9), we have the following lower bound

$$\pi_k = (1-\epsilon)\mathbb{P}[|\mathcal{N}(0,1)| \geq \lambda] + \epsilon\mathbb{P}[|\mathcal{N}(\mu_{\min}, 1)| \geq \lambda] \geq \epsilon\mathbb{P}[\mathcal{N}(\mu_{\min}, 1) \geq \lambda].$$

We prove the two cases separately.
Case 1: *Large number of tasks.* By direct calculation

$$\pi_k \geq \epsilon\mathbb{P}[\mathcal{N}(\mu_{\min}, 1) \geq \lambda] = \frac{1}{\sqrt{4\pi \log k}\big(\sqrt{1 + C_{k,p,s}} - \sqrt{r}\big)} k^{-\beta - \left(\sqrt{1+C_{k,p,s}} - \sqrt{r}\right)^2} =: \underline{\pi_k}.$$

Since $1 - \beta > \left(\sqrt{1 + C_{k,p,s}} - \sqrt{r}\right)^2$, we have that $\mathbb{P}[\mathrm{Bin}(k, \pi_k) = 0] \xrightarrow{n\to\infty} 0$. We can conclude that as soon as $k\underline{\pi_k} \geq \ln(s/\delta')$, it holds that $\mathbb{P}[S(\widehat{\mu}^{\ell_1}) \neq S] \leq \alpha$.
Case 2: *Medium number of tasks.* When $\mu_{\min} \geq \lambda$, it holds that

$$\pi_k \geq \epsilon\mathbb{P}[\mathcal{N}(\mu_{\min}, 1) \geq \lambda] \geq \frac{k^{-\beta}}{2}.$$

We can conclude that as soon as $k^{1-\beta}/2 \geq \ln(s/\delta')$, it holds that $\mathbb{P}[S(\widehat{\mu}^{\ell_1}) \neq S] \leq \alpha$.

12.6.3 Proof of Theorem 12.3

Using a Chernoff bound, $\mathbb{P}[\mathrm{Bin}(k, k^{-\beta}) \leq (1-c)k^{1-\beta}] \leq \delta'/2s$ for $c = \sqrt{2\ln(2s/\delta')/k^{1-\beta}}$. For $i \in S$, we have that

$$\mathbb{P}[S_k(i) \leq \lambda] \leq \frac{\delta'}{2s} + \Big(1 - \frac{\delta'}{2s}\Big)\mathbb{P}\Big[S_k(i) \leq \lambda \;\Big|\; \Big\{||\theta_i||_2^2 \geq (1-c)k^{1-\beta}\mu_{\min}^2\Big\}\Big].$$

Therefore, using lemma 12.1 with $\delta = \delta'/(2s - \delta')$, if follows that $\mathbb{P}[S_k(i) \leq \lambda] \leq \delta'/(2s)$ for all $i \in S$ when

$$\mu_{\min} \geq \sigma\sqrt{2(\sqrt{5}+4)}\sqrt{\frac{k^{-1/2+\beta}}{1-c}}\sqrt{\ln\frac{2e(2s-\delta')(p-s)}{\alpha'\delta'}}.$$

Since $\lambda = t_{n,\alpha'/(p-s)}\sigma^2$, $\mathbb{P}[S_k(i) \geq \lambda] \leq \alpha'/(p-s)$ for all $i \in S^c$. We can conclude that $\mathbb{P}[S(\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_2}) \neq S] \leq \alpha$.

12.6.4 Proof of Theorem 12.4

Without loss of generality, we can assume that $\sigma = 1$. Proceeding as in the proof of theorem 12.3, $\mathbb{P}[\mathrm{Bin}(k, k^{-\beta}) \leq (1-c)k^{1-\beta}] \leq \delta'/2s$ for $c = \sqrt{2\ln(2s/\delta')/k^{1-\beta}}$. Then for $i \in S$ it holds that

$$\mathbb{P}[\sum_j |Y_{ij}| \leq \lambda] \leq \frac{\delta'}{2s} + \Big(1 - \frac{\delta'}{2s}\Big)\mathbb{P}[(1-c)k^{1-\beta}\mu_{\min} + z_k \leq \lambda],$$

where $z_k \sim \mathcal{N}(0,k)$. Since $(1-c)k^{1-\beta}\mu_{\min} \geq (1+\tau)\lambda$, the right-hand side of the above display can upper bounded as

$$\frac{\delta'}{2s} + \Big(1 - \frac{\delta'}{2s}\Big)\mathbb{P}[\mathcal{N}(0,1) \geq \tau\lambda/\sqrt{k}] \leq \frac{\delta'}{2s} + \Big(1 - \frac{\delta'}{2s}\Big)\frac{\delta'}{2s-\delta'} \leq \frac{\delta'}{s}.$$

The above display gives us the desired control of the type two error, and we can conclude that $\mathbb{P}[S(\widehat{\boldsymbol{\mu}}^{\ell_1/\ell_\infty}) \neq S] \leq \alpha$.

Chapter 13

Feature Screening With Forward Regression

In this chapter, we propose a novel application of the Simultaneous Orthogonal Matching Pursuit (S-OMP) procedure for sparsistant variable selection in ultra-high dimensional multi-task regression problems. Screening of variables, as introduced in [62], is an efficient and highly scalable way to remove many irrelevant variables from the set of all variables, while retaining all the relevant variables. S-OMP can be applied to problems with hundreds of thousands of variables and once the number of variables is reduced to a manageable size, a more computationally demanding procedure can be used to identify the relevant variables for each of the regression outputs. To our knowledge, this is the first attempt to utilize relatedness of multiple outputs to perform fast screening of relevant variables. As our main theoretical contribution, we prove that, asymptotically, S-OMP is guaranteed to reduce an ultra-high number of variables to below the sample size without losing true relevant variables. We also provide formal evidence that a modified Bayesian information criterion (BIC) can be used to efficiently determine the number of iterations in S-OMP. We further provide empirical evidence on the benefit of variable selection using multiple regression outputs jointly, as opposed to performing variable selection for each output separately. The finite sample performance of S-OMP is demonstrated on extensive simulation studies, and on a genetic association mapping problem.

13.1 Introduction

Multiple output regression, also known as multi-task regression, with *ultra-high dimensional* inputs commonly arise in problems such as genome-wide association (GWA) mapping in genetics, or stock portfolio prediction in finance. For example, in a GWA mapping problem, the goal is to find a small set of relevant single-nucleotide polymorphisms (SNP) (*covariates, or inputs*) that account for variations of a large number of gene expression or clinical traits (*responses, or outputs*), through a response function that is often modeled via a regression. However, this is a very challenging problem for current statistical methods since the number of input variables is likely to reach millions, prohibiting even usage of scalable implementation of Lasso-like procedures for model selection, which are a convex relaxation of a combinatorial subset selection search.

Furthermore, the outputs in a typical multi-task regression problem are not independent of each other, therefore the discovery of truly relevant inputs has to take into consideration of potential joint effects induced by coupled responses. To appreciate this better, consider again the GWA example. Typically, genes in a biological pathway are co-expressed as a module and it is often assumed that a causal SNP affects multiple genes in one pathway, but not all of the genes in the pathway. In order to effectively reduce the dimensionality of the problem and to detect the causal SNPs, it is very important to look at how SNPs affect all genes in a biological pathway. Since the experimentally collected data is usually very noisy, regressing genes individually onto SNPs may not be sufficient to identify the relevant SNPs that are only weakly marginally correlated with each individual gene in a module. However, once the whole biological pathway is examined, it is much easier to find such causal SNPs. In this paper, we demonstrate that the Simultaneous Orthogonal Matching Pursuit (S-OMP) [174] can be used to quickly reduce the dimensionality of such problems, without losing any of the relevant variables.

From a computational point of view, as the dimensionality of the problem and the number of outputs increase, it can become intractable to solve the underlying convex programs commonly used to identify relevant variables in multi-task regression problems. Previous work by [120], [123] and [116], for example, do not scale well to settings when the number of variables exceeds $\gtrsim 10000$ and the number of outputs exceeds $\gtrsim 1000$, as in typical genome-wide association studies. Furthermore, since the estimation error of the regression coefficients depends on the number of variables in the problem, variable selection can improve convergence rates of estimation procedures. These concerns motivate us to propose and study the S-OMP as a fast way to remove irrelevant variables from an ultra-high dimensional space.

Formally, the GWA mapping problem, which we will use as an illustrative example both in here for model formulation and later for empirical experimental validation, can be cast as a variable selection problem in a multiple output regression model:

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{W} \tag{13.1}$$

where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_T] \in \mathbb{R}^{n \times T}$ is a matrix of outputs, whose column $\mathbf{y}_t$ is an n-vector for the t-th output (i.e., gene), $\mathbf{X} \in \mathbb{R}^{n \times p}$ is a random design matrix, of which each row $\mathbf{x}_i$ denotes a p-dimensional input, $\mathbf{B} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T] \in \mathbb{R}^{p \times T}$ is the matrix of regression coefficients and $\mathbf{W} = [\epsilon_1, \ldots, \epsilon_T] \in \mathbb{R}^{n \times T}$ is a matrix of IID random noise, independent of $\mathbf{X}$. Throughout the paper, we will assume that the columns of $\mathbf{B}$ are jointly sparse, as we precisely specify below. Note that if different columns of $\mathbf{B}$ do not share any underlying structure, the model in (13.1) can be estimated by fitting each of the tasks separately.

We are interested in estimating the regression coefficients, under the assumption that they share a common structure, for example, there exist a subset of variables with non-zero coefficients for more than one regression output. We informally refer to such outputs as related. Such a variable selection problem can be formalized in two ways: i) the *union support* recovery of $\mathbf{B}$, as defined in [144], where a subset of variables is selected that affect at least one output; ii) the *exact support* recovery of $\mathbf{B}$, where the exact positions of non-zero elements in $\mathbf{B}$ are estimated. In this paper, we concern ourselves with exact support recovery, which is of particular importance in problems like GWA mapping [115] or biological network estimation [147]. Under such a multi-task setting, two interesting questions naturally follow: i) how can information be

shared between related outputs in order to improve the predictive accuracy and the rate of convergence of the estimated regression coefficients over the independent estimation on each output separately; ii) how to select relevant variables more accurately based on information from related outputs. To address these two questions, one line of research [for example, 120, 123, 204] has looked into the following estimation procedure leveraging a *multi-task regularization*:

$$\widehat{\mathbf{B}} = \underset{\boldsymbol{\beta}_t \in \mathbb{R}^p, t \in [T]}{\operatorname{argmin}} \sum_{t=1}^{T} ||\mathbf{y}_t - \mathbf{X}\boldsymbol{\beta}_t||_2^2 + \lambda \sum_{j=1}^{p} \operatorname{pen}(\beta_{1,j}, \ldots, \beta_{T,j}), \tag{13.2}$$

with $\operatorname{pen}(a_1, \ldots, a_T) = \max_{t \in [T]} |a_t|$ or $\operatorname{pen}(a_1, \ldots, a_T) = \sqrt{\sum_{t \in [T]} a_t^2}$ for a vector $\mathbf{a} \in \mathbb{R}^T$. Under an appropriate choice of the penalty parameter λ, the estimator $\widehat{\mathbf{B}}$ has many rows equal to zero, which correspond to irrelevant variables. However, solving (13.2) can be computationally prohibitive.

In this chapter, we consider an ultra-high dimensional setting for the aforementioned multi-task regression problem, where the number of variables p is much higher than the sample size n, for example, $p = \mathcal{O}(\exp(n^{\delta_p}))$ for a positive constant δ_p, but the regression coefficients $\boldsymbol{\beta}_t$ are sparse, that is, for each task t, there exist a very small number of variables that are relevant to the output. Under the sparsity assumption, it is highly important to efficiently select the relevant variables in order to improve the accuracy of the estimation and prediction, and to facilitate the understanding of the underlying phenomenon for domain experts. In the seminal paper of [62], the concept of *sure screening* was introduced, which leads to a sequential variable selection procedure that keeps all the relevant variables with high probability in ultra-high dimensional *uni-output regression*. In this paper, we propose the S-OMP procedure, which enjoys *sure screening* property in ultra-high dimensional *multiple output regression* as defined in (13.1). To perform *exact support* recovery, we further propose a two-step procedure that first uses S-OMP to screen the variables, i.e., select a subset of variables that contain all the true variables; and then use the adaptive Lasso (ALasso) [97] to further select a subset of screened variables for each task. We show, both theoretically and empirically, that our procedure ensures sparsistant recovery of relevant variables. To the best of our knowledge, this is the first attempt to analyze the sure screening property in the ultra-high dimensional space using the shared information from the multiple regression outputs.

In this chapter, we make the following novel contributions: i) we prove that the S-OMP can be used for the ultra-high dimensional variable screening in multiple output regression problems and demonstrate its performance on extensive numerical studies; ii) we show that a two step procedure can be used to select exactly the relevant variables for each task; and iii) we prove that a modification of the BIC score [31] can be used to select the number of steps in the S-OMP.

13.2 Methodology

13.2.1 The model and notation

We will consider a slightly more general model

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{X}_1\boldsymbol{\beta}_1 + \boldsymbol{\epsilon}_1 \\ \mathbf{y}_2 &= \mathbf{X}_2\boldsymbol{\beta}_2 + \boldsymbol{\epsilon}_2 \\ &\dots \\ \mathbf{y}_T &= \mathbf{X}_T\boldsymbol{\beta}_T + \boldsymbol{\epsilon}_T, \end{aligned} \tag{13.3}$$

than the one given in (13.1). The model in (13.1) is a special case of the model in (13.3), with all the design matrices $\{\mathbf{X}_t\}_{t\in[T]}$ being equal and $[T]$ denoting the set $\{1,\dots,T\}$. Assume that for all $t \in [T]$, $\mathbf{X}_t \in \mathbb{R}^{n\times p}$. For the design $\mathbf{X}_t$, we denote $\mathbf{X}_{t,j}$ the j-th column (i.e., dimension), $\mathbf{x}_{t,i}$ the i-th row (i.e., instance) and $x_{t,ij}$ the element at (i,j). Denote $\boldsymbol{\Sigma}_t = \mathrm{Cov}(\mathbf{x}_{t,i})$. Without loss of generality, we assume that $\mathrm{Var}(y_{t,i}) = 1$, $\mathbb{E}(x_{t,ij}) = 0$ and $\mathrm{Var}(x_{t,ij}) = 1$. The noise $\boldsymbol{\epsilon}_t$ is zero mean and $\mathrm{Cov}(\boldsymbol{\epsilon}_t) = \sigma^2\mathbf{I}_{n\times n}$. We assume that the number of variables $p \gg n$ and that the vector of regression coefficients $\boldsymbol{\beta}_t$ are jointly sparse, that is, there exist a small number of variables that are relevant for the most of the T regression problems. Put another way, the matrix $B = [\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_T]$ has only a small number of non-zero rows. Let $\mathcal{M}_{*,t}$ denote the set of non-zero coefficients of $\boldsymbol{\beta}_t$ and $\mathcal{M}_* = \cup_{t=1}^T \mathcal{M}_{*,t}$ denote the set of all relevant variables, i.e., variables with non-zero coefficient in at least one of the tasks. For an arbitrary set $\mathcal{M} \subseteq \{1,\dots,p\}$, $\mathbf{X}_{t,\mathcal{M}}$ denotes the design with columns indexed by $\mathcal{M}$, $\mathbf{B}_{\mathcal{M}}$ denotes the rows of $\mathbf{B}$ indexed by $\mathcal{M}$ and $\mathbf{B}_j = (\boldsymbol{\beta}_{1,j},\dots,\boldsymbol{\beta}_{T,j})'$. The cardinality of the set $\mathcal{M}$ is denoted as $|\mathcal{M}|$. Let $s := |\mathcal{M}_*|$ denote the total number of relevant variables, so under the sparsity assumption we have $s < n$. For a matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{p\times T}$, we define $||\mathbf{A}||_{2,1} := \sum_{i\in[p]} \sqrt{\sum_{j\in[T]} a_{ij}^2}$.

13.2.2 Simultaneous Orthogonal Matching Pursuit

We propose a Simultaneous Orthogonal Matching Pursuit procedure for ultra high-dimensional variable selection in the multi-task regression problem, which is outlined in Algorithm 4. Before describing the algorithm, we introduce some additional notation. For an arbitrary subset $\mathcal{M} \subseteq [p]$ of variables, let $\mathbf{H}_{t,\mathcal{M}}$ be the orthogonal projection matrix onto $\mathrm{Span}(\mathbf{X}_{t,\mathcal{M}})$, i.e.,

$$\mathbf{H}_{t,\mathcal{M}} = \mathbf{X}_{t,\mathcal{M}}(\mathbf{X}'_{t,\mathcal{M}}\mathbf{X}_{t,\mathcal{M}})^{-1}\mathbf{X}'_{t,\mathcal{M}},$$

and define the residual sum of squares (RSS) as

$$\mathrm{RSS}(\mathcal{M}) = \sum_{t=1}^{T} \mathbf{y}'_t(\mathbf{I}_{n\times n} - \mathbf{H}_{\mathcal{M}})\mathbf{y}_t.$$

The algorithm starts with an empty set $\mathcal{M}^{(0)} = \emptyset$. We recursively define the set $\mathcal{M}^{(k)}$ based on the set $\mathcal{M}^{(k-1)}$. The set $\mathcal{M}^{(k)}$ is obtained by adding a variable indexed by $\widehat{f}_k \in [p]$, which minimizes $\mathrm{RSS}(\mathcal{M}^{(k-1)} \cup j)$ over the set $[p]\backslash\mathcal{M}^{(k-1)}$, to the set $\mathcal{M}^{(k-1)}$. Repeating the algorithm for $n-1$ steps, a sequence of nested sets $\{\mathcal{M}^{(k)}\}_{k=0}^{n-1}$ is obtained, with $\mathcal{M}^{(k)} = \{\widehat{f}_1,\dots,\widehat{f}_k\}$.

Input: Dataset $\{\mathbf{X}_t, \mathbf{y}_t\}_{t=1}^T$
Output: Sequence of selected models $\{\mathcal{M}^{(k)}\}_{k=0}^{n-1}$

Set $\mathcal{M}^{(0)} = \emptyset$
for $k = 1$ *to* $n-1$ **do**
  for $j = 1$ *to* p **do**
    $\widetilde{\mathcal{M}}_j^{(k)} = \mathcal{M}^{(k-1)} \cup \{j\}$
    $\mathbf{H}_{t,j} = \mathbf{X}_{t,\widetilde{\mathcal{M}}_j^{(k)}} (\mathbf{X}'_{t,\widetilde{\mathcal{M}}_j^{(k)}} \mathbf{X}_{t,\widetilde{\mathcal{M}}_j^{(k)}})^{-1} \mathbf{X}'_{t,\widetilde{\mathcal{M}}_j^{(k)}}$
    $\mathrm{RSS}(\widetilde{\mathcal{M}}_j^{(k)}) = \sum_{t=1}^T \mathbf{y}'_t (\mathbf{I}_{n\times n} - \mathbf{H}_{t,j}) \mathbf{y}_t$
  end
  $\widehat{f}_k = \mathrm{argmin}_{j \in \{1,\ldots,p\} \setminus \mathcal{M}^{(k-1)}} \mathrm{RSS}(\widetilde{\mathcal{M}}_j^{(k)})$
  $\mathcal{M}^{(k)} = \mathcal{M}^{(k-1)} \cup \{\widehat{f}_k\}$
end

Algorithm 4: Group Forward Regression

To practically select one of the sets of variables from $\{\mathcal{M}^{(k)}\}_{k=0}^{n-1}$, we minimize the modified BIC criterion [31], which is defined as

$$\mathrm{BIC}(\mathcal{M}) = \log\left(\frac{\mathrm{RSS}(\mathcal{M})}{nT}\right) + \frac{|\mathcal{M}|(\log(n) + 2\log(p))}{n} \tag{13.4}$$

with $|\mathcal{M}|$ denoting the number of elements of the set $\mathcal{M}$. Let

$$\widehat{s} = \underset{k \in \{0,\ldots,n-1\}}{\mathrm{argmin}} \ \mathrm{BIC}(\mathcal{M}^{(k)}),$$

so that the selected model is $\mathcal{M}^{(\widehat{s})}$.

The S-OMP algorithm is outlined only conceptually in this section. The steps 5 and 6 of the algorithm can be implemented efficiently using the progressive Cholesky decomposition (see, for example, [34]). A computationally costly step 5 involves inversion of the matrix $\mathbf{X}'_{t,\mathcal{M}}\mathbf{X}_{t,\mathcal{M}}$, however, it can be seen from the algorithm that the matrix $\mathbf{X}'_{t,\mathcal{M}}\mathbf{X}_{t,\mathcal{M}}$ is updated in each iteration by simply appending a row and a column to it. Therefore, its Cholesky factorization can be efficiently computed based on calculation involving only the last row. A detailed implementation of the orthogonal matching pursuit algorithm based on the progressive Cholesky decomposition can be found in [153].

13.2.3 Exact variable selection

After many of the irrelevant variables have been removed using Algorithm 4, we are left with the variables in the set $\mathcal{M}^{(\widehat{s})}$, whose size is smaller than the sample size n. These variables are candidates for the relevant variables for each of the regressions. Now, we can address the problem of estimating the regression coefficients and recovering the exact support of $\mathbf{B}$ using

a lower dimensional selection procedure. In this paper, we use the adaptive Lasso as a lower dimensional selection procedure, which was shown to have oracle properties [97]. The ALasso solves the penalized least square problem

$$\widehat{\boldsymbol{\beta}}_t = \underset{\boldsymbol{\beta}_t \in \mathbb{R}^{\widehat{s}}}{\operatorname{argmin}} ||\mathbf{y}_t - \mathbf{X}_{t,\mathcal{M}^{(\widehat{s})}} \boldsymbol{\beta}_t||_2^2 + \lambda \sum_{j \in \mathcal{M}^{(\widehat{s})}} w_j |\beta_{t,j}|,$$

where $(w_j)_{j \in \mathcal{M}^{(\widehat{s})}}$ is a vector of known weight and λ is a tuning parameter. Usually, the weights are defined as $w_j = 1/|\widehat{\beta}_{t,j}|$ where $\widehat{\boldsymbol{\beta}}_t$ is a $\sqrt{n}$-consistent estimator of $\boldsymbol{\beta}_t$. In a low dimensional setting, we know from [89] that the adaptive Lasso can recover exactly the relevant variables. Therefore, we can use the ALasso on each output separately to recover the exact support of $\mathbf{B}$. However, in order to ensure that the exact support of $\mathbf{B}$ is recovered with high probability, we need to ensure that the total number of tasks is $o(n)$. The exact support recovery of $\mathbf{B}$ is established using the union bound over different tasks, therefore we need the number of tasks to remain relatively small in comparison to the sample size n. However, simulation results presented in Section 13.4.1 show that the ALasso procedure succeeds in the exact support recovery even when the number of tasks are much larger than the sample size, which indicates that our theoretical considerations could be improved. Figure 13.1 illustrates the two step procedure.

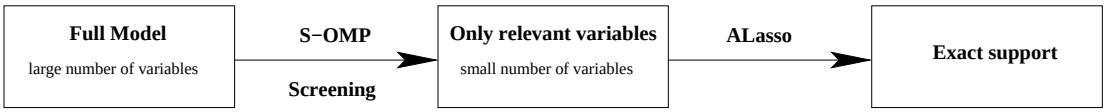


Figure 13.1: Framework for exact support recovery

We point out that solving the multi-task problem defined in (13.2) can be efficiently done on the reduced set of variables, but it is not obvious how to obtain the estimate of the exact support using (13.2). In Section 13.4.1, our numerical studies show that the ALasso applied to the reduced set of variables can be used to estimate the exact support of $\mathbf{B}$.

13.3 Theory

In this section, we state conditions under which Algorithm 4 is screening consistent, i.e.,

$$\mathbb{P}[\exists k \in \{0, 1, \ldots, n-1\} : \mathcal{M}_* \subseteq \mathcal{M}^{(k)}] \to 1, \text{ as } n \to \infty.$$

Furthermore, we also show that the model selected using the modified BIC criterion contains all the relevant variables, i.e.,

$$\mathbb{P}[\mathcal{M}_* \subseteq \mathcal{M}^{(\widehat{s})}] \to 1, \text{ as } n \to \infty.$$

Note that we can choose trivially $\mathcal{M}^{(n)}$ since it holds that $\mathcal{M}_* \subseteq \mathcal{M}^{(n)}$. However, we will be able to prove that $\widehat{s}$ chosen by the modified BIC criterion is much smaller than the sample size n.

13.3.1 Assumptions

Before we state the theorem characterizing the performance of the S-OMP, we give some technical conditions that are needed for our analysis.

A1: The random noise vectors $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T$ are independent Gaussian with zero mean and covariance matrix $\sigma^2 \mathbf{I}_{n \times n}$.

A2: Each row of the design matrix $\mathbf{X}_t$ is IID Gaussian with zero mean and covariance matrix $\boldsymbol{\Sigma}_t$. Furthermore, there exist two positive constants $0 < \phi_{\min} < \phi_{\max} < \infty$ such that

$$\phi_{\min} \leq \min_{t \in [T]} \Lambda_{\min}(\boldsymbol{\Sigma}_t) \leq \max_{t \in [T]} \Lambda_{\max}(\boldsymbol{\Sigma}_t) \leq \phi_{\max}.$$

A3: The true regression coefficients are bounded, i.e., there exists a positive constant $C_{\boldsymbol{\beta}}$ such that $||\mathbf{B}||_{2,1} \leq C_{\boldsymbol{\beta}}$. Furthermore, the norm of any non-zero row of the matrix $\mathbf{B}$ is bounded away from zero, that is, there exist positive constants $c_{\boldsymbol{\beta}}$ and $\delta_{\min}$ such that

$$T^{-1} \min_{j \in \mathcal{M}_*} \sum_{t \in [T]} \beta_{t,j}^2 \geq c_{\boldsymbol{\beta}} n^{-\delta_{\min}}.$$

A4: There exist positive constants C_s, C_p, δ_s and δ_p such that $|\mathcal{M}_*| \leq C_s n^{\delta_s}$ and $\log(p) \leq C_p n^{\delta_p}$.

