# Some Advances on Modeling High-Dimensional Data with Complex Structures

by

Cheng Qian

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2017

Doctoral Committee:

Professor Ji Zhu, Chair
Professor Judy Jin
Professor Elizaveta Levina
Professor Kerby A. Shedden

qianche@umich.edu
ORCID iD: 0000-0002-9909-2130

---

2017

# ACKNOWLEDGMENTS

First I wish to express my great appreciation to my advisor, Prof. Ji Zhu. Without his encouragement and guidance, this dissertation could not have been completed. I am also very grateful to my friends and classmates at Michigan, especially to Tianxi Li and Jiahe Lin, for their constant help and incomparable friendship during my graduate studies. I would also like to thank the dissertation committee members, Prof. Jionghua Jin, Prof. Liza Levina, and Prof. Kerby Shedden, for their time to serve on the committee and their useful comments on my research. Finally, I would like to thank my parents for their love and confidence in me. My parents' support is a big part of everything that I accomplish.

# TABLE OF CONTENTS

**Acknowledgments** . . . . . **ii**

**List of Figures** . . . . . **v**

**List of Tables** . . . . . **vii**

**List of Appendices** . . . . . **viii**

**Abstract** . . . . . **ix**

**Chapter**

**1 Introduction** . . . . . **1**

1.1 Regularized least squares regression . . . . . 1
1.2 Gaussian graphical model . . . . . 3
1.3 Outline of the thesis . . . . . 4

**2 Gaussian Graphical Models on Network-linked Data** . . . . . **6**

2.1 Introduction . . . . . 6
2.2 Gaussian graphical model with network cohesion . . . . . 9
2.2.1 Model estimation . . . . . 11
2.2.2 Model selection . . . . . 12
2.3 Theoretical properties . . . . . 13
2.3.1 Cohesive assumptions on the observation network . . . . . 13
2.3.2 Mean estimation error bounds . . . . . 16
2.3.3 Inverse covariance estimation error bounds . . . . . 17
2.3.4 Oracle mean estimation and sufficiency of two-stage estimation . 20
2.4 Simulation studies . . . . . 23
2.4.1 Performance under different Gaussian graphs . . . . . 23
2.4.2 Comparison with other methods under different cohesion settings 24
2.5 Data example: learning associations between statistical terms . . . . . 27
2.6 Conclusion . . . . . 32

**3 A Two-Step Approach for Estimating Directed Acyclic Graphs** . . . . . **33**

3.1 Introduction . . . . . 33
3.2 The proposed two-step methodology . . . . . 34
3.2.1 Some useful theoretical results . . . . . 35
3.2.2 Step 1: estimating the moral graph . . . . . 38

3.2.3 Step 2: reconstructing the DAG on the restricted space . . . . . 39
3.3 Simulation studies . . . . . 43
3.4 Data example . . . . . 46
3.5 Summary . . . . . 48

**4 Estimating Cointegrated Vectors with Structured Sparsity . . . . . 49**

4.1 Introduction . . . . . 49
4.2 Problem formulation and the proposed method . . . . . 51
4.2.1 Estimation . . . . . 52
4.2.2 Tuning parameter selection. . . . . 56
4.3 Simulation studies . . . . . 57
4.4 Data examples . . . . . 61
4.4.1 Financial sector stock data . . . . . 62
4.4.2 Treasury yield data . . . . . 65
4.5 Summary. . . . . 67

**5 Sparse Rank Support Vector Machines . . . . . 68**

5.1 Introduction . . . . . 68
5.2 Sparse rank support vector machines . . . . . 70
5.2.1 Algorithm . . . . . 71
5.2.2 Tuning parameter selection . . . . . 74
5.3 Simulation studies . . . . . 75
5.3.1 Effects of $n$ and $m$ . . . . . 76
5.3.2 Effects of $p$ and signal-to-noise ratio . . . . . 79
5.4 Data example . . . . . 81
5.5 Summary . . . . . 86

**6 Future Work . . . . . 88**

**Bibliography . . . . . 89**

**Appendices . . . . . 95**

# LIST OF FIGURES


2.1 $n = 400, p = 500$, varying sparsisty, 200 replications . . . . . 25
2.2 $n = 400$, sparsity is 0.01, varying $p$, 200 replications . . . . . 25
2.3 Nontrivial cohesion in mean, $n = 400$, $p = 500$ . . . . . 26
2.4 Trivial cohesion in mean (constant), $n = 400$, $p = 500$. . . . . 26
2.5 The coauthorship network of 635 statisticians based on four statistical journals. Both the size and the color of each node indicate the degree of the node (number of connections), with larger and darker nodes being statisticians with more coauthors in the network. . . . . 29
2.6 Partial correlation graphs estimated using Glasso . . . . . 31
2.7 Partial correlation graphs estimated using GNC-lasso . . . . . 31

3.1 ROC curves for $\widehat{\Theta}$ in identifying edges in the moral graph of $A$. Solid red curves correspond to the weighted graphical lasso, and the dashed black curves correspond to the standard graphical lasso. . . . . 45
3.2 ROC curves for $\widehat{\Theta}$ when using $A$ as the baseline. $n = 100$, $p =$30, 50, 100, 200. 45
3.3 Estimated dependence structures among 10 stocks using the one-step (left) and two-step (right) methods . . . . . 47

4.1 Estimated cointegrated series 1 for financial stocks . . . . . 64
4.2 Estimated cointegrated series 2 for financial stocks . . . . . 64
4.3 Treasury yields of different maturities over time . . . . . 66
4.4 Estimated cointegrated series for treasury yields . . . . . 67

5.1 Cumulative return curves when long top 100 stocks and short bottom 100 stocks using individual features (features 1-6). . . . . 82
5.2 Cumulative return curves when long top 100 stocks and short bottom 100 stocks using individual features (features 7-12). . . . . 83
5.3 Cumulative return curves when long top 100 stocks and short bottom 100 stocks using individual features (features 13-18). . . . . 84
5.4 Cumulative return curves when long top 100 stocks and short bottom 100 stocks using individual features (features 19-21). . . . . 85
5.5 Correlations between 21 features' daily returns when the long-short strategy is used for each individual feature . . . . . 86
5.6 Cumulative return curves when long top 100 stocks and short bottom 100 stocks using the fitted model to rank stocks. The two vertical lines indicate the separation of training, validation and testing sets. . . . . 87

A.1 The $20 \times 20$ grid network in simulation studies. The size of the node indicates the corresponding mean value in the nontrivial cohesion setting. . . . . 107

# LIST OF TABLES


3.1 Simulation results based on 50 replications . . . . . 46

4.1 Simulation results for S1 with $p = 100$, $r = 20$ over 50 replications . . . . . 61
4.2 Simulation results for S2 with $p = 20$, $r = 5$, $\|\beta_{\cdot,j}\|_0 = 6$, over 50 replications . 61
4.3 Simulation results for S2 with $p = 30$, $r = 5$, $\|\beta_{\cdot,j}\|_0 = 4$, over 50 replications . 61
4.4 Correlation matrix for log-returns of financial sector stocks . . . . . 63
4.5 $p$-value from the ADF test on identified cointegrated series . . . . . 63
4.6 Estimated cointegrating vectors for financial sector stocks . . . . . 63
4.7 Estimated cointegrating vector for treasury yields with constant maturity . . . 66

5.1 Simulation results under 3 correlation structures. We set $p = 100$ and $\sigma^2$'s are set such that the signal-to-noise ratio is equal to 1. Three methods are compared, the standard rank SVM, the $\ell_1$-norm rank SVM and the elastic-net rank SVM. All results are averages over 50 replications. . . . . 78
5.2 Simulation result under 3 correlation structures. We fix $n = 100, m = 100$. Three methods are compared, the standard rank SVM, the $\ell_1$-norm rank SVM and the elastic-net rank SVM. All results are averages over 50 replications. . . 80
5.3 Cumulative returns and Sharpe ratios (SR) of the three methods in training, validation and testing periods . . . . . 86

# LIST OF APPENDICES

**A Proofs of the Main Results in Chapter 2** . . . . . **95**

**B Proofs of the Main Results in Chapter 4** . . . . . **108**

# ABSTRACT

Recent advances in technology have created an abundance of high-dimensional data and made its analysis possible (gene arrays, stock prices, text retrieval, recommender systems, and many others). These data require new, computationally efficient methodology and new kind of asymptotic analysis. This thesis consists of four projects that deal with high-dimensional data with complex structures.

The first project focuses on the graph estimation problem for Gaussian graphical models. Graphical models are commonly used in representing conditional independence between random variables, and learning the conditional independence structure from data has attracted much attention in recent years. However, almost all commonly used graph learning methods rely on the assumption that the observations share the same mean vector. In the first project, we extend the Gaussian graphical model to the setting where the observations are connected by a network and the mean vector can be different for different observations. We propose an efficient estimation method for the model, and under the assumption of network cohesion, we show that our method can accurately estimate the inverse covariance matrix as well as the corresponding graph structure, both from the theoretical perspective and using numerical studies. To further demonstrate the effectiveness of the proposed method, we also analyze a statisticians' coauthorship network data to learn the term dependency based on statistics publications.

The second project addresses the directed acyclic graph (DAG) estimation problem. DAG is a commonly used tool to encode causal relationships between random variables. Estimation of the DAG structure is often a challenging problem as the computational com-

plexity scales exponentially in the graph size when the total ordering of the DAG is unknown. To reduce the computational cost, and also with the aim of improving the estimation accuracy via the bias-variance trade-off, we propose a two-step approach for estimating the DAG, when data are generated from a linear structural equation model. In the first step, we infer the moral graph of the DAG via estimation of the inverse covariance matrix, which reduces the parameter space that one would search for the DAG. In the second step, we apply a penalized likelihood method for estimating the DAG restricted in the reduced space. Numerical studies indicate that the proposed method compares favorably with the one-step method in terms of both computational cost and estimation accuracy.

The third and fourth projects investigate supervised learning problems. Specifically, in the third project, we study the cointegration problem for multivariate time series data and propose a method for identifying cointegrating vectors with simultaneously group and elementwise sparse structures. Such a sparsity structure enables the elimination of certain coordinates of the original multivariate series from all cointegrated series, leading to parsimonious and potentially more interpretable cointegrating vectors. Specifically, we formulate an optimization problem based on the profile likelihood and propose an iterative algorithm for solving the optimization problem. The proposed method has been evaluated on synthetic data and also applied to two real world data examples involving daily prices of financial sector stocks and monthly treasury yields of different maturities. In the fourth project, we focus on the learning to rank problem with sparse feature selection. In particular, we extend the rank support vector machine method to the sparse setting, by applying the lasso and elastic-net penalties. We also employ the bundle method and the order statistic tree data structure to reduce the computational complexity. Numerical results indicate that the proposed method works well in both simulation studies and a real world stock selection problem.

# CHAPTER 1

# Introduction

Recent advances in technology have created an abundance of high-dimensional data (gene arrays, stock prices, text retrieval, recommender systems, and many others) and posed both computational and theoretical challenges that traditional statistical methods do not address. This thesis consists of four projects that deal with high-dimensional data with complex structures.

Before delving into specific individual projects, we first briefly summarize some of the existing methods for high-dimensional data that are directly relevant to our developments.

## 1.1 Regularized least squares regression

In standard regression problems, we are given a set of training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where the input (prediction variables) $x_i \in \mathbb{R}^p$ and the output (response variable) $y_i \in \mathbb{R}$. The aim is to find a model $\hat{f}(x)$ from the training data, so that when given a new input $x$, we can make a prediction for the output.

Consider the linear model

$$y_i = \beta_0 + x_i^T \beta + \epsilon_i, \tag{1.1}$$

where $\epsilon_i$'s are i.i.d. $N(0, \sigma^2)$. The standard least squares estimate for $\beta$ is given by

$$\hat{\beta}^{\text{ols}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)^2. \tag{1.2}$$

It is well-known that when two or more of the prediction variables are highly correlated, the ordinary least squares estimate tends to be unstable, and hence the prediction accuracy is jeopardized due to the bias-variance trade-off. To address this multi-collinearity problem, Hoerl and Kennard (1970) proposed the ridge regression, which penalizes the $\ell_2$-norm of

the regression coefficient vector, i.e.

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)^2 + \lambda ||\beta||_2^2, \tag{1.3}$$

where $\lambda$ is a tuning parameter that controls the bias-variance trade-off. When $\lambda$ is 0, we gain back the ordinary least squares estimate, which has the smallest bias but potentially large variance, while when $\lambda$ goes to $\infty$, we have $\hat{\beta}^{\text{ridge}} \to 0$, which has zero variance but large bias. One can show that there always exists a $\lambda$ such that the mean squared error (MSE) of $\hat{\beta}^{\text{ridge}}$ is less that of $\hat{\beta}^{\text{ols}}$.

Tibshirani (1997) proposed another regularized variation of the least squares regression, i.e. lasso, by penalizing the $\ell_1$-norm of the regression coefficient vector:

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)^2 + \lambda ||\beta||_1, \tag{1.4}$$

where $||\beta||_1 = |\beta_1| + \cdots + |\beta_p|$. Note that the absolute value function is non-differentiable at 0, and when $\lambda$ is large enough, this non-differentiability renders some of the estimated $\hat{\beta}_j$ to be exact zero, which implies that lasso can do automatic variable selection (i.e. offering a sparse model), in addition to improving the prediction accuracy via the bias-variance trade-off. For that reason, lasso is often preferred over the ridge regression in high-dimensional data modeling.