The normality condition **A1** is assumed here only to facilitate presentation of theoretical results, as is commonly assumed in literature [see, for example, 62, 199]. The normality assumption can be avoided at the cost of more technical proofs (see, for example, [123]). Under the condition **A2**, we will be able to show that the empirical covariance matrix satisfies the sparse eigenvalue condition with probability tending to one. The assumption that the rows of the design are Gaussian can be easily relaxed to the case when the rows are sub-Gaussian, without any technical difficulties in proofs, since we would still obtain exponential bounds on the tail probabilities. The condition **A3** states that the regression coefficients are bounded, which is a technical condition likely to be satisfied in practice. Furthermore, it is assumed that the row norms of $\mathbf{B}_{\mathcal{M}_*}$ do not decay to zero too fast or, otherwise, they would not be distinguishable from noise. If every non-zero coefficient is bounded away from zero by a constant, the condition **A3** is trivially satisfied with $\delta_{\min} = 0$. However, we allow the coefficients of the relevant variables to get smaller as the sample size increases and still guarantee that the relevant variable will be identified, which suggests that the condition is not too restrictive. The condition **A4** sets the upper bound on the number of relevant variables and the total number of variables. While the total number of variables can diverge to infinity much faster than the sample size, the number of relevant variables needs to be smaller than the sample size. Conditions **A3** and **A4** implicitly relate different outputs and control the number of non-zero coefficients shared between different outputs. Intuitively, if the upper bound in **A4** on the size of $\mathcal{M}_*$ is large, this immediately implies that the constant C_β in **A3** should be large as well, since otherwise there would exist a row of $\mathbf{B}$ whose ℓ_2 norm would be too small to be detected by Algorithm 4.

13.3.2 Screening consistency

Our first result states that after a small number of iterations, compared to the dimensionality p, the S-OMP procedure will include all the relevant variables.

Theorem 13.1. *Assume the model in* (13.3) *and that the conditions* **A1**-**A4** *are satisfied. Furthermore, assume that*

$$\frac{n^{1-6\delta_s-6\delta_{\min}}}{\max\{\log(p),\log(T)\}} \to \infty, \text{ as } n \to \infty.$$

Then there exists a number $m^*_{\max} = m^*_{\max}(n)$, so that in $m^*_{\max}$ steps of S-OMP iteration, all the relevant variables are included in the model, that is, as $n \to \infty$

$$\mathbb{P}[\mathcal{M}_* \subseteq \mathcal{M}^{(m^*_{\max})}] \geq 1 - C_1 \exp\left(-C_2 \frac{n^{1-6\delta_s-6\delta_{\min}}}{\max\{\log(p),\log(T)\}}\right),$$

for some positive constants C_1 and C_2. The exact value of $m^*_{\max}$ is given as

$$m^*_{\max} = \lfloor 2^4 \phi^{-2}_{\min} \phi_{\max} C^2_{\boldsymbol{\beta}} C^2_s c^{-2}_{\boldsymbol{\beta}} n^{2\delta_s+2\delta_{\min}} \rfloor. \tag{13.5}$$

Under the assumptions of Theorem 13.1, $m^*_{\max} \leq n-1$, so that the procedure effectively reduces the dimensionality below the sample size. From the proof of the theorem, it is clear how multiple outputs help to identify the relevant variables. The crucial quantity in identifying all relevant variables is the minimum non-zero row norm of $\mathbf{B}$, which allows us to identify weak variables if they are relevant for a large number of outputs even though individual coefficients may be small. It should be noted that the main improvement over the ordinary forward regression is in the size of the signal that can be detected, as defined in **A3** and **A4**.

Theorem 13.1 guarantees that one of the sets $\{\mathcal{M}^{(k)}\}$ will contain all relevant variables, with high probability. However, it is of practical importance to select one set in the collection that contains all relevant variables and does not have too many irrelevant ones. Our following theorem shows that the modified BIC criterion can be used for this purpose, that is, the set $\mathcal{M}^{(\widehat{s})}$ is screening consistent.

Theorem 13.2. *Assume that the conditions of Theorem 13.1 are satisfied. Let*

$$\widehat{s} = \underset{k \in \{0,\ldots,n-1\}}{\operatorname{argmin}} \ \mathrm{BIC}(\mathcal{M}^{(k)})$$

be the index of the model selected by optimizing the modified BIC criterion. Then, as $n \to \infty$

$$\mathbb{P}[\mathcal{M}_* \subseteq \mathcal{M}^{(\widehat{s})}] \to 1.$$

Combining results from Theorem 13.1 and Theorem 13.2, we have shown that the S-OMP procedure is screening consistent and can be applied to problems where the dimensionality of the problem p is exponential in the number of observed samples. In the next section, we also show that the S-OMP has great empirical performance.

13.4 Numerical studies

In this section, we perform simulation studies on an extensive number of synthetic data sets. Furthermore, we demonstrate the application of the procedure on the genome-wide association mapping problem.

13.4.1 Simulation studies

We conduct an extensive number of numerical studies to evaluate the finite sample performance of the S-OMP. We consider three procedures that perform estimation on individuals outputs: Sure Independence Screening (SIS), Iterative SIS (ISIS) [62], and the OMP, for comparison purposes. The evaluation is done on the model in (13.1). SIS and ISIS are used to select a subset of variables and then the ALasso is used to further refine the selection. We denote this combination as SIS-ALasso and ISIS-ALasso. The size of the model selected by SIS is fixed as $n-1$, while the ISIS selects $\lfloor n/\log(n) \rfloor$ variables in each of the $\lfloor \log(n) - 1 \rfloor$ iterations. From the screened variables, the final model is selected using the ALasso, together with the BIC criterion (13.4) to determine the penalty parameter λ. The number of variables selected by the OMP is determined using the BIC criterion, however, we do not further refine the selected variables using the ALasso, since from the numerical studies in [182] it was observed that the further refinement does not result in improvement. The S-OMP is used to reduce the dimensionality below the sample size jointly using the regression outputs. Next, the ALasso is used on each of the outputs to further perform the estimation. This combination is denoted SOMP-ALasso.

Let $\widehat{\mathbf{B}} = [\widehat{\boldsymbol{\beta}}_1, \ldots, \widehat{\boldsymbol{\beta}}_T] \in \mathbb{R}^{p \times T}$ be an estimate obtained by one of the estimation procedures. We evaluate the performance averaged over 200 simulation runs. Let $\widehat{\mathbb{E}}_n$ denote the empirical average over the simulation runs. We measure the size of the union support $\widehat{S} = S(\widehat{\mathbf{B}}) := \{j \in [p] : ||\widehat{\mathbf{B}}_j||_2^2 > 0\}$. Next, we estimate the probability that the screening property is satisfied $\widehat{\mathbb{E}}_n[\mathbb{I}\{\mathcal{M}_* \subseteq S(\widehat{\mathbf{B}})\}]$, which we call coverage probability. For the union support, we define fraction of correct zeros $(p-s)^{-1}\widehat{\mathbb{E}}_n[|S(\widehat{\mathbf{B}})^C \cap \mathcal{M}_*^C|]$, fraction of incorrect zeros $s^{-1}\widehat{\mathbb{E}}_n[|S(\widehat{\mathbf{B}})^C \cap \mathcal{M}_*|]$ and fraction of correctly fitted $\widehat{\mathbb{E}}_n[\mathbb{I}\{\mathcal{M}_* = S(\widehat{\mathbf{B}})\}]$ to measure the performance of different procedures. Similar quantities are defined for the exact support recovery. In addition, we measure the estimation error $\widehat{\mathbb{E}}_n[||\mathbf{B} - \widehat{\mathbf{B}}||_2^2]$ and the prediction performance on the test set. On the test data $\{\mathbf{x}_i^*, \mathbf{y}_i^*\}_{i \in [n]}$, we compute

$$R^2 = 1 - \frac{\sum_{i \in [n]} \sum_{t \in [T]} (y_{t,i}^* - (\mathbf{x}_{t,i}^*)' \widehat{\boldsymbol{\beta}}_t)^2}{\sum_{i \in [n]} \sum_{t \in [T]} (y_{t,i}^* - \overline{y_t^*})^2},$$

where $\overline{y_t^*} = n^{-1} \sum_{i \in [n]} y_{t,i}$.

The following simulation studies are used to comparatively assess the numerical performance of the procedures. Due to space constraints, tables with detailed numerical results are given in the Appendix. In this section, we outline main findings.

Simulation 1: [Model with uncorrelated variables] The following toy model is based on the simulation I in [62] with $(n, p, s, T) = (400, 20000, 18, 500)$. Each $\mathbf{x}_i$ is drawn independently from a standard multivariate normal distribution, so that the variables are mutually independent. For $j \in [s]$ and $t \in [T]$, the non-zero coefficients of $\mathbf{B}$ are given as $\beta_{t,j} = (-1)^u (4n^{-1/2} \log n + |z|)$, where $u \sim \text{Bernoulli}(0.4)$ and $z \sim \mathcal{N}(0,1)$. The number of non-zero elements in $\mathbf{B}_j$ is given as a parameter $T_{\text{non-zero}} \in \{500, 300, 100\}$. The positions of non-zero elements are chosen uniformly at random from $[T]$. The noise is Gaussian with the standard deviation σ set to control the signal-to-noise ratio (SNR). SNR is defined as $\text{Var}(\mathbf{x}\boldsymbol{\beta})/\text{Var}(\boldsymbol{\epsilon})$ and we vary $\text{SNR} \in \{15, 10, 5, 1\}$.

Simulation 2: [Changing the number of non-zero elements in $\mathbf{B}_j$] The following model is used to evaluate the performance of the methods as the number of non-zero elements in a row of $\mathbf{B}$ varies. We set $(n, p, s) = (100, 500, 10)$ and vary the number of outputs $T \in \{500, 750, 1000\}$. For each number of outputs T, we vary $T_{\text{non-zero}} \in \{0.8T, 0.5T, 0.2T\}$. The samples $\mathbf{x}_i$ and regression coefficients $\mathbf{B}$ are given as in Simulation 1, that is, $\mathbf{x}_i$ is drawn from a multivariate standard normal distribution and the non-zero coefficients $\mathbf{B}$ are given as $\beta_{t,j} = (-1)^u(4n^{-1/2}\log n + |z|)$, where $u \sim \text{Bernoulli}(0.4)$ and $z \sim \mathcal{N}(0, 1)$. The noise is Gaussian, with the standard deviation defined through the SNR, which varies in $\{10, 5, 1\}$.

Simulation 3: [Model with the decaying correlation between variables] The following model is borrowed from [182]. We assume a correlation structure between variables given as

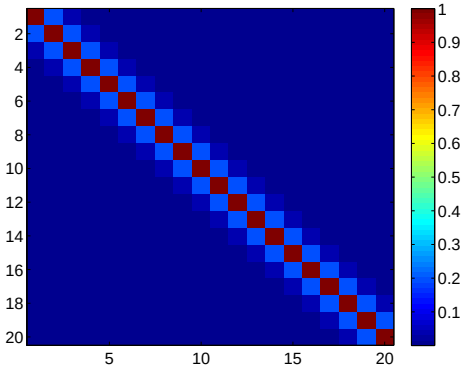
$$\text{Var}(\mathbf{X}_{j_1}, \mathbf{X}_{j_2}) = \rho^{|j_1 - j_2|},$$

where $\rho \in \{0.2, 0.5, 0.7\}$. This correlation structure appears naturally among ordered variables. We set $(n, p, s, T) = (100, 5000, 3, 150)$ and $T_{\text{non-zero}} = 80$. The relevant variables are at positions $(1, 4, 7)$ and non-zero coefficients are given as $3, 1.5$ and 2 respectively. The SNR varies in $\{10, 5, 1\}$. A heat map of the correlation matrix between different covariates is given in Figure 13.2.

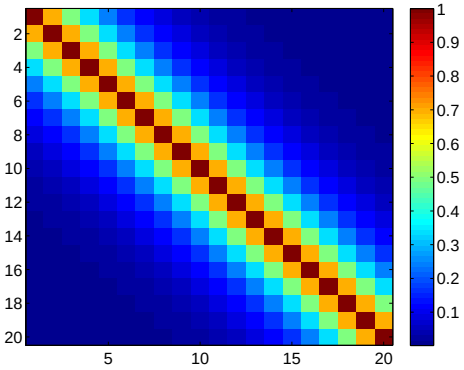
Simulation 4: [Model with the block-compound correlation structure] The following model assumes a block compound correlation structure. For a parameter ρ, the correlation between two variables $\mathbf{X}_{j_1}$ and $\mathbf{X}_{j_2}$ is given as ρ, ρ^2 or ρ^3 when $|j_1 - j_2| \leq 10$, $|j_1 - j_2| \in (10, 20]$ or $|j_1 - j_2| \in (20, 30]$ and is set to 0 otherwise. We set $(n, p, s, T) = (150, 4000, 8, 150)$, $T_{\text{non-zero}} = 80$ and the parameter $\rho \in \{0.2, 0.5\}$. The relevant variables are located at positions 1, 11, 21, 31, 41, 51, 61, 71 and 81, so that each block of highly correlated variables has exactly one relevant variable. The values of relevant coefficients are given in Simulation 1. The noise is Gaussian and the SNR varies in $\{10, 5, 1\}$. A heat map of the correlation matrix between different covariates is shown in Figure 13.3.

Simulation 5: [Model with a 'masked' relevant variable] This model represents a difficult setting. It is modified from [182]. We set $(n, p, s, T) = (200, 10000, 5, 500)$. The number of non-zero elements in each row varies is $T_{\text{non-zero}} \in \{400, 250, 100\}$. For $j \in [s]$ and $t \in [T]$, the non-zero elements equal $\beta_{t,j} = 2j$. Each row of $\mathbf{X}$ is generated as follows. Draw independently $\mathbf{z}_i$ and $\mathbf{z}'_i$ from a p-dimensional standard multivariate normal distribution. Now, $x_{ij} = (z_{ij} + z'_{ij})/\sqrt{(2)}$ for $j \in [s]$ and $x_{ij} = (z_{ij} + \sum_{j' \in [s]} z_{ij'})/2$ for $j \in [p] \backslash [s]$. Now, $\text{Corr}(x_{i,1}, y_{t,i})$ is much smaller then $\text{Corr}(x_{i,j}, y_{t,i})$ for $j \in [p] \backslash [s]$, so that it becomes difficult to select variable 1. The variable 1 is 'masked' with the noisy variables. This setting is difficult for screening procedures as they take into consideration only marginal information. The noise is Gaussian with standard deviation $\sigma \in \{1.5, 2.5, 4.5\}$.

In the next section, we summarize results of our experimental findings. Our simulation setting transitions from a simple scenario considered in Simulation 1 towards a challenging one in Simulation 5. Simulation 1 is adopted from [62] as a toy model on which all algorithms should work well. Simulation 2 examines the influence of the number of non-zero elements in a relevant row of the matrix $\mathbf{B}$. We expect that Algorithm 4 will outperform procedures that perform estimation on individual outputs when $T_{\text{non-zero}}$ is large, while when $T_{\text{non-zero}}$ is small the single-task screening procedures should have an advantage. Our intuition is also supported by recent results of [110]. Simulations 3 and 4 represent more challenging situations with structured

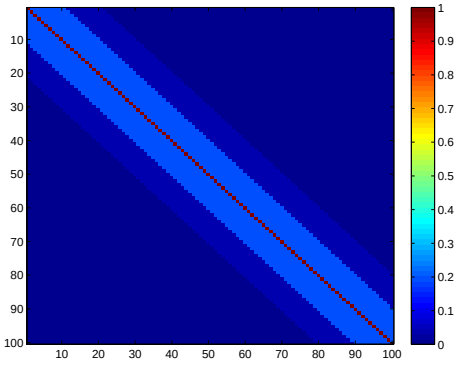


(a) $\rho = 0.2$

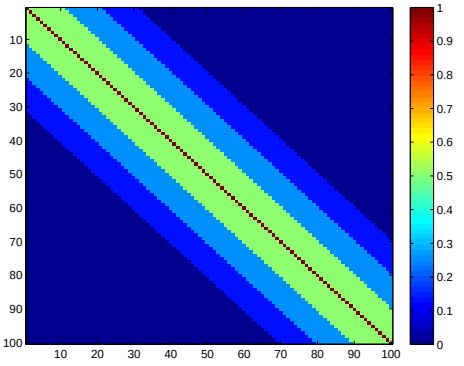


(b) $\rho = 0.7$

Figure 13.2: Visualization of the correlation matrix in Simulation 3. Only an upper left corner is presented corresponding to 20 of the 5000 variables.



(a) $\rho = 0.2$



(b) $\rho = 0.5$

Figure 13.3: Visualization of the correlation matrix in Simulation 4. Only an upper left corner is presented corresponding to 100 of the 4000 variables.

correlation that naturally appears in many data sets, for example, a correlation between gene measurements that are closely located on a chromosome. Finally Simulation 5 is constructed in such a way such that procedures which use only marginal information are going to include irrelevant variables before relevant ones.

13.4.2 Results of simulations

Tables giving detailed results of the above described simulations are given in [104]. In this section, we outline main findings and reproduce some parts of the tables that we think are insightful.

Table 13.1 shows parts of the results for simulation 1. We can see that all methods perform well in the setting when the input variables are mutually uncorrelated and the SNR is high. Note that even though the variables are uncorrelated, the sample correlation between variables can be quite high due to large p and small n, which can result in selection of spurious variables. As we can see from the table, comparing to SIS, ISIS and OMP, the S-OMP is able to select the correct union support, while the procedures that select variables based on different outputs separately also include additional spurious variables into the selection. Furthermore, we can see that the S-OMP-ALasso procedure does much better on the problem of exact support recovery compared to the other procedures. The first simulations suggests that somewhat higher computational cost of the S-OMP procedure can be justified by the improved performance on the problem of union and exact support recovery as well as on the error in the estimated coefficients.

Table 13.2 shows parts of the results for simulation 2. In this simulation, we measured the performance of estimation procedures as the amount of shared input variables between different outputs varies. The parameter $T_{\mathrm{non-zero}}$ controls the amount of information that is shared between different tasks as defined in the previous subsection. In particular, the parameter controls the number of non-zero elements in a row of the matrix $\mathbf{B}$ corresponding to a relevant variable. When the number of non-zero elements is high, a variable is relevant to many tasks and we say that outputs overlap. In this setting, the S-OMP procedure is expected to outperform the other methods, however, when $T_{\mathrm{non-zero}}$ is low, the noise coming from the tasks for which the variable is irrelevant can actually harm the performance. The table shows results when the overlap of shared variables is small, that is, a relevant variable is only relevant for 10% of outputs. As one would expect, the S-OMP procedure does as well as other procedures. This is not surprising since the amount of shared information between different outputs is limited. Therefore, if one expects little variable sharing across different outputs, using the SIS or ISIS may result in similar accuracy, but an improved computational efficiency. It is worth pointing out that in our simulations, the different tasks are correlated since the same design $\mathbf{X}$ is used for all tasks. However, we expect the same qualitative results even under the model given in equation (13.3) where different tasks can have different designs $\mathbf{X}_t$ and the outputs are uncorrelated.

Simulation 3 represents a situation that commonly occurs in nature, where there is an ordering among input variables and the correlation between variables decays as the distance between variables increases. The model in simulation 4 is a modification of the model in simulation 3 where the variables are grouped and there is some correlation between different groups. Table 13.3 gives results for simulation 3 for the parameter $\rho = 0.5$. In this setting, the S-OMP performs much better that the other procedures. The improvement becomes more pronounced with increase of the correlation parameter ρ. Similar behavior is observed in simulation 4 as

well, see table 13.4. Results of simulation 5, given in Table 13.5, further reinforce our intuition that the S-OMP procedure does well even on problems with high-correlation between the set of relevant input variables and the set of irrelevant ones.

To further compare the performance of the S-OMP procedure to the SIS, we explore the minimum number of iterations needed for the algorithm to include all the relevant variables into the selected model. From our limited numerical experience, we note that the simulation parameters do not affect the number of iterations for the S-OMP procedure. This is unlike the SIS procedure, which occasionally requires a large number of steps before all the true variables are included, see Figure 3 in [62]. We note that while the S-OMP procedure does include, in many cases, all the relevant variables before the irrelevant ones, the BIC criterion is not able to correctly select the number of variables to include when the SNR is small. As a result, we can see the drop in performance as the SNR decreases.

13.4.3 Real data analysis

We demonstrate an application of the S-OMP to a genome-wide association mapping problem. The data were collected by our collaborator Judie Howrylak, M.D. at Harvard Medical School from 200 individuals that are suffering from asthma. For each individual, we have a collection of about $\sim$350,000 genetic markers[1], which are called single nucleotide polymorphisms (SNPs), and a collection of 1,424 gene expression measurements. The goal of this study is to identify a small number of SNPs that can help explain variations in gene expressions. Typically, this type of analysis is done by regressing each gene individually on the measured SNPs, however, since the data are very noisy, such an approach results in selecting many variables. Our approach to this problem is to regress a group of genes onto the SNPs instead. There has been some previous work on this problem [115], that considered regressing groups of genes onto SNPs, however, those approaches use variants of the estimation procedure given in Eq. (13.2), which is not easily scalable to the data we analyze here.

We use the spectral relaxation of the k-means clustering [200] to group 1424 genes into 48 clusters according to their expression values, so that the minimum, maximum and median number of genes per cluster is 4, 90 and 19, respectively. The number of clusters was chosen somewhat arbitrarily, based on the domain knowledge of the medical experts. The main idea behind the clustering is that we want to identify genes that belong to the same regulatory pathway since they are more likely to be affected with the same SNPs. Instead of clustering, one may use prior knowledge to identify interesting groups of genes. Next, we want to use the S-OMP procedure to identify relevant SNPs for each of the gene clusters. Since we do not have the ground truth for the data set, we use predictive power on the test set and the size of estimated models to access their quality. We randomly split the data into a training set of size 170 and a testing set of size 30 and report results over 500 runs. We compute the R^2 coefficient on the test set defined as $1 - 30^{-1}T^{-1}\sum_{t\in[T]} ||\mathbf{y}_{t,\text{test}} - \mathbf{X}_{t,\text{test}}\widehat{\boldsymbol{\beta}}_t||_2^2$ (because the data have been normalized).

[1]These markers were preprocessed, by imputing missing values and removing duplicate SNPs that were perfectly correlated with other SNPs.

Table 13.1: Results for simulation 1 with parameters $(n, p, s, T) = (500, 20000, 18, 500)$, $T_{\text{non-zero}} = 500$

	Method name	Prob. (%) of $\mathcal{M}_* \subseteq \widehat{S}$	Fraction (%) of Correct zeros	Fraction (%) of Incorrect zeros	Fraction (%) of $\mathcal{M}_* = \widehat{S}$	$\lvert\widehat{S}\rvert$	Est. error $\lvert\lvert\mathbf{B} - \widehat{\mathbf{B}}\rvert\rvert_2^2$	Test error R^2
				SNR = 15				
Union Support	SIS-ALASSO	100.0	100.0	0.0	10.0	20.2	-	-
	ISIS-ALASSO	100.0	100.0	0.0	18.0	19.6	-	-
	OMP	100.0	100.0	0.0	0.0	23.9	-	-
	S-OMP	100.0	100.0	0.0	100.0	18.0	-	-
	S-OMP-ALASSO	100.0	100.0	0.0	100.0	18.0	-	-
Exact Support	SIS-ALASSO	0.0	100.0	0.7	0.0	8940.5	0.97	0.93
	ISIS-ALASSO	100.0	100.0	0.0	18.0	9001.6	0.33	0.93
	OMP	100.0	100.0	0.0	0.0	9005.9	0.20	0.93
	S-OMP-ALASSO	100.0	100.0	0.0	100.0	9000.0	0.20	0.93

Table 13.2: Results for simulation 2 with parameters $(n, p, s, T) = (200, 5000, 10, 1000)$, $T_{\text{non-zero}} = 200$

	Method name	Prob. (%) of $\mathcal{M}_* \subseteq \widehat{S}$	Fraction (%) of Correct zeros	Fraction (%) of Incorrect zeros	Fraction (%) of $\mathcal{M}_* = \widehat{S}$	$\lvert\widehat{S}\rvert$	Est. error $\lvert\lvert\mathbf{B} - \widehat{\mathbf{B}}\rvert\rvert_2^2$	Test error R^2
				SNR = 5				
Union Support	SIS-ALASSO	100.0	100.0	0.0	100.0	10.0	-	-
	ISIS-ALASSO	100.0	100.0	0.0	100.0	10.0	-	-
	OMP	100.0	97.4	0.0	0.0	139.6	-	-
	S-OMP	100.0	100.0	0.0	100.0	10.0	-	-
	S-OMP-ALASSO	100.0	100.0	0.0	100.0	10.0	-	-
Exact Support	SIS-ALASSO	100.0	100.0	0.0	100.0	2000.0	0.04	0.72
	ISIS-ALASSO	100.0	100.0	0.0	100.0	2000.0	0.04	0.72
	OMP	100.0	100.0	0.0	0.0	2131.6	0.05	0.71
	S-OMP-ALASSO	100.0	100.0	0.0	100.0	2000.0	0.03	0.72

Table 13.3: Results for simulation 3 with parameters $(n, p, s, T) = (100, 5000, 3, 150)$, $T_{\text{non-zero}} = 80$, $\rho = 0.5$

	Method name	Prob. (%) of $\mathcal{M}_* \subseteq \widehat{S}$	Fraction (%) of Correct zeros	Fraction (%) of Incorrect zeros	Fraction (%) of $\mathcal{M}_* = \widehat{S}$	$\lvert\widehat{S}\rvert$	Est. error $\lvert\lvert\mathbf{B} - \widehat{\mathbf{B}}\rvert\rvert_2^2$	Test error R^2
				SNR = 5				
Union Support	SIS-ALASSO	100.0	100.0	0.0	97.0	3.0	-	-
	ISIS-ALASSO	100.0	100.0	0.0	96.0	3.0	-	-
	OMP	100.0	99.8	0.0	0.0	19.6	-	-
	S-OMP	100.0	100.0	0.0	100.0	3.0	-	-
	S-OMP-ALASSO	100.0	100.0	0.0	100.0	3.0	-	-
Exact Support	SIS-ALASSO	60.0	100.0	0.2	57.0	239.5	0.10	0.61
	ISIS-ALASSO	84.0	100.0	0.1	80.0	239.8	0.08	0.61
	OMP	100.0	100.0	0.0	0.0	256.6	0.06	0.61
	S-OMP-ALASSO	100.0	100.0	0.0	100.0	240.0	0.03	0.62

Table 13.4: Results of simulation 4 with parameters $(n, p, s, T) = (150, 4000, 8, 150)$, $T_{\text{non-zero}} = 80$, $\rho = 0.5$

	Method name	Prob. (%) of $\mathcal{M}_* \subseteq \widehat{S}$	Fraction (%) of Correct zeros	Fraction (%) of Incorrect zeros	Fraction (%) of $\mathcal{M}_* = \widehat{S}$	$\lvert\widehat{S}\rvert$	Est. error $\lvert\lvert\mathbf{B} - \widehat{\mathbf{B}}\rvert\rvert_2^2$	Test error R^2
				SNR = 10				
Union Support	SIS-ALASSO	100.0	100.0	0.0	100.0	8.0	-	-
	ISIS-ALASSO	100.0	100.0	0.0	97.0	8.0	-	-
	OMP	100.0	99.9	0.0	2.0	11.7	-	-
	S-OMP	100.0	100.0	0.0	100.0	8.0	-	-
	S-OMP-ALASSO	100.0	100.0	0.0	100.0	8.0	-	-
Exact Support	SIS-ALASSO	35.0	100.0	1.4	35.0	631.3	0.55	0.88
	ISIS-ALASSO	100.0	100.0	0.0	97.0	640.0	0.14	0.89
	OMP	100.0	100.0	0.0	2.0	643.7	0.10	0.89
	S-OMP-ALASSO	100.0	100.0	0.0	100.0	640.0	0.09	0.89

Table 13.5: Results of simulation 5 with parameters $(n, p, s, T) = (200, 10000, 5, 500)$, $T_{\text{non-zero}} = 400$

	Method name	Prob. (%) of $\mathcal{M}_* \subseteq \widehat{S}$	Fraction (%) of Correct zeros	Fraction (%) of Incorrect zeros	Fraction (%) of $\mathcal{M}_* = \widehat{S}$	$\lvert\widehat{S}\rvert$	Est. error $\lvert\lvert\mathbf{B} - \widehat{\mathbf{B}}\rvert\rvert_2^2$	Test error R^2
				σ = 1.5				
Union Support	SIS-ALASSO	53.0	99.6	9.4	0.0	41.1	-	-
	ISIS-ALASSO	100.0	99.8	0.0	0.0	28.1	-	-
	OMP	100.0	99.9	0.0	12.0	10.0	-	-
	S-OMP	100.0	100.0	0.0	44.0	5.6	-	-
	S-OMP-ALASSO	100.0	100.0	0.0	100.0	5.0	-	-
Exact Support	SIS-ALASSO	0.0	100.0	68.9	0.0	936.0	84.66	0.66
	ISIS-ALASSO	0.0	100.0	16.2	0.0	1791.9	5.80	0.96
	OMP	100.0	100.0	0.0	12.0	2090.3	0.06	0.99
	S-OMP-ALASSO	100.0	100.0	0.0	100.0	2000.0	0.05	0.99

We give results on few clusters in Table 13.6 and note that, qualitatively, the results do not vary much between different clusters. While the fitted models have limited predictive performance, which results from highly noisy data, we observe that the S-OMP is able to identify on average one SNP per gene cluster that is related to a large number of genes. Other methods, while having a similar predictive performance, select a larger number of SNPs, which can be seen from the size of the union support. On this particular data set, the S-OMP seems to produce results that are more interpretable from a specialist's points of view. Further investigation needs to be done to verify the biological significance of the selected SNPs, however, the details of such an analysis are going to be reported elsewhere.