Two major limitations of lasso are: 1) the number of selected variables by lasso is upper bounded by the sample size $n$, which is often considered as undesirable in the high-dimensional setting, where the number of variables $p$ can be much larger than $n$; 2) when there are highly correlated prediction variables, lasso tends to select only one or few from the group, and this again is considered as undesirable as which variables are selected (from the highly correlated group) are a little arbitrary.

To address these two limitations, Zou and Hastie (2005) proposed the elastic-net method, which penalizes a combination of the $\ell_2$-norm and the $\ell_1$-norm of the regression coefficients:

$$\hat{\beta}^{\text{e-net}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)^2 + \lambda_1 ||\beta||_1 + \lambda_2 ||\beta||_2^2. \tag{1.5}$$

Now the number of selected variables is no longer bounded by $n$, and it can be as large as $p$ (when $n < p$), and when there are highly correlated prediction variables, they tend to be either selected together in the fitted model or removed together from the fitted model.

## 1.2 Gaussian graphical model

We consider the following Gaussian graphical model with $p$ variables, denoted by $X$:

$$X = (X_1, \cdots, X_p)^T \sim N_p(0, \Sigma), \tag{1.6}$$

where $\Sigma$ is a positive-definite covariance matrix. Denote $\Theta$ as the inverse covariance matrix, i.e. $\Theta \equiv \Sigma^{-1}$. It is well-known that under the Gaussianity assumption, $\Theta$ encodes the conditional dependence/independence between the variables, i.e., if $\Theta_{jj'} = 0$ ($\Theta_{jj'} \neq 0$), it implies that $X_j$ and $X_{j'}$ are independent (dependent) of each other conditional on all other variables $X_{\setminus\{j,j'\}}$, and vice versa. Thus, $\Theta$ is often the primary parameter of interest.

Here we briefly review two methods for estimating $\Theta$ that serve as building blocks for our work. Specifically, one is the neighborhood selection method by Meinshausen and Bühlmann (2006), and the other is the graphical lasso method which has been investigated by several researchers (Banerjee et al., 2008, Friedman et al., 2008, Yuan and Lin, 2007).

Given the model in (1.6), the relationship among the variables can be equivalently written as follows:

$$X_j = \sum_{j' \neq j} \beta_{jj'} X_{j'} + Z_j, \;\; j = 1, \ldots, p, \tag{1.7}$$

where $Z_j \sim \mathcal{N}(0, \sigma^2_{Z_j})$ and is independent of $\{X_{j'}, j' \neq j\}$. Moreover, the regression coefficients $\beta_{jj'}$'s and the variance $\sigma^2_{Z_j}$'s are given by:

$$\beta_{jj'} = -\Theta_{jj'}/\Theta_{jj}, \quad \sigma^2_{Z_j} = 1/\Theta_{jj}. \tag{1.8}$$

According to (1.8), identifying the zeros (or equivalently nonzeros) in $\Theta$ can be equivalently accomplished by determining whether $\beta_{jj'}$'s are zeros or not, where $\beta_{jj'}$ can be estimated by regressing $X_j$ on $\{X_{j'}, j' \neq j\}$.

Under certain sparsity assumptions, Meinshausen and Bühlmann (2006) proposed to fit $p$ separate lasso regressions to obtain sparse estimates of $\beta_{jj'}$'s, i.e.

$$\left\{\widehat{\beta}_{jj'}\right\} = \underset{\beta_{jj'}}{\operatorname{argmin}} \sum_{i=1}^{n} (x_{ij} - \sum_{j' \neq j} \beta_{jj'} x_{ij'})^2 + \lambda \sum_{j' \neq j} |\beta_{jj'}|, \;\; j = 1, \ldots, p. \tag{1.9}$$

With certain appropriate post-processing steps, one is able to obtain an estimate of the zero/non-zero positions in $\Theta$, as well as an estimate of $\Theta$.

Another type of methods for estimating $\Theta$ is based on regularized likelihood; specifically, Yuan and Lin (2007) and Banerjee et al. (2008) proposed to apply the $\ell_1$-norm penalty

to the negative log-likelihood, i.e.

$$\widehat{\Theta} = \underset{\Theta \succ 0}{\operatorname{argmin}} \operatorname{tr}(\Theta \widehat{\Sigma}) - \log\det\Theta + \lambda \sum_{j \neq j'} |\Theta_{jj'}|, \tag{1.10}$$

where $\widehat{\Sigma}$ is an estimate for $\Sigma$, e.g., the sample covariance matrix of $X$. The optimization in (1.10) is non-trivial, as $\Theta$ is required to be positive definite. To that end, Friedman et al. (2008) developed an efficient algorithm using the block coordinate descent strategy, and the algorithm is often referred as "glasso" in the literature.

## 1.3 Outline of the thesis

Chapter 2 focuses on the graph estimation problem for Gaussian graphical models. Graphical models are commonly used in representing conditional independence between random variables, and learning the conditional independence structure from data has attracted much attention in recent years. However, almost all commonly used graph learning methods rely on the assumption that the observations share the same mean vector. In Chapter 2, we extend the Gaussian graphical model to the setting where the observations are connected by a network and the mean vector can be different for different observations. We propose an efficient estimation method for the model, and under the assumption of network cohesion, we show that our method can accurately estimate the inverse covariance matrix as well as the corresponding graph structure, both from the theoretical perspective and using numerical studies. To further demonstrate the effectiveness of the proposed method, we also analyze a statisticians' coauthorship network data to learn the term dependency based on statistics publications.

Chapter 3 addresses the directed acyclic graph (DAG) estimation problem. DAG is a commonly used tool to encode causal relationships between random variables. Estimation of the DAG structure is often a challenging problem as the computational complexity scales exponentially in the graph size when the total ordering of the DAG is unknown. To reduce the computational cost, and also with the aim of improving the estimation accuracy via the bias-variance trade-off, we propose a two-step approach for estimating the DAG, when data are generated from a linear structural equation model. In the first step, we infer the moral graph of the DAG via estimation of the inverse covariance matrix, which reduces the parameter space that one would search for the DAG. In the second step, we apply a penalized likelihood method for estimating the DAG restricted in the reduced space. Numerical studies indicate that the proposed method compares favorably with the one-step method in

terms of both computational cost and estimation accuracy.

Chapters 4 and 5 investigate supervised learning problems. Specifically, in Chapter 4, we study the cointegration problem for multivariate time series data and propose a method for identifying cointegrating vectors with simultaneously group and elementwise sparse structures. Such a sparsity structure enables the elimination of certain coordinates of the original multivariate series from all cointegrated series, leading to parsimonious and potentially more interpretable cointegrating vectors. Specifically, we formulate an optimization problem based on the profile likelihood and propose an iterative algorithm for solving the optimization problem. The proposed method has been evaluated on synthetic data and also applied to two real world data examples involving daily prices of financial sector stocks and monthly treasury yields of different maturities. In Chapter 5, we focus on the learning to rank problem with sparse feature selection. In particular, we extend the rank support vector machine method to the sparse setting, by applying the lasso and elastic-net penalties. We also employ the bundle method and the order statistic tree data structure to reduce the computational complexity. Numerical results indicate that the proposed method works well in both simulation studies and a real world stock selection problem.

# CHAPTER 2

# Gaussian Graphical Models on Network-linked Data

## 2.1 Introduction

Network data provide information about pair-wise relations or interactions between units, such as friendship or collaboration between people, neighborhood between locations etc. Nowadays, modern data collection techniques make it possible to collect such network information in more and more situations on top of the traditional covariates such as characteristics of each person, gene expressions of each patient etc. Incorporating the network information in statistical modeling is expected to be able to improve statistical estimation or prediction performance as the network offers additional information about the relation between different observations. However, one challenge lies in that traditional methods for data analysis, such as regression models, density estimation and clustering methods typically assume the samples are independent and do not extend to situations when the samples are connected by a network. Though there are many earlier work for specific settings (Lee, 2007, Manski, 1993, Raducanu and Dornaika, 2012, Vural and Guillemot, 2016, Yang et al., 2011), substantial progress has been made only recently on extending many of the standard statistical methods to incorporate network structures, such as Li et al. (2016) for regression, Tang et al. (2013) for classification, and Yang et al. (2013), Binkiewicz et al. (2014) for clustering. In this chapter, we generalize the widely used Gaussian graphical model to incorporate network information.

Graphical models are commonly used to represent pairwise relationship between a group of random variables, in which each node of the graph corresponds to a random variable and an edge between two nodes represents conditional or marginal dependence between the two random variables. Graphical models have received extensive attention in the fields of statistics and machine learning, due to its wide application in biological

problems, text mining and causal inference, to name a few. The Gaussian graphical model is a special member of the family of undirected graphical models (a.k.a. Markov random field) where the joint distribution of random variables is assumed to be Gaussian. In such a graphical model, two disconnected nodes are interpreted to be conditionally independent given all the other random variables. When Gaussian distribution is assumed for the data, the conditional independence is fully characterized by the covariance matrix. In particular, random variables $j$ and $j'$ are conditionally independent given the rest if and only if the $(j, j')$th entry of the inverse covariance matrix (a.k.a precision matrix) is zero. As a result, estimating the graph structure for Gaussian graphical model is equivalent to identifying the zero positions of the precision matrix. There has been a large body of work in estimating the graph structure under Gaussian graphical models, especially in high dimensional situations when the number of variables is close to or much larger than the number of observations. Meinshausen and Bühlmann (2006) proposed a node-wise regression method with the lasso penalty that is fast and renders good asymptotic properties in estimating the graph. A wide class of methods based on penalized likelihood (Banerjee et al., 2008, d'Aspremont et al., 2008, Friedman et al., 2008, Rothman et al., 2008, Yuan and Lin, 2007) are proposed later. In particular, the graphical lasso algorithm of Friedman et al. (2008) is one of the most widely used algorithms for the problem due to its computational efficiency.

The above line of work on Gaussian graphical models assume the observations come from the same distribution, an assumption that is restrictive in many real-world situations. Zhou et al. (2010) extend the model by allowing time stamps along with the observations, and the covariance matrix varies smoothly over time. On the other hand, Guo et al. (2011), Danaher et al. (2014) and Mohan et al. (2014) assume there are multiple groups of observations for which the covariance matrices are different but still similar across the groups. In these extensions, the data observations are assumed to share the same mean vector. However, this may not be the case in many real-world applications. One special case that was considered before is when in addition to the multivariate Gaussian random variables, additional covariate vectors associated with each observation are also given, in which a sparse linear relationship is assumed between the mean vectors and the covariates (Cai et al., 2013, Lee and Liu, 2012, Lin et al., 2016, Rothman et al., 2010, Yin and Li, 2011, 2013). Though many applications in gene analysis involve data sets in this format, the assumption of having covariates and a sparse linear mapping between mean vectors and covariates is still restrictive in many problems.

In this chapter, we consider the problem of estimating a graphical model with het-

erogeneous mean vectors when a network connecting the observations is available. For example, in analyzing the word frequencies of researchers in writing papers, the conditional dependence may represent certain conceptual connections between different words and is expected to be universal for all people. However, as people have personal writing styles and different research interests, it is more reasonable to assume the expected word frequencies for different researchers are different. In such problems, we may have the collaboration network of the researchers as additional information which can be used to help in model estimation. In this chapter, we propose a generalization of the Gaussian graphical model to such setting where each data point can have individual expectation but all the data points share the same covariance structure. In addition to the data matrix, we assume a network connecting all of the observations is available. We thus define a Gaussian graphical model with network cohesion (GNC), where the term "network cohesion" refers to the phenomenon that connected nodes have similar mean vectors (Li et al., 2016). As the network is one of the most general data structure to represent pairwise relationship, our model can also be used to handle many other types of data commonly encountered in data analysis that are not necessarily directly represented in the network format, such as multivariate functional data and high dimensional spatial observations. Our contributions in this chapter include:

1. We propose a generalized Gaussian graphical model for multivariate network-linked data as well as an efficient algorithm to estimate the model.

2. Under the network cohesion assumption, we provide theoretical guarantees for our method to consistently estimate the Gaussian covariance matrix and the corresponding graph structure in high dimensional settings where the number of variables can be much larger than the number of observations.

3. We show that the graphical model estimation admits an oracle estimation property, in the sense that, even if the true covariance estimate is known, one cannot achieve a better estimation of the the mean vectors than our method under the same penalized maximum likelihood framework. In particular, our estimate cannot be improved by a more complicated and computationally demanding penalized likelihood estimation, which is intuitively expected to be better.

The rest of the chapter is organized as follows. Section 2.2 introduces a Gaussian graphical model on network-linked observations and the corresponding two-stage model estimation procedure. Section 2.3 presents a rigorous definition of network cohesion and error bounds of our estimation under the assumption of network cohesion. It also shows

that the estimation cannot be improved by a joint penalized likelihood estimation with the network cohesion penalty, though intuitively the latter uses the data more thoroughly. Section 2.4 presents simulation studies on the performance of the proposed method under various settings and comparisons with its iterative version as well as the standard graphical lasso. Section 2.5 presents an example of applying the proposed method to analyze dependences beteween statistical terms, and Section 2.6 summarizes the chapter with discussion.

**Notations.** Given a matrix $X \in \mathbb{R}^{n\times p}$, let $X_{\cdot j}$ denote the $j$th column and $X_{i\cdot}$ denote the $i$th row. Let $||X||_0 = \#\{(i,j) : X_{ij} \neq 0\}$ be the $L_0$ norm and $||X||_1 = \sum_{ij} |X_{ij}|$ be the $L_1$ norm. A special variant that will be used is the $L_1$ norm constrained on off-diagonal elements, which is $||X||_{1,\text{off}} = \sum_{i\neq j} |X_{ij}|$. For a square matrix $\Sigma$, let $\text{tr}(\Sigma)$ and $\det(\Sigma)$ be the trace and determinant of $\Sigma$ respectively. By default, we treat all vectors as column vectors. A network or graph connecting $n$ nodes are denoted by $\mathcal{G}_n$, with the subscript $n$ usually being omitted when it is clear in context. Furthermore, if two nodes $i$ and $i'$ of $\mathcal{G}_n$ are connected, we write $i \sim_{\mathcal{G}_n} i'$. The adjacency matrix of a network $\mathcal{G}_n$ is an $n \times n$ matrix $A$, such that $A_{ii'} = 1$ if $i \sim_{\mathcal{G}_n} i'$ and 0 otherwise. Note that for any undirected network (which will be the network we consider in this chapter), the adjacency matrix is symmetric. For the adjacency matrix $A$, we define its Laplacian by $L = D - A$ where $D = \textbf{diag}(d_1, d_2, \cdots, d_n)$ and $d_i = \sum_{i'=1}^n A_{ii'}$ is the degree of node $i$. In addition, define the normalized Laplacian $\mathcal{L}_n = D^{-1/2}LD^{-1/2}$, as well as the approximately normalized Laplacian $\mathcal{L}_s = \frac{1}{\bar{d}}L$ where $\bar{d}$ is the average degree of the network $\mathcal{G}$, given by $\frac{1}{n}\sum_i d_i$. Assume $\tau_1 \geq \tau_2 \geq \cdots, \geq \tau_{n-1} > \tau_n = 0$ be the eigenvalues of $\mathcal{L}_s$, with corresponding eigenvectors $u_i$'s.

## 2.2 Gaussian graphical model with network cohesion

Assume that the data matrix we have is $X$, recording $n$ independent observations $X_{i\cdot} \in \mathbb{R}^p, i = 1, 2, \cdots, n$, such that each $X_{i\cdot}$ is a random vector from a multivariate Gaussian distribution

$$X_{i\cdot} \sim \mathcal{N}(\mu_{i\cdot}, \Sigma), i = 1, 2, \cdots, n,$$

where $\mu_{i\cdot} \in \mathbb{R}^p$ is a $p$-dimensional vector and $\Sigma \in \mathbb{S}_+^p$, in which $\mathbb{S}_+^p$ is the set of $p \times p$ symmetric positive definite matrices. Let $\Theta = \Sigma^{-1}$ be the precision matrix and

$$M = (\mu_{1\cdot}, \mu_{2\cdot}, \cdots, \mu_{n\cdot})^T = (\mu_{\cdot 1}, \mu_{\cdot 2}, \cdots, \mu_{\cdot p})$$

be the mean matrix. The log-likelihood on the observations can be written up to a constant as

$$\log\det(\Theta) - \text{tr}(\Theta(X-M)^T(X-M))/n. \tag{2.1}$$

The Gaussian distribution is naturally associated with an undirected graph $\mathcal{G}^{(v)}$ in which each of the $p$ coordinates corresponds to a node in the graph and the pairs of nodes $j, j'$ such that $j \nsim_{\mathcal{G}^{(v)}} j'$ corresponds to the conditional independence relationship $x_j \perp x'_j | \{x_k, k \neq j, j'\}$ (Lauritzen, 1996). Note that we assume all the observations share the same covariance structure but may have different mean vector $\mu_i$'s. In the special case when $\mu_{i\cdot} = \mu$ for all $i = 1, 2, \cdots, n$, the model is the standard i.i.d. multivariate Gaussian model. In general, the model defined above involves $np + p(p-1)/2$ parameters and is not estimable with only $n$ observation. We make two further assumptions on the model:

- In a graphical model, one typically assumes that the graph structure between random variables is sparse, which is equivalent to $\Theta$ being a sparse matrix. This is the same assumption made in almost all Gaussian graphical model estimation methods.

- Further, let $\mathcal{G}^{(o)}$ be the network connecting all $n$ observations $X_{i\cdot}$; we assume that $\mu_{i\cdot}$'s are cohesive over the network. That means $\mu_{ik}$ and $\mu_{i'k}$ are similar if node $i, i'$ are connected $i \sim_{\mathcal{G}^{(o)}} i'$. We measure the network incohesiveness of vector $\mu_{\cdot k}$ by

$$\sum_{i \sim_{\mathcal{G}} i'} d(|\mu_{ik} - \mu_{i'k}|),$$

  and assume this incohesiveness measure to be small, where $d(\cdot)$ can be defined to be any reasonable function to measure the magnitude of a scalar. Two natural choices are $d(|\mu_{ik} - \mu_{i'k}|) = (\mu_{ik} - \mu_{i'k})^2$ (Li et al., 2016) and $d(|\mu_{ik} - \mu_{i'k}|) = |\mu_{ik} - \mu_{i'k}|$ (Hallac et al., 2015). For computational efficiency, we will focus on the squared difference in this chapter. We specifically write the cohesiveness measure by its equivalent form

$$\mu^T \mathcal{L}_s \mu = \frac{1}{\bar{d}} \sum_{i \sim i'} (\mu_i - \mu'_i)^2$$

  and refer this quantity as the *cohesion penalty*.

As there are two graphs involved in our setting, we assign specific names to them in order to avoid confusion in later discussion: the observed network that connects $n$ observations, $\mathcal{G}^{(o)}$, is referred as the "observation network"; the graph structure among $p$ random variables that is to be estimated, $\mathcal{G}^{(v)}$, is referred as the "Gaussian graph".

### 2.2.1 Model estimation

We use the following two-stage procedure to estimate the model and refer it as the Gaussian graphical estimation with Network Cohesion and lasso penalty (GNC-lasso) .

**Algorithm 1:** Two-stage GNC-lasso algorithm

1 Given $X$, $\lambda$ and $\alpha$.

1. Mean estimation: Minimize

$$||x_{\cdot j} - \mu_{\cdot j}||_2^2 + \alpha \mu_{\cdot j}^T \mathcal{L}_s \mu_{\cdot j} \tag{2.2}$$

separately for each $j = 1, \ldots, p$. Denote the estimate as $\hat{M}$.

2. Covariance estimation: Let $\hat{S} = \frac{1}{n}(X - \hat{M})^T(X - \hat{M})$. Solve for $\hat{\Theta}$ by

$$\min_{\Theta \in \mathbb{S}_+^n} \log\det(\Theta) - \text{tr}(\Theta \hat{S}) - \lambda ||\Theta||_{1,\text{off}}. \tag{2.3}$$

Note criterion (2.2) is a univariate Laplacian smoothing and has a closed form solution

$$\hat{\mu}_{\cdot j} = (I_n + \alpha \mathcal{L}_s)^{-1} x_{\cdot j}. \tag{2.4}$$

In practice, we usually need to compute the estimate on a sequence of $\alpha$ values, so we will first calculate the eigen-decomposition of $\mathcal{L}_s$, then each $(I + \alpha \mathcal{L}_s)^{-1}$ can be directly obtained in linear time. Since $\mathcal{L}_s$ is often sparse, taking advantage of that and using the fact that $\mathcal{L}_s$ is symmetrically diagonal dominant, the eigen-decomposition of $\mathcal{L}_s$ can be computed very efficiently (Cohen et al., 2014).

Given $\hat{M}$, criterion (2.3) is a graphical lasso problem that uses the lasso penalty (Tibshirani, 1996) to encourage sparsity in estimation. It can be solved by the glasso algorithm of Friedman et al. (2008). One can also include the diagonal elements of $\Theta$ in the penalty as was done in the original glasso algorithm. This subtle modification does not make much difference so we will not distinguish the two versions in later discussion.

Note that

$$\sum_j \mu_{\cdot j}^T \mathcal{L}_s \mu_{\cdot j} = \text{tr}(M^T \mathcal{L}_s M).$$

Therefore criterion (2.2) is essentially solving

$$\min_{M\in\mathbb{R}^{n\times p}} \text{tr}((X-M)^T(X-M)) + \alpha\text{tr}(M^T\mathcal{L}_s M)$$

which is a cohesion penalized Gaussian log-likelihood with the covariance being $\sigma^2 I_p$ for some $\sigma$. It is thus natural to instead use a penalized log-likelihood to estimate both $M$ and $\Theta$ jointly by

$$\max_{\Theta,M} \ \log\det(\Theta) - \frac{1}{n}\text{tr}(\Theta(X-M)^T(X-M)) - \lambda||\Theta||_{1,\text{off}} - \frac{\alpha}{n}\text{tr}(M^T\mathcal{L}_s M). \quad (2.5)$$

The above optimization is bi-convex and we can iteratively solve $M$ by fixing $\Theta$ and then solve $\Theta$ by fixing $M$, with initialization done by the two-stage procedure. We refer this method as "iterative GNC-lasso". The computational complexity of the iterative method is significantly higher than the two-stage estimation and can hardly handle problems in which either $n$ or $p$ is large. To see this, notice that in each iteration when fixing $\Theta$ and maximizing over $M$, all $p$ coordinates are coupled resulting in a problem of scale $np \times np$ (see Proposition 2). Thus for even moderate $n$ and $p$, one needs either huge memory or resorting to certain Gauss-Seidel type algorithms that further increase the number of iterations. This disadvantage of the iterative estimation will be amplified when one needs to select $\lambda$ and $\alpha$, as will be discussed in next subsection. More importantly, although intuitively the iterative method is expected to provide better model estimation, we will show later by both theoretical and empirical results that such iterative estimation does not bring improvements. Therefore, we will focus on the two-stage GNC-lasso throughout this chapter.

Finally, we emphasize that, between the two model parameters $\Theta$ and $M$, our primary interest is to estimate $\Theta$ and will treat $M$ as a nuisance parameter to some extent. This is because $\Theta$ depicts the underlying population, while $\mu_{i\cdot}$ only provides individual effects that may not have generalizable knowledge of the data.

### 2.2.2 Model selection

There are two tuning parameters $\lambda$ and $\alpha$ in GNC-lasso. Note $\alpha$ is the tuning parameter controlling the mean estimation and such estimation can be easily validated by checking predictive performance. In particular, in this chapter, we always tune $\alpha$ from a sequence of candidate values by 10-fold cross-validation. In each validation, the sum of squared prediction errors on validation set $\sum(X_{ij}-\hat{\mu}_{ij})^2$ is computed and the $\alpha$ that renders the minimum average error in all folds is used. Given $\alpha$, we obtain $\hat{M}$ and use $\hat{S} = \frac{1}{n}(X-\hat{M})^T(X-\hat{M})$

as the input of the glasso problem in (2.3); therefore $\lambda$ can be selected by standard parameter tuning methods for glasso. Such tuning methods can be chosen according to specific applications. For example, if prediction is the major interest, it can be selected again by cross-validation. On the other hand, Gaussian graphical model is very often used as an exploratory tool to obtain interpretable partial correlations between variables, and in this situation $\lambda$ can also be selected to achieve a predefined sparsity level of the estimated graph or according to the easiness of interpretation.

Note that the model selection step illustrates another important advantage of using two-stage estimation instead of the iterative method. In the iterative method, the coupling of $\alpha$ and $\lambda$ makes it difficult to tune them separately by different criteria. Moreover, even if one is willing to tune the parameters under the same criterion, due to the coupling, the model fitting must be done on a grid of $(\alpha, \lambda)$ pairs, which involves $m_1 \times m_2$ model fittings if $m_1$ $\alpha$ values and $m_2$ $\lambda$ values are to be considered. In contrast, the two-stage method only needs $m_1 + m_2$ times of model fittings instead, which is substantially more efficient, in addition to the fact that the two-stage estimation can be many times faster than the iterative method for each model fitting.

## 2.3 Theoretical properties

In this section, we investigate theoretical properties of the GNC-lasso estimator. Throughout this section, we always assume the observation network $\mathcal{G}^{(o)}$ is connected, without loss of generality. Recall that $\tau_1 \geq \tau_2 \geq \cdots, \geq \tau_{n-1} > \tau_n = 0$ are the eigenvalues of $\mathcal{L}_s$, with corresponding eigenvectors $u_i$'s. We also need the following notations for matrix norms: given a matrix $M$, let $||M||$ be its spectral norm and $||M||_F$ be its Frobenius norm. In addition, given two quantities $a_n$ and $b_n$, which depend on $n$, we use $a_n \gtrsim b_n$ to denote the fact that $b_n \leq C a_n$ for some constant $C$, which can also be written as $b_n = O(a_n)$.