13.5 Discussion

In this work, we analyze the Simultaneous Orthogonal Matching Pursuit as a method for variable selection in an ultra-high dimensional space. We prove that the S-OMP is screening consistent and provide a practical way to select the number of steps in the procedure using the modified Bayesian information criterion. A limited number of experiments suggests that the method performs well in practice and that the joint estimation from multiple outputs often outperforms methods that use one regression output at a time. Furthermore, we can see the S-OMP procedure as a way to improve the variable selection properties of the SIS without having to solve a costly complex optimization procedure in Eq. (13.2), therefore, balancing the computational costs and the estimation accuracy.

13.6 Technical Proofs

13.6.1 Proof of Theorem 13.1

Under the assumptions of the theorem, the number of relevant variables s is relatively small compared to the sample size n. The proof strategy can be outlined as follows: i) we are going to show that, with high probability, at least one relevant variable is going to be identified within the following m^*_{one} steps, conditioning on the already selected variables $\mathcal{M}^{(k)}$ and this holds uniformly for all k; ii) we can conclude that all the relevant variables are going to be selected within $m^*_{\text{max}} = sm^*_{\text{one}}$ steps. Exact values for m^*_{one} and m^*_{max} are given below. Without loss of generality, we analyze the first step of the algorithm, that is, we show that the first relevant variable is going to be selected within the first m^*_{one} steps.

Assume that in the first $m^*_{\text{one}} - 1$ steps, there were no relevant variables selected. Assuming that the variable selected in the m^*_{one}-th step is still an irrelevant one, we will arrive at a contradiction, which shows that at least one relevant variable has been selected in the first m^*_{one} steps. For any step k, the reduction of the squared error is given as

$$\Delta(k) := \text{RSS}(k-1) - \text{RSS}(k) = \sum_t ||\mathbf{H}^{(k)}_{t,\widehat{f}_k}(\mathbf{I}_{n\times n} - \mathbf{H}_{t,\mathcal{M}^{(k)}})\mathbf{y}_t||_2^2 \tag{13.6}$$

with $\mathbf{H}^{(k)}_{t,j} = \mathbf{X}^{(k)}_{t,j}\mathbf{X}^{(k)'}_{t,j}||\mathbf{X}^{(k)}_{t,j}||^{-2}$ and $\mathbf{X}^{(k)}_{t,j} = (\mathbf{I}_{n\times n} - \mathbf{H}_{t,\mathcal{M}^{(k)}})\mathbf{X}_{t,j}$. We are interested in the quantity $\sum_{k=1}^{m^*_{\text{one}}} \Delta(k)$, when all the selected variables $\widehat{f}_k$ (see Algorithm 4) belong to $[p]\backslash\mathcal{M}_*$.

Table 13.6: Results on the asthma data

	Method name	Union support	R^2
Cluster 9 Size = 18	SIS-ALASSO	18.0 (1.0)	0.178 (0.006)
	OMP	17.5 (2.9)	0.167 (0.002)
	S-OMP	1.0 (0.0)	0.214 (0.005)
Cluster 16 Size = 31	SIS-ALASSO	31.0 (1.0)	0.160 (0.007)
	OMP	29.0 (1.8)	0.165 (0.002)
	S-OMP	1.0 (0.0)	0.209 (0.005)
Cluster 17 Size = 19	SIS-ALASSO	18.5 (0.9)	0.173 (0.006)
	OMP	19.5 (0.8)	0.146 (0.003)
	S-OMP	1.0 (0.0)	0.184 (0.004)
Cluster 19 Size = 17	SIS-ALASSO	17.0 (1.2)	0.270 (0.017)
	OMP	11.0 (4.1)	0.213 (0.008)
	S-OMP	1.0 (0.0)	0.280 (0.017)
Cluster 22 Size = 34	SIS-ALASSO	34.0 (0.9)	0.153 (0.005)
	OMP	30.0 (7.3)	0.142 (0.000)
	S-OMP	1.0 (0.0)	0.145 (0.002)
Cluster 23 Size = 35	SIS-ALASSO	35.0 (0.9)	0.238 (0.018)
	OMP	33.0 (9.9)	0.208 (0.009)
	S-OMP	1.0 (0.0)	0.229 (0.014)
Cluster 24 Size = 28	SIS-ALASSO	28.0 (1.0)	0.123 (0.003)
	OMP	28.0 (2.6)	0.114 (0.001)
	S-OMP	1.0 (0.0)	0.129 (0.003)
Cluster 32 Size = 15	SIS-ALASSO	15.0 (0.9)	0.188 (0.010)
	OMP	10.0 (2.6)	0.211 (0.006)
	S-OMP	1.0 (0.0)	0.215 (0.008)
Cluster 36 Size = 33	SIS-ALASSO	34.0 (1.4)	0.147 (0.005)
	OMP	29.0 (5.3)	0.157 (0.002)
	S-OMP	1.0 (0.0)	0.168 (0.004)
Cluster 37 Size = 19	SIS-ALASSO	19.0 (0.9)	0.207 (0.015)
	OMP	22.0 (2.5)	0.175 (0.006)
	S-OMP	1.0 (0.0)	0.235 (0.014)
Cluster 39 Size = 24	SIS-ALASSO	24.0 (0.9)	0.131 (0.006)
	OMP	27.0 (1.9)	0.141 (0.003)
	S-OMP	1.0 (0.0)	0.160 (0.005)
Cluster 44 Size = 35	SIS-ALASSO	35.0 (0.9)	0.177 (0.010)
	OMP	26.5 (6.6)	0.183 (0.005)
	S-OMP	1.0 (0.0)	0.170 (0.011)
Cluster 49 Size = 23	SIS-ALASSO	23.0 (1.0)	0.124 (0.004)
	OMP	23.0 (1.2)	0.140 (0.000)
	S-OMP	1.0 (0.0)	0.159 (0.004)

In what follows, we will derive a lower bound for $\Delta(k)$. We perform our analysis on the event

$$\begin{aligned}\mathcal{E} = &\{\min_{t\in[T]}\min_{\mathcal{M}\subseteq[p],|\mathcal{M}|\leq m^*_{\max}} \Lambda_{\min}(\widehat{\mathbf{\Sigma}}_{\mathcal{M}}) \geq \phi_{\min}/2\} \\ &\bigcap\{\max_{t\in[T]}\max_{\mathcal{M}\subseteq[p],|\mathcal{M}|\leq m^*_{\max}} \Lambda_{\max}(\widehat{\mathbf{\Sigma}}_{\mathcal{M}}) \leq 2\phi_{\max}\}.\end{aligned}$$

From the definition of $\widehat{f}_k$, we have

$$\begin{aligned}\Delta(k) &\geq \max_{j\in\mathcal{M}_*}\sum_t ||\mathbf{H}^{(k)}_{t,j}(\mathbf{I}_{n\times n} - \mathbf{H}_{t,\mathcal{M}^{(k)}})\mathbf{y}_t||_2^2 \\ &\geq \max_{j\in\mathcal{M}_*}\Big(\sum_t ||\mathbf{H}^{(k)}_{t,j}(\mathbf{I}_{n\times n} - \mathbf{H}_{t,\mathcal{M}^{(k)}})\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*}||_2^2 \\ &\qquad - \sum_t ||\mathbf{H}^{(k)}_{t,j}(\mathbf{I}_{n\times n} - \mathbf{H}_{t,\mathcal{M}^{(k)}})\boldsymbol{\epsilon}_t||_2^2\Big) \\ &\geq \max_{j\in\mathcal{M}_*}\sum_t ||\mathbf{H}^{(k)}_{t,j}(\mathbf{I}_{n\times n} - \mathbf{H}_{t,\mathcal{M}^{(k)}})\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*}||_2^2 \\ &\qquad - \max_{j\in\mathcal{M}_*}\sum_t ||\mathbf{H}^{(k)}_{t,j}(\mathbf{I}_{n\times n} - \mathbf{H}_{t,\mathcal{M}^{(k)}})\boldsymbol{\epsilon}_t||_2^2 \\ &= (I) - (II).\end{aligned} \tag{13.7}$$

We deal with these two terms separately. Let $\mathbf{H}^{\perp}_{t,\mathcal{M}} = \mathbf{I}_{n\times n} - \mathbf{H}_{t,\mathcal{M}}$ denote the projection matrix. We have that the first term (I) is lower bounded by

$$\begin{aligned}&\max_{j\in\mathcal{M}_*}\sum_t ||\mathbf{H}^{(k)}_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*}||_2^2 \\ &= \max_{j\in\mathcal{M}_*}\sum_t ||\mathbf{X}^{(k)}_{t,j}||_2^{-2}|\mathbf{X}^{(k)'}_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*}|^2 \\ &\geq \min_{t\in[T],j\in\mathcal{M}_*}\{||\mathbf{X}^{(k)}_{t,j}||_2^{-2}\}\max_{j\in\mathcal{M}_*}\sum_t |\mathbf{X}^{(k)'}_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*}|^2 \\ &\geq \{\max_{t\in[T],j\in\mathcal{M}_*}||\mathbf{X}_{t,j}||_2^2\}^{-1}\max_{j\in\mathcal{M}_*}\sum_t |\mathbf{X}'_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*}|^2,\end{aligned} \tag{13.8}$$

where the last inequality follows from the fact that $||\mathbf{X}_{t,j}||_2 \geq ||\mathbf{X}^{(k)}_{t,j}||_2$ and $\mathbf{X}^{(k)'}_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}} = \mathbf{X}'_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}$. A simple calculation shows that

$$\begin{aligned}&\sum_t ||\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*}||_2^2 \\ &\quad= \sum_t\sum_{j\in\mathcal{M}_*}\beta_{t,j}\mathbf{X}_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*} \\ &\quad\leq \sum_{j\in\mathcal{M}_*}\sqrt{\sum_t\beta^2_{t,j}}\sqrt{\sum_t(\mathbf{X}_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*})^2} \\ &\quad\leq ||\boldsymbol{\beta}||_{2,1}\max_{j\in\mathcal{M}_*}\sqrt{\sum_t(\mathbf{X}_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*})^2}.\end{aligned} \tag{13.9}$$

Plugging (13.9) back into (13.8), the following lower bound is achieved

$$(I) \geq \{\max_{t\in[T],j\in\mathcal{M}_*} ||\mathbf{X}_{t,j}||_2^2\}^{-1} \frac{(\sum_t ||\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*}||_2^2)^2}{||\mathbf{B}||_{2,1}^2}. \tag{13.10}$$

On the event $\mathcal{E}$, $\max_{t\in[T],j\in\mathcal{M}_*} ||\mathbf{X}_{t,j}||_2^2 \leq 2n\phi_{\max}$. Since we have assumed that no additional relevant predictors have been selected by the procedure, it holds that $\mathcal{M}_* \not\subseteq \mathcal{M}^{(k)}$. This leads to

$$\sum_t ||\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,\mathcal{M}_*}\boldsymbol{\beta}_{t,\mathcal{M}_*}||_2^2 \geq 2^{-1}n\phi_{\min}\min_{j\in\mathcal{M}^*}\sum_{t\in[T]}\beta_{t,j}^2,$$

on the event $\mathcal{E}$. Using the Cauchy-Schwarz inequality, $||\mathbf{B}||_{2,1}^{-2} \geq s^{-1}T^{-1}C_{\boldsymbol{\beta}}^{-2}$. Plugging back into (13.10), we have that

$$\begin{aligned}(I) &\geq 2^{-3}\phi_{\min}^2\phi_{\max}^{-1}C_{\boldsymbol{\beta}}^{-2}ns^{-1}T^{-1}(\min_{j\in\mathcal{M}^*}\sum_{t\in[T]}\beta_{t,j}^2)^2 \\ &\geq 2^{-3}\phi_{\min}^2\phi_{\max}^{-1}C_{\boldsymbol{\beta}}^{-2}C_s^{-1}n^{1-\delta_s}T^{-1}(\min_{j\in\mathcal{M}^*}\sum_{t\in[T]}\beta_{t,j}^2)^2\end{aligned}$$

Next, we deal with the second term in (13.7). Recall that $\mathbf{X}_{t,j}^{(k)} = \mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\mathbf{X}_{t,j}$, so that $||\mathbf{X}_{t,j}^{(k)}||_2^2 \geq 2^{-1}n\phi_{\min}$, on the event $\mathcal{E}$. We have

$$\begin{aligned}&\sum_t ||\mathbf{H}_{t,j}^{(k)}(\mathbf{I}_{n\times n} - \mathbf{H}_{t,\mathcal{M}^{(k)}})\boldsymbol{\epsilon}_t||_2^2 \\ &\quad= \sum_t ||\mathbf{X}_{t,j}^{(k)}||^{-2}(\mathbf{X}'_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}^{(k)}}\boldsymbol{\epsilon}_t)^2 \\ &\quad\leq 2\phi_{\min}^{-1}n^{-1}\max_{j\in\mathcal{M}_*}\max_{|\mathcal{M}|\leq m^*_{\max}}\sum_t(\mathbf{X}'_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}}\boldsymbol{\epsilon}_t)^2.\end{aligned} \tag{13.11}$$

Under the conditions of the theorem, $\mathbf{X}'_{t,j}\mathbf{H}^{\perp}_{t,\mathcal{M}}\boldsymbol{\epsilon}_t$ is normally distributed with mean 0 and variance $||\mathbf{H}^{\perp}_{t,\mathcal{M}}\mathbf{X}_{t,j}||_2^2$. Furthermore,

$$\max_{j\in\mathcal{M}_*}\max_{|\mathcal{M}|\leq m^*_{\max}}\max_{t\in[T]}||\mathbf{H}^{\perp}_{t,\mathcal{M}}\mathbf{X}_{t,j}||_2^2 \leq 2n\phi_{\max}.$$

Plugging back in (13.11), we have

$$(II) \leq 2^2\phi_{\min}^{-1}\phi_{\max}\max_{j\in\mathcal{M}_*}\max_{|\mathcal{M}|\leq m^*_{\max}}\chi_T^2,$$

where χ_T^2 denotes a chi-squared random variable with T degrees of freedom. The total number of possibilities for $j \in \mathcal{M}_*$ and $|\mathcal{M}| \leq m^*_{\max}$ is bounded by $p^{m^*_{\max}+2}$. Using a tail bound for χ^2 random variable together with the union bound, we obtain

$$\begin{aligned}(II) &\leq 2^3\phi_{\min}^{-1}\phi_{\max}T(m^*_{\max}+2)\log p \\ &\leq 9\phi_{\min}^{-1}\phi_{\max}C_p n^{\delta_p}Tm^*_{\max}\end{aligned} \tag{13.12}$$

with probability at least

$$1 - p^{m^*_{\max}+2} \exp\left(-2T(m^*_{\max}+2)\log(p)\left(1 - 2\sqrt{\frac{1}{2(m^*_{\max}+2)\log(p)}}\right)\right).$$

Going back to (13.7), we have the following

$$\begin{aligned} n^{-1}T^{-1}\Delta(k) &\geq 2^{-3}\phi^2_{\min}\phi^{-1}_{\max}C^{-2}_{\boldsymbol{\beta}}C^{-1}_s n^{-\delta_s}T^{-2}(\min_{j\in\mathcal{M}^*}\sum_{t\in[T]}\beta^2_{t,j})^2 \\ &\quad - 9\phi^{-1}_{\min}\phi_{\max}C_p n^{\delta_p-1}m^*_{\max} \\ &\geq 2^{-3}\phi^2_{\min}\phi^{-1}_{\max}C^{-2}_{\boldsymbol{\beta}}C^{-1}_s c^2_{\boldsymbol{\beta}} n^{-\delta_s-2\delta_{\min}} \\ &\quad - 9\phi^{-1}_{\min}\phi_{\max}C_p n^{\delta_p-1}m^*_{\max} \\ &\geq 2^{-3}\phi^2_{\min}\phi^{-1}_{\max}C^{-2}_{\boldsymbol{\beta}}C^{-1}_s c^2_{\boldsymbol{\beta}} n^{-\delta_s-2\delta_{\min}} \\ &\quad \times (1 - 72\phi^{-3}_{\min}\phi^2_{\max}C^2_{\boldsymbol{\beta}}C_p C_s c^{-2}_{\boldsymbol{\beta}} n^{\delta_s+2\delta_{\min}+\delta_p-1}m^*_{\max}). \end{aligned} \tag{13.13}$$

Since the bound in (13.13) holds uniformly for $k \in \{1, \ldots, m^*_{\text{one}}\}$, we have that

$$n^{-1}T^{-1}\sum_{t\in[T]}||\mathbf{y}_t||^2_2 \geq n^{-1}T^{-1}\sum_{k=1}^{m^*_{\text{one}}}\Delta(k).$$

Setting

$$m^*_{\text{one}} = \lfloor 2^4\phi^{-2}_{\min}\phi_{\max}C^2_{\boldsymbol{\beta}}C_s c^{-2}_{\boldsymbol{\beta}} n^{\delta_s+2\delta_{\min}}\rfloor$$

and recalling that $m^*_{\max} = sm^*_{\text{one}}$, the lower bound becomes

$$n^{-1}T^{-1}\sum_{t\in[T]}||\mathbf{y}_t||^2_2 \geq 2(1 - Cn^{3\delta_s+4\delta_{\min}+\delta_p-1}), \tag{13.14}$$

for a positive constant C independent of p, n, s and T. Under the conditions of the theorem, the right side of (13.14) is bounded below by 2. We have arrived at a contradiction, since under the assumptions $\text{Var}(y_{t,i}) = 1$ and by the weak law of large numbers, $n^{-1}T^{-1}\sum_{t\in[T]}||\mathbf{y}_t||^2_2 \to 1$ in probability. Therefore, at least one relevant variable will be selected in m^*_{one} steps.

To complete the proof, we lower bound the probability in (13.12) and the probability of the event $\mathcal{E}$. Plugging in the value for $m^*_{\max}$, the probability in (13.12) can be lower bounded by $1 - \exp(-C(2T-1)n^{2\delta_s+2\delta_{\min}+\delta_p})$ for some positive constant C. The probability of the event $\mathcal{E}$ is lower bounded as $1 - C_1\exp(-C_2\frac{n^{1-6\delta_s-6\delta_{\min}}}{\max\{\log p, \log T\}})$, for some positive constants C_1 and C_2. Both of these probabilities converge to 1 under the conditions of the theorem.

13.6.2 Proof of Theorem 13.2

To prove the theorem, we use the same strategy as in [182]. From Theorem 13.1, we have that $\mathbb{P}[\exists k \in \{0, \ldots, n-1\} : \mathcal{M}_* \subseteq \mathcal{M}^{(k)}] \to 1$, so $k_{\min} := \min_{k\in\{0,\ldots,n-1\}}\{k : \mathcal{M}_* \subseteq \mathcal{M}^{(k)}\}$ is well defined and $k_{\min} \leq m^*_{\max}$, for $m^*_{\max}$ defined in (13.5). We show that

$$\mathbb{P}[\min_{k\in\{0,\ldots,k_{\min}-1\}}(\text{BIC}(\mathcal{M}^{(k)}) - \text{BIC}(\mathcal{M}^{(k+1)})) > 0] \to 1,$$

so that $\mathbb{P}[\widehat{s} < k_{\min}] \to 0$ as $n \to \infty$. We proceed by lower bounding the difference in the BIC scores as

$$\begin{aligned}\mathrm{BIC}(\mathcal{M}^{(k)}) - \mathrm{BIC}(\mathcal{M}^{(k+1)}) &= \log\left(\frac{\mathrm{RSS}(\mathcal{M}^{(k)})}{\mathrm{RSS}(\mathcal{M}^{(k+1)})}\right) - \frac{\log(n) + 2\log(p)}{n} \\ &\geq \log\left(1 + \frac{\mathrm{RSS}(\mathcal{M}^{(k)}) - \mathrm{RSS}(\mathcal{M}^{(k+1)})}{\mathrm{RSS}(\mathcal{M}^{(k+1)})}\right) - 3n^{-1}\log(p),\end{aligned}$$

where we have assumed $p > n$. Define the event $\mathcal{A} := \{n^{-1}T^{-1}\sum_{t\in[T]} ||\mathbf{y}_t||_2^2 \leq 2\}$. Note that $\mathrm{RSS}(\mathcal{M}^{(k+1)}) \leq \sum_{t\in[T]} ||\mathbf{y}_t||_2^2$, so on the event $\mathcal{A}$ the difference in the BIC scores is lower bounded as

$$\log(1 + 2n^{-1}T^{-1}\Delta(k)) - 3n^{-1}\log(p),$$

where $\Delta(k)$ is defined in (13.6). Using the fact that $\log(1+x) \geq \min(\log(2), 2^{-1}x)$ and the lower bound from (13.13), we have

$$\mathrm{BIC}(\mathcal{M}^{(k)}) - \mathrm{BIC}(\mathcal{M}^{(k+1)}) \geq \min(\log 2, Cn^{-\delta_s - 2\delta_{\min}}) - 3n^{-1}\log p, \tag{13.15}$$

for some positive constant C. It is easy to check that $\log 2 - 3n^{-1}\log p > 0$ and $Cn^{-\delta_s - 2\delta_{\min}} - 3n^{-1}\log p > 0$ under the conditions of the theorem. The lower bound in (13.15) is uniform for $k \in \{0, \ldots, k_{\min}\}$, so the proof is complete if we show that $\mathbb{P}[\mathcal{A}] \to 1$. But this easily follows from the tail bounds on the central chi-squared random variable.

Chapter 14

Marginal Regression For Multi-task Learning

Variable selection is an important practical problem that arises in analysis of many high dimensional datasets. Convex optimization procedures, that arise from relaxing the NP-hard subset selection procedure, e.g., the Lasso or Dantzig selector, have become the focus of intense theoretical investigations. Although many efficient algorithms exist that solve these problems, finding a solution when the number of variables is large, e.g., several hundreds of thousands in problems arising in genome-wide association analysis, is still computationally challenging. A practical solution for these high-dimensional problems is the marginal regression, where the output is regressed on each variable separately. We investigate theoretical properties of the marginal regression in a multitask framework. Our contribution include: i) sharp analysis for the marginal regression in a single task setting with random design, ii) sufficient conditions for the multitask screening to select the relevant variables, iii) a lower bound on the Hamming distance convergence for multitask variable selection problems. A simulation study further demonstrates the performance of the marginal regression.

14.1 Introduction

Recent technological advances are allowing scientists in a variety of disciplines to collect data of unprecedented size and complexity. Examples include data from biology, genetics, astronomy, brain imaging and high frequency trading. These novel applications are often characterized by large number of variables p, which can be much larger than the number of observations n, and are currently driving the development of statistical and machine learning procedures. The sparsity assumption has been recognized to play a critical role in effective high-dimensional inference in classification and regression problems, that is, the statistical inference is possible in the underdetermined problems under the assumption that only a few variables contribute to the response. Therefore, the variable selection is of fundamental importance in the high-dimensional problems.

Consider a regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{14.1}$$

with response $\mathbf{y} = (y_1, \ldots, y_m)'$, $m \times p$ design matrix $\mathbf{X}$, noise vector $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_m)'$ and

coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$. For simplicity of presentation, we assume that $m = 2n$ and use first n samples to estimate the parameters and use remaining parameters to optimally select the tuning parameters. The high dimensional setting assumes $p \gg n$ and the sparsity assumption roughly states that the coefficient vector $\boldsymbol{\beta}$ has a few non-zero components or that it can be well approximated by such a vector. In the context of linear regression, there has been a lot of recent work focusing on variable selection under the sparsity assumption, such as, [175], [64], [28], [97], [201], [202], [198], [38], [35], [48], [190], [205], [49], [74], [173], and [137], to name a few. Many of these methods are based on constrained or penalized optimization procedures in which solutions are biased to have many zero coefficients. One of the main tools for variable selection in a regression model is the Lasso estimator defined by

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \, ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_1 \tag{14.2}$$

where $\lambda \geq 0$ is a user defined regularization parameter. Theoretical properties of the estimator $\widehat{\beta}$ are now well understood and the optimization problem (14.2) can be efficiently solved for medium sized problems. However, finding a solution in problems involving hundreds of thousands variables, which commonly arise in genome-wide association mapping problems, still remains a computationally challenging task, even when many variables can be pruned using rules based on the KKT conditions [57, 176].