### 2.3.1 Cohesive assumptions on the observation network

Before proceeding to our estimators, we first answer the question: given a vector $\mu \in \mathbb{R}^n$ over the network, what is a reasonable mathematical representation for the assumption "$\mu$ is cohesive on the network?" Intuitively, we can define cohesion as a condition that the cohesion penalty $\mu^T \mathcal{L}_s \mu$ is small in certain sense. Alternatively, an equivalent way is to require $||\mathcal{L}_s \mu||_2$ to be small, because $\mathcal{L}_s \mu$ is the gradient of the cohesion penalty up to a

constant and

$$||\mathcal{L}_s\mu||_2 \to 0 \iff \mu^T\mathcal{L}_s\mu \to 0.$$

It turns out that defining cohesion with respect to $\mathcal{L}_s\mu$ is easier for later derivations, so we will take this option. The vector $\mathcal{L}_s\mu$ itself also has nice interpretation. Note the $i$th coordinate of $\mathcal{L}_s\mu$ is given by

$$\frac{d_i}{\bar{d}}(\mu_i - \frac{1}{d_i}\sum_{i'\sim_{\mathcal{G}^{(o)}} i}\mu_i'),$$

therefore $\mathcal{L}_s\mu$ represents the difference between $\mu_i$ and its local average for all nodes $i$. Let $\mathcal{L}_s = U\Lambda U^T$ be the eigen-decomposition of $\mathcal{L}_s$ in which the eigenvalues $\tau_i$ are sorted in decreasing order. For any $\mu \in \mathbb{R}^n$, we can represent $\mu$ by its basis expansion $\mu = U\beta = \sum_{i=1}^n \beta_i u_i$ where $\beta \in \mathbb{R}^n$ and each $\beta_i$ is the magnitude of $\mu$ in the direction of $u_i$. In any reasonable cohesion assumption, we expect $||\mathcal{L}_s\mu||_2^2$ to be much smaller than $||\mu||_2^2$. Notice that

$$||\mu||_2^2 = ||\beta||_2^2 \text{ and } ||\mathcal{L}_s\mu||_2^2 = \sum_i \tau_i^2\beta_i^2.$$

Therefore we specifically make the follow assumption as our requirement of a vector $\mu$ being cohesive over the network:

**Assumption 1** (Cohesion assumption)**.** *Given a network $\mathcal{G}^{(o)}$. Let $\mu = \sum_{i=1}^n \beta_i u_i$ be the basis expansion of $\mu$ according to the eigenvectors of $\mathcal{L}_s$. For some positive constants $N_G$ and $\delta$, we have*

$$\textbf{Scale: } ||\mu||_2^2 = ||\beta||_2^2 = \sum_i \beta_i^2 = N_G \cdot n, \tag{2.6}$$

$$\textbf{Cohesion: } \tau_i|\beta_i| \le n^{-\frac{1+\delta}{3}}, \forall i \in [n]. \tag{2.7}$$

*These two indicate*

$$||\mathcal{L}_s\mu||_2^2 \le n^{\frac{1-2\delta}{3}} \ll N_G n = ||\mu||_2^2.$$

While the cohesion assumption specifies what vectors are considered as cohesive in our theoretical analysis, the assumption alone is not enough to ensure the estimability of the model. For a given integer $m$, define $t(m) := \tau_{n-m}$. To ensure the cohesion assumption can effectively control the model complexity, we make the following assumption about the observation network $\mathcal{G}^{(o)}$ itself.

**Assumption 2.** *There exists $m$ such that*

$$c_G \cdot t(m) \geq \frac{1}{\sqrt{m}} \tag{2.8}$$

*for some constant $c_G$. In particular, define $m_G$ to be the smallest $m$ that satisfies* (2.8)*. We call such $m_G$ the* ***cohesive dimension*** *of the network.*

Assumption 2 is more a definition than assumption, as we can always take $m = n - 1$ and the fact that $\tau_1 \geq \frac{n}{n-1}$ ensures the assumption trivially holds. Therefore, we always have the cohesive dimension $m_G \leq n - 1$. We call $m_G$ the cohesive dimension since it in some sense measures the effective number of parameters we need to estimate after assuming network cohesion. It indicates that if we require $||\mathcal{L}_s\mu||_2^2 = O(\frac{1}{m_G})$, then we are at least free to pick up an arbitrary vector $\mu$ from a subspace of dimension $m_G$ spanned by the last $m_G$ eigenvectors of $\mathcal{L}_s$ without any constraint. From this interpretation, it is clear that we hope $m_G$ to be small, otherwise the complexity of the model will not be effectively reduced by the cohesion assumption.

We now distinguish two different situations of network cohesion. Obviously, if $\mu = c\mathbf{1}$ for a scalar $c$, it must be perfectly cohesive since all elements are the same and this is exactly what standard graphical model estimation methods such as glasso assume. However, this is ***trivial cohesion*** since there is no heterogeneity between observations. If the vector is trivially cohesive, one does not need to use the more general GNC-lasso. In our setting, a vector $\mu$ is said to be ***nontrivially cohesive*** if

$$||\mu - P_{\mathbf{1}}\mu||_2 \gtrsim ||P_{\mathbf{1}}\mu||_2$$

where $P_{\mathbf{1}}\mu$ is the projection of $\mu$ onto the subspace spanned by $\mathbf{1}$, i.e. a constant vector with all coordinates being the average of $\mu$. The nontrivial cohesion setting is the regime where GNC-lasso is primarily designed for and where we extend the standard graphical model estimation framework.

Now we justify that Assumptions 1 and 2 are reasonable and realistic by the following proposition. Specifically, we show that a lattice network's cohesive dimension is $O(n^{2/3})$, and nontrivial cohesion of $\mu$ over the lattice network is allowed under the two assumptions.

**Proposition 1** (Cohesive properties of a lattice network)**.** *Assume $\sqrt{n}$ is an integer and $\mathcal{G}^{(o)}$ is a $\sqrt{n} \times \sqrt{n}$ lattice network. Then we have:*

1. *The cohesive dimension $m_G \leq cn^{2/3}$ for some constant $c$.*

2. *There exists $\mu$ satisfying Assumption 1 with $\delta < 1/2$ such that $||\mu - P_{\mathbf{1}}\mu||_2^2 \geq c'n$ for some constant $c'$. In particular, this indicates that $||\mu - P_{\mathbf{1}}\mu||_2 \gtrsim ||P_{\mathbf{1}}\mu||_2$.*

### 2.3.2 Mean estimation error bounds

Now we proceed to discuss the estimation bound for the $n \times p$ mean estimate $\hat{M}$ in Algorithm 1. Formally, we assume a heterogeneous multivariate Gaussian model for the observed data matrix $X$:

**Assumption 3.** *Assume $X = M^* + E$ where $M^* = (\mu^*_{\cdot 1}, \mu^*_{\cdot 2}, \cdots, \mu^*_{\cdot p})$ and $E = (\epsilon_{ij})$ such that $\epsilon_{i\cdot} \sim_{i.i.d} \mathcal{N}(0, \Sigma^*)$ for some $\Sigma^* \in \mathbb{S}_n^+$. Let $\sigma^2 := \max_j \Sigma^*_{jj} > 0$. Moreover, assume $\log p < cn$ for some constant $0 < c < 1$.*

Finally, we make the same cohesive assumption for each column of $M^*$.

**Assumption 4.** *Assume Assumptions 1 holds for all $\mu^*_{\cdot j}$, $j = 1, 2, \cdots, p$.*

**Theorem 1** (Mean matrix error bound)**.** *Under Assumptions 2 - 4, let $\hat{M}$ be the estimated mean matrix $M$ from* (2.2)*. Then we have*

$$||\hat{M} - M||_\infty \leq (2\sqrt{2}\sigma + 1)[c_G\sqrt{m_G}n^{\frac{2-\delta}{3}} + \sqrt{\log p}m_G] \tag{2.9}$$

*with probability at least $1 - \exp(-cn) - \exp(-Cm_G \log p)$ for some constants $c$ and $C$. In Frobenius norm, we have*

$$||\hat{M} - M^*||_F \leq \sqrt{(1 + 4\sigma^2)(c_G^2 m_G n^{\frac{1-2\delta}{3}} + 1)p} \tag{2.10}$$

*with probability at least $1 - \exp(-p(n - m_G)) - \exp(-pm_G)$.*

Note the theorem shows that we may not achieve vanishing errors for all entries of $M$. This is expected as the cohesion penalty is a ridge-type penalty and it is known that ridge regression does not enjoy vanishing estimation errors in general. Nevertheless, the average error measured by $||\hat{M} - M^*||_F/\sqrt{np}$ can still be vanishing as long as the cohesive dimension $m_G = O(n^{\frac{2}{3}})$ as in the case when $\mathcal{G}^{(o)}$ is a lattice network. Then as we will show in the next subsection, $\hat{M}$ is an adequately accurate estimate of $M$ that ensures good estimation properties of the precision matrix and the corresponding Gaussian graph structure, which is our primary target.

### 2.3.3 Inverse covariance estimation error bounds

For properties on the inverse covariance estimation, we need a few more notations and assumptions. Let $\Gamma^*$ be the Fisher information matrix of the model, defined as

$$\Gamma^* = \Sigma^* \otimes \Sigma^* \tag{2.11}$$

where $\otimes$ is the Kronecker product. In particular, under the multivariate Gaussian distribution, we have $\Gamma^*_{(j,k),(\ell,m)} = \mathsf{Cov}(X_j X_k, X_\ell X_m)$. Define the set of nonzero entries in $\Theta^*$ to be

$$S(\Theta^*) = \{(j, j') \in [n] \times [n] : \Theta^*_{jj'} \neq 0\}. \tag{2.12}$$

We use $S_o(\Theta^*)$ to denote the set of nonzero off-diagonal elements of $\Theta^*$ and $S^c(\Theta^*)$ to denote the complement of $S(\Theta^*)$. Let $s = |S_o(\Theta^*)|$ be the number of nonzero off-diagonal elements in $\Theta^*$. For any two sets $T_1, T_2 \subset [n] \times [n]$, let $\Gamma^*_{T_1,T_2}$ denote the submatrix with rows and columns taken in $T_1, T_2$ respectively. When it is clear in the context, we may suppress the notation $\Theta^*$ in $S(\Theta^*)$ and just write it as $S$. Let $\psi$ be the maximum number of nonzeros in each row of $\Theta^*$, which is also the maximum node degree of the Gaussian graph plus 1 :

$$\psi = \max_j ||\Theta^*_{j.}||_0. \tag{2.13}$$

Moreover, define

$$\kappa_{\Sigma^*} = ||\Sigma^*||_{\infty,\infty}, \tag{2.14}$$

which measures the overall magnitude of the covariances. We also define the parameter

$$\kappa_{\Gamma^*} = ||(\Gamma^*_{SS})^{-1}||_{\infty,\infty} \tag{2.15}$$

Finally, it is known that a necessary and sufficient condition for lasso regression to succeed in support recovery is the *irrepresentability* condition (Wainwright, 2009). Similarly, we need an edge-level irrepresentability condition here.

**Assumption 5.** *There exists some* $0 < \rho \leq 1$ *such that*

$$\max_{e \in S^c} ||\Gamma^*_{eS}(\Gamma^*_{SS})^{-1}||_1 \leq 1 - \rho.$$

When only Frobenius norm error bound is considered, a much weaker assumption is adequate without the requirements on $\psi, \kappa_{\Sigma^*}, \kappa_{\Gamma^*}$ and Assumption 5.

**Assumption 6.** *Let* $\eta_{\min}(\Sigma^*)$ *and* $\eta_{\max}(\Sigma^*)$ *be the minimum and maximum eigenvalues of*

$\Sigma^*$, *respectively. There exists a constant* $\bar{k}$ *such that*

$$\frac{1}{\bar{k}} \leq \eta_{\min}(\Sigma^*) \leq \eta_{\max}(\Sigma^*) \leq \bar{k}. \tag{2.16}$$

Let $\hat{S} = \frac{1}{n}(X - \hat{M})^T(X - \hat{M})$. We use $\hat{S}$ as the input for the glasso estimation of (2.3). The difference from our estimation and the standard glasso lies in the fact that our $\hat{S}$ is obtained by plug in our estimate $\hat{M}$ instead of using the true $M^*$. We would expect that if $\hat{M}$ is a reasonable estimate of $M^*$, $\Theta^*$ can still be accurately estimated. The following theorem confirms this intuition, based on a few concentration properties of $\hat{S}$ around $\Sigma^*$ and the proof strategy of Ravikumar et al. (2011).

**Theorem 2.** *Under the conditions of Theorem 1 and Assumption 5, there exist some positive constants* $C, c, c', c''$ *that only depend on* $N_G, c_G$ *and* $\sigma$*, such that if* $\hat{\Theta}$ *is the solution of the two-stage procedure of Algorithm 1 with* $\alpha = n^{\frac{1+\delta}{3}}$, $\lambda = \frac{8}{\rho}\nu(n,p)$ *where*

$$\begin{aligned}\nu(n,p) := C \max\Big( & \sqrt{\log pn} m_G n^{-\frac{2+2\delta}{3}}, \sqrt{\log pn}\sqrt{\log p} m_G^{3/2} n^{-\frac{4+\delta}{3}}, \\ & \sqrt{\log pn}\sqrt{m_G} n^{-\frac{1+\delta}{3}}, \sqrt{\log pn}\sqrt{\log p}\frac{m_G}{n}, \sqrt{\frac{\log p}{n}}\Big)\end{aligned}$$

*and* $n$ *is large enough to ensure*

$$\nu(n,p) < \frac{1}{6(1+8/\rho)\psi \max\{\kappa_{\Sigma^*}\kappa_{\Gamma^*}, (1+8/\rho)\kappa_{\Sigma^*}^3\kappa_{\Gamma^*}^2\}},$$

*then with probability at least* $1-\exp(-c\log(p(n-m_G)))-\exp(-c'\log(pm_G))-\exp(-c''\log p)$, *we have*

1. *The edge set is a subset of the true edge set, i.e.*

$$S_o(\hat{\Theta}) \subset S_o(\Theta^*).$$

2. *The estimate* $\hat{\Theta}$ *satisfies*

$$||\hat{\Theta} - \Theta^*||_\infty \leq 2(1+8/\rho)\kappa_{\Gamma^*}\nu(n,p). \tag{2.17}$$

3. *If in addition, all of the nonzero off-diagonal elements of* $\Theta^*$ *satisfy*

$$\max_{(j,j')\in S_o(\Theta^*)} |\Theta^*_{jj'}| > 2(1+8/\rho)\kappa_{\Gamma^*}\nu(n,p),$$

*then the edge set is exactly recovered by* $S(\hat{\Theta})$.

4. *In Frobenius norm, the estimate satisfies*

$$||\hat{\Theta} - \Theta^*||_F \leq 2(1 + 8/\rho)\kappa_{\Gamma^*}\nu(n,p)\sqrt{s+p}. \tag{2.18}$$

5. *In row-wise* $L_\infty$ *norm, the estimate satisfies*

$$||\hat{\Theta} - \Theta^*||_{\infty,\infty} \leq 2(1 + 8/\rho)\kappa_{\Gamma^*}\nu(n,p)\psi. \tag{2.19}$$

6. *In spectral norm, the estimate satisfies*

$$||\hat{\Theta} - \Theta^*|| \leq 2(1 + 8/\rho)\kappa_{\Gamma^*}\nu(n,p)\min(\sqrt{s+p}, \psi). \tag{2.20}$$

**Remark 1.** *1. To make the sample size requirement practical, we mostly need* $\kappa_{\Gamma^*}$, $\kappa_{\Sigma^*}$ *and* $\rho$ *to be constants or restricted in a bounded region, as in Ravikumar et al. (2011).*

*2. To achieve the Frobenius bound* (2.18)*, we do not need the irrepresentability assumption or the information of* $\kappa_{\Gamma^*}$ *and* $\kappa_{\Sigma^*}$*. Following the proof strategy in Rothman et al. (2008), we can have the same bound with Assumption 6.*

Compared with the standard glasso error bound (Ravikumar et al., 2011), the prices we pay for assuming different mean vectors under the cohesion assumption are the first four terms in the formula of $\nu$. As a result, we need a stronger requirement on the $p/n$ ratio in the high-dimensional setting, depending on the observation network $\mathcal{G}^{(o)}$. To see the comparison in a simpler format, we assume $m_G = O(n^{2/3})$, which holds for lattice networks and path networks.

**Corollary 1.** *Under the assumption of Theorem 2, if we have* $m_G \leq cn^{2/3}$ *for some constant* $c$ *and* $\delta < \frac{1}{2}$*, then all the results of Theorem 2 hold with*

$$\nu(n,p) \leq C\sqrt{\log np}n^{-\frac{\delta}{3}}$$

*where* $C$ *is a constant that only depends on* $N_G, c_G, \sigma$.

**Remark 2.** *1. According to Corollary 1, for estimation consistency in a Gaussian graphical model with nontrivially cohesive mean vectors, we need* $\log p = o(n^{\frac{2\delta}{3}})$ *for* $\delta < 1/2$*. This is strictly stronger than the condition* $\log p = o(n)$ *required in the case of i.i.d multivariate Gaussian problems (Ravikumar et al., 2011).*

2. *Note that the stronger requirement for $p/n$ ratio is simply because we want to allow for nontrivial cohesion. On the other hand, if we let $\delta \to \infty$, the cohesion assumption becomes trivial cohesion. The error bound in our theorem becomes the same as glasso (Ravikumar et al., 2011) and the requirement is still $\log p = o(n)$, so we do not sacrifice accuracy by using GNC-lasso.*

### 2.3.4 Oracle mean estimation and sufficiency of two-stage estimation

Given the error bound from the two-stage procedure, a natural question to ask is whether we can achieve better performance by optimizing the penalized joint log-likelihood (2.5). A good estimate of $\Theta$ reflects the covariance structure in the penalized joint likelihood and may help to produce a better estimation of $M$ than the separable estimation in the two-stage procedure, and such improved estimate of $M$ will in turn help to improve the estimate of $\Theta$ again. In this section, however, we will give a negative answer to that question. Let $(\tilde{\Theta}, \tilde{M})$ be the maximizer of (2.5). Note given $\tilde{\Theta}$, $\tilde{M}$ must be the maximizer of (2.5) as a function of $M$, and vice versa. We will show that such $\tilde{M}$ will not improve $\hat{M}$ produced by our two-stage estimation. In particular, under diagonal dominance assumption of the precision matrix, we will show that even if the true $\Theta$ is given by an oracle and used in place of $\tilde{\Theta}$, the maximizer of (2.5) over $M$ cannot reduce the estimation error of $\hat{M}$ by more than a trivial constant scale. Since the oracle estimate of $M$ cannot be better than $\hat{M}$, nor will $\tilde{M}$. Based on the theoretical discussion of Section 2.3.3, we can see the inverse covariance estimate will not be improved by $\tilde{M}$ either.

For the ease of derivation, we use the basis expansion in the spectrum of $\mathcal{L}_s$ again. Recall that $U$ is the matrix of eigenvectors of $\mathcal{L}_s$ and for any $M \in \mathbb{R}^{n\times p}$, we can write it as $M = UB$. As $U$ is orthonormal, estimating $M$ is equivalent to estimating $B$. We now specifically define the following two estimation objectives to estimate $B$:

$$\min_{B\in\mathbb{R}^{n\times p}} \text{tr}((X-UB)^T(X-UB)) + \alpha\text{tr}(B^T\Lambda B), \text{ and} \tag{2.21}$$

$$\min_{B\in\mathbb{R}^{n\times p}} \text{tr}(\Theta^*(X-UB)^T(X-UB)) + \alpha\text{tr}(B^T\Lambda B), \tag{2.22}$$

in which $\Lambda = \textbf{diag}(\tau_1, \tau_2, \cdots, \tau_n)$. It is not difficult to see that (2.21) is the mean estimation step (2.2) in the two-stage procedure (up to $U$), while (2.22) is the mean estimation procedure of maximizing (2.5), with $\Theta$ replaced by true $\Theta^*$. Therefore the estimation of (2.22) is an oracle estimate in the sense that we assume the true covariance matrix is already known. Though (2.22) is not applicable in practice, it serves as a benchmark for the

best performance one could expect in estimating $B$ (or equivalently $M$). Let $\hat{B}_1$ and $\hat{B}_2$ be the estimates from (2.21) and (2.22) respectively and let $W_k = B^* - \hat{B}_k$, $k = 1, 2$ be the estimation error matrices. Then we have the following result.

**Proposition 2.** *For the estimate from* (2.21)*, we have*

$$W_1 I_p + \alpha \Lambda W_1 = \alpha \Lambda B^* + \tilde{E}, \tag{2.23}$$

*where* $\tilde{E} = (\tilde{\epsilon}_{1\cdot}, \tilde{\epsilon}_{2\cdot}, \cdots, \tilde{\epsilon}_{n\cdot})$ *and* $\tilde{\epsilon}_{i\cdot} \sim_{\text{i.i.d}} \mathcal{N}(0, \Sigma^*)$*. For the estimate from* (2.22)*, we have*

$$W_2 \Theta^* + \alpha \Lambda W_2 = \alpha \Lambda B^* + \dot{E}, \tag{2.24}$$

*where* $\dot{E} = (\dot{\epsilon}_{1\cdot}, \dot{\epsilon}_{2\cdot}, \cdots, \dot{\epsilon}_{n\cdot})$ *and* $\dot{\epsilon}_{i\cdot} \sim_{\text{i.i.d}} \mathcal{N}(0, \Theta^*)$.

Intuitively, one can see from the formula in Proposition 2 that there is no reason to expect significant improvement of $W_2$ over $W_1$. The random parts of the two errors come from $\mathcal{N}(0, \Sigma^*)$ and $\mathcal{N}(0, \Theta^*)$ and in general either can be larger than the other, depending on specific $\Sigma^*$. For better illustration, we define two additional estimating equations:

$$W_3 I_p + \alpha \Lambda W_3 = \alpha \Lambda B^* + \dot{E} \tag{2.25}$$

$$W_4 \mathbf{diag}(\Theta^*) + \alpha \Lambda W_4 = \alpha \Lambda B^* + \dot{E} \tag{2.26}$$

Note equation (2.25) is almost identical to (2.23), except that the covariance of the random noises is now $\Theta^*$ instead of $\Sigma^*$. Which of $W_1$ and $W_3$ has a smaller norm is a random event and there is no clear winner between them when $||\Sigma^*||$ and $||\Theta^*||$ are in similar magnitudes, as assumed in (2.16). Thus we can say $W_1$ and $W_3$ are equivalent error matrices in distribution.

On the other hand, equation (2.26) corresponds to the situation when we carry $p$ separate Laplacian smoothing estimations but adjust $\alpha$ for each variable so that it is proportional to $1/\Theta^*_{jj}$. Intuitively, when the off-diagonals are of small magnitudes, the estimation $W_2$ should not be very different from $W_4$ and when the diagonals of $\Theta^*$ are in the similar magnitude as in Assumption 2.16, $W_3$ and $W_4$ should also be similar. The next theorem verifies this intuition under the assumption that $\Theta$ is diagonally dominant. As a result, we see that using $\Theta$ in estimation (2.22) does not bring real improvement in this situation and $W_1, W_2, W_3, W_4$ are essentially equivalent.

**Theorem 3.** *Under Assumption 3, assume* $W_2$*,* $W_3$*, and* $W_4$ *are the estimation errors from* (2.24)*,* (2.25) *and* (2.26) *respectively, with the same* $\alpha$*. If* $\Theta^*$ *is diagonally dominant with*

$\max_j \frac{\sum_{j' \neq j} |\Theta^*_{j'j}|}{\Theta^*_{jj}} \leq \rho < 1$, *then we have*

$$(1-\rho)\min(1, \min_j \Theta^*_{jj}) \leq \frac{||W_3||_\infty}{||W_2||_\infty} \leq (1+\rho)\max(1, \max_j \Theta^*_{jj}). \tag{2.27}$$

*In particular, under Assumption 2.16, we always have*

$$(1-\rho)\frac{1}{\bar{k}} \leq \frac{||W_3||_\infty}{||W_2||_\infty} \leq (1+\rho)\bar{k}$$

*for the constant* $\bar{k}$.

Theorem 3 assumes diagonal dominance of $\Theta^*$. Here we give a brief justification for this assumption. Given a general multivariate Gaussian $y \sim \mathcal{N}(0, \Sigma)$, it is known that the element-wise conditional distribution can be written in an element-wise regression form:

$$y_j = \sum_{j' \neq j} \zeta^j_{j'} y_{j'} + \xi_j$$

where $\zeta^j \in \mathbb{R}^p$ such that $\zeta^j_{j'} = -\frac{\Theta_{jj'}}{\Theta_{j'j'}}$ for $j' \neq j$ and $\zeta^j_j = 0$, and further $\xi_j$ is a Gaussian random variable with zero mean and variance as the conditional variance of $y_j | \{y_{j'}\}_{j' \neq j}$. Thus the diagonal dominance assumption of Theorem 3 is essentially assuming

$$\max_j ||\zeta^j||_1 = \max_j \sum_{j' \neq j} |\zeta^j_{j'}| < \rho < 1.$$

Specifically, the assumption is in the same form as the Assumption 4 of Meinshausen and Bühlmann (2006) when one uses the node-wise regression method to estimate the Gaussian graphical model. In that situation, $\rho < 1$ is actually an assumption needed for the node-wise regression method to consistently estimate the graph structure (see Proposition 4 of Meinshausen and Bühlmann (2006)). Though the glasso estimation does not directly rely on this assumption, it can still be expected that such assumption is not strong as long as good graph estimation performance is expected.

We conjecture that the phenomenon of the maximum penalized joint likelihood failing to provide further improvements over the two-stage estimation may be expected in a much wider class of problems. For instance, in a different problem of using sparse regression to adjust the Gaussian graphical model estimation, this phenomenon was observed in numerical results of Yin and Li (2013), though no explanation was given there. Here we provide an intuitive explanation of why we do not expect improvement by (2.22) even when the

diagonal dominance does not hold. The explanation is expected to be applicable in many other situations as well, such as the the sparse regression problem of Yin and Li (2013).