One computationally superior alternative to the Lasso is marginal regression, also known as correlation learning, marginal learning and sure screening. This is a very old and simple procedure, which has recently gained popularity due to its desirable properties in high-dimensional setting [62, 66, 68, 83, 184]. Marginal regression is based on regressing the response variable on each variable separately

$$\widehat{\mu}_j = (\mathbf{X}_j'\mathbf{X}_j)^{-1}\mathbf{X}_j'\mathbf{y}, \tag{14.3}$$

where $\mathbf{X}_j = (x_{1j}, \ldots, x_{nj})'$. Next, the values $\{|\widehat{\mu}_j|\}$ are sorted in decreasing order, with $\{\widehat{r}_j\}$ denoting the ranks, and the set of estimated variables is

$$\widehat{S}(k) := \{1 \leq j \leq p : \widehat{r}_j \leq k\}, \qquad 1 \leq k \leq p.$$

Note that in Eq. (14.3) we use the first n samples only to compute $\widehat{\mu}_j$. Under a condition, related to the faithfulness conditions used in causal literature [157, 165], it can be shown that the set $\widehat{S}(k)$ correctly estimates the relevant variables $S := \{1 \leq j \leq p : \beta_j \neq 0\}$, see [184]. The following result provides the conditions under which the exact variable selection is possible if the size of the support $s := |S|$ is known.

Theorem 14.1. *Consider the regression model in* (14.1) *with* $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$, $\mathbf{x}_i \overset{iid}{\sim} \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$, *and* $\boldsymbol{\epsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2\mathbf{I}_n)$, $\mathbf{X}$ *independent of* $\boldsymbol{\epsilon}$. *Assume that*

$$\max_{j \in S^C} |\boldsymbol{\Sigma}_{jS}\boldsymbol{\beta}_S| + \gamma_n(p, s, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \delta) < \min_{j \in S} |\boldsymbol{\Sigma}_{jS}\boldsymbol{\beta}_S| \tag{14.4}$$

with $\gamma_n = \mathcal{O}(\sqrt{\log(p-s)/n})$, *then*

$$\mathbb{P}[\widehat{S}(s) = S] \geq 1 - \delta.$$

The above theorem is based on the asymptotic result in [184]. We provide a finite sample analysis and explicit constants for the term $\gamma_n(p, s, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \delta)$ in Appendix. The condition like the one in Eq. (14.4) is essentially unavoidable for marginal regression, since it can be seen that in the noiseless setting ($\boldsymbol{\epsilon} = \mathbf{0}$) the condition (14.4) with $\gamma_n = 0$ is necessary and sufficient for successful recovery. See [83] for discussion of cases where the faithfulness condition is weaker than the irrepresentable condition, which is necessary and sufficient for exact recovery of the support using the Lasso [190, 205].

Besides computational simplicity, another practical advantage of the marginal regression is that the number of relevant variables s can be estimated from data efficiently as we show below. This corresponds to choosing the tuning parameter λ in the Lasso problem (14.2) from data. To estimate the number of relevant variables, we will use the samples indexed by $\{n+1, \ldots, 2n\}$, which are independent from those used to estimate $\{\widehat{\mu}_j\}_j$. For a fixed $1 \leq k \leq p$, let j_k denote the index of the variable for which $\widehat{r}_{j_k} = k$. Let $\widehat{V}_n(k) = \text{span}\{\mathbf{X}_{j_1}, \ldots, \mathbf{X}_{j_k}\}$ be the linear space spanned by k variables whose empirical correlation with the response is the highest, and let $\widehat{\mathbf{H}}(k)$ be the projection matrix from $\mathbb{R}^n$ to $\widehat{V}_n(k)$. Note that $\mathbf{X}_{j_k} = (x_{n+1,j_k}, \ldots, x_{2n,j_k})$. Define

$$\widehat{\xi}_n(k) := ||(\widehat{\mathbf{H}}(k+1) - \widehat{\mathbf{H}}(k))\mathbf{y}||_2^2, \quad 1 \leq k \leq p-1,$$

which is then used to estimate the number of relevant variables as

$$\widehat{s}_n = \max\{1 \leq k \leq p-1 : \widehat{\xi}_n(k) \leq 2\sigma^2 \log \tfrac{4n}{\delta}\} + 1.$$

Using an independent sample to select the number of relevant variables is needed so that the projection matrix is independent of the noise $\boldsymbol{\epsilon}$. With these definitions, we have the following result.

Theorem 14.2. *Assume that the conditions of Theorem 14.1 are satisfied. Furthermore, assume that*

$$\min_{j \in S} |\beta_j| = \Omega(\sqrt{\log n}).$$

Then $\mathbb{P}[\widehat{S}(\widehat{s}_n) = S] \xrightarrow{n \to \infty} 1$.

The above results builds on Theorem 3 in [83].

In the next few sections, we study properties of the marginal regression in a multitask setting.

14.2 Multitask Learning with Marginal Regression

In this section, we analyze properties of the marginal regression in a multitask setting. We will consider the following multitask regression model

$$\mathbf{y}_t = \mathbf{X}\boldsymbol{\beta}_t + \boldsymbol{\epsilon}_t \qquad t = 1, \ldots, T \tag{14.5}$$

where $\mathbf{y}_t, \boldsymbol{\epsilon} \in \mathbb{R}^m$ and , $\mathbf{X} \in \mathbb{R}^{m \times p}$. Again, we assume that $m = 2n$ and use half of the samples to rank the variables and the other half to select the correct number of relevant variables. The subscript t indexes tasks and $\boldsymbol{\beta}_t \in \mathbb{R}^p$ is the unknown regression coefficient for the t-th task. We assume that there is a shared design matrix $\mathbf{X}$ for all tasks, a situation that arises, for example, in

genome-wide association studies. Alternatively, one can have one design matrix $\mathbf{X}_t$ for each task. We assume that the regression coefficients are jointly sparse. Let $S_t := \{1 \le j \le p : \beta_{tj} \neq 0\}$ be the set of relevant variables for the t-th task and let $S = \cup_t S_t$ be the set of all relevant variables. Under the joint sparsity assumption $s := |S| \ll n$.

To perform marginal regression in the multitask, one computes correlation between each variable and each task using the first half of the samples

$$\widehat{\mu}_{tj} = (\mathbf{X}_j'\mathbf{X}_j)^{-1}\mathbf{X}_j'\mathbf{y}_t, \tag{14.6}$$

for each $t = 1, \ldots, T,\ j = 1, \ldots, p$. Let $\Phi : \mathbb{R}^T \mapsto \mathbb{R}_+$ be a scoring function, which is used to sort the values $\{\Phi(\{\widehat{\mu}_{tj}\}_t)\}_j$ in decreasing order. Let $\{\widehat{r}_{\Phi,j}\}$ denote the rank of variable j in the ordering, then the set of estimated variables is

$$\widehat{S}_\Phi(k) := \{1 \le j \le p : \widehat{r}_{\Phi,j} \le k\}, \qquad 1 \le k \le p.$$

For concreteness, we will use the norm $||\cdot||_1, ||\cdot||_2$ and $||\cdot||_\infty$ as our scoring functions and denote the sets of estimated variables $\widehat{S}_{\ell_1}(k)$, $\widehat{S}_{\ell_2}(k)$ and $\widehat{S}_{\ell_\infty}(k)$ respectively.

With the notation introduced, we focus on providing conditions for the marginal regression to exactly select the relevant variables S. We start our analysis in the fixed design setting. Let $\mathbf{\Sigma} = n^{-1}\mathbf{X}'\mathbf{X}$ and assume that the variables are standardized to have zero mean and unit variance, so that the diagonal elements of $\mathbf{\Sigma}$ are equal to 1. Now it simply follows from (14.6) that

$$\widehat{\mu}_{tj} = n^{-1}\mathbf{X}_j'\mathbf{y}_t = \mathbf{\Sigma}_{jS_t}\boldsymbol{\beta}_{tS_t} + n^{-1}\mathbf{X}_j'\boldsymbol{\epsilon}_t.$$

In order to show that marginal regression exactly recovers the set of relevant variables, we need to have

$$\max_{j \in S^C} \Phi(\{\widehat{\mu}_{tj}\}_t) \le \min_{j \in S} \Phi(\{\widehat{\mu}_{tj}\}_t). \tag{14.7}$$

It is easy to see that (14.7) is necessary for exact recovery. The following theorem provides sufficient conditions for (14.7) to hold.

Theorem 14.3. *Consider the model* (14.5) *with* $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_n)$ *and* $\sigma > 0$ *known. The following three claims hold: i) Define* $\nu_j = \sigma^{-2}n\sum_{t=1}^T(\mathbf{\Sigma}_{jS_t}\boldsymbol{\beta}_{tS_t})^2$. *If*

$$\begin{aligned}\max_{j \in S^C} \nu_j + 2\log\frac{2(p-s)}{\delta} + \max_{j \in S} 2\sqrt{(T+2\nu_j)\log\frac{2s}{\delta}} + \max_{j \in S^C} 2\sqrt{(T+2\nu_j)\log\frac{2(p-s)}{\delta}} \\ \le \min_{j \in S} \nu_j\end{aligned} \tag{14.8}$$

then $\mathbb{P}[\widehat{S}_{\ell_2}(s) = S] \ge 1 - \delta$. *ii) If*

$$\begin{aligned}\max_{j \in S^C} \sum_{t=1}^T |\mathbf{\Sigma}_{jS_t}\boldsymbol{\beta}_{tS_t}| + n^{-1/2}\sigma\sqrt{T^2 + 2T\sqrt{T\log\frac{2(p-s)}{\delta}} + 2T\log\frac{2(p-s)}{\delta}} \\ + n^{-1/2}\sigma\sqrt{T^2 + 2T\sqrt{T\log\frac{2s}{\delta}} + 2T\log\frac{2s}{\delta}} \\ \le \min_{j \in S} \sum_{t=1}^T |\mathbf{\Sigma}_{jS_k}\boldsymbol{\beta}_{kS_k}|\end{aligned} \tag{14.9}$$

then $\mathbb{P}[\widehat{S}_{\ell_1}(s) = S] \geq 1 - \delta$. *iii) If*

$$\max_{j \in S^C} \max_{1 \leq t \leq T} |\mathbf{\Sigma}_{jS_t} \boldsymbol{\beta}_{tS_t}| + n^{-1/2}\sigma \left(\sqrt{2 \log \frac{2(p-s)T}{\delta}} + \sqrt{2 \log \frac{2sT}{\delta}} \right) \leq \min_{j \in S} \max_{1 \leq t \leq T} |\mathbf{\Sigma}_{jS_t} \boldsymbol{\beta}_{tS_t}| \tag{14.10}$$

then $\mathbb{P}[\widehat{S}_{\ell_\infty}(s) = S] \geq 1 - \delta$.

Theorem 14.3 extends Theorem 14.1 to the multitask setting and provides sufficient conditions for the marginal regression to perform exact variable selection. We will discuss how the three different scoring procedures compare to each other in the following section.

Theorem 14.3 assumes that the number of relevant variables is known, as in Theorem 14.1. Therefore, we need to estimate the number of relevant variables in a data-dependent way. This is done using the remaining n samples, indexed by $\{n+1, \ldots, 2n\}$. Recall the definitions from p. 213, where j_k denotes the index of the variable for which $\widehat{r}_{\Phi, j_k} = k$, $\widehat{V}_n(k) = \text{span}\{\mathbf{X}_{j_1}, \ldots, \mathbf{X}_{j_k}\}$ and $\widehat{\mathbf{H}}(k)$ is the projection matrix from $\mathbb{R}^n$ to $\widehat{V}_n(k)$. Define

$$\widehat{\xi}_{\ell_2,n}(k) := \sum_{t=1}^{T} ||(\widehat{\mathbf{H}}(k+1) - \widehat{\mathbf{H}}(k))\mathbf{y}_t||_2^2, \quad 1 \leq k \leq p-1,$$

which is then used to estimate the number of relevant variables as

$$\widehat{s}_{\ell_2,n} = 1 + \max\{1 \leq k \leq p-1 : \\ \widehat{\xi}_{\ell_2,n}(k) \leq (T + 2\sqrt{T \log(2/\delta)} + 2\log(2/\delta))\sigma^2\}.$$

Let $V_S = \text{span}\{\mathbf{X}_j : j \in S\}$ be the subspace spanned by columns of $\mathbf{X}$ indexed by S and similarly define $V_{S,-j} = \text{span}\{\mathbf{X}_{j'} : j' \in S \backslash \{j\}\}$. Let $\mathbf{X}_j^{(2)}$ denote the projection of $\mathbf{X}_j$ to $V_S \cap V_{S,-j}^{\perp}$. With these definitions, we have the following result.

Theorem 14.4. *Consider the model* (14.5) *with* $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$ *and* $\sigma > 0$ *known. Suppose that one of the following three claims hold: i) Eq.* (14.8) *holds and variables are ranked as* $\{\widehat{r}_{\ell_2,j}\}_j$*, ii) Eq.* (14.9) *holds and variables are ranked as* $\{\widehat{r}_{\ell_1,j}\}_j$*, or iii) Eq.* (14.10) *holds and variables are ranked as* $\{\widehat{r}_{\ell_1,j}\}_j$*. Furthermore assume that*

$$\min_{j \in S} \sum_{t=1}^{T} ||\mathbf{X}_j^{(2)} \beta_{tj}||_2^2 > \left[2\sqrt{5} \log^{1/2}\left(\frac{4}{\delta^2}\right) \sqrt{T} + 8 \log\left(\frac{4}{\delta^2}\right) \right] \sigma^2. \tag{14.11}$$

Then $\mathbb{P}[\widehat{s}_{\ell_2,n} = s] \geq 1 - 2\delta$ *and* $\mathbb{P}[\widehat{S}_\phi(\widehat{s}_{\ell_2,n}) = S] \geq 1 - 2\delta$.

Theorem 14.4 provides a way to select the number of relevant variables in a multitask setting. It is assumed that one of the conditions given in Theorem 14.3 are satisfied and that the corresponding scoring procedure is used to rank features. Condition (14.11) is required in order to distinguish relevant variables from noise. If the signal strength is small compared to the noise, there is no hope to select the relevant variables. Comparing to Theorem 14.2, we can quantify improvement over applying marginal regression to each task individually. First, the minimal signal strength for each variable, quantified as $\min_{j \in S} \sum_{t=1}^{T} ||\mathbf{X}_j^{(2)} \beta_{tj}||_2^2$ needs to increase only as $\mathcal{O}(\sqrt{T})$ in multitask setting compared to $\mathcal{O}(T)$ when the marginal regression is applied to each task individually.

Theorem 14.3 and 14.4 assume that the design is fixed. However, given proofs of Theorem 14.1 and 14.2, extending the proofs of the multitask marginal regression is straight forward.

14.2.1 Comparing Different Scoring Procedures

In this section, we compare the three scoring procedures based on $||\cdot||_1$, $||\cdot||_2$ and $||\cdot||_\infty$. Theorem 14.3 provides sufficient conditions under which $\widehat{S}_{\ell_1}$, $\widehat{S}_{\ell_2}$ and $\widehat{S}_{\ell_\infty}$ exactly recover the set of relevant variables S. In order to provide more intuition, we will focus on conditions (14.8), (14.9) and (14.10) when $\mathbf{\Sigma} = \mathbf{I}$. Furthermore, we assume that $s = O(1)$.

From (14.8), we have that

$$\max_{j\in S^C} T^{-1}||\beta_{\cdot j}||_2^2 + \mathcal{O}(\frac{\log p}{nT}) + \mathcal{O}(\frac{\sqrt{(T + n\max_j ||\beta_{\cdot j}||_2^2)\log p}}{nT}) \leq \min_{j\in S} T^{-1}||\beta_{\cdot j}||_2^2$$

is sufficient for $\widehat{S}_{\ell_2}$ to recover S. Condition (14.9) simplifies to

$$\max_{j\in S^C} T^{-1}||\beta_{\cdot j}||_1 + \mathcal{O}(\sqrt{\frac{1 + T^{-1}\log p + T^{-1/2}\sqrt{\log p}}{n}}) \leq \min_{j\in S} T^{-1}||\beta_{\cdot j}||_1.$$

Finally, condition (14.10) simplifies to

$$\max_{j\in S^C} ||\beta_{\cdot j}||_\infty + \mathcal{O}(\sqrt{\frac{\log pT}{n}}) \leq \min_{j\in S} ||\beta_{\cdot j}||_\infty.$$

Comparing the sufficient condition in this simplified form, we can observe that the $\widehat{S}_{\ell_2}$ requires weaker conditions for exact support recovery than $\widehat{S}_{\ell_\infty}$. Furthermore, it can be seen that the estimator $\widehat{S}_{\ell_\infty}$ is the most related to the support recovered using the marginal regression on each task separately. From Theorem 14.1, if we stack regression coefficients for different tasks into a big vector, we have that

$$\max_{j\in S^C}\max_{1\leq t\leq T} |\beta_{tj}| + \mathcal{O}(\sqrt{\frac{\log pT}{n}}) \leq \min_{j\in S}\min_{1\leq t\leq T} |\beta_{tj}|$$

is sufficient for the exact support recovery. This is a stronger requirement that the one needed for $\widehat{S}_{\ell_\infty}$. Still, from the numerical results, we observe that $\widehat{S}_{\ell_1}$ and $\widehat{S}_{\ell_2}$ perform better than $\widehat{S}_{\ell_\infty}$.

14.3 Universal Lower Bound for Hamming distance

So far, we have focused on the exact variable selection. Although the exact variable selection has been focus of many studies, the exact recovery of variables is not possible in many practical applications with low signal to noise ratio. Therefore, it is more natural to measure performance using a distance between the sets of selected variables and the true set S.

In this section, let $\mathbf{X}$, $\mathbf{y}_1, \ldots, \mathbf{y}_T$, $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T$, $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T$ be the same as before. Here $\mathbf{X}$ could be either deterministic or random satisfying $\mathbf{X}_j'\mathbf{X}_j = 1$ for $j = 1, \ldots, p$. We are interested in studying the lower bound for variable selection problem measured by Hamming distance. To construct lower bound, we need to clearly define the model family we are studying. We use the following random coefficient model which is adapted from [83]:

$$\beta_{tj} \overset{\text{i.i.d.}}{\sim} (1 - \eta_p)\nu_0 + \eta_p\nu_{\tau_p},$$

for all $t = 1, \ldots, T,\ j = 1, \ldots, p$, where ν_0 is the point mass at 0 and ν_{τ_p} is the point mass at τ_p. Both η_p and τ_p vary with p. We set

$$\eta_p = p^{-v}, \ \ 0 < v < 1,$$

so that the expected number of signals is $s_p = p\eta_p = p^{1-v}$. Let $r > 0$ be some fixed constant and set $\tau_p = \sqrt{2r \log p}$ the signal strength. Such a setting has been extensively explored in the community of modern statistics to explore the theoretical limit of many problems including classification, density estimation, and multiple hypothesis testing [36, 50, 98].

Let $\widehat{S}$ be the index set of selected variables for any variable selection procedure and S be the index set of true relevant variables. We define the Hamming distance

$$H_p(\widehat{S}, S \mid \mathbf{X}) = \mathbb{E}_{\eta_p, \pi_p}\left[\left|(\widehat{S} \setminus S) \cup (S \setminus \widehat{S})\right|\right].$$

Let

$$\begin{aligned}\lambda_p &:= \frac{1}{\tau_p}\left[\log\left(\frac{1-\eta_p}{\eta_p}\right) + \frac{T\tau_p^2}{2}\right] \\ &= \frac{1}{\sqrt{2r\log p}}\log(p^v - 1) + T\sqrt{\frac{r\log p}{2}} \\ &\le \frac{(v + Tr)\sqrt{\log p}}{\sqrt{2r}}.\end{aligned}$$

Our main result in this section provides a universal lower bound of $H_p(\widehat{S}, S \mid \mathbf{X})$ for all sample size n and design matrix $\mathbf{X}$. Let $F(\cdot)$ and $\overline{F}(\cdot)$ be the distribution function and survival function of the standard Gaussian distribution and let $\phi(\cdot)$ denote the density function of the standard Gaussian distribution. We have the following lower bound results.

Theorem 14.5. (Universal lower bound) *Fix $v \in (0, 1), r > 0$ and a sufficiently large p. For any n and design matrix $\mathbf{X}$ such that $\mathbf{X}'\mathbf{X}$ has unit diagonals, we have the following lower bound:*

$$\frac{H_p(\widehat{S}, S \mid \mathbf{X})}{s_p} \ \ge \left[\frac{1-\eta_p}{\eta_p}\overline{F}\left(\frac{\lambda_p}{\sqrt{T}}\right) + F\left(\frac{\lambda_p}{\sqrt{T}} - \sqrt{T}\tau_p\right)\right]. \tag{14.12}$$

This can be further written as

$$\frac{H_p(\widehat{S}, S \mid \mathbf{X})}{s_p} \ \ge \begin{cases} \dfrac{\sqrt{rT}}{2(v + Tr)\sqrt{\pi \log p}} \cdot p^{-(v - Tr)^2/(4rT)}, & v < rT \\ 1 + o(1), & v > rT. \end{cases}$$

One thing to note is that in the above theorem is that such a lower bound simultaneously holds for any sample size n. The main reason for this is that we constraint $\mathbf{X}_j'\mathbf{X}_j = 1$ for all $j = 1, \ldots, p$. Such a standardization essentially fixes the signal-to-noise ratio under asymptotic framework where p increases. Therefore, the lower bound does not depend on sample size n.

14.3.1 Comparing with Single Task Screening

It would be instructive to compare the lower bounds for multitask screening with that for single task screening. By setting $T = 1$, we can obtain from Theorem 14.5 that the Hamming distance lower bound for single task screening takes the form:

$$\frac{H_p^{\text{single}}(\widehat{S}, S \mid \mathbf{X})}{s_p} \geq \begin{cases} \dfrac{\sqrt{r}}{2(v+r)\sqrt{\pi \log p}} \cdot p^{-(v-r)^2/(4r)}, & v < r \\ 1 + o(1), & v > r. \end{cases}$$

Comparing the lower bounds for both settings, we see that for single task screening. If $v > r$, $H_p^{\text{single}}(\widehat{S}, S \mid \mathbf{X}) \geq s_p + o(1)$. This means no procedure can recovery any information of the true signal at all. On the other hand, the corresponding no recovery condition for multitask screening is strengthened to be $r > Tr$ and such a condition rarely holds when T is larger. Therefore, one effect of the multitask setting is that the signal-to-noise ratio is improved by jointly considering multiple tasks. For the case that $r < vT$ and $r < T$ in both settings, it can be seen that the rate for multitask screening is much faster than that for single-task screening.

14.3.2 Upper Bound on Hamming Distance

Though the lower bound result in 14.5 is illustrative, it would be more interesting if we could match the lower bound with a certain algorithm procedure. If we only consider the screening error made by the multitask regression (i.e., the screening procedure should miss important variables), it's straightforward to match the lower bound by setting a conservative threshold using any of the $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$-procedures. However, it is still an open problem to see which procedure could match the Hamming distance lower bound.

14.4 Empirical Results

We conduct an extensive number of numerical studies to evaluate the finite sample performance of the marginal regression on the multitask model given in (14.5). We consider marginal regression using the three scoring procedures outlined in Section 14.2. The variables are ranked using $||\cdot||_1$, $||\cdot||_2$ and $||\cdot||_\infty$ norms and the resulting sets of variables are denoted $\widehat{S}_{\ell_1}$, $\widehat{S}_{\ell_2}$ and $\widehat{S}_{\ell_\infty}$. The number of active variables is set using the result of Theorem 14.4.

Let $\widehat{S}$ be an estimate obtained by one of the scoring methods. We evaluate the performance averaged over 200 simulation runs. Let $\widehat{\mathbb{E}}_n$ denote the empirical average over the simulation runs. We measure the size of the support $\widehat{S}$. Next, we estimate the probability that the estimated set contains the true set S, that is, $\widehat{\mathbb{E}}_n[\mathbb{I}\{S \subseteq \widehat{S}\}]$, which we call coverage probability. We define fraction of correct zeros $(p-s)^{-1}\widehat{\mathbb{E}}_n[|\widehat{S}^C \cap S^C|]$, fraction of incorrect zeros $s^{-1}\widehat{\mathbb{E}}_n[|\widehat{S}^C \cap S|]$ and fraction of correctly fitted $\widehat{\mathbb{E}}_n[\mathbb{I}\{S = \widehat{S}\}]$ to measure the performance of different scoring procedures.

We outline main findings using the following simulation studies. Due to space constraints, tables with detailed numerical results are given in the Appendix.

Simulation 1: The following toy model is based on the simulation I in [62] with $(n, p, s, T) = (400, 20000, 18, 500)$. Each $\mathbf{x}_i$ is drawn independently from a standard multivariate normal distribution, so that the variables are mutually independent. For $j \in S$ and $t \in 1, \ldots, T$, the non-zero coefficients are given as $\beta_{tj} = (-1)^u(4n^{-1/2}\log n + |z|)$, where $u \sim \text{Bernoulli}(0.4)$ and $z \sim \mathcal{N}(0, 1)$. The number of non-zero elements in $\{\beta_{tj}\}_t$ is given as a parameter $T_{\text{non-zero}} \in \{500, 300, 100\}$. The positions of non-zero elements are chosen uniformly at random from $\{1, \ldots, T\}$. The noise is Gaussian with the standard deviation σ set to control the signal-to-noise ratio (SNR). SNR is defined as $\text{Var}(\mathbf{x}\boldsymbol{\beta})/\text{Var}(\boldsymbol{\epsilon})$ and we vary $\text{SNR} \in \{15, 10, 5, 1\}$.

Simulation 2: The following model is used to evaluate the performance of the methods as the number of non-zero elements in $\{\beta_{tj}\}_t$ varies. We set $(n, p, s) = (100, 500, 10)$ and vary the number of outputs $T \in \{500, 750, 1000\}$. For each number of outputs T, we vary $T_{\text{non-zero}} \in \{0.8T, 0.5T, 0.2T\}$. The samples $\mathbf{x}_i$ and regression coefficients are given as in Simulation 1, that is, $\mathbf{x}_i$ is drawn from a multivariate standard normal distribution and the non-zero coefficients are given as $\beta_{tj} = (-1)^u(4n^{-1/2}\log n + |z|)$, where $u \sim \text{Bernoulli}(0.4)$ and $z \sim \mathcal{N}(0, 1)$. The noise is Gaussian, with the standard deviation defined through the SNR, which varies in $\{10, 5, 1\}$.

Simulation 3: The following model is borrowed from [182]. We assume a correlation structure between variables given as $\text{Var}(\mathbf{X}_{j_1}, \mathbf{X}_{j_2}) = \rho^{|j_1 - j_2|}$, where $\rho \in \{0.2, 0.5, 0.7\}$. This correlation structure appears naturally among ordered variables. We set $(n, p, s, T) = (100, 5000, 3, 150)$ and $T_{\text{non-zero}} = 80$. The relevant variables are at positions $(1, 4, 7)$ and non-zero coefficients are given as $3, 1.5$ and 2 respectively. The SNR varies in $\{10, 5, 1\}$.