We hope to use $\Theta$ in estimating $M$ as information may be pooled across $p$ variables so that the estimation in each dimension can be improved. It is straightforward to see that in the Gaussian log-likelihood (2.1), the maximizer over $M$ is always $X$ which does not involve $\Theta$. On the other hand, the cohesion penalty is separable for the $p$ dimensions so that it does not help to pool information between the $p$ coordinates either. As a result, though the solution of the penalized likelihood problem after including the cohesion penalty does depend on $\Theta$, we cannot expect fundamental improvements over the simple estimation (2.2) since the objective does not effectively pool information between variables.

## 2.4 Simulation studies

In this section, we investigate the performance of the proposed method using several simulation examples. We first demonstrate the effectiveness of the proposed method by varying the sparsity and dimensionality of the underlying Gaussian graph. Then we compare the performance of the two-stage GNC-lasso with iterative GNC-lasso and standard glasso under both nontrivial cohesion and trivial cohesion (constant mean vector) settings over the observation network.

### 2.4.1 Performance under different Gaussian graphs

We first generate data from a model with nontrivial network cohesion. The observation network $\mathcal{G}^{(o)}$ in our simulation study is a lattice network with $n = 20 \times 20 = 400$ nodes. Each node corresponds to a random vector with dimension $p = 500$. For the $j$th variable, $j = 1, 2 \cdots, p$, $\mu^*_{\cdot j}$ is assumed to vary smoothly over the lattice, as shown in Figure A.1 of the Appendix. We constrain the range of $\mu^*_{\cdot j}$ to be between 0 and 1 and the resulting $\mu^*_{\cdot j}$'s are nontrivially cohesive. In the first example, we evaluate the performance of GNC-lasso by varying the underlying sparsity of $\Theta^*$. In particular, the Gaussian graph $\mathcal{G}^{(v)}$ is generated as an Erdos-Renyi graph on $p = 500$ nodes such that each node pair is connected independently with probability 0.005, 0.01, and 0.02 respectively. The Gaussian noise is then from $\mathcal{N}(0, \Sigma^*)$ where $\Theta^* = \Sigma^{*-1}$ is consistent with the Gaussian graph and we set $\Theta^* = A * 0.3 + (e + 0.1) * I$ where $A$ is the adjacency matrix of the variable graph, and $e$ is the absolute value of the minimum eigenvalue of $A * 0.3$. Such that the noises are comparable with $\mu^*_{\cdot j}$ in magintude. We evaluate the performance of the proposed method

in recovering the true underlying Gaussian graph, measured by the receiver operating characteristic (ROC) curve, along a graph estimation path by varying $\lambda$. An ROC curve depicts the tradeoff between *True Positive Rate* (TPR) and *False Positive Rate* (FPR), where

$$\text{TPR} = \frac{\#\{(j,j') : j \neq j', \Theta^*_{jj'} \neq 0, \hat{\Theta}_{jj'} \neq 0\}}{\#\{(j,j') : j \neq j', \Theta^*_{jj'} \neq 0\}}$$

and

$$\text{FPR} = \frac{\#\{(j,j') : j \neq j', \Theta^*_{jj'} = 0, \hat{\Theta}_{jj'} \neq 0\}}{\#\{(j,j') : j \neq j', \Theta^*_{jj'} = 0\}}.$$

In each setting, we repeat the data generation and model estimation independently 200 times. Figure 2.1 shows the ROC curves of the two-stage GNC-lasso for the three sparsity levels. As expected from Theorem 2, the graph selection performance of GNC-lasso increases as the true graph structure becomes sparser. When the sparsity level is 0.005, the GNC-lasso correctly recovers almost all true edges while only falsely setting 0.4% of the null-pairs to be edges. Even for the denser case of 0.02, it correctly detects more than 60% true edges when the FPR is 0.4%.

In the second example, we fix the sparsity level of the Gaussian graph to be 0.01 and evaluate the graph selection performance of GNC-lasso when $p = 200$, 500, and 800 respectively. The observation network and the corresponding $\mu^*_{\cdot j}$ values are still generated as in the previous example. The ROC curves are shown in Figure 2.2. Again, as expected from the theory, the performance degrades gradually as $p$ grows. When $p = 200$, GNC-lasso is able to detect almost all true edges with only 0.4% FPR. For the higher dimension setting when $p = 800$, it is also able to achieve more than 75% TPR when FPR is controlled at 0.4%.

### 2.4.2 Comparison with other methods under different cohesion settings

In this example, we compare several estimation methods under nontrivial cohesion. The methods we compare include: 1) the proposed method where $\alpha$ is tuned by 10-fold cross-validation ("two-stage GNC-lasso") ; 2) the proposed method where $\alpha$ is set to be the one that gives the best ROC curve ("optimal two-stage GNC-lasso"); 3) the jointly penalized likelihood method (2.5) using iterative optimization where $\alpha$ is also set to be the optimal ("optimal iterative GNC-lasso"); 4) the glasso estimate using the algorithm of Friedman et al. (2008) ("glasso"). Note that the optimal $\alpha$ is unknown in practice, but the performance under the optimal $\alpha$ provides a benchmark to evaluate the effectiveness of tuning $\alpha$ by

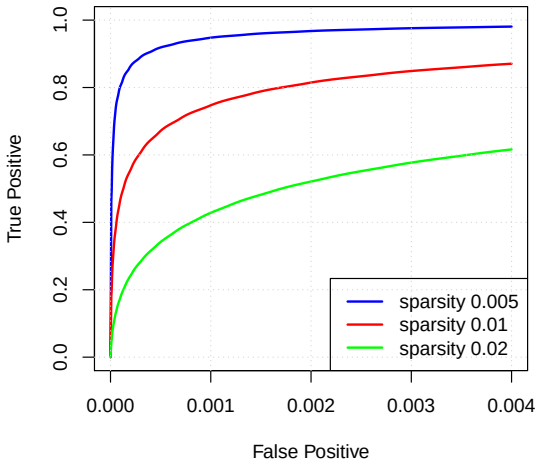



Figure 2.1: $n = 400, p = 500$, varying sparsisty, 200 replications

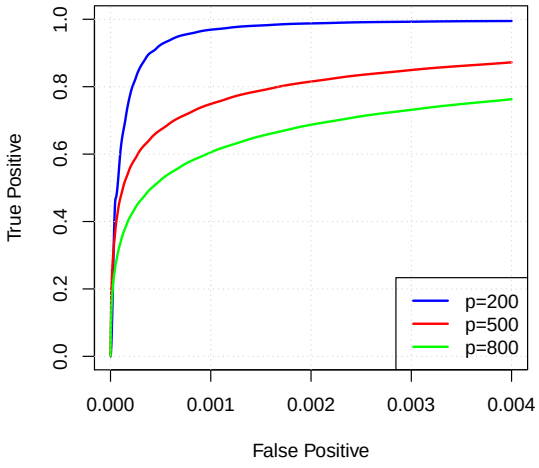



Figure 2.2: $n = 400$, sparsity is 0.01, varying $p$, 200 replications

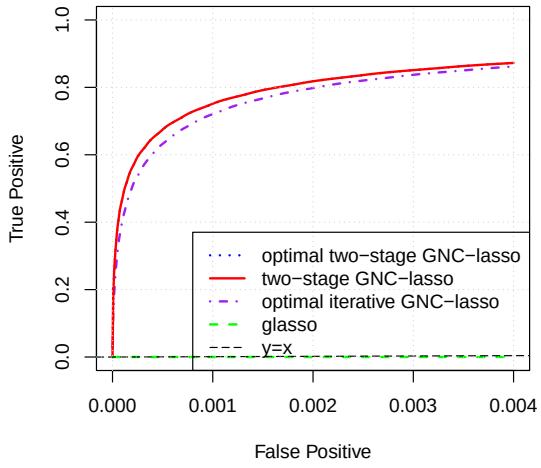



Figure 2.3: Nontrivial cohesion in mean, $n = 400$, $p = 500$

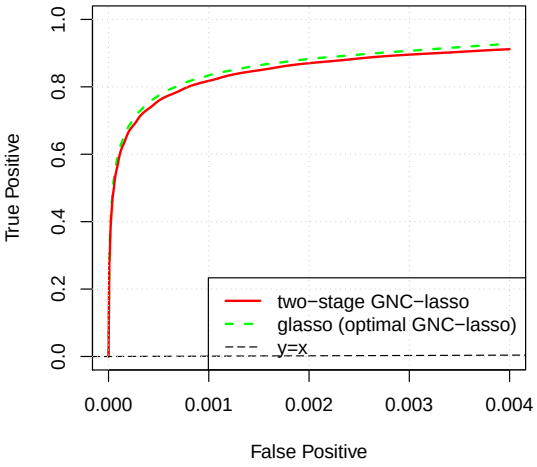



Figure 2.4: Trivial cohesion in mean (constant), $n = 400$, $p = 500$.

cross-validation. Moreover, iterative GNC-lasso is computationally expensive, especially when tuned by cross-validattion. Therefore we only compare with its optimal version.

Figure 2.3 shows the ROC curves of the four methods obtained from 200 independent replications. It can be seen that glasso completely fails in such setting with an ROC curve almost identical to the straight line $y = x$, i.e. the ROC curve for random guessing. This is expected as in the nontrivial cohesion setting, the mean vectors are quite different from be-

ing constant over the network. On the other hand, the three variants of GNC-lasso achieve reasonable model selection performances, by correctly recovering more than 80% of the true edges while only falsely setting 0.2% of the null-pairs to be edges. The cross-validated GNC-lasso is only negligibly worse than its oracle version, indicating that cross-validation is effective at selecting a good value of $\alpha$. Moreover, the performance of the two-stage GNC-lasso and the iterative GNC-lasso are similar, with the iterative GNC-lasso being slightly inferior. This agrees with what we have observed in general and can be explained by our theory, i.e. there is no significant difference between the two - either can be slightly better than the other by chance.

Next, we evaluate the performance of the proposed method under the trivial cohesion setting, i.e. when the i.i.d model assumption is correct. Note this is the setting where the standard glasso is expected to perform the best. We still use the same Gaussian model as before, except for setting $\mu_{ij} = 0$ for all pairs of $(i, j)$, thus the data we observe are i.i.d. samples from $\mathcal{N}(0, \Sigma^*)$. It is clear that the optimal version of GNC-lasso is just the glasso estimate, by letting $\alpha \to \infty$. Figure 2.4 shows the ROC curves of GNC-lasso where $\alpha$ is tuned by cross-validation and the standard glasso. As one can see, glasso is effective at recovering the graph structure in this setting as it now assumes the correct model. Further, the cross-validated GNC-lasso remains competitive when comparing with its optimal version (glasso).

## 2.5 Data example: learning associations between statistical terms

In this section, we apply the proposed method to a statisticians network data based on bibliography from four statistical journals collected by Ji and Jin (2014). In this data set, the author-to-paper bipartite graph, as well as the titles of the published papers are available. We demonstrate the proposed method by learning partial correlations between statistical terms that have appeared in paper titles and we treat the coauthorship network as the observation network.

In data pre-processing, we remove all authors who have only one paper in the data set and filter out common stopping words as well as terms that have appeared in fewer than 10 papers. For each author, we then calculate his/her average term frequency across all papers for which he/she is a coauthor. The coauthorship network is constructed by checking whether or not two authors have coauthored at least one paper, and we focus on the largest

connected component of the network. Finally, to focus on more informative terms, we sort the terms according to their term frequency-inverse document frequency score (tf-idf), one of the most commonly used approach in natural language processing to measure how informative a term is (Leskovec et al., 2014). We keep the top 300 terms with the highest tf-idf scores. The final data set we use has $n = 635$ authors and $p = 300$ terms. Each observation is a 300-dimensional vector showing the average frequency of term usage for a specific author. The coauthorship network is shown in Figure 2.5.

The interpretation of the proposed method is natural in this setting. Treating each author as an observation, the mean vector of the Gaussian distribution corresponds to the author-specific term usage habit. If two authors have collaborations, their writing habits are potentially similar due to common research interests and personal interactions. Given the term usage habit of each author, the observed average term frequency deviates randomly from the person's habit and the correlation between the deviations of different terms is depicted by a graph $\mathcal{G}^{(v)}$ due to connections between statistical concepts, assumed to be common across all authors. On the other hand, the standard Gaussian graphical model assumes all the authors share the same term preference and the deviation of observation from this term preference is due to common statistical concepts, which is to be learned. Intuitively, the proposed model interpretation is potentially more interpretable and flexible and offers more information.