Simulation 4: The following model assumes a block compound correlation structure. For a parameter ρ, the correlation between two variables $\mathbf{X}_{j_1}$ and $\mathbf{X}_{j_2}$ is given as ρ, ρ^2 or ρ^3 when $|j_1 - j_2| \leq 10$, $|j_1 - j_2| \in (10, 20]$ or $|j_1 - j_2| \in (20, 30]$ and is set to 0 otherwise. We set $(n, p, s, T) = (150, 4000, 8, 150)$, $T_{\text{non-zero}} = 80$ and the parameter $\rho \in \{0.2, 0.5\}$. The relevant variables are located at positions 1, 11, 21, 31, 41, 51, 61, 71 and 81, so that each block of highly correlated variables has exactly one relevant variable. The values of relevant coefficients are given in Simulation 1. The noise is Gaussian and the SNR varies in $\{10, 5, 1\}$.

Simulation 5: This model represents a difficult setting. It is modified from [182]. We set $(n, p, s, T) = (200, 10000, 5, 500)$. The number of non-zero elements in each row varies is $T_{\text{non-zero}} \in \{400, 250, 100\}$. For $j \in [s]$ and $t \in [T]$, the non-zero elements equal $\beta_{tj} = 2j$. Each row of $\mathbf{X}$ is generated as follows. Draw independently $\mathbf{z}_i$ and $\mathbf{z}'_i$ from a p-dimensional standard multivariate normal distribution. Now, $x_{ij} = (z_{ij} + z'_{ij})/\sqrt{(2)}$ for $j \in [s]$ and $x_{ij} = (z_{ij} + \sum_{j' \in [s]} z_{ij'})/2$ for $j \in [p] \backslash [s]$. Now, $\text{Corr}(x_{i,1}, y_{t,i})$ is much smaller then $\text{Corr}(x_{i,j}, y_{t,i})$ for $j \in [p] \backslash [s]$, so that it becomes difficult to select variable 1. The variable 1 is 'masked' with the noisy variables. This setting is difficult for screening procedures as they take into consideration only marginal information. The noise is Gaussian with standard deviation $\sigma \in \{1.5, 2.5, 4.5\}$.

Our simulation setting transitions from a simple scenario considered in Simulation 1 towards a challenging one in Simulation 5. Simulation 1 represents a toy model, where variables are independent. Simulation 2 examines the influence of the number of non-zero elements in the set $\{\beta_{tj}\}_t$. Simulations 3 and 4 represent more challenging situations with structured correlation that naturally appears in many data sets, for example, a correlation between gene measurements that are closely located on a chromosome. Finally Simulation 5 is constructed in such a way such that an irrelevant variable is more correlated with the output than a relevant variable. Tables giving

detailed results of the above described simulations are given in Appendix. We reproduce some parts of the tables below. We observe that the sets $\widehat{S}_{\ell_1}$ and $\widehat{S}_{\ell_2}$ perform similarly across different simulation settings. Except for the simulation 5, $\widehat{S}_{\ell_\infty}$ has worse performance than the other two estimators. The performance difference is increased as the signal to noise ratio decreases. However, when the signal to noise ratio is large there is little difference between the procedures.

	$\widehat{S}$	Prob. (%) of $S \subseteq \widehat{S}$	Fraction (%) of Correct zeros	Fraction (%) of Incorrect zeros	Fraction (%) of $S = \widehat{S}$	$\lvert\widehat{S}\rvert$
Simulation 1: $(n, p, s, T) = (500, 20000, 18, 500)$, $T_{\mathrm{non-zero}} = 300$						
SNR = 5	$\widehat{S}_{\ell_\infty}$	100.0	100.0	0.0	76.0	18.3
	$\widehat{S}_{\ell_1}$	100.0	100.0	0.0	91.0	18.1
	$\widehat{S}_{\ell_2}$	100.0	100.0	0.0	92.0	18.1
Simulation 2.a: $(n, p, s, T) = (200, 5000, 10, 500)$, $T_{\mathrm{non-zero}} = 400$						
SNR = 5	$\widehat{S}_{\ell_\infty}$	100.0	100.0	0.0	82.0	10.2
	$\widehat{S}_{\ell_1}$	100.0	100.0	0.0	91.0	10.1
	$\widehat{S}_{\ell_2}$	100.0	100.0	0.0	91.0	10.1
Simulation 3: $(n, p, s, T) = (100, 5000, 3, 150)$, $T_{\mathrm{non-zero}} = 80$, $\rho = 0.7$						
SNR = 5	$\widehat{S}_{\ell_\infty}$	96.0	100.0	1.3	95.0	3.0
	$\widehat{S}_{\ell_1}$	99.0	100.0	0.3	97.0	3.0
	$\widehat{S}_{\ell_2}$	97.0	100.0	1.0	95.0	3.0
Simulation 4: $(n, p, s, T) = (150, 4000, 8, 150)$, $T_{\mathrm{non-zero}} = 80$, $\rho = 0.5$						
SNR = 5	$\widehat{S}_{\ell_\infty}$	100.0	100.0	0.0	84.0	8.2
	$\widehat{S}_{\ell_1}$	100.0	100.0	0.0	87.0	8.1
	$\widehat{S}_{\ell_2}$	100.0	100.0	0.0	87.0	8.1
Simulation 5: $(n, p, s, T) = (200, 10000, 5, 500)$, $T_{\mathrm{non-zero}} = 250$						
σ = 2.5	$\widehat{S}_{\ell_\infty}$	87.0	100.0	2.6	39.0	5.9
	$\widehat{S}_{\ell_1}$	0.0	99.9	90.6	0.0	14.8
	$\widehat{S}_{\ell_2}$	0.0	99.9	55.0	0.0	12.5

14.5 Discussion

This chapter has focused on the analysis of the marginal regression in the multitask setting. Due to its simplicity and computational efficiency, the marginal regression is often applied in practice. Therefore, it is important to understand under what assumptions it can be expected to work well. Using multiple related tasks, the signal in data can be more easily detected and the estimation procedure is more efficient. Our theoretical results support this intuition. One open question still

remains. It is still not clear how to match the lower bound on the Hamming distance given in Section 14.3, but we suspect that recent developments in [98] could provide tools to match the lower bound.

14.6 Technical Proofs

14.6.1 Tail bounds for Chi-squared variables

Throughout the proofs we will often use one of the following tail bounds for central χ^2 random variables. These are well known and proofs can be found in the original papers.

Lemma 14.1 ([118]). *Let $X \sim \chi^2_d$. For all $x \geq 0$,*

$$\mathbb{P}[X - d \geq 2\sqrt{dx} + 2x] \leq \exp(-x) \tag{14.13}$$
$$\mathbb{P}[X - d \leq -2\sqrt{dx}] \leq \exp(-x).$$

Lemma 14.2 ([100]). *Let $X \sim \chi^2_d$, then*

$$\mathbb{P}[|d^{-1}X - 1| \geq x] \leq \exp(-\frac{3}{16}dx^2), \quad x \in [0, \frac{1}{2}). \tag{14.14}$$

The following result provide a tail bound for non-central χ^2 random variable with non-centrality parameter ν.

Lemma 14.3 ([18]). *Let $X \sim \chi^2_d(\nu)$, then for all $x > 0$*

$$\mathbb{P}[X \geq (d + \nu) + 2\sqrt{(d + 2\nu)x} + 2x] \leq \exp(-x) \tag{14.15}$$
$$\mathbb{P}[X \leq (d + \nu) - 2\sqrt{(d + 2\nu)x}] \leq \exp(-x). \tag{14.16}$$

14.6.2 Spectral norms for random matrices

The following results can be found in literature on random matrix theory. We collect some useful results.

Lemma 14.4 ([54]). *Let $\mathbf{A} \in \mathbb{R}^{n\times k}$ be a random matrix from the standard Gaussian ensemble with $k < n$. Then for all $t > 0$*

$$\mathbb{P}[\Lambda_{\max}(n^{-1}\mathbf{A}'\mathbf{A} - \mathbf{I}_k) \geq f(n, k, t)] \leq 2\exp(-nt^2/2)$$

where $f(n, k, t) = 2(\sqrt{\frac{k}{n}} + t) + (\sqrt{\frac{k}{n}} + t)^2$.

The above results holds for random matrices whose elements are independent and identically distributed $\mathcal{N}(0, 1)$. The result can be extended to random matrices with correlated elements in each row.

Lemma 14.5 ([190]). *Let $A \in \mathbb{R}^{n\times k}$ be a random matrix with rows sampled iid from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Then for all $t > 0$*

$$\mathbb{P}[\Lambda_{\max}(n^{-1}\mathbf{A}'\mathbf{A} - \mathbf{\Sigma}) \geq \Lambda_{\max}(\mathbf{\Sigma})f(n, k, t)] \leq 2\exp(-nt^2/2). \tag{14.17}$$

Corollary 14.1. *Let $A \in \mathbb{R}^{n\times k}$ be a random matrix with rows sampled iid from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Then*

$$\mathbb{P}\left[|\Lambda_{\max}\left(n^{-1}\mathbf{A}'\mathbf{A}\right)| \geq 9\Lambda_{\max}\left(\mathbf{\Sigma}\right)\right] \leq 2\exp(-n/2).$$

14.6.3 Sample covariance matrix

Let $\mathbf{X} \in \mathbb{R}^{n\times p}$ be a random matrix whose rows are independent and identically distributed $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. The matrix $\Sigma = (\sigma_{ab})$ and denote $\rho_{ab} = (\sigma_{aa}\sigma_{bb})^{-1/2}\sigma_{ab}$. The following result provides element-wise deviation of the empirical covariance matrix $\widehat{\mathbf{\Sigma}} = n^{-1}\mathbf{X}'\mathbf{X}$ from the population quantity $\mathbf{\Sigma}$.

Lemma 14.6. *Let* $\nu_{ab} = \max\{(1-\rho_{ab})\sqrt{\sigma_{aa}\sigma_{bb}}, (1+\rho_{ab})\sqrt{\sigma_{aa}\sigma_{bb}}\}$. *Then or all* $t \in [0, \nu_{ab}/2)$

$$\mathbb{P}\left[|\widehat{\sigma}_{ab} - \sigma_{ab}| \geq t\right] \leq 4\exp\left(-\frac{3nt^2}{16\nu_{ab}^2}\right).$$

The proof is based on Lemma A.3. in [25] with explicit constants.

Proof. Let $x'_{ia} = x_{ia}/\sqrt{\sigma_{aa}}$. Then using (14.14)

$$\begin{aligned}
&\mathbb{P}[|\frac{1}{n}\sum_{i=1}^{n} x_{ia}x_{ib} - \sigma_{ab}| \geq t] \\
&= \mathbb{P}[|\frac{1}{n}\sum_{i=1}^{n} x'_{ia}x'_{ib} - \rho_{ab}| \geq \frac{t}{\sqrt{\sigma_{aa}\sigma_{bb}}}] \\
&= \mathbb{P}[|\sum_{i=1}^{n}((x'_{ia}+x'_{ib})^2 - 2(1+\rho_{ab})) - ((x'_{ia}-x'_{ib})^2 - 2(1-\rho_{ab}))| \geq \frac{4nt}{\sqrt{\sigma_{aa}\sigma_{bb}}}] \\
&\leq \mathbb{P}[|\sum_{i=1}^{n}((x'_{ia}+x'_{ib})^2 - 2(1+\rho_{ab}))| \geq \frac{2nt}{\sqrt{\sigma_{aa}\sigma_{bb}}}] \\
&\quad + \mathbb{P}[|\sum_{i=1}^{n}((x'_{ia}-x'_{ib})^2 - 2(1-\rho_{ab}))| \geq \frac{2nt}{\sqrt{\sigma_{aa}\sigma_{bb}}}] \\
&\leq 2\mathbb{P}[|\chi^2_n - n| \geq \frac{nt}{\nu_{ab}}] \leq 4\exp(-\frac{3nt^2}{16\nu_{ab}^2}),
\end{aligned}$$

where $\nu_{ab} = \max\{(1-\rho_{ab})\sqrt{\Sigma_{aa}\Sigma_{bb}}, (1+\rho_{ab})\sqrt{\Sigma_{aa}\Sigma_{bb}}\}$ and $t \in [0, \nu_a/2)$. □

This result implies that, for any $\delta \in (0,1)$, we have

$$\mathbb{P}\left[\sup_{0\leq a<b\leq p}|\widehat{\sigma}_{ab} - \sigma_{ab}| \leq 4\max_{ab}\nu_{ab}\sqrt{\frac{2\log 2d + \log(1/\delta)}{3n}}\right] \geq 1-\delta.$$

As a corollary of Lemma 14.6, we have a tail bound for sum of product-normal random variables.

Corollary 14.2. *Let* Z_1 *and* Z_2 *be two independent Gaussian random variables and let* $X_i \overset{iid}{\sim} Z_1Z_2$, $i = 1\ldots n$. *Then for* $t \in [0, 1/2)$

$$\mathbb{P}[|n^{-1}\sum_{i\in[n]} X_i| > t] \leq 4\exp(-\frac{3nt^2}{16}). \tag{14.18}$$

14.6.4 Proof of Theorem 14.1

We introduce some notation before providing the proof of Theorem 14.1. Consider a $p+1$ dimensional random vector $(Y, \mathbf{X}') = (Y, X_1, \ldots, X_p)$ and assume that

$$\begin{pmatrix} Y \\ \mathbf{X} \end{pmatrix} \sim \mathcal{N}\left(0, \mathbf{\Sigma}_F\right), \qquad \mathbf{\Sigma}_F = \begin{pmatrix} \sigma_{00} & \mathbf{C}' \\ \mathbf{C} & \mathbf{\Sigma} \end{pmatrix}$$

with $\mathbf{C} = (\sigma_{0b})_{b=1}^p = \mathbb{E}Y\mathbf{X} \in \mathbb{R}^p$ and $\mathbf{\Sigma} = (\sigma_{ab})_{a,b=1}^p = \mathbb{E}\mathbf{X}\mathbf{X}'$. Define

$$\mathbf{\Sigma}_F^{-1} = \mathbf{\Omega}_F = \begin{pmatrix} \omega_{00} & \mathbf{P}' \\ P & \mathbf{\Omega} \end{pmatrix},$$

with $\mathbf{P} = (\omega_{0b})_{b=1}^p$ and $\mathbf{\Omega} = (\omega_{ab})_{a,b=1}^p$. The partial correlation between Y and X_j is defined as

$$\rho_j \equiv \mathrm{Corr}\left(Y, X_j \mid X_{\setminus\{j\}}\right) = -\frac{\omega_{0j}}{\sqrt{\omega_{00}\omega_{jj}}}$$

Therefore, nonzero entries of the inverse covariance matrix correspond to nonzero partial correlation coefficients. For Gaussian models, $\rho_j = 0$ correspond to Y and X_j are conditionally independent given $X_{\setminus\{j\}}$. The relationship between the partial correlation estimation and a regression problem can be formulated by the following well-known proposition [130].

Proposition 14.1. *Consider the following regression model:*

$$Y = \sum_{j=1}^{p} \beta_j X_j + \epsilon, \;\; \epsilon \sim N(0, \mathrm{Var}(\epsilon))$$

Then ϵ is independent of $X_1, \ldots, X_d$ if and only if for all $j = 1, \ldots, p$

$$\beta_j = -\frac{\omega_{0j}}{\omega_{00}} = \rho_j \sqrt{\frac{\omega_{jj}}{\omega_{00}}}.$$

Furthermore, $\mathrm{Var}(\epsilon) = 1/\omega_{00}$.

Let $\mathbf{\Sigma}_{S^C|S} = \mathbf{\Sigma}_{S^C S^C} - \mathbf{\Sigma}_{S^C S}(\mathbf{\Sigma}_{SS})^{-1}\mathbf{\Sigma}_{SS^C}$ be the conditional covariance of $(X_{S^C}|X_S)$. We are now ready to prove Theorem 14.1.

Theorem 14.1. *Consider the regression model in* (14.1) *with* $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$, $\mathbf{x}_i \overset{iid}{\sim} \mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma})$, *and* $\boldsymbol{\epsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2\mathbf{I}_n)$ *with known* $\sigma > 0$, $\mathbf{X}$ *independent of* $\boldsymbol{\epsilon}$. *Assume that*

$$\max_{j \in S^C} |\mathbf{\Sigma}_{jS}\boldsymbol{\beta}_S| + \gamma_n(p, s, \boldsymbol{\beta}, \mathbf{\Sigma}, \delta) < \min_{j \in S} |\mathbf{\Sigma}_{jS}\boldsymbol{\beta}_S|$$

with

$$\begin{aligned}
\gamma_n(p, s, \boldsymbol{\beta}, \mathbf{\Sigma}, \delta) &= 8\Lambda_{\max}(\mathbf{\Sigma}_{SS})\sqrt{\frac{s}{n}}||\boldsymbol{\beta}_S||_2 \max_{j \in S^C}(1 + ||\mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}||_2) \\
&+ 4\left(\max_{j \in S^C}\sqrt{\frac{\mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}\mathbf{\Sigma}_{Sj}}{\omega_{00}}} + \max_{j \in S^C}\sqrt{[\mathbf{\Sigma}_{S^C|S}]_{jj}\sigma_{00}}\right)\sqrt{\frac{\log\frac{4(p-s)}{\delta}}{3n}} \\
&+ 4\max_{j \in S}\sqrt{\frac{\sigma_{jj}}{\omega_{00}}}\sqrt{\frac{\log\frac{4s}{\delta}}{3n}}
\end{aligned}$$

then

$$\mathbb{P}[\widehat{S}(s) = S] \geq 1 - 3\delta - 2\exp(-s/2).$$

Proof. Denoting $\widehat{c}_j = n^{-1}\sum_{i=1}^n y_i x_{ij}$, we would like to establish that

$$\max_{j \notin S} |\widehat{c}_j| \leq \min_{j \in S} |\widehat{c}_j|.$$

Using Proposition 14.1, for $j \in S^C$ we have $\mathbf{X}'_j = \mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}\mathbf{X}'_S + \mathbf{E}'_j$ with $\mathbf{E}_j = (e_{ij})$, $e_{ij} \sim \mathcal{N}(0, [\mathbf{\Sigma}_{S^C|S}]_{jj})$. Now

$$\begin{aligned}
\widehat{c}_j &= n^{-1}\mathbf{X}_j\mathbf{X}_S\boldsymbol{\beta}_S + n^{-1}\mathbf{X}_j\boldsymbol{\epsilon} \\
&= n^{-1}\mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}\mathbf{X}'_S(\mathbf{X}_S\boldsymbol{\beta}_S + \boldsymbol{\epsilon}) + n^{-1}\mathbf{E}'_j(\mathbf{X}_S\boldsymbol{\beta}_S + \boldsymbol{\epsilon}) \\
&= \mathbf{\Sigma}_{jS}\boldsymbol{\beta}_S + \mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}(\widehat{\mathbf{\Sigma}}_{SS} - \mathbf{\Sigma}_{SS})\boldsymbol{\beta}_S \\
&\quad + n^{-1}\mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}\mathbf{X}'_S\boldsymbol{\epsilon} + n^{-1}\mathbf{E}'_j(\mathbf{X}_S\boldsymbol{\beta}_S + \boldsymbol{\epsilon}),
\end{aligned} \tag{14.19}$$

where $\widehat{\mathbf{\Sigma}} = n^{-1}\mathbf{X}'\mathbf{X}$ is the empirical covariance matrix. Using (14.17) with $t = \sqrt{s/n}$ we have that

$$\begin{aligned}
&\max_{j \in S^C} |\mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}(\widehat{\mathbf{\Sigma}}_{SS} - \mathbf{\Sigma}_{SS})\boldsymbol{\beta}_S| \\
&\qquad \leq 8\Lambda_{\max}(\mathbf{\Sigma}_{SS})\sqrt{\frac{s}{n}}||\boldsymbol{\beta}_S||_2 \max_{j \in S^C} ||\mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}||_2
\end{aligned}$$

with probability at least $1 - 2\exp(-s/2)$. From (14.18) it follows that

$$\max_{j \in S^C} |n^{-1}\mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}\mathbf{X}'_S\boldsymbol{\epsilon}| \leq 4\max_{j \in S^C}\sqrt{\frac{\mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}\mathbf{\Sigma}_{Sj}}{\omega_{00}}}\sqrt{\frac{\log\frac{4(p-s)}{\delta}}{3n}}$$

with probability $1 - \delta$ and

$$\max_{j \in S^C} |n^{-1}\mathbf{E}'_j(\mathbf{X}_S\boldsymbol{\beta}_S + \boldsymbol{\epsilon})| \leq 4\max_{j \in S^C}\sqrt{[\mathbf{\Sigma}_{S^C|S}]_{jj}\sigma_{00}}\sqrt{\frac{\log\frac{4(p-s)}{\delta}}{3n}} \tag{14.20}$$

with probability $1 - \delta$. Combining (14.19)-(14.20)

$$\begin{aligned}
\max_{j \in S^C} |\widehat{c}_j| \leq\ & |\mathbf{\Sigma}_{jS}\boldsymbol{\beta}_S| + 8\Lambda_{\max}(\mathbf{\Sigma}_{SS})\sqrt{\frac{s}{n}}||\boldsymbol{\beta}_S||_2 \max_{j \in S^C} ||\mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}||_2 \\
&+ 4\max_{j \in S^C}\sqrt{\frac{\mathbf{\Sigma}_{jS}(\mathbf{\Sigma}_{SS})^{-1}\mathbf{\Sigma}_{Sj}}{\omega_{00}}}\sqrt{\frac{\log\frac{4(p-s)}{\delta}}{3n}} \\
&+ 4\max_{j \in S^C}\sqrt{[\mathbf{\Sigma}_{S^C|S}]_{jj}\sigma_{00}}\sqrt{\frac{\log\frac{4(p-s)}{\delta}}{3n}}
\end{aligned} \tag{14.21}$$

with probability $1 - 2\delta - 2\exp(-s/2)$.

Similarly we can show for $j \in S$ that

$$\begin{aligned}\min_{j\in S}|\widehat{c}_j| &\geq \min|\boldsymbol{\Sigma}_{SS}\boldsymbol{\beta}_S| - \Lambda_{\max}(\widehat{\boldsymbol{\Sigma}}_{SS}-\boldsymbol{\Sigma}_{SS})||\boldsymbol{\beta}_S||_2 - \max|n^{-1}\mathbf{X}_S'\boldsymbol{\epsilon}| \\ &\geq \min|\boldsymbol{\Sigma}_{SS}\boldsymbol{\beta}_S| - 8\Lambda_{\max}(\boldsymbol{\Sigma}_{SS})\sqrt{\frac{s}{n}}||\boldsymbol{\beta}_S||_2 - 4\max_{j\in S}\sqrt{\frac{\sigma_{jj}}{\omega_{00}}}\sqrt{\frac{\log\frac{4s}{\delta}}{3n}}\end{aligned} \tag{14.22}$$

with probability $1-\delta-2\exp(-s/2)$. The theorem now follows from (14.21) and (14.22). □

14.6.5 Proof of Theorem 14.2

In this section we prove Theorem 14.2. Define $S_{-j} := S\backslash\{j\}$ and let

$$\widetilde{\sigma}_j^2 := \sigma_{jj} - \boldsymbol{\Sigma}_{jS_{-j}}(\boldsymbol{\Sigma}_{S_{-j}S_{-j}})^{-1}\boldsymbol{\Sigma}_{S_{-j}j}$$

denote the variance of $(X_{j_s}|\mathbf{X}_{S_{-j_s}})$, $j \in S$. The theorem is restated below.

Theorem 14.2. *Assume that the conditions of Theorem 14.1 are satisfied. Let*

$$\iota = \sqrt{\frac{16\log(16/\delta)}{3(n-s+1)}}$$

and assume that $\iota < \frac{1}{2}$. Furthermore, assume that

$$\max_{j\in S}\left\{\frac{2\sigma^2\log(4n/\delta)}{\beta_j^2\widetilde{\sigma}_j^2(1-\iota)} + \frac{2\sigma\sqrt{2(1+\iota)\log(8n/\delta)}}{\beta_j\widetilde{\sigma}_j(1-\iota)}\right\} < 1.$$

Then

$$\mathbb{P}[\widehat{S}(\widehat{s}_n) = S] \geq 1 - 4\delta - 2\exp(-s/2).$$

Proof. Define the event

$$\mathcal{E}_n = \{\widehat{S}(s) = S\}.$$

From Theorem 14.1,

$$\mathbb{P}[\mathcal{E}_n^C] \leq 3\delta + 2\exp(-s/2). \tag{14.23}$$

We proceed to show that for some small $\delta' > 0$

$$\mathbb{P}[\widehat{s}_n \neq s] \leq \mathbb{P}[\widehat{s}_n \neq s|\mathcal{E}_n]\mathbb{P}[\mathcal{E}_n] + \mathbb{P}[\mathcal{E}_n^C] \leq \delta',$$

which will prove the theorem together with (14.23). An upper bound on $\mathbb{P}[\widehat{s}_n \neq s|\mathcal{E}_n]$ is constructed by combining upper bounds on $\mathbb{P}[\widehat{s}_n > s|\mathcal{E}_n]$ and $\mathbb{P}[\widehat{s}_n < s|\mathcal{E}_n]$.

Let $\tau = 2\sigma^2\log\frac{4n}{\delta}$. From $\{\widehat{s}_n > s|\mathcal{E}_n\} \subseteq \cup_{k=s}^{p-1}\{\widehat{\xi}_n(k) \geq \tau|\mathcal{E}_n\}$ follows that

$$\mathbb{P}[\widehat{s}_n > s|\mathcal{E}_n] \leq \sum_{k=s}^{p-1}\mathbb{P}[\widehat{\xi}_n(k) \geq \tau|\mathcal{E}_n]. \tag{14.24}$$

Recalling definitions of $\widehat{V}_n(k)$ and $\widehat{\mathbf{H}}_n(k)$ from p. 213, for a fixed $s \leq k \leq p-1$, $\widehat{\mathbf{H}}(k+1)-\widehat{\mathbf{H}}(k)$ is the projection matrix from $\mathbb{R}^n$ to $\widehat{V}_n(k+1) \cap \widehat{V}_n(k)^{\perp}$. Recall also that we are using the second half of the sample to estimate $\widehat{s}_n$, which implies that the projection matrix $\widehat{\mathbf{H}}(k)$ is independent of $\boldsymbol{\epsilon}$ for all k. Now, exactly one of the two events $\{\widehat{V}_n(k) = \widehat{V}_n(k+1)\}$ and $\{\widehat{V}_n(k) \subsetneq \widehat{V}_n(k+1)\}$ occur. On the event $\{\widehat{V}_n(k) = \widehat{V}_n(k+1)\}$, $\widehat{\xi}_n(k) = 0$. We analyze the event $\{\widehat{V}_n(k) \subsetneq \widehat{V}_n(k+1)\} \cap \mathcal{E}_n$ by conditioning on $\mathbf{X}$. Since $\widehat{\mathbf{H}}(k+1) - \widehat{\mathbf{H}}(k)$ is a rank one projection matrix

$$\widehat{\xi}_n(k) = ||(\widehat{\mathbf{H}}(k+1) - \widehat{\mathbf{H}}(k))\mathbf{y}||_2^2 = ||(\widehat{\mathbf{H}}(k+1) - \widehat{\mathbf{H}}(k))\boldsymbol{\epsilon}||_2^2 \stackrel{d}{=} \sigma^2\chi_1^2.$$

Furthermore, $(\widehat{\mathbf{H}}(k+1) - \widehat{\mathbf{H}}(k))\boldsymbol{\epsilon} \perp\!\!\!\perp (\widehat{\mathbf{H}}(k'+1) - \widehat{\mathbf{H}}(k'))\boldsymbol{\epsilon}$, $k \neq k'$. It follows that for any realization of the sequences $\widehat{V}_n(1), \ldots, \widehat{V}_n(p)$,

$$\begin{aligned}
&\sum_{k=s}^{p-1} \mathbb{P}[\widehat{\xi}_n(k) \geq \tau | \mathcal{E}_n] \\
&\quad = \sum_{k=s}^{p-1} \mathbb{P}[\widehat{\xi}_n(k) \geq \tau | \{\widehat{V}_n(k) \subsetneq \widehat{V}_n(k+1)\} \cap \mathcal{E}_n] \mathbb{P}[\widehat{V}_n(k) \subsetneq \widehat{V}_n(k+1)] \\
&\quad = \mathbb{P}[\sigma^2\chi_1^2 \geq \tau] \mathbb{E} \sum_{k=s}^{p-1} \mathbb{I}\{\widehat{V}_n(k) \subsetneq \widehat{V}_n(k+1)\} \\
&\quad \leq n\mathbb{P}[\sigma^2\chi_1^2 \geq \tau],
\end{aligned}$$

where the first equality follows since $\{\widehat{V}_n(k) \subsetneq \widehat{V}_n(k+1)\}$ is independent of $\mathcal{E}_n$. Combining with (14.24) gives

$$\mathbb{P}[\widehat{s}_n > s | \mathcal{E}_n] \leq n\mathbb{P}[\sigma^2\chi_1^2 \geq \tau] \leq \delta/2$$

using a standard normal tail bound.

Next, we focus on bounding $\mathbb{P}[\widehat{s}_n < s | \mathcal{E}_n]$. Since $\{\widehat{s}_n < s | \mathcal{E}_n\} \subset \{\widehat{\xi}_n(s-1) < \tau | \mathcal{E}_n\}$, we can bound $\mathbb{P}[\widehat{\xi}_n(s-1) < \tau | \mathcal{E}_n]$. Using the definition of $\widehat{\mathbf{H}}(s)$ it is straightforward to obtain that

$$(\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))\mathbf{y} = (\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))(\mathbf{X}_{j_s}\beta_{j_S} + \boldsymbol{\epsilon}).$$

Using Proposition 14.1, we can write $\mathbf{X}'_{j_s} = \boldsymbol{\Sigma}_{j_s S_{-j_s}}(\boldsymbol{\Sigma}_{S_{-j_s}S_{-j_s}})^{-1}\mathbf{X}'_{S_{-j_s}} + \mathbf{E}'$ where $\mathbf{E} = (e_i)$, $e_i \stackrel{iid}{\sim} \mathcal{N}(0, \widetilde{\sigma}^2_{j_s})$. Then

$$\begin{aligned}
(\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))\mathbf{y} &= (\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))(\mathbf{E}\beta_{j_S} + \boldsymbol{\epsilon}) \\
&= (\mathbf{I}_n - \widehat{\mathbf{H}}(s-1))\mathbf{E}\beta_{j_S} + (\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))\boldsymbol{\epsilon}.
\end{aligned}$$

Define

$$T_1 = \beta_{j_S}^2 \mathbf{E}'(\mathbf{I}_n - \widehat{\mathbf{H}}(s-1))\mathbf{E}$$

and

$$T_2 = \boldsymbol{\epsilon}'(\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))\boldsymbol{\epsilon}.$$

Conditional on $\mathbf{X}_{S_{-j_s}}$, $T_1 \stackrel{d}{=} \beta^2_{j_S}\widetilde{\sigma}^2_{j_s}\chi^2_{n-s+1}$ since $\mathbf{E} \perp\!\!\!\perp \mathbf{X}_{S_{-j_s}}$, and conditional on $\mathbf{X}_S$, $T_2 \stackrel{d}{=} \sigma^2\chi^2_1$. Define the events

$$\mathcal{A}_1 = \{\beta^2_{j_S}\widetilde{\sigma}^2_{j_s}(1-\iota) \leq T_1 \leq \beta^2_{j_S}\widetilde{\sigma}^2_{j_s}(1+\iota)\}$$

and

$$\mathcal{A}_2 = \{T_2 \leq 2\sigma^2 \log\frac{8n}{\delta}\}.$$

From Eq. (14.14), $\mathbb{P}[\mathcal{A}_1(\iota)^C] \leq \delta/4$, and using a normal tail bound, $\mathbb{P}[\mathcal{A}_2^C] < \delta/4$. Setting

$$\widetilde{\tau} = \tau + 2\beta_{j_S}\widetilde{\sigma}_{j_s}\sigma\sqrt{2(1+\iota)\log\frac{8n}{\delta}},$$

under the assumptions of theorem

$$\begin{aligned}\mathbb{P}[\widehat{\xi}_n(s-1) < \tau|\mathcal{E}_n] &\leq \mathbb{P}[T_1 + T_2 < \tau + 2\sqrt{T_1T_2}|\mathcal{E}_n] \\ &\leq \mathbb{P}[\beta^2_{j_S}\widetilde{\sigma}^2_{j_s}(1-\iota) < \widetilde{\tau}] + \mathbb{P}[\mathcal{A}_1^C] + \mathbb{P}[\mathcal{A}_2^C] \\ &\leq \frac{\delta}{2}.\end{aligned} \tag{14.25}$$

Combining (14.23)-(14.25), we have that $\mathbb{P}[\widehat{S}(\widehat{s}_n) = S] \geq 1 - 4\delta - 2\exp(-s/2)$, which completes the proof.

□

14.6.6 Proof of Theorem 14.3

We proceed to show that (14.7) holds with high probability under the assumptions of the theorem. We start with the case when $\Phi(\cdot) = ||\cdot||_2$. Let $\sigma_n^2 = \sigma^2/n$ and $\nu_j = \sigma_n^{-2}\sum_{k\in[T]}(\Sigma_{jS_k}\beta_{kS_k})^2$. With this notation, it is easy to observe that $\Phi^2(\{\widehat{\mu}_{kj}\}_k) \sim \sigma_n^2\chi^2_T(\nu_j)$ where $\chi^2_T(\nu_j)$ is a non-central chi-squared random variable with T degrees of freedom and non-centrality parameter ν_j. From (14.15),

$$\sigma_n^{-2}\max_{j\in S^C}\Phi^2(\{\widehat{\mu}_{kj}\}_k) \leq T + 2\log\frac{2(p-s)}{\delta} + \max_{j\in S^C}\nu_j + 2\sqrt{(T+2\nu_j)\log\frac{2(p-s)}{\delta}}$$

with probability at least $1-\delta/2$. Similarly, from (14.16),

$$\sigma_n^{-2}\min_{j\in S}\Phi^2(\{\widehat{\mu}_{kj}\}_k) \geq T + \min_{j\in S}\nu_j - \max_{j\in S}2\sqrt{(T+2\nu_j)\log\frac{2s}{\delta}}$$

with probability at least $1-\delta/2$. Combining the last two displays we have shown that (14.8) is sufficient to show that $\mathbb{P}[\widehat{S}_{\ell_2}(s) = S] \geq 1-\delta$.

Next, we proceed with $\Phi(\cdot) = ||\cdot||_1$, which can be dealt with similarly as the previous case. Using (14.13) together with $||\mathbf{a}||_1 \leq \sqrt{p}||\mathbf{a}||_2$, $a \in \mathbb{R}^p$,

$$\max_{j\in S^C}\sum_{k\in[T]}|\widehat{\mu}_{kj}| \leq \max_{j\in S^C}\sum_{k\in[T]}|\boldsymbol{\Sigma}_{jS_k}\boldsymbol{\beta}_{kS_k}| + \sigma_n\sqrt{T^2 + 2T\sqrt{T\log\frac{2(p-s)}{\delta}} + 2T\log\frac{2(p-s)}{\delta}}$$

with probability at least $1-\delta/2$. Similarly,

$$\min_{j\in S}\sum_{k\in[T]}|\widehat{\mu}_{kj}| \geq \min_{j\in S}\sum_{k\in[T]}|\boldsymbol{\Sigma}_{jS_k}\boldsymbol{\beta}_{kS_k}| - \sigma_n\sqrt{T^2+2T\sqrt{T\log\frac{2s}{\delta}}+2T\log\frac{2s}{\delta}}$$

with probability $1-\delta/2$. Combining the last two displays we have shown that (14.9) is sufficient to show that $\mathbb{P}[\widehat{S}_{\ell_1}(s)=S]\geq 1-\delta$.

We complete the proof with the case when $\Phi(\cdot)=||\cdot||_\infty$. Using a standard normal tail bound together with union bound

$$\max_{j\in S^C}\Phi(\{\widehat{\mu}_{kj}\}_k) \leq \max_{j\in S^C}\max_{k\in[T]}|\boldsymbol{\Sigma}_{jS_k}\boldsymbol{\beta}_{kS_k}| + \sigma_n\sqrt{2\log\frac{2(p-s)T}{\delta}}$$

with probability $1-\delta/2$ and

$$\min_{j\in S}\Phi(\{\widehat{\mu}_{kj}\}_k) \geq \min_{j\in S}\max_{k\in[T]}|\boldsymbol{\Sigma}_{jS_k}\boldsymbol{\beta}_{kS_k}| - \sigma_n\sqrt{2\log\frac{2sT}{\delta}}$$

with probability $1-\delta/2$, where $\sigma_n^2=\sigma^2/n$. This shows that (14.10) is sufficient to show that $\mathbb{P}[\widehat{S}_{\ell_\infty}(s)=S]\geq 1-\delta$.

14.6.7 Proof of Theorem 14.4

We proceed as in the proof of 14.2. Define the event

$$\mathcal{E}_n=\{\widehat{S}_\phi(s)=S\}.$$

Irrespective of which scoring function Φ is used, Theorem 14.1 provides the sufficient conditions under which $\mathbb{P}[\mathcal{E}_n^C]\leq\delta$. It remains to upper bound $\mathbb{P}[\widehat{s}_n\neq s|\mathcal{E}_n]$, since

$$\mathbb{P}[\widehat{s}_n\neq s]\leq\mathbb{P}[\widehat{s}_n\neq s|\mathcal{E}_n]\mathbb{P}[\mathcal{E}_n]+\mathbb{P}[\mathcal{E}_n^C]. \tag{14.26}$$

An upper bound on $\mathbb{P}[\widehat{s}_n\neq s|\mathcal{E}_n]$ is constructed by combining upper bounds on $\mathbb{P}[\widehat{s}_n>s|\mathcal{E}_n]$ and $\mathbb{P}[\widehat{s}_n<s|\mathcal{E}_n]$.

Let $\tau=(T+2\sqrt{T\log(2/\delta)}+2\log(2/\delta))\sigma^2$. From $\{\widehat{s}_n>s|\mathcal{E}_n\}\subseteq\cup_{k=s}^{p-1}\{\widehat{\xi}_{\ell_2,n}(k)\geq\tau|\mathcal{E}_n\}$ follows that

$$\mathbb{P}[\widehat{s}_n>s|\mathcal{E}_n]\leq\sum_{k=s}^{p-1}\mathbb{P}[\widehat{\xi}_{\ell_2,n}(k)\geq\tau|\mathcal{E}_n]. \tag{14.27}$$

For a fixed $s\leq k\leq p-1$, $\widehat{\mathbf{H}}(k+1)-\widehat{\mathbf{H}}(k)$ is the projection matrix from $\mathbb{R}^n$ to $\widehat{V}_n(k+1)\cap\widehat{V}_n(k)^\perp$. Since we are estimating $\widehat{s}_n$ on the second half of the samples, the projection matrix $\widehat{\mathbf{H}}(k)$ is independent of $\boldsymbol{\epsilon}$ for all k. Now, exactly one of the two events $\{\widehat{V}_n(k)=\widehat{V}_n(k+1)\}$ and $\{\widehat{V}_n(k)\subsetneq\widehat{V}_n(k+1)\}$ occur. On the event $\{\widehat{V}_n(k)=\widehat{V}_n(k+1)\}$, $\widehat{\xi}_{\ell_2,n}(k)=0$. Next we analyze the event $\{\widehat{V}_n(k)\subsetneq\widehat{V}_n(k+1)\}\cap\mathcal{E}_n$. Since $\widehat{\mathbf{H}}(k+1)-\widehat{\mathbf{H}}(k)$ is a rank one projection matrix

$$\widehat{\xi}_{\ell_2,n}(k)=\sum_{t\in[T]}||(\widehat{\mathbf{H}}(k+1)-\widehat{\mathbf{H}}(k))\mathbf{y}_t||_2^2=\sum_{t\in[T]}||(\widehat{\mathbf{H}}(k+1)-\widehat{\mathbf{H}}(k))\boldsymbol{\epsilon}_t||_2^2\overset{d}{=}\sigma^2\chi_T^2.$$

Furthermore, $\widehat{\xi}_{\ell_2,n}(k) \perp\!\!\!\perp \widehat{\xi}_{\ell_2,n}(k')$, $k \neq k'$. It follows that for any realization of the sequences $\widehat{V}_n(1), \ldots, \widehat{V}_n(p)$,

$$\begin{aligned}
&\sum_{k=s}^{p-1} \mathbb{P}[\widehat{\xi}_{\ell_2,n}(k) \geq \tau | \mathcal{E}_n] \\
&\quad = \sum_{k=s}^{p-1} \mathbb{P}[\widehat{\xi}_{\ell_2,n}(k) \geq \tau | \{\widehat{V}_n(k) \subsetneq \widehat{V}_n(k+1)\} \cap \mathcal{E}_n] \mathbb{P}[\widehat{V}_n(k) \subsetneq \widehat{V}_n(k+1)] \\
&\quad = \mathbb{P}[\sigma^2 \chi_T^2 \geq \tau] \mathbb{E} \sum_{k=s}^{p-1} \mathbb{I}\{\widehat{V}_n(k) \subsetneq \widehat{V}_n(k+1)\} \\
&\quad \leq n \mathbb{P}[\sigma^2 \chi_T^2 \geq \tau],
\end{aligned}$$

where the first equality follows since $\{\widehat{V}_n(k) \subsetneq \widehat{V}_n(k+1)\}$ is independent of $\mathcal{E}_n$. Combining with (14.27) gives

$$\mathbb{P}[\widehat{s}_n > s | \mathcal{E}_n] \leq n \mathbb{P}[\sigma^2 \chi_T^2 \geq \tau] \leq \delta/2 \tag{14.28}$$

using (14.13).

Next, we focus on bounding $\mathbb{P}[\widehat{s}_n < s | \mathcal{E}_n]$. Since $\{\widehat{s}_n < s | \mathcal{E}_n\} \subset \{\widehat{\xi}_{\ell_2,n}(s-1) < \tau | \mathcal{E}_n\}$, it is sufficient to bound $\mathbb{P}[\widehat{\xi}_{\ell_2,n}(s-1) < \tau | \mathcal{E}_n]$. Using the definition of $\widehat{\mathbf{H}}(s)$ it is straightforward to obtain that

$$(\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))\mathbf{y}_t = (\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))(\mathbf{X}_{j_s}\beta_{tj_s} + \boldsymbol{\epsilon}_t).$$

Write $\mathbf{X}_{j_s} = \mathbf{X}_{j_s}^{(1)} + \mathbf{X}_{j_s}^{(2)}$ where $\mathbf{X}_{j_s}^{(1)} \in \widehat{V}_n(s-1)$ and $\mathbf{X}_{j_s}^{(2)} \in \widehat{V}_n(s) \cap \widehat{V}_n(s-1)^{\perp}$. Then

$$(\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))\mathbf{y}_t = (\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))(\mathbf{X}_{j_s}^{(2)}\beta_{tj_s} + \boldsymbol{\epsilon}_t).$$

Furthermore we have that

$$||(\widehat{\mathbf{H}}(s) - \widehat{\mathbf{H}}(s-1))(\mathbf{X}_{j_s}^{(2)}\beta_{tj_s} + \boldsymbol{\epsilon}_t)||_2^2 = (||\mathbf{X}_{j_s}^{(2)}\beta_{tj_s}||_2 + Z_t)^2$$

where $Z_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$. It follows that $\widehat{\xi}_{\ell_2,n}(s-1) \sim \sigma^2 \chi_T^2(\nu)$ with $\nu = \sigma^{-2} \sum_{t\in[T]} ||\mathbf{X}_{j_s}^{(2)}\beta_{tj_s}||_2^2$. It is left to show that

$$\mathbb{P}[\sigma^2 \chi_T^2(\nu) < \tau] \leq \delta/2. \tag{14.29}$$

Using (14.16) and following the proof of Theorem 2 in [22], we have that (14.29) holds if

$$\nu > 2\sqrt{5}\log^{1/2}\left(\frac{4}{\delta^2}\right)\sqrt{T} + 8\log\left(\frac{4}{\delta^2}\right).$$

Under the assumptions, we have that

$$\min_{j\in S} \sum_{t\in[T]} ||\mathbf{X}_j^{(2)}\beta_{tj}||_2^2 > \left[2\sqrt{5}\log^{1/2}\left(\frac{4}{\delta^2}\right)\sqrt{T} + 8\log\left(\frac{4}{\delta^2}\right)\right]\sigma^2$$

which shows (14.29). Combining (14.28) and (14.29), we obtain (14.26) which completes the proof.

14.6.8 Proof of Theorem 14.5

We have

$$H_p(\widehat{S}, S \mid \mathbf{X}) \geq \sum_{j=1}^{p} \left[\mathbb{P}\left(\|\beta_{\cdot j}\|_2 = 0, \|\widehat{\beta}_{\cdot j}\|_2 \neq 0 \right) + \mathbb{P}\left(\|\beta_{\cdot j}\|_2 \neq 0, \|\widehat{\beta}_{\cdot j}\|_2 = 0 \right) \right].$$

For $1 \leq j \leq p$, we consider the hypothesis testing:

$$H_{0,j}: \ \|\beta_{\cdot j}\|_2 = 0 \ \text{ vs. } \ \|\beta_{\cdot j}\|_2 \neq 0.$$

For $1 \leq t \leq T$, we denote by $\boldsymbol{\beta}_t$ any empirical realization of the coefficient vector. Let $\widetilde{\boldsymbol{\beta}}_t := \boldsymbol{\beta}_t - \beta_{tj} e_j$ where e_j is the j-th canonical basis of $\mathbb{R}^p$. We define

$$h(\mathbf{y}; \widetilde{\beta}, \boldsymbol{\alpha}) := h(\mathbf{y}_1, \ldots, \mathbf{y}_T; \widetilde{\boldsymbol{\beta}}_1, \ldots, \widetilde{\boldsymbol{\beta}}_T, \alpha_1, \ldots, \alpha_T)$$

to be the joint distribution of

$$\mathbf{y}_1, \ldots, \mathbf{y}_T \sim \prod_{t=1}^{T} \mathcal{N}\left(\mathbf{X}\left(\widetilde{\boldsymbol{\beta}}_t + \alpha_t e_j \right), \mathbf{I}_n \right).$$

We then have

$$h(\mathbf{y}; \widetilde{\boldsymbol{\beta}}, \boldsymbol{\alpha}) = h(\mathbf{y}; \widetilde{\boldsymbol{\beta}}, 0) \cdot \exp\left(\sum_{t=1}^{T} \alpha_t x_j'(\mathbf{y}_t - \mathbf{X}\widetilde{\boldsymbol{\beta}}_t) - \sum_{t=1}^{T} \frac{\alpha_t^2}{2} \right).$$

Let $\max_{1 \leq t \leq T} |\alpha_t| \leq \tau_p$. We define

$$h(\mathbf{y}; \widetilde{\beta}, \tau_p) = h(\mathbf{y}; \widetilde{\beta}, 0) \cdot \exp\left(\tau_p \sum_{t=1}^{T} x_j'(\mathbf{y}_t - \mathbf{X}\widetilde{\boldsymbol{\beta}}_t) - \frac{T\tau_p^2}{2} \right).$$

Let $G(\widetilde{\boldsymbol{\beta}})$ be the joint distribution of $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T$. Using Neyman-Pearson Lemma, Fubinni's Theorem and some basic calculus, we have

$$\begin{aligned}
&\mathbb{P}\left(\|\beta_{\cdot j}\|_2 = 0, \|\widehat{\beta}_{\cdot j}\|_2 \neq 0 \right) + \mathbb{P}\left(\|\beta_{\cdot j}\|_2 \neq 0, \|\widehat{\beta}_{\cdot j}\|_2 = 0 \right) \\
&\geq \frac{1}{2} - \frac{1}{2} \int \left[\int \left| (1-\eta_p) h(\mathbf{y}; \widetilde{\boldsymbol{\beta}}, 0) - \eta_p h(\mathbf{y}; \widetilde{\boldsymbol{\beta}}, \boldsymbol{\alpha}) \right| d\mathbf{y} \right] d\pi_p(\boldsymbol{\alpha}) dG(\widetilde{\boldsymbol{\beta}}) \\
&= \frac{1}{2} - \frac{1}{2} \int H(\widetilde{\boldsymbol{\beta}}, \boldsymbol{\alpha}) d\pi_p(\boldsymbol{\alpha}) dG(\widetilde{\boldsymbol{\beta}}),
\end{aligned}$$

where

$$H(\widetilde{\boldsymbol{\beta}}, \boldsymbol{\alpha}) \equiv \int \left| (1-\eta_p) h(\mathbf{y}; \widetilde{\boldsymbol{\beta}}, 0) - \eta_p h(\mathbf{y}; \widetilde{\boldsymbol{\beta}}, \boldsymbol{\alpha}) \right| d\mathbf{y}.$$

It can be seen that

$$H(\widetilde{\boldsymbol{\beta}}, \boldsymbol{\alpha}) \leq H(\widetilde{\boldsymbol{\beta}}, \tau_p).$$

We then have

$$\mathbb{P}\left(\|\beta_{\cdot j}\|_2 = 0, \|\widehat{\beta}_{\cdot j}\|_2 \neq 0\right) + \mathbb{P}\left(\|\beta_{\cdot j}\|_2 \neq 0, \|\widehat{\beta}_{\cdot j}\|_2 = 0\right) \geq \frac{1}{2} - \frac{1}{2}\int H(\widetilde{\boldsymbol{\beta}}, \tau_p) dG(\widetilde{\boldsymbol{\beta}}).$$

For any realization of $\widetilde{\beta}_1, \ldots, \widetilde{\beta}_p$, we define

$$D_p(\widetilde{\boldsymbol{\beta}}) := \left\{ \mathbf{y}_1, \ldots, \mathbf{y}_T : \eta_p \cdot \exp\left(\tau_p \sum_{t=1}^{T} x_j'(\mathbf{y}_t - \mathbf{X}\widetilde{\boldsymbol{\beta}}_t) - \frac{T\tau_p^2}{2} \right) > (1-\eta_p) \right\}.$$

We know that $\mathbf{y}_1, \ldots, \mathbf{y}_T \in D_p(\widetilde{\boldsymbol{\beta}})$ if and only if

$$W_j = \sum_{t=1}^{T} x_j^{'}(\mathbf{y}_t - \mathbf{X}\widetilde{\boldsymbol{\beta}}_t) > \lambda_p.$$

It is then easy to see that

$$W_j \sim \mathcal{N}(0, T) \text{ under } H_{0,j}$$
$$W_j \sim \mathcal{N}(T\tau_p, T) \text{ under } H_{1,j}.$$

Following exactly the same argument as in Lemma 6.1 from Ji and Jin (2011), we obtain the lower bound:

$$\frac{1}{2} - \frac{1}{2}H(\widetilde{\boldsymbol{\beta}}, \tau_p) \geq (1-\eta_p)\overline{F}\left(\frac{\lambda_p}{\sqrt{T}}\right) + \eta_p F\left(\frac{\lambda_p}{\sqrt{T}} - \sqrt{T}\tau_p\right).$$

Thus we finish the proof of the main argument (14.12).

To obtain more detailed rate, we have

$$\frac{1}{\eta_p} - 1 = p^v - 1.$$

Also,

$$\begin{aligned}
\overline{\Phi}\left(\frac{\lambda_p}{\sqrt{T}}\right) &\geq \frac{\sqrt{T}}{2\lambda_p}\phi\left(\frac{\lambda_p}{\sqrt{T}}\right) \\
&\geq \frac{\sqrt{rT}}{(v+Tr)\sqrt{2\log p}} \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(v+Tr)^2 \log p}{4rT}\right) \\
&= \frac{\sqrt{rT}}{2(v+Tr)\sqrt{\pi \log p}} \cdot p^{-(v+Tr)^2/(4rT)}.
\end{aligned}$$

Therefore

$$\frac{1-\eta_p}{\eta_p}\overline{F}\left(\frac{\lambda_p}{\sqrt{T}}\right) \asymp \frac{\sqrt{rT}}{2(v+Tr)\sqrt{\pi \log p}} \cdot p^{v-(v+Tr)^2/(4rT)}$$

$$= \frac{\sqrt{rT}}{2(v+Tr)\sqrt{\pi \log p}} \cdot p^{-(v-Tr)^2/(4rT)}.$$

We then evaluate the second term

$$F\left(\frac{\lambda_p}{\sqrt{T}} - \sqrt{T}\tau_p\right) = \overline{F}\left(\sqrt{T}\tau_p - \frac{\lambda_p}{\sqrt{T}}\right).$$

First, we have that

$$\frac{\lambda_p}{\sqrt{T}} - \sqrt{T}\tau_p = \frac{(v+Tr)\sqrt{\log p}}{\sqrt{2Tr}} - \sqrt{2rT\log p}.$$

If $v > Tr$, we have

$$\frac{\lambda_p}{\sqrt{T}} - \sqrt{T}\tau_p \to \infty,$$

which implies that

$$F\left(\frac{\lambda_p}{\sqrt{T}} - \sqrt{T}\tau_p\right) \geq 1 + o(1).$$

Now, we consider the case that $v < Tr$,

$$\begin{aligned}
\overline{F}\left(\sqrt{T}\tau_p - \frac{\lambda_p}{\sqrt{T}}\right) &= \overline{F}\left(\frac{(Tr-v)\sqrt{\log p}}{\sqrt{2Tr}}\right) \\
&\geq \frac{\sqrt{2Tr}}{(Tr-v)\sqrt{\log p}} \frac{1}{\sqrt{2\pi}} \cdot p^{-(v-Tr)^2/(4rT)} \\
&= \frac{\sqrt{Tr}}{(Tr-v)\sqrt{\pi \log p}} \cdot p^{-(v-Tr)^2/(4rT)}
\end{aligned}$$

This finishes the whole proof.