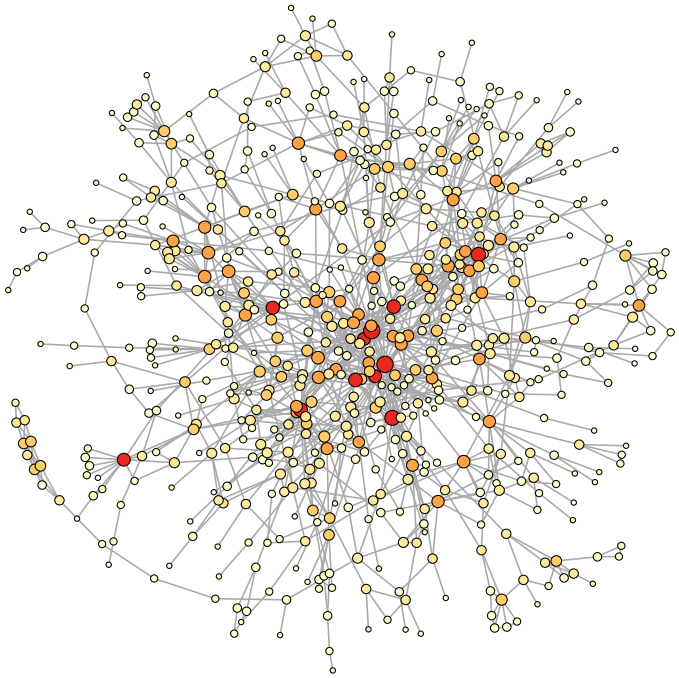


Figure 2.5: The coauthorship network of 635 statisticians based on four statistical journals. Both the size and the color of each node indicate the degree of the node (number of connections), with larger and darker nodes being statisticians with more coauthors in the network.

Again, we select $\alpha$ using 10-fold cross-validation. The cross-validation on $\lambda$ for GNC-lasso and glasso recovers 4 and 6 edges respectively, which are a little sparse. As discussed in previous sections, cross-validation may not necessarily provide easily interpretable results and in this setting, the cross-validated Gaussian graphs for both methods are too sparse to interpret. Instead, we apply both methods to obtain 25 edges in the final Gaussian graphs, which is a number that gives interpretable results for both methods. Figure 2.6 and Figure 2.7 shows the two estimated graphs of statistical terms. For visualization, we only plot the terms that have connections in at least one of the two graphs. This results in 47 terms in the figure.

Overall, most of the estimated edges represent valid concepts, such as "Markov chain Monte Carlo", "exponential families", "measurement error", "least absolute (deviation)". Regarding reasonable discoveries, "high dimensional", "gene expression", "covariance matrices", "partially linear model", "maximum likelihood", "confidence bands" and "bivariate associate" are discovered in the GNC-lasso graph but are missed in the glasso

graph, while the concept of "moving average" is discovered by glasso but missed by GNC-lasso. While glasso connects "false-discover-control", GNC-lasso misses the edge between "false-discovery" and "control", both of which can be reasonable. On the other hand, regarding potential false discoveries, the 4-connected component of "orthogonal-construction-computer-experiment" in the glasso graph may not correspond to any well-established concepts in statistics, though combing the terms together can be meaningful. The edge between "moving average" and "least absolute" is also questionable. In the GNC-lasso graph, the connections of "monotone-count" and "alternative-composite" are likely to be false.

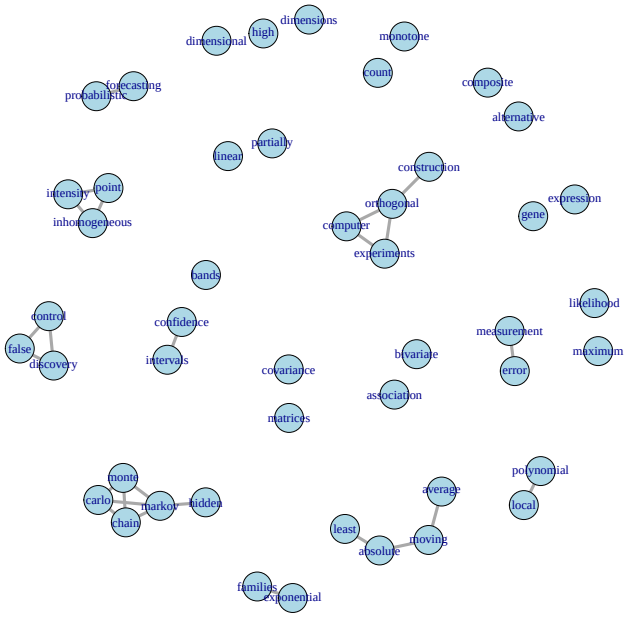



Figure 2.6: Partial correlation graphs estimated using Glasso

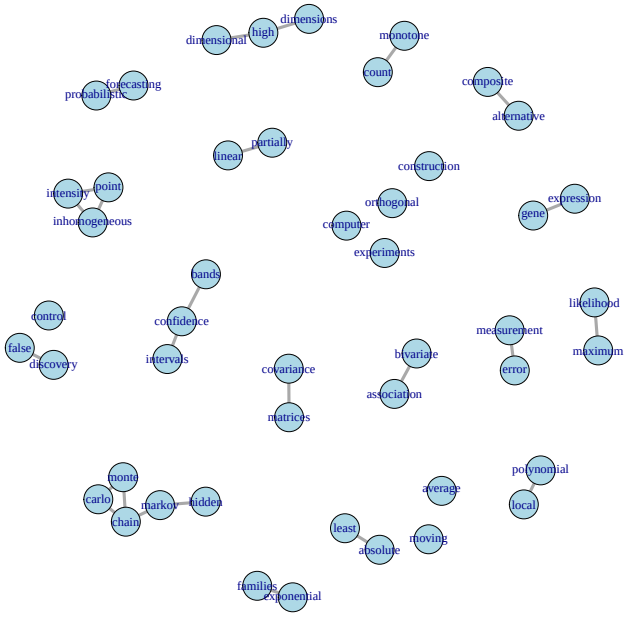



Figure 2.7: Partial correlation graphs estimated using GNC-lasso

## 2.6 Conclusion

In this chapter, we extend the standard graphical lasso problem and the corresponding estimation to the more general setting in which each individual observation has its own mean vector while all observations share the same covariance matrix under the assumption of network cohesion. Both the problem and the cohesion assumption are well motivated by many real world examples. Though the model involves more parameters than the number of observations, estimation can still be done with the help of network cohesion. The method is computationally efficient with theoretical guarantees to precisely estimate the inverse covariance matrix and the graph structure. In addition, we show that the estimation cannot be improved by using the maximum penalized joint likelihood estimator, though such estimator may be preferred at first look. The effectiveness of our proposal is demonstrated in both simulation studies and a real world application.

# CHAPTER 3

# A Two-Step Approach for Estimating Directed Acyclic Graphs

## 3.1 Introduction

Graphical model is one of the most popular tools for representing the probabilistic structure of the underlying random variables. Using nodes to denote the variables and edges to denote their conditional dependence, graphical models provide a way for analyzing and visualizing interactions between these variables. In particular, directed graphs can be used to encode causal relations among variables. In this chapter, we focus our discussion on the directed acyclic graph (DAG), that is, all edges are directed and there is no cycle in the graph. The interpretation of DAG is based on the directed Markov property (Lauritzen, 1996), and it has been widely used in a large variety of applications, such as, genetic networks (Hughes et al., 2000), social networks (Friedman et al., 2008) and time series analysis (Aalen et al., 2012).

Reconstructing DAG from observational data is a computationally NP-hard problem, and the number of candidate DAGs grows super-exponentially with the number of nodes. There has been a number of methods proposed to estimate DAGs with relatively small number of nodes, which are essentially based on searching through all possible graphs in the space. Two examples of such kind of methods are the max-min hill climbing algorithm (Tsamardinos et al., 2006) and the Peter-Clark (PC) algorithm (Spirtes et al., 2000). If given a natural ordering of the variables, the estimation of DAGs reduces to the estimation of their skeletons, and two important such methods include the implementation of the PC-algorithm, proposed by Kalisch and Buhlmann (2007), and penalized likelihood approachs, proposed by Shojaie and Michailidis (2010) and Van de Geer et al. (2013). Note that both methods assume a high-dimensional sparse setting. In particular, for multivariate Gaussian random variables, the estimation problem resembles the estimation of the inverse covariance matrix, or the precision matrix.

How to reduce the computational cost for estimating DAG poses a challenging problem, and an effective method which leads to such reduction will facilitate the analysis of large-scale networks. In this chapter, we propose a two-step approach of estimating DAGs, which will expedite the procedure of reconstructing DAGs from data. Specifically, we introduce the following two-step approach, which involves an initial screening step and a reconstruction step:

- Step 1: Screening. In this step, we develope a method for estimating the inverse covariance matrix corresponding to the DAG, which provides us a reduced space for possibly existing edges in the DAG.
- Step 2: Reconstruction. Based on the result in step 1, we reconstruct the DAG on a reduced parameter space, which is of a much lower dimension compared with the original one. Consequently the reconstruction of the DAG is expected to be much faster on this reduced parameter space, and due to the bias-variance tradeoff, the estimation accuracy is also possibly better for the two-step estimate.

The rest of the chapter is organized as follows: in Section 2, we propose the two-step method in greater details together with theoretical justifications; in Sections 3 and 4, we evaluate the performance of the proposed method using simulation studies and a real-world S&P500 stock data; Section 5 summarizes the chapter.

## 3.2 The proposed two-step methodology

In this section, we introduce the model set-up for DAG, and describe in detail how our two-step method is applied to reconstruct the DAG.

Consider a Gaussian DAG, which can be represented using the following linear structural equation model (SEM) (Peters and Bühlmann, 2012):

$$X_j = \sum_{k \in \text{pa}(j)} A_{jk} X_k + Z_j; \qquad j = 1, 2, \cdots, p, \tag{3.1}$$

where $\text{pa}(j)$ denotes the parental nodes of node $X_j$, $Z_j$ is the noise term and independent of $\{X_k : k \in \text{pa}(j)\}$. We additionally assume that $Z_1, \ldots, Z_p$ are independent Gaussian with equal variance, that is, $Z_j \sim \mathcal{N}(0, \sigma^2)$. Note that the equal variance assumption on $Z_j$'s enables identifiability of the graph (Peters and Bühlmann, 2012). With the notation $\boldsymbol{X} = (X_1, \cdots, X_p)^T$, the model can be equivalently written in the following matrix form:

$$\boldsymbol{X}_{p\times 1} = A_{p\times p}\boldsymbol{X}_{p\times 1} + \boldsymbol{Z}, \quad \Leftrightarrow \quad \boldsymbol{X} = (I - A)^{-1}\boldsymbol{Z} \equiv \Lambda \boldsymbol{Z},$$

with $A \in \mathbb{R}^{p\times p}$ contains the adjacency information, and $\Lambda = (I - A)^{-1}$. Here, we impose several constraints on $A$, including 1) the corresponding graph is sparse (i.e., $A$ being sparse), 2) the graph does not contain self-loops (i.e., $\text{diag}(A) = 0$), and 3) the graph is acyclic.

Further, note that since $\boldsymbol{X} = (I-A)^{-1}\boldsymbol{Z}$, then we have $\mathsf{Cov}(\boldsymbol{X}, \boldsymbol{X}) = (I-A)^{-1}D(I-A)^{-T}$, where $D = \text{diag}(\sigma^2, \ldots, \sigma^2)$. Consequently, the inverse covariance matrix of $\boldsymbol{X}$ (denoted by $\Theta$) is given by

$$\Theta = (I - A)^T D^{-1}(I - A). \tag{3.2}$$

Note that $\Theta$ encodes the moral graph corresponding to the DAG encoded by $A$. For a directed acyclic graph, its corresponding moral graph is obtained by adding edges between all pairs of nodes that have a common child, then removing the direction of all edges. In other words, if we only focus on the skeleton of a DAG, its edge set is a subset of its moralized graph counterpart. With the DAG represented in the linear SEM form, its moral graph is encoded by $\Theta$, corresponding to the inverse covariance matrix of $\boldsymbol{X}$. By estimating the inverse covariance matrix first then restricting the DAG estimation on the space governed by the inverse covariance estimate, we effectively reduce the estimation space of the directed edges, from which we attain computational gain.

### 3.2.1 Some useful theoretical results

Our primary interest is to estimate $A$ using iid observations of $\boldsymbol{X}$ arranging in the rows of $X \in \mathbb{R}^{n\times p}$. Throughout this chapter, we use $X = [x_{ij}]_{1\le i\le n, 1\le j\le p}$ to denote the data matrix, and bold $\boldsymbol{X}$ to denote the underlying random vector.

In this subsection, we introduce assumptions and describe some theoretical results that justify the validity of the proposed two-step method.

**Sparsity Assumption (A1).** Let $s_i$ be the number of nonzero elements in the $i$th row of $A$ and $s_j^*$ be the number of nonzero elements in the $j$th column of $A$, then $s = O(1)$ where $s$ is defined as:

$$s := \max\{s_1, \cdots, s_p, s_1^*, \cdots, s_p^*\}$$

This assumption implies that none of the nodes has many children, nor do we allow the case where a node has many parents.

Denote the inverse covariance matrix of $\boldsymbol{X}$ by $\Theta$. The following proposition then ensures that under the sparsity assumption (A1), given a sparse $A$, its corresponding moral graph, which is reflected by $\Theta$, is also sparse.

**Proposition 3.** *Suppose the adjancency matrix $A \in \mathbb{R}^{p\times p}$ for the DAG satisfies sparsity assumption (A1), then its corresponding moral graph, denoted by $\Theta$ is also sparse, with $s_\Theta \sim O(p)$, where $s_\Theta$ is the total number of nonzero off-diagonal entries in $\Theta$.*

To prove Proposition 3, we need the following Lemma 3.2.1.