Part III

Conclusions and Future Work

Chapter 15

Conclusions and Future Directions

Black-box models are not useful for scientific discovery because they do not provide insights about a system under consideration. Finding interesting and interpretable structure is important in many sciences, ranging from systems biology, statistical genetics, and computer science to various social sciences. In this thesis, we have developed principled machine learning techniques, with strong theoretical guarantees, that are capable of uncovering mechanisms underlying complex systems. When data are high-dimensional and generated by some unknown process, it is important to have flexible models that provide insights into data generating mechanisms and allow for discovering of new scientific facts.

In this thesis, we have focused on two specific problems where experimental techniques are expensive, not sufficient, or not available to uncover mechanisms underlying a complex system. In these cases, statistical tools are needed. We have addressed the following questions:

1. Given noisy observations collected from a complex system, how can we find a dynamic network, which encodes and explains relationships of interest in the system?
2. How can we identify features that are relevant for a number of high-dimensional, noisy learning tasks in a fast and reliable way?

For all of these problems, an important question is under what circumstances are statistical methods going to reliably identify the underlying structure of interest; and, furthermore, which procedure can be used to identify the structure quickly.

In the first part of the thesis, we have focused on methods for uncovering dynamic network structure from nodal observations, while in the second part, we have analyzed methods for variable selection in multi-task learning problems. We have focused on applications arising in systems biology, social media analysis and economics; However, our results are applicable in many other modern scientific fields, ranging from cognitive neuroscience to computational meteorology.

15.1 Learning and exploring network structure

In this thesis, we have developed a comprehensive framework of time-varying networks for uncovering structure of dynamic networks from noisy observational data, based on rigorous statistical formalism with provable guarantees. We see it as the first step towards building a dynamic

network analysis system for understanding complex network entities and how they evolve and interact over time. The framework is especially useful in scientific domains where interactions cannot be easily measured, but noisy and indirect versions of nodal attributes are available, which prevents scientists from an in-depth investigation of the mechanisms underlying a system of interest. Using the framework of time-varying networks, researchers can reverse-engineer active interactions between entities in a system from observed longitudinal data and postulate more precise hypotheses about processes undergoing changes in networks. As an exploratory tool, they are indispensable for capturing transient events in the dynamic system, and have the potential to change the way people analyze complex, dynamic systems and networks.

The new framework of time-varying networks is a semiparametric generalization of the classical framework of probabilistic graphical models, which allows for both flexibility in modeling many effects of interest and development of efficient and scalable estimation procedures. Estimation in the framework is done by solving convex optimization programs, based on penalized empirical risk minimization, for which we have developed efficient methods, including the proximal gradient descent. Furthermore, we have identified sufficient conditions for correct recovery of the underlying network structure with high probability, for different models in the framework. The framework can model a number of interesting scenarios that could arise in a dynamic system, e.g., a smoothly evolving network during regular development of a biological organism; or a network undergoing dramatic reorganization, possibly in response to harsh economic and political changes during a crisis, or a cell response to a virus. We used the time-varying network framework to identify patterns of interactions between genes in fruit flies as they go through the developmental process. We have also demonstrated applicability to social media analysis by learning a latent time-varying network between senators from the US Senate voting records.

We have also studied a couple of related network structure learning problems. We have studied uncovering structure of covariate indexed networks, where interactions between nodes depend on an external covariate. For example, when nodes represent stock prices, it is of interest to understand which stock prices jointly rise or fall, and this relationship may change depending on oil price or the price of some other commodity. Estimation of network structure from multi-attribute data often arises in practice, however, existing methods largely ignore this aspect of data. We have develop a new principled framework for estimating network structure from multi-attribute data based on partial canonical correlation. Finally, we have develop an estimation method, based on a convex optimization program, for learning network structure from data with missing values that runs 20 to 40 times faster than the existing Expectation-Maximization approach.

15.2 Identifying relevant variables for a large number of related high-dimensional tasks

In different scientific fields, such as neuroscience and genetics, it has been empirically observed that learning jointly from related tasks (i.e., multi-task learning) improves estimation performance. For example, in biology, a genome-wide association mapping study aims to find a small set of causal single-nucleotide polymorphisms (SNPs) that account for genetic variations of a

large number of genes. Identifying causal SNPs is a challenging problem for current statistical methods due to a large number of variables and low signal-to-noise ratio. However, genes in a biological pathway are co-expressed as a module and it is often assumed that a causal SNP affects multiple genes in one pathway. Therefore, once the whole biological pathway is examined, it is much easier to find the causal SNPs.

Prior to my work, despite many investigations, the theory of variable selection in multi-task regression models was far from settled, and there was no clear picture that explained when variable selection can be done more efficiently by considering multiple tasks. Using the framework of the Normal means model, we were able to sharply characterize the theoretical properties of different estimation procedures. In particular, we have provided a sharp characterization of the variable selection properties of two commonly used procedures for variable selection in high-dimensional problems, the lasso and group lasso. Interestingly, two distinct regimes emerged showing that one or the other procedure is optimal, in the minimax sense, depending on the amount of relatedness between the tasks.

Finally, we have explored efficient greedy methods for quick identification of relevant variables in multi-task learning problems. When faced with problems that involve hundreds of thousands input variables, classical methods for variable selection based on convex programming are too slow. Due to their simplicity and computational efficiency, the marginal and forward regressions are often applied in practice. Our investigation provides understanding under what assumptions these methods can be expected to work well. This understanding will hopefully lead to design of better and faster variable selection procedures in the future.

15.3 Future Directions

It is clear that in the future, statistical and machine learning models will become even more prevalent in the analysis of high-dimensional functional data. Although capable of discovering complex structures underlying noisy data, machine learning methods still need human guidance and expertise to instruct them for what to search. The challenge is therefore to develop methods capable of posing hypotheses on what constitutes an interesting structure and trying to identify it in data, reducing the need for human supervision. We see an opportunity in continuing research on flexible models capable of extracting useful and interpretable patterns from complex systems. Here, we provide examples of several research problems that represent important future directions:

1. *Uncertainty quantification* of learned structure. Most of the current literature on high-dimensional structure recovery provides only a point estimate of the underlying structure without providing confidence intervals, which could be used to assess uncertainty on different parts of the structure. Assessed uncertainty is important for domain scientists, who can use it to guide the design of future experiments and data collection processes.
2. *Develop network tools* that would allow researchers to reason about meta-level semantic aspects underlying network structures and their dynamical behaviors. In my current research, I have tackled the problem of learning network structure. Once the structure is uncovered, scientists will need network tools capable of answering useful analytic questions, like:

(a) *Function identification* – What role(s) do individuals play when they interact with different peers? Over the course of a cellular process, such as a cell cycle or an immune response, what is each molecules function and relationship with other molecules?

(b) *System robustness* – How do social groups form and dissolve as a response to external stimuli? How do biological networks rewire to respond to external stress?

(c) *Forecasting* – Based on current activity, can we predict changes in social structure (e.g., emerging or dissolving of subpopulations)? How will a disease progress based on current expression levels of genes in different pathways?

3. *Nonparametric and semiparametric methods* for uncovering structure. Nonparametric and semiparametric models are rather flexible in representing various phenomena, however, due to the amount of samples and computational resources needed to fit them, they have not been used often in the analysis of high-dimensional data. More recent findings show that in many cases the problem under study has a special structure, which can be exploited to effectively fit a nonparametric method. We plan to investigate how nonparametric methods can be used to learn the structure underlying a high-dimensional non-stationary time-series, extending the applicability of time-varying dynamic Bayesian networks.

Bibliography

[1] A. A. Afifi and R. M. Elashoff. Missing Observations In Multivariate Statistics. i. review Of The Literature. *J. Am. Stat. Assoc.*, 61:595–604, 1966. 9.2

[2] A. Ahmed and E. P. Xing. Recovering Time-varying Networks Of Dependencies In Social And Biological Studies. *Proc. Natl. Acad. Sci. U.S.A.*, 106(29):11878–11883, 2009. 3.3, 7.1

[3] A. Argyriou, T. Evgeniou, and M. Pontil. Convex Multi-task Feature Learning. *Mach. Learn.*, 73(3):243–272, 2008. 11.1, 12.1

[4] C. Andrieu, M. Davy, and A. Doucet. Efficient Particle Filtering For Jump markov Systems. application To Time-varying Autoregressions. *IEEE Trans. Signal Proces.*, 51(7):1762–1770, 2003. 3.3

[5] P. Abbeel, D. Koller, and A. Y. Ng. Learning Factor Graphs In Polynomial Time And Sample Complexity. *J. Mach. Learn. Res.*, 7:1743–1788, 2006. 2.2.1

[6] S. Arlot and F. Bach. Data-driven Calibration Of Linear Estimators With Minimal Penalties. *Proc. of NIPS*, pages 46–54, Y. Bengio, D. Schuurmans, John D. Lafferty, C. K. I. Williams, and A. Culotta, eds., 2009. 12.4

[7] T. W. Anderson. Maximum Likelihood Estimates For A Multivariate Normal Distribution When Some Observations Are Missing. *J. Am. Stat. Assoc.*, 52:200–203, 1957. 9.2

[8] J. R. Andrews-Hanna, A. Z. Snyder, J. L. Vincent, C. Lustig, D. Head, M. E. Raichle, and R. L. Buckner. Disruption Of Large-scale Brain Systems In Advanced Aging. *Neuron*, 56(5):924-935, 2007. 10.6.2

[9] D. J. Aldous. Exchangeability And Related Topics. *École d'été de probabilités de Saint-Flour, XIII—1983*, volume 1117 of *Lecture Notes in Math.*, pages 1–198, Springer, Berlin, 1985. 12.6.1

[10] M. N. Arbeitman, E. E. M. Furlong, F. Imam, E. Johnson, B. H. Null, B. S. Baker, M. A. Krasnow, M. P. Scott, R. W. Davis, and K. P. White. Gene Expression During The Life Cycle Of Drosophila Melanogaster. *Science*, 297(5590):2270–2275, American Association for the Advancement of Science, 2002. 4.7.2, 4.7.2, 4.7.2

[11] A. Beck and M. Teboulle. A Fast Iterative Shrinkage-thresholding Algorithm For Linear Inverse Problems. *SIAM J. Imag. Sci.*, 2:183-202, 2009. 7.2.1, 7.2.1, 10.2.2, 10.8.1

[12] F. Bunea. Honest Variable Selection In Linear And Logistic Regression Models Via ℓ_1 And $\ell_1 + \ell_2$ Penalization. *Electron. J. Stat.*, 2:1153–1194, 2008. 6.2, 6.3.1, 6.3.1, 6.1, 7.3

[13] G. Bresler, E. Mossel, and A. Sly. Reconstruction Of markov random fields From samples: some observations And algorithms. *SIAM J. Comput.*, 42(2):563–578, 2013. 2.2.1, 3.1

[14] J. Baxter. Learning Internal Representations. *Proc. of COLT*, pages 311–320, 1995. 11.1

[15] J. Bai and P. Perron. Estimating And Testing Linear Models With Multiple Structural Changes. *Econometrica*, 66(1):47–78, 1998. 7.1

[16] K. Bertin and G. Lecué. Selection Of Variables And Dimension Reduction In High-dimensional Non-parametric Regression. *Electron. J. Stat.*, 2:1224–1241, 2008. 6.3.1

[17] L. Breiman. Heuristics Of Instability And Stabilization In Model Selection. *Ann. Stat.*, 24(6):2350–2383, 1996. 6.3

[18] L. Birgé. An Alternative Point Of View On lepski's Method. *State of the art in probability and statistics (Leiden, 1999)*, volume 36 of *IMS Lecture Notes Monogr. Ser.*, pages 113–133, Inst. Math. Statist., Beachwood, OH, 2001. 14.3

[19] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model Selection Through Sparse Maximum Likelihood Estimation. *J. Mach. Learn. Res.*, 9(3):485-516, 2008. 2.2.2, 3.1, 4.7.1, 4.7.1, 7.2, 7.4.2, 10.1

[20] P. Brucker. An $O(n)$ Algorithm For Quadratic Knapsack Problems. *Oper. Res. Lett.*, 3(3):163–166, 1984. 7.4.1

[21] R. B. Partial Canonical Correlations. *Trabajos de Estadstica y de Investigacin Operativa*, 20(2):211–219, 1969. 10.2.1

[22] Y. Baraud. Non-asymptotic Minimax Rates Of Testing In Signal Detection. *Bernoulli*, 8(5):577–606, 2002. 12.3.2, 12.3.2, 12.6.1, 14.6.7

[23] A. R. Barron, A. Cohen, W. Dahmen, and R. A. DeVore. Approximation And Learning By Greedy Algorithms. *Ann. Stat.*, 36(1):64–94, 2008. 11.1

[24] L. D. Brown and M. G. Low. Asymptotic Equivalence Of Nonparametric Regression And White Noise. *Ann. Stat.*, 24(6):2384–2398, 1996. 12.1.1

[25] P. J. Bickel and E. Levina. Regularized Estimation Of Large Covariance Matrices. *Ann. Stat.*, 36(1):199–227, 2008. 7.7.8, 14.6.3

[26] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. pages xiv+716, Cambridge University Press, Cambridge, 2004. 7.2.1

[27] C. Chow and C. Liu. Approximating Discrete Probability Distributions With Dependence Trees. *IEEE Trans. Inf. Theory*, 14(3):462–467, IEEE, 1968. 2.2.1

[28] E. Candes and T. Tao. The dantzig Selector: Statistical Estimation When p Is Much Larger Than n. *Ann. Stat.*, 35(6):2313–2351, 2007. 14.1

[29] I. Csiszár and Z. Talata. Consistent Estimation Of The Basic Neighborhood Of markov Random Fields. *Ann. Stat.*, 34(1):123–145, 2006. 2.2.1

[30] J. Chiquet, Y. Grandvalet, and C. Ambroise. Inferring Multiple Graphical Structures. *Stat. Comput.*, 21(4):537–553, 2011. 10.1

[31] J. Chen and Z. Chen. Extended bayesian Information Criteria For Model Selection With Large Model Spaces. *Biometrika*, 95(3):759–771, 2008. 8.4, 13.1, 13.2.2

[32] R. Caruana. Multitask Learning. *Mach. Learn.*, 28(1):41–75, Springer, 1997. 11.1

[33] D. M. Chickering. Learning Bayesian Networks Is Np-complete. *Learning from Data*, volume 112 of *Lecture Notes in Statistics*, pages 121-130, Doug Fisher and Hans J. Lenz, eds., Springer New York, 1996. 2.2.1

[34] S. F. Cotter, B. D. Rao, K. Kreutz-Delgado, and J. Adler. Forward Sequential Algorithms For Best Basis Selection. *IEE Proc. Vision, Image and Signal Proces.*, 146(5):235–244, IET, 1999. 13.2.2

[35] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic Decomposition By Basis Pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998. 14.1

[36] T. T. Cai, J. Jin, and M. G. Low. Estimation And Confidence Sets For Sparse Normal Mixtures. *Ann. Stat.*, 35(6):2421–2449, 2007. 14.3

[37] T. T. Cai, W. Liu, and X. Luo. A Constrained ℓ_1 Minimization Approach To Sparse Precision Matrix Estimation. *J. Am. Stat. Assoc.*, 106(494):594-607, 2011. 2.2.2, 9.6

[38] T. T. Cai, L. Wang, and G. Xu. Shifting Inequality And Recovery Of Sparse Signals. *IEEE Trans. Signal Proces.*, 58(3, part 1):1300–1308, 2010. 14.1

[39] T. T. Cai, C.-H. Zhang, and H. H. Zhou. Optimal Rates Of Convergence For Covariance Matrix Estimation. *Ann. Stat.*, 38(4):2118–2144, Institute of Mathematical Statistics, 2010. 2.2.2

[40] W. S. Cleveland, E. Grosse, and W. M. Shyu. Local Regression Models. *Statistical Models in S*, pages 309–376, J. M. Chambers and Trevor J. Hastie, eds., 1991. 4.1

[41] F. Dondelinger, S. Lebre, and D. Husmeier. Heterogeneous Continuous Dynamic Bayesian Networks With Flexible Structure And Inter-time Segment Information Sharing. *Proc. of ICML*, Johannes Fürnkranz and Thorsten Joachims, eds., Haifa, Israel, 2010. 3.3

[42] F. Dondelinger, S. Lébre, and D. Husmeier. Non-homogeneous Dynamic Bayesian Networks With Bayesian Regularization For Inferring Gene Regulatory Networks With Gradually Time-varying Structure. *Mach. Learn.*, page 1-40, Springer US, 2012. 3.3

[43] M. Drton and M. D. Perlman. Model Selection For gaussian Concentration Graphs. *Biometrika*, 91(3):591–602, 2004. 2.2.2

[44] N. Dobigeon, J. Y. Tourneret, and M. Davy. Joint Segmentation Of Piecewise Constant Autoregressive Processes By Using A Hierarchical Model And A Bayesian Sampling Approach. *IEEE Trans. Signal Proces.*, 55(4):1251–1263, IEEE, 2007. 3.3

[45] P. Danaher, P. Wang, and D. M. Witten. The Joint Graphical Lasso For Inverse Covariance Estimation Across Multiple Classes. University of Washington, 2011. 10.1, 10.2.3, 10.5, 10.1b, 10.2b, 10.3b, 10.4b, 10.5, 10.5.1

[46] S. Dasgupta. Learning Polytrees. *Proc. of UAI*, pages 134–141, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. 2.2.1

[47] A. P. Dempster. Covariance Selection. *Biometrics*, 28:157-175, 1972. 2.2.2, 8.1

[48] D. L. Donoho. For Most Large Underdetermined Systems Of Linear Equations The Minimal l_1-norm Solution Is Also The Sparsest Solution. *Comm. Pure Appl. Math.*, 59(6):797–

829, 2006. 14.1

[49] D. L. Donoho and M. Elad. Optimally Sparse Representation In General (nonorthogonal) Dictionaries Via l^1 Minimization. *Proc. Natl. Acad. Sci. U.S.A.*, 100(5):2197–2202 (electronic), 2003. 14.1

[50] D. L. Donoho and J. Jin. Higher Criticism For Detecting Sparse Heterogeneous Mixtures. *Ann. Stat.*, 32(3):962–994, 2004. 14.3

[51] E. H. Davidson. *Genomic Regulatory Systems: In Development And Evolution*. Academic Press, 2001. 4.1

[52] J. C. Duchi, S. Gould, and D. Koller. Projected Subgradient Methods For Learning Sparse Gaussians. *Proc. of UAI*, pages 145–152, 2008. 2.2.2, 4.2

[53] K. E. D. *Statistical Analysis Of Network Data*. pages xii+386, Springer, New York, Methods and models, 2009. 1.1

[54] K. R. Davidson and S. J. Szarek. Local Operator Theory, Random Matrices And banach Spaces. *Handbook of the geometry of Banach spaces, Vol. I*, pages 317–366, North-Holland, Amsterdam, 2001. 7.7.8, 14.4

[55] B. Efron, T. J. Hastie, I. M. Johnstone, and R. J. Tibshirani. Least Angle Regression. *Ann. Stat.*, 32(2):407–499, With discussion, and a rejoinder by the authors, 2004. 2.2.2

[56] D. Edwards. *Introduction To Graphical Modelling*. pages xvi+333, Springer-Verlag, New York, 2000. 2.2.2

[57] L. El Ghaoui, V. Viallon, and T. Rabbani. Safe Feature Elimination In Sparse Supervised Learning. *Pac. J. Optim.*, 8(4):667–698, 2012. 14.1

[58] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, P. A. Morettin, M. C. Sogayar, and C. E. Ferreira. Time-varying Modeling Of Gene Expression Regulatory Networks Using The Wavelet Dynamic Vector Autoregressive Method. *Bioinformatics*, 23(13):1623–1630, Oxford Univ Press, 2007. 3.3

[59] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Bayesian Nonparametric Inference Of Switching Dynamic Linear Models. *IEEE Trans. Signal Proces.*, 59(4):1569–1585, IEEE, 2011. 3.3

[60] J. Fan. Local Linear Regression Smoothers And Their Minimax Efficiencies. *Ann. Stat.*, 21(1):196–216, 1993. 8.2

[61] J. Fan and T. Huang. Profile Likelihood Inferences On Semiparametric Varying-coefficient Partially Linear Models. *Bernoulli*, 11(6):1031–1057, 2005. 8.4

[62] J. Fan and J. Lv. Sure Independence Screening For Ultrahigh Dimensional Feature Space. *J. R. Stat. Soc. B*, 70(5):849–911, 2008. 11.1, 13, 13.1, 13.3.1, 13.4.1, 13.4.1, 13.4.2, 14.1, 14.4

[63] J. Fan and J. Lv. Nonconcave Penalized Likelihood With np-dimensionality. *IEEE Trans. Inf. Theory*, 57(8):5467–5484, 2011. 6.2

[64] J. Fan and R. Li. Variable Selection Via Nonconcave Penalized Likelihood And Its Oracle Properties. *J. Am. Stat. Assoc.*, 96(456):1348–1360, 2001. 2.2.2, 5.5, 12.1.1, 14.1

[65] J. Fan and Q. Yao. Efficient Estimation Of Conditional Variance Functions In Stochastic Regression. *Biometrika*, 85(3):645–660, 1998. 8.1

[66] J. Fan, Y. Feng, and R. Song. Nonparametric Independence Screening In Sparse Ultra-high-dimensional Additive Models. *J. Am. Stat. Assoc.*, 106(494):544–557, 2011. 11.1, 14.1

[67] J. Fan, Y. Feng, and Y. Wu. Network Exploration Via The Adaptive Lasso And scad Penalties. *Ann. Appl. Stat.*, 3(2):521–541, 2009. 2.2.2, 3.1, 7.5

[68] J. Fan, R. Samworth, and Y. Wu. Ultrahigh Dimensional Feature Selection: Beyond The Linear Model. *J. Mach. Learn. Res.*, 10:2013–2038, 2009. 11.1, 14.1

[69] P. Fearnhead. Exact And Efficient Bayesian Inference For Multiple Changepoint Problems. *Stat. Comput.*, 16(2):203–213, Springer, 2006. 3.3

[70] J. H. Friedman, T. J. Hastie, and R. J. Tibshirani. Regularization Paths For Generalized Linear Models Via Coordinate Descent. *Department of Statistics, Stanford University, Tech. Rep*, 2008. 4.2, 4.2, 8.3

[71] J. H. Friedman, T. J. Hastie, and R. J. Tibshirani. Sparse Inverse Covariance Estimation With The Graphical Lasso. *Biostatistics*, 9(3):432-441, 2008. 2.2.2, 3.1, 9.1, 10.1

[72] J. H. Friedman, T. J. Hastie, and R. J. Tibshirani. A Note On The Group Lasso And A Sparse Group Lasso. *ArXiv e-prints, arXiv:1001.0736*, 2010. 8.3, 12.3.2

[73] J. H. Friedman, T. J. Hastie, H. Höfling, and R. J. Tibshirani. Pathwise Coordinate Optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007. 2.2.2

[74] J. J. Fuchs. Recovery Of Exact Sparse Representations In The Presence Of Bounded Noise. *IEEEit*, 51(10):3601–3608, 2005. 14.1

[75] F. Guo, S. Hanneke, W. Fu, and E. P. Xing. Recovering Temporally Rewiring Networks: A Model-based Approach. *Proc. of ICML*, pages 321–328, Corvalis, Oregon, 2007. 3.3

[76] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint Structure Estimation For Categorical Markov Networks. *Unpublished manuscript*, 2010. 3.1, 5.2, 6, 6.3

[77] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint Estimation Of Multiple Graphical Models. *Biometrika*, 98(1):1-15, 2011. 10.1, 10.5, 10.1c, 10.2c, 10.3c, 10.4c, 10.5.1

[78] M. D. Greicius, G. Srivastava, A. L. Reiss, and V. Menon. Default-mode Network Activity Distinguishes Alzheimer's Disease From Healthy Aging: Evidence From Functional mri. *Proc. Natl. Acad. Sci. U.S.A.*, 101(13):4637–4642, 2004. 10.6.2

[79] M. Grzegorczyk and D. Husmeier. Improvements In The Reconstruction Of Time-varying Gene Regulatory Networks: Dynamic Programming And Regularization By Information Sharing Among Genes. *Bioinformatics*, 27(5):693–699, Oxford Univ Press, 2011. 3.3

[80] M. Grzegorczyk and D. Husmeier. Non-homogeneous Dynamic Bayesian Networks For Continuous Data. *Mach. Learn.*, 83(3):355–419, Kluwer Academic Publishers, Hingham, MA, USA, June 2011. 3.3

[81] M. Grzegorczyk and D. Husmeier. Bayesian Regularization Of Non-homogeneous Dynamic Bayesian Networks By Globally Coupling Interaction Parameters. *Proc. of AIS-*

TATS, pages 467–476, Neil Lawrence and Mark Girolami , eds., 2012. 3.3

[82] R. L. Gould, B. Arroyo, R. G. Brown, A. M. Owen, E. T. Bullmore, and R. J. Howard. Brain Mechanisms Of Successful Compensation During Learning In Alzheimer Disease. *Neurology*, 67(6):1011–1017, 2006. 10.6.2

[83] C. R. Genovese, J. J.in, L. Wasserman, and Z. Yao. A Comparison Of The Lasso And Marginal Regression. *J. Mach. Learn. Res.*, 13:2107–2143, 2012. 11.1, 14.1, 14.1, 14.1, 14.3

[84] M. C. Grant and S. P. Boyd. cvx: Matlab Software For Disciplined Convex Programming, Version 2.0 Beta. `http://cvxr.com/cvx`, September 2012. 4.3

[85] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. Sparse Inverse Covariance Matrix Estimation Using Quadratic Approximation. *Proc. of NIPS*, pages 2330–2338, http://nips.cc/, 2011. 2.2.2

[86] D. Husmeier, F. Dondelinger, and S. Lébre. Inter-time Segment Information Sharing For Non-homogeneous Dynamic Bayesian Networks. *Proc. of NIPS*, pages 901–909, John D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, eds., 2010. 3.3

[87] H. O. Hartley and R. R. Hocking. The Analysis Of Incomplete Data. *Biometrics*, 27(4):783–823, JSTOR, 1971. 9.2

[88] J. Honorio and D. Samaras. Multi-task Learning Of gaussian Graphical Models. *Proc. of ICML*, pages 447–454, Johannes Fürnkranz and Thorsten Joachims, eds., Omnipress, Haifa, Israel, June 2010. 10.1

[89] J. Huang, S. Ma, and C.-H. Zhang. Adaptive lasso For Sparse High-dimensional Regression Models. *Stat. Sinica*, 18(4):1603–1618, 2008. 13.2.3

[90] R. R. Hocking and W. B. Smith. Estimation Of Parameters In The Multivariate Normal Distribution With Missing Observations. *J. Am. Stat. Assoc.*, 63:159–173, 1968. 9.2

[91] S. Huang, J. Li, L. Sun, J. Liu, T. Wu, K. Chen, A. Fleisher, E. Reiman, and J. Ye. Learning Brain Connectivity Of Alzheimer's Disease From Neuroimaging Data. *Proc. of NIPS*, pages 808–816, Y. Bengio, D. Schuurmans, John D. Lafferty, C. K. I. Williams, and A. Culotta, eds., 2009. 10.6.2, 10.6.2

[92] S. Hanneke, W. Fu, and E. P. Xing. Discrete Temporal Models Of Social Networks. *Electron. J. Stat.*, 4:585–605, 2010. 3.3

[93] T. Hedden, K. R. A. V. Dijk, J. A. Becker, A. Mehta, R. A. Sperling, K. A. Johnson, and R. L. Buckner. Disruption Of Functional Connectivity In Clinically Normal Older Adults Harboring Amyloid Burden. *J. Neurosci.*, 29(40):12686–12694, 2009. 10.6.2

[94] Z. Harchaoui and C. Lévy-Leduc. Multiple Change-point Estimation With A Total Variation Penalty. *J. Am. Stat. Assoc.*, 105(492):1480–1493, 2010. 7.3.2, 7.7.2

[95] H. J. A. L. F. L. A Bayesian Markov-switching Model For Sparse Dynamic Network Estimation. *Proc. 2012 SIAM Int. Conf. Data Mining*, pages 506–515, 2012. 3.3

[96] T. J. Hastie and R. J. Tibshirani. Varying-coefficient Models. *J. R. Stat. Soc. B*, 55(4):757–796, 1993. 4.1, 6.1, 7.1

[97] hou. The Adaptive Lasso And Its Oracle Properties. *J. Am. Stat. Assoc.*, 101(476):1418–1429, 2006. 13.1, 13.2.3, 14.1

[98] P. Ji and J. Jin. Ups Delivers Optimal Phase Diagram In High-dimensional Variable Selection. *Ann. Stat.*, 40(1):73–103, 2012. 14.3, 14.5

[99] Y. Jia and J. Huan. Constructing Non-stationary Dynamic Bayesian Networks With A Flexible Lag Choosing Mechanism. *BMC Bioinformatics*, 11(Suppl 6):S27, 2010. 3.3

[100] I. M. Johnstone. Chi-square Oracle Inequalities. *State of the art in probability and statistics (Leiden, 1999)*, volume 36 of *IMS Lecture Notes Monogr. Ser.*, pages 399–418, Inst. Math. Statist., Beachwood, OH, 2001. 14.2

[101] K. Koh, S.-J. Kim, and S. P. Boyd. An Interior-point Method For Large-scale l_1-regularized Logistic Regression. *J. Mach. Learn. Res.*, 8:1519–1555, 2007. 4.2

[102] M. Kolar and H. Liu. Marginal Regression For Multitask Learning. *Proc. of ICML*, pages 647–655, John Langford and Joelle Pineau, eds., Edinburgh, Scotland, GB, 2012. 1.3

[103] M. Kolar and E. P. Xing. Sparsistent Estimation Of Time-varying Discrete Markov Random Fields. *ArXiv e-prints, arXiv:0907.2337*, July 2009. 1.3, 3.3

[104] M. Kolar and E. P. Xing. Ultra-high Dimensional Multiple Output Learning With Simultaneous Orthogonal Matching Pursuit: Screening Approach. *Proc. of AISTATS*, pages 413–420, 2010. 1.3, 11.1, 12.1, 13.4.2

[105] M. Kolar and E. P. Xing. On Time Varying Undirected Graphs. *Proc. of AISTATS*, 2011. 1.3, 3.3, 6.2.1, 6.2.1, 6.3.1, 6.3.1, 6.3.1

[106] M. Kolar and E. P. Xing. Estimating Networks With Jumps. *Electron. J. Stat.*, 6:2069–2106, Institute of Mathematical Statistics, 2012. 1.3, 3.3

[107] M. Kolar and E. P. Xing. Consistent Covariance Selection From Data With Missing Values. *Proc. of ICML*, pages 551–558, John Langford and Joelle Pineau, eds., Omnipress, Edinburgh, Scotland, GB, July 2012. 1.3, 10.2.2, 10.5.2

[108] M. Kolar, H. Liu, and E. P. Xing. Graph Estimation From Multi-attribute Data. *ArXiv e-prints, arXiv:1210.7665*, October 2012. 10.6.2

[109] M. Kolar, H. Liu, and E. P. Xing. Markov Network Estimation From Multi-attribute Data. *Proc. of ICML*, 2013. 1.3

[110] M. Kolar, J. D. Lafferty, and L. Wasserman. Union Support Recovery In Multi-task Learning. *J. Mach. Learn. Res.*, 12:2415–2435, 2011. 1.3, 12, 13.4.1

[111] M. Kolar, A. P. Parikh, and E. P. Xing. On Sparse Nonparametric Conditional Covariance Selection. *Proc. of ICML*, Johannes Fürnkranz and Thorsten Joachims, eds., Haifa, Israel, 2010. 1.3, 3.3

[112] M. Kolar, L. Song, A. Ahmed, and E. P. Xing. Estimating time-varying Networks. *Ann. Appl. Stat.*, 4(1):94–123, 2010. 1.3, 3.3, 5.4, 6.3, 10.1

[113] M. Kolar, L. Song, and E. P. Xing. Sparsistent Learning Of Varying-coefficient Models With Structural Changes. *Proc. of NIPS*, pages 1006–1014, Y. Bengio, D. Schuurmans, John D. Lafferty, C. K. I. Williams, and A. Culotta, eds., 2009. 1.3

[114] N. Katenka and E. D. Kolaczyk. Multi-attribute Networks And The Impact Of Partial Information On Inference And Characterization. *Ann. Appl. Stat.*, 6(3):1068–1094, 2011. 10.1, 10.2.1, 10.2.1, 10.4, 10.6.1, 10.6.1

[115] S. Kim and E. P. Xing. Statistical Estimation Of Correlated Genome Associations To A Quantitative Trait Network. *PLoS genetics*, 5(8):e1000587, Public Library of Science, 2009. 13.1, 13.4.3

[116] S. Kim, K.-A. Sohn, and E. P. Xing. A Multivariate Regression Approach To Association Analysis Of A Quantitative Trait Network. *Bioinformatics*, 25(12):i204–i212, Oxford Univ Press, 2009. 11.1, 12.1, 13.1

[117] A. Lozano, G. Swirszcz, and N. Abe. Grouped Orthogonal Matching Pursuit For Variable Selection And Prediction. *Proc. of NIPS*, pages 1150–1158, Y. Bengio, D. Schuurmans, John D. Lafferty, C. K. I. Williams, and A. Culotta, eds., 2009. 11.1

[118] B. Laurent and P. Massart. Adaptive Estimation Of A Quadratic Functional By Model Selection. *Ann. Stat.*, 28(5):1302–1338, 2000. 14.1

[119] H. Li and J. Gui. Gradient Directed Regularization For Sparse Gaussian Concentration Graphs, With Applications To Inference Of Genetic Networks. *Biostatistics*, 7(2):302–317, Biometrika Trust, 2006. 5.4, 7.5, 10.5

[120] H. Liu, M. Palatucci, and J. Zhang. Blockwise Coordinate Descent Procedures For The Multi-task Lasso, With Applications To Neural Semantic Basis Discovery. *Proc. of ICML*, pages 649–656, New York, NY, USA, 2009. 11.1, 12.1, 12.3.3, 12.2, 12.3.3, 13.1, 13.1

[121] J. Liu, S. Wu, and J. V. Zidek. On Segmented Multivariate Regression. *Stat. Sinica*, 7(2):497–525, 1997. 7.1

[122] K. Lounici. High-dimensional Covariance Matrix Estimation With Missing Observations. *ArXiv e-prints, arXiv:1201.2577*, January 2012. 9.1, 9.3.2, 9.6

[123] K. Lounici, M. Pontil, A. B. Tsybakov, and S. A. van de Geer. Oracle Inequalities And Optimal Inference Under Group Sparsity. *Ann. Stat.*, 39:2164–204, 2011. 7.7.8, 10.3, 11.1, 12.1, 12.1, 2, 13.1, 13.1, 13.3.1

[124] P.-L. Loh and M. J. Wainwright. High-dimensional Regression With Noisy And Missing Data: Provable Guarantees With Nonconvexity. *Ann. Stat.*, 40(3):1637–1664, 2012. 9.1, 9.3.2, 9.5, 9.5.1

[125] R. Li and H. Liang. Variable Selection In Semiparametric Regression Modeling. *Ann. Stat.*, 36(1):261–286, 2008. 8.2

[126] S. Lébre, J. Becq, F. Devaux, M. Stumpf, and G. Lelandais. Statistical Inference Of The Time-varying Structure Of Gene-regulation Networks. *BMC Systems Biology*, 4(1):130, 2010. 3.3

[127] N. M. Luscombe, M. M. Babu, H. Yu, M. Snyder, S. A. Teichmann, and M. Gerstein. Genomic Analysis Of Regulatory Network Dynamics Reveals Large Topological Changes. *Nature*, 431(7006):308–312, Nature Publishing Group, 2004. 4.7.2

[128] R. J. A. Little. A Test Of Missing Completely At Random For Multivariate Data With Missing Values. *J. Am. Stat. Assoc.*, 83(404):1198–1202, 1988. 9.5.3

[129] R. J. A. Little and D. B. Rubin. *Statistical Analysis With Missing Data*. pages xviii+381, Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2002. 9.1

[130] S. L. Lauritzen. *Graphical Models*. pages x+298, The Clarendon Press Oxford University Press, New York, Oxford Science Publications, 1996. 2.1, 2.2.2, 6.3, 10.8.3, 14.6.4

[131] E. Mammen and S. A. van de Geer. Locally Adaptive Regression Splines. *Ann. Stat.*, 25(1):387–413, 1997. 7.2

[132] H. Markowitz. Portfolio Selection. *J. Finance*, 7(1):77–91, Wiley Online Library, 1952. 8, 8.6

[133] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Learning For Matrix Factorization And Sparse Coding. *J. Mach. Learn. Res.*, 11:19–60, 2010. 7.4.1

[134] N. Meinshausen. A Note On The Lasso For Graphical Gaussian Model Selection. *Statist. Probab. Lett.*, 78(7):880–884, 2008. 6.3

[135] N. Meinshausen and P. Bühlmann. High Dimensional Graphs And Variable Selection With The Lasso. *Ann. Stat.*, 34(3):1436–1462, 2006. 2.2.2, 3.1, 5.2, 6, 6.1, 6.3, 6.3, 6.4, 7.3, 7.4.2, 7.5, 8.5.1, 10.1

[136] N. Meinshausen and P. Bühlmann. Stability Selection. *J. R. Stat. Soc. B*, 72(4):417–473, 2010. 10.6.1, 10.6.2

[137] N. Meinshausen and B. Yu. Lasso-type Recovery Of Sparse Representations For High-dimensional Data. *Ann. Stat.*, 37(1):246–270, 2009. 14.1

[138] R. Mazumder and D. K. Agarwal. A Flexible, Scalable And Efficient Algorithmic Framework For Primal Graphical Lasso. Stanford University, 2011. 10.2.2

[139] R. Mazumder and T. J. Hastie. Exact Covariance Thresholding Into Connected Components For Large-scale Graphical Lasso. *J. Mach. Learn. Res.*, 13:781–794, 2012. 10.2.3

[140] M. Nussbaum. Asymptotic Equivalence Of Density Estimation And Gaussian White Noise. *Ann. Stat.*, 24(6):2399–2430, 1996. 12.1.1

[141] Y. Nesterov. Gradient Methods For Minimizing Composite Objective Function. Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Technical Report 76:2007, 2007. 7.2.1

[142] Y. Nesterov. Smooth Minimization Of Non-smooth Functions. *Math. Program.*, 103(1, Ser. A):127–152, 2005. 7.2.1, 7.2.1

[143] S. N. Negahban and M. J. Wainwright. Simultaneous Support Recovery In High Dimensions: Benefits And Perils Of Block ℓ_1/ℓ_∞-regularization. *IEEE Trans. Inf. Theory*, 57(6):3841–3863, IEEE Press, Piscataway, NJ, USA, June 2011. 11.1, 12.1, 12.1, 3

[144] G. Obozinski, M. J. Wainwright, and M. I. Jordan. Support Union Recovery In High-dimensional Multivariate Regression. *Ann. Stat.*, 39(1):1–47, 2011. 11.1, 12.1, 12.1, 2, 13.1

[145] E. Punskaya, C. Andrieu, A. Doucet, and W. J. Fitzgerald. Bayesian Curve Fitting Using Mcmc With Applications To Signal Segmentation. *IEEE Trans. Signal Proces.*, 50(3):747–758, IEEE, 2002. 3.3

[146] J. Peng, P. Wang, N. Zhou, and J. Zhu. Partial Correlation Estimation By Joint Sparse Regression Models. *J. Am. Stat. Assoc.*, 104(486):735–746, 2009. 2.2.2, 3.1, 4.6, 5.2, 6.3, 7.5, 8.3, 10.1

[147] J. Peng, J. Zhu, A. Bergamaschi, W. Han, D.-Y. Noh, J. R. Pollack, and P. Wang. Regularized Multivariate Regression For Identifying Master Predictors With Application To Integrative Genomics Study Of Breast Cancer. *Ann. Appl. Stat.*, 4(1):53–77, 2010. 11.1, 13.1

[148] A. Rinaldo. Properties And Refinements Of The Fused Lasso. *Ann. Stat.*, 37(5B):2922–2952, 2009. 4.3, 7.2

[149] A. Rao, A. O. Hero, III, D. J. States, and J. D. Engel. Inferring Time-varying Network Topologies From Gene Expression Data. *EURASIP J. Bioinformatics Syst. Bio.*, 2007(1):51947, 2007. 3.3

[150] D. Ruppert, M. P. Wand, U. Holst, and O. Hössjer. Local Polynomial Variance-function Estimation. *Technometrics*, 39(3):262–273, 1997. 8.1

[151] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional ising Model Selection Using ℓ_1-regularized Logistic Regression. *Ann. Stat.*, 38(3):1287–1319, 2010. 2.2.1, 3.1, 4.1, 4.6, 4.8, 5.2, 5.3, 6.2, 6.3

[152] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional Covariance Estimation By Minimizing ℓ_1-penalized Log-determinant Divergence. *Electron. J. Stat.*, 5:935-980, 2011. 2.2.2, 3.1, 6.2, 6.2, 6.2.1, 6.3, 6.4, 7.4.2, 9.3, 9.4, 9.4, 9.4, 10.3, 10.3, 10.8.4, 10.8.4, 10.8.4, 10.8.4, 10.8.4

[153] R. R., Z. M., and E. M. Efficient Implementation Of The K-svd Algorithm Using Batch Orthogonal Matching Pursuit. *CS Technion, Tech. Rep.*, 2008. 13.2.2

[154] A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse Permutation Invariant Covariance Estimation. *Electron. J. Stat.*, 2:494–515, 2008. 2.2.2, 3.1, 5.6.5

[155] D. B. Rubin. Inference And Missing Data. *Biometrika*, 63(3):581–592, With comments by R. J. A. Little and a reply by the author, 1976. 9.1

[156] J. M. Robins, A. Rotnitzky, and L. P. Zhao. Estimation Of Regression Coefficients When Some Regressors Are Not Always Observed. *J. Am. Stat. Assoc.*, 89(427):846–866, Taylor & Francis Group, 1994. 9.6

[157] J. M. Robins, R. Scheines, P. Spirtes, and L. Wasserman. Uniform Consistency In Causal Inference. *Biometrika*, 90(3):491–515, 2003. 14.1

[158] J. W. Robinson and A. J. Hartemink. Learning Non-stationary Dynamic bayesian Networks. *J. Mach. Learn. Res.*, 11:3647–3680, 2010. 3.3

[159] L. Song, M. Kolar, and E. P. Xing. Keller: Estimating Time-varying Interactions Between Genes. *Bioinformatics*, 25(12):i128–i136, Oxford Univ Press, 2009. 1.3, 5.1

[160] L. Song, M. Kolar, and E. P. Xing. Time-varying Dynamic Bayesian Networks. *Proc. of NIPS*, pages 1732–1740, Y. Bengio, D. Schuurmans, John D. Lafferty, C. K. I. Williams, and A. Culotta, eds., 2009. 3.3

[161] M. R. Siracusa and J. W. Fisher. Tractable Bayesian Inference Of Time-series Dependence Structure. *Proc. of AISTATS*, 2009. 3.3

[162] M. Sjöbeck and E. Englund. Alzheimer's Disease And The Cerebellum: A Morphologic Study On Neuronal And Glial Changes. *Dementia and geriatric cognitive disorders*, 12(3):211–218, 2001. 10.6.2

[163] N. Srebro. Maximum Likelihood Bounded Tree-width Markov Networks. *Proc. of UAI*, pages 504–511, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001. 2.2.1

[164] N. Städler and P. Bühlmann. Missing Values: Sparse Inverse Covariance Estimation And An Extension To Sparse Regression. *Stat. Comput.*, 22(1):219–235, 2012. 9.1, 9.2, 9.5, 9.5.2, 9.5.3

[165] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, And Search*. pages xxii+543, MIT Press, Cambridge, MA, With additional material by David Heckerman, Christopher Meek, Gregory F. Cooper and Thomas Richardson, A Bradford Book, 2000. 14.1

[166] P. Sarkar and A. W. Moore. Dynamic Social Network Analysis Using Latent Space Models. *ACM SIGKDD Explor. Newsl.*, 7(2):31–40, ACM, 2005. 3.3

[167] S. S. W. Moments And Distributions Of Estimates Of Population Parameters From Fragmentary Samples. *Ann. Math. Stat.*, 3(3):163–195, JSTOR, 1932. 9.2

[168] M. Talih and N. Hengartner. Structural Learning With Time-varying Components: Tracking The Cross-section Of The Financial Time Series. *J. R. Stat. Soc. B*, 67(3):321–341, 2005. 3.3

[169] P. Tseng. Convergence Of A Block Coordinate Descent Method For Nondifferentiable Minimization. *J. Optim. Theory Appl.*, 109(3):475–494, Plenum Press, New York, NY, USA, 2001. 4.3, 10.2.2

[170] S. Thrun and J. O'Sullivan. Discovering Structure In Multiple Learning Tasks: The Tc Algorithm. *Proc. of ICML*, pages 489–497, 1996. 11.1

[171] A. B. Tsybakov. *Introduction To Nonparametric Estimation*. pages xii+214, Springer, New York, Revised and extended from the 2004 French original, Translated by Vladimir Zaiats, 2009. 12.6.1

[172] B. A. Turlach, W. N. Venables, and S. J. Wright. Simultaneous Variable Selection. *Technometrics*, 47(3):349–363, 2005. 11.1, 12.1

[173] J. A. Tropp. Greed Is Good: Algorithmic Results For Sparse Approximation. *IEEEit*, 50(10):2231–2242, 2004. 14.1

[174] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms For Simultaneous Sparse Approximation. Part I: Greedy Pursuit. *Signal Proces.*, 86(3):572–588, Elsevier, 2006. 11.1, 13.1

[175] R. J. Tibshirani. Regression Shrinkage And Selection Via The Lasso. *J. R. Stat. Soc. B*, 58(1):267–288, 1996. 2.2.2, 14.1

[176] R. J. Tibshirani, J. Bien, jfriedman, T. J. Hastie, N. Simon, J. E. Taylor, and R. J. Tibshi-

rani. Strong Rules For Discarding Predictors In Lasso-type Problems. *J. R. Stat. Soc. B*, 74(2):245–266, 2012. 14.1

[177] R. J. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity And Smoothness Via The Fused Lasso. *J. R. Stat. Soc. B*, 67(1):91–108, 2005. 4.5, 7.2

[178] D. Vogel and R. Fried. On Robust Gaussian Graphical Modelling. *Recent Developments in Applied Probability and Statistics*, pages 155-182, L. Devroye et al. (Eds.), ed., Berlin, Heidelberg: Springer-Verlag, 2010. 3.3

[179] G. Varoquaux, A. Gramfort, J.-B. Poline, and B. Thirion. Brain Covariance Selection: Better Individual Functional Connectivity Models Using Population Prior. *Proc. of NIPS*, pages 2334–2342, John D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, eds., 2010. 10.1

[180] M. A. J. van Duijn, K. J. Gile, and M. S. Handcock. A Framework For The Comparison Of Maximum Pseudo-likelihood And Maximum Likelihood Estimation Of Exponential Family Random Graph Models. *Social Networks*, 31(1):52–62, Elsevier, 2009. 4.1

[181] S. A. van de Geer and P. Bühlmann. On The Conditions Used To Prove Oracle Results For The lasso. *Electron. J. Stat.*, 3:1360–1392, 2009. 6.3, 7.3.1, 12.1.1

[182] H. Wang. Forward Regression For Ultra-high Dimensional Variable Screening. *J. Am. Stat. Assoc.*, 104(488):1512–1524, 2009. 11.1, 13.4.1, 13.4.1, 13.6.2, 14.4

[183] H. Wang and Y. Xia. Shrinkage Estimation Of The Varying Coefficient Model. *J. Am. Stat. Assoc.*, 104(486):747–757, 2009. 8.4

[184] L. Wasserman and K. Roeder. High-dimensional Variable Selection. *Ann. Stat.*, 37(5A):2178–2201, 2009. 11.1, 14.1, 14.1, 14.1

[185] P. Wang, D. L. Chao, and L. Hsu. Learning Networks From High Dimensional Binary Data: An Application To Genomic Instability Data. *ArXiv e-prints, arXiv:0908.3882*, 2009. 3.1

[186] X. Wu, R. Li, A. S. Fleisher, E. M. Reiman, X. Guan, Y. Zhang, K. Chen, and L. Yao. Altered Default Mode Network Connectivity In Alzheimer's Diseasea Resting Functional Mri And Bayesian Network Study. *Human brain mapping*, 32(11):1868–1881, Wiley Online Library, 2011. 10.6.2

[187] Z. Wang, E. E. Kuruoglu, X. Yang, Y. Xu, and T. S. Huang. Time Varying Dynamic Bayesian Network For Nonstationary Events Modeling And Online Inference. *IEEE Trans. Signal Proces.*, 59(4):1553–1568, IEEE, 2011. 3.3

[188] D. J. Watts and S. H. Strogatz. Collective Dynamics Of small-worldnetworks. *nature*, 393(6684):440–442, Nature Publishing Group, 1998. 4.7.2

[189] D. M. Witten, J. H. Friedman, and N. Simon. New Insights And Faster Computations For The Graphical Lasso. *J. Comput. Graph. Stat.*, 20(4):892–900, ASA, 2011. 10.2.3

[190] M. J. Wainwright. Sharp Thresholds For High-dimensional And Noisy Sparsity Recovery Using ℓ_1-constrained Quadratic Programming (lasso). *IEEE Trans. Inf. Theory*, 55(5):2183–2202, 2009. 6.3, 6.3.1, 6.3.1, 7.3, 7.7.8, 14.1, 14.1, 14.5

[191] M. J. Wainwright and M. I. Jordan. Graphical Models, Exponential Families, And Variational Inference. *Found. and Trends Mach. Learn.*, 1(1-2):1–305, Now Publishers Inc., Hanover, MA, USA, 2008. 4.1

[192] X. Xuan and K. Murphy. Modeling Changing Dependency Structure In Multivariate Time Series. *Proc. of ICML*, pages 1055–1062, ACM, New York, NY, USA, 2007. 3.3

[193] J. Yin, Z. Geng, R. Li, and H. Wang. Nonparametric Covariance Model. *Stat. Sinica*, 20:469–479, 2010. 3.3, 8.1

[194] M. Yuan and Y. Lin. Model Selection And Estimation In Regression With Grouped Variables. *J. R. Stat. Soc. B*, 68:49–67, 2006. 8.2, 10.2.2

[195] M. Yuan and Y. Lin. Model Selection And Estimation In The Gaussian Graphical Model. *Biometrika*, 94(1):19-35, 2007. 2.2.2, 2.2.2, 3.1, 7.4.2, 9.2, 9.3.1

[196] R. Yoshida, S. Imoto, and T. Higuchi. Estimating Time-dependent Gene Networks From Time Series Microarray Data By Dynamic Linear Models With Markov Switching. *Proc. 2005 IEEE Comp. Comput. Syst. Bioinformatics. Conf.*, pages 289–298, IEEE Computer Society, Washington, DC, USA, 2005. 3.3

[197] S. Yaakov. Cognitive Reserve And Alzheimer Disease. *Alzheimer Disease & Associated Disorders*, 20(2):112–117, LWW, 2006. 10.6.2

[198] C.-H. Zhang. Nearly Unbiased Variable Selection Under Minimax Concave Penalty. *Ann. Stat.*, 38(2):894–942, 2010. 14.1

[199] C.-H. Zhang and J. Huang. The Sparsity And Bias Of The lasso Selection In High-dimensional Linear Regression. *Ann. Stat.*, 36(4):1567–1594, 2008. 13.3.1

[200] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral Relaxation For K-means Clustering. *Proc. of NIPS*, pages 1057–1064, 2001. 13.4.3

[201] H. Zou and T. J. Hastie. Regularization And Variable Selection Via The Elastic Net. *J. R. Stat. Soc. B*, 67(2):301–320, 2005. 14.1

[202] H. Zou and R. Li. One-step Sparse Estimates In Nonconcave Penalized Likelihood Models. *Ann. Stat.*, 36(4):1509–1533, 2008. 2.2.2, 14.1

[203] H. Zou and M. Yuan. The F_∞-norm Support Vector Machine. *Stat. Sinica*, 18(1):379–398, 2008. 11.1, 12.1

[204] J. Zhang. A Probabilistic Framework For Multitask Learning. Ph.D. Thesis, Carnegie Mellon University, 2006. 11.1, 12.1, 13.1

[205] P. Zhao and B. Yu. On Model Selection Consistency Of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, 2006. 6.3, 6.3.1, 7.3.1, 12.1.1, 14.1, 14.1

[206] S. Zhou, J. D. Lafferty, and L. Wasserman. Time Varying Undirected Graphs. *Mach. Learn.*, 80(2-3):295–319, Springer, 2010. 3.3, 6, 6.1, 6.1, 6.2, 6.2, 6.4, 7.1

[207] T. Zhang. On The Consistency Of Feature Selection Using Greedy Least Squares Regression. *J. Mach. Learn. Res.*, 10:555–568, 2009. 11.1