**Lemma 3.2.1.** *Suppose $M \in \mathbb{R}^{p\times p}$ satisfies the sparsity assumption (A1), then $\Omega \triangleq MM^T$ is sparse, that is, $s_\Omega \sim O(p)$.*

*Proof of Lemma 3.2.1.* First we note

$$\Omega_{ij} = \sum_{k=1}^{p} M_{ik}M_{kj}^T = \sum_{k=1}^{p} M_{ij}M_{jk}.$$

The total number of nonzero elements in $\Omega$ is thus given by

$$\begin{aligned}
\sum_i \sum_j \mathbf{1}(\Omega_{ij} \neq 0) &= \sum_i \sum_j \mathbf{1}(\sum_k M_{ik}M_{jk} \neq 0) \\
&\leq \sum_i \sum_j \sum_k \mathbf{1}(M_{ik} \neq 0)\mathbf{1}(M_{jk} \neq 0) \\
&= \sum_k \left\{\textstyle\sum_i \mathbf{1}(M_{ik} \neq 0)\right\}\left\{\textstyle\sum_j \mathbf{1}(M_{jk} \neq 0)\right\} \\
&\leq p \cdot s \cdot s \sim O(p)
\end{aligned}$$

□

With Lemma 3.2.1, we can prove Proposition 3 as follows.

*Proof of Proposition 3.* By (3.2), $\Theta$ can be written as:

$$\Theta = (I - A)^T D^{-1}(I - A), \tag{3.3}$$

where $D$ is a diagonal matrix. Without loss of generality, we can assume $D = I$, the identity matrix. Now given $A$ that satisfies the sparsity assumption (A1), it immediately follows that $I - A$ also satisfies the sparsity assumption, and so does $(I - A)^T$. Substituting $M$ in Lemma 3.2.1 by $(I - A)^T$ and applying the lemma, it directly follows that for $\Theta$ defined as in (3.2), $s_\Theta \sim O(p)$, which means $\Theta$ is sparse. □

Next we introduce the faithfulness assumption (A2) and show that under the faithfulness assumption, if node pair $(i, j)$ are not linked in the moral graph encoded by $\Theta$, then the corresponding pair $(i, j)$ are not linked in the adjancency matrix $A$ of the DAG neither.

**Faithfulness Assumption (A2).** Consider the Gaussian DAG given in (3.1). We say the DAG satisfies the *faithfulness assumption* if the following equality holds for all $i < j$:

$$-\sigma_j^{-2} A_{ij} + \sum_{k>j} \sigma_k^{-2} A_{ik} A_{jk} = 0 \tag{3.4}$$

if and only if $A_{ij} = 0$ and $A_{ik}A_{jk} = 0$ for all $k > j$.

Note that if the nonzero entries in $A$ are independently sampled continuous random variables, the assumption holds for all choices of $A$ almost surely. Further, it is not difficult to see that under the faithfulness assumption (A2), if $\Theta_{ij} = 0$, then $A_{ij} = 0$ as well for $i, j = 1, 2, \cdots, p; i \neq j$ (Loh and Buhlmann, 2014, Spirtes et al., 2000), where $\Theta$ is the inverse covariance matrix corresponding to the DAG in (3.1), whose adjacency matrix is $A$.

This result is a population level statement, which guarantees that the true (undirected) edge set $E_\Theta \equiv \{(i,j) : \Theta_{ij} \neq 0, i \neq j\}$ is at least a superset of the true edge set of $A$, defined as $E_A \equiv \{(i,j) : A_{ij} \neq 0, i \neq j\}$. This implies that if we knew $E_\Theta$, we could simply restrict the searching of $E_A$ within $E_\Theta$, and if the size of $E_\Theta$ is small or $\Theta$ is sparse (implied by Proposition 3), the parameter space for estimating $A$ can be signficantly reduced. Of course, in practice, we do not know $\Theta$ (or more precisely $E_\Theta$), but we can estimate it using many of the graph learning algorithms that have been developed for estimating the nonzero entries of the inverse covariance matrix. Therefore, we propose a two-step approach for estimating $A$. In the first step, we apply an undirected graph learning algorithm to estimate $E_\Theta$, and in the second step, we apply a directed graph learning algorithm to estimate $E_A$ (or $A$) but with the nonzero entries restricted within $\widehat{E}_\Theta$. If the size of $\widehat{E}_\Theta$ is small, this approach will obviously reduce the computational cost. Further, since the nonzero parameter space is also reduced, due to the bias-variance tradeoff, the statistical estimation accuracy may also be improved, even though in practice, $\widehat{E}_\Theta$ may not be a superset of $E_A$. Specifically, using the symmetric differece notation, we have

$$\Delta \equiv E_A \Delta \widehat{E}_\Theta = \underbrace{(E_A \backslash \widehat{E}_\Theta)}_{\Delta_1} \cup \underbrace{(\widehat{E}_\Theta \backslash E_A)}_{\Delta_2} \equiv \Delta_1 \cup \Delta_2. \tag{3.5}$$

If $\widehat{E}_\Theta = E_\Theta$, $\Delta_1 = \emptyset$, but in practice, $\widehat{E}_\Theta$ may not be the same as $E_\Theta$, and $\Delta_1$ may not be empty, which implies that we will not be able to identify the edges in $\Delta_1$ in the second step of our method. Nevertheless, since the parameter space is reduced in the second step of our method, due to the bias-variance tradeoff, statistical estimation accuracy may still be improved, as long as the size of $\Delta_1$ is not large.

### 3.2.2 Step 1: estimating the moral graph

Given the data matrix $X \in \mathbb{R}^{n\times p}$ (assuming it is centered), we first obtain an estimate of the inverse covariance matrix corresponding to $\boldsymbol{X}$, denoted by $\Theta$, which will be used to provide $\widehat{E}_\Theta$. As noted above, $\Theta$ reflects the moral graph corresponding to the DAG, whose information is encoded by $A$.

Note given the model in (3.1), the inverse covariance matrix $\Theta$ corresponding to $\boldsymbol{X}$ can be written as follows:

$$\Theta = \Lambda^{-T} D^{-1} \Lambda^{-1} = (I - A)^T D^{-1} (I - A), \quad \text{where } D = \text{diag}(\sigma_1^2, \cdots, \sigma_p^2). \tag{3.6}$$

Note based on the discussion in the previous subsection, if the sparsity assumption (A1) holds, $\Theta$ is also sparse. Then to obtain an estimate $\widehat{\Theta}$ for $\Theta$, we propose to estimate $\Theta$ in the following way. First, we use graphical lasso (Friedman et al., 2008) combined with stability selection (Meinshausen and Buhlmann, 2010) to obtain a weight matrix $W$. Specifically, we apply graphical lasso with a sequence of tuning parameters $\{\lambda_k\}_{k=1}^K$ over $B$ bootstrapped samples of the original data $X$, denoted by $\{X^{(b)}\}_{b=1}^B$. For each bootstrapped sample $X^{(b)}$, given tuning parameter $\lambda_k$, we are able to obtain an estimate $\widehat{\Theta}^{k,b}$ for $\Theta$ via graphical lasso. Define the selection probability corresponding to $\lambda_k$, denoted by $\Pi^k$, as

$$\Pi^k \equiv \frac{1}{B} \sum_{b=1}^{B} \mathbb{I}(\widehat{\Theta}^{k,b} \neq 0),$$

where both the inequality and indicator function are taken entry-wise. Note that with the definition in Meinshausen and Buhlmann (2010), $\Pi^k, k \in \{1, \ldots, K\}$ forms the stability path of each edge, and each entry in $\Pi^k$ can be roughly interpreted as the probability of an edge being selected over $B$ bootstrapped samples. Further, let

$$W = [W_{ij}] \equiv \max_{k=1,2,\cdots,K} \Pi^k,$$

where the maximum is taken over the entire stability path and the operation is also performed entry-wise. Each entry in $W$ encodes the maximum selection probability over the corresponding stability path, and can be viewed as an empirical measure of the "existing probability" for each edge. This facilitates us to assign different penalties on different entries of $\Theta$ when using the graphical lasso to estimate $\Theta$, which is also expected to yield more accurate estimates. Specifically, $\widehat{\Theta}$ is obtained by solving the following optimization

problem:

$$\widehat{\Theta} = \underset{\Theta \in \mathbb{S}_+^p}{\operatorname{argmin}} \left\{ \log\det\Theta - \operatorname{tr}(S\Theta) + \rho \|(1-W) * \Theta\|_{1,\text{off}} \right\}, \tag{3.7}$$

where $S = X^T X/n$, the sample covariance matrix of $X$, $*$ denotes the entry-wise product between matracies, and $\rho$ is a tuning parameter.

Note that the weighted graphical lasso penalizes more on edges that have small values of $W_{ij}$ and penalizes less on edges that have large values of $W_{ij}$. Since stability selection avoids the complication of choosing a proper tuning parameter and provides a good sense for which edges are likely to exist and which are not, we expect the weighted graphical lasso to perform better than the standard graphical lasso. Further, note if an edge is removed in the first step, it will never be recovered back in the second step, thus if one is concerned about selecting most true edges (i.e. sensitivity), a small value of the tuning parameter $\rho$ in (3.7) would be preferred. On the other hand, a small value of $\rho$ tends to result in a larger set of $\widehat{E}_\Theta$, which implies that the parameter space in the second step will be larger, and then both the variance of the estimate in the second step and the computational cost will also be larger. Thus, there are considerations in both the bias-variance tradeoff and the computational cost when selecting $\rho$.

### 3.2.3 Step 2: reconstructing the DAG on the restricted space

In this step, we reconstruct the DAG, or equivalently estimate $A$, using the estimated $\widehat{\Theta}$ from step 1. Specifically, let

$$\mathcal{M}_{\widehat{\Theta}} = \{M \in \mathbb{R}^{p\times p} : M_{ij} = 0 \text{ if } \widehat{\Theta}_{ij} = 0\}, \tag{3.8}$$

i.e. the collection of $p \times p$ matrices with certain entries being 0, and the reconstruction of $A$ will be restricted to the reduced space $\mathcal{M}_{\widehat{\Theta}}$. Further, we let

$$\ell(A) = \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \left( x_{ij} - \sum_{k:k\neq j} x_{ik} A_{jk} \right)^2,$$

which is the negative log-likelihood up to a constant, and we consider to estimate $A$ using the following criterion:

$$\begin{aligned} \min_{A} \quad & \ell(A) && (3.9)\\ \text{subject to} \quad & \sum_{i\neq j} |A_{ij}| \leq K, && (3.10)\\ & \sum_{j_1=j_L:1\leq k\leq L} \mathbb{I}(A_{j_{k-1},j_k} \neq 0) \leq L-1, && (3.11) \end{aligned}$$

for any $\{j_1, \cdots, j_L\} \subseteq \{1, 2, \cdots, p\}$. Note the first constraint corresponds to the sparsity assumption on $A$, while the second one guarantees that the estimated $\widehat{A}$ satisfies the acyclic property of DAG. However the second constraint in fact includes a total number of $O(p^p)$ constraints, which is super-exponential in $p$ and renders the optimization computationally infeasible. It turns out that by introducing an intermediate dual variable matrix, the number of constraints can be reduced to $p^3 - p^2$ (Yuan et al., 2014), and the criterion (3.9)-(3.11) can be written in an equivalent form

$$\begin{aligned} \min_{A,\lambda} \quad & \ell(A) && \\ \text{subject to} \quad & \sum_{i\neq j} |A_{ij}| \leq K, && (3.12)\\ & \lambda_{ik} + \mathbb{I}(j \neq k) - \lambda_{jk} \geq \mathbb{I}(A_{ij} \neq 0), \quad i, j, k = 1, \ldots, p, i \neq j. && (3.13) \end{aligned}$$

Note that now the acyclic property of $A$ is satisfied as long as the set of constraints in (3.13) are satisfied.

Further, to ease the optimization, we replace the indicator function $\mathbb{I}(A_{ij} \neq 0)$ with the truncated $\ell_1$ function $J_\tau(A_{ij}) = \min(|A_{ij}|/\tau, 1)$ (Shen et al., 2012), where $\tau$ is a tuning parameter and $J_\tau(x)$ approximates the indicator function as $\tau \to 0^+$. Then following the derivations in Yuan et al. (2014) and using techniques in ADMM, we obtain the augmented Lagurangian for the ADMM update:

$$\begin{aligned} L_\rho(A, B, U, \lambda, \xi, y) \quad = \quad & \ell(A) + \mu\|B\|_1 + \frac{\rho}{2}\|A - B + U\|_F^2 \\ & + \frac{\rho}{2}\sum_k \sum_{i\neq j} (|B_{ij}|w_{ij} + \tau(1 - w_{ij}) + \xi_{ijk} \\ & - \tau\lambda_{ik} - \tau\mathbb{I}(j \neq k) + \tau\lambda_{jk} + y_{ijk})^2, \end{aligned} \tag{3.14}$$

where $y = \{y_{ijk}\}_{p\times p\times p}$ is a scaled dual variable tensor, and $U = \{u_{ij}\}_{p\times p}$ is a scaled dual variable matrix.

Then optimizing (3.14) can be solved by iteratively minimizing each of the six blocks $(A, B, \lambda, \xi, y, U)$ sequentially while holding others fixed until convergence. Specifically, at iteration $s+1$, we have the following update rules:

$$
\begin{aligned}
A^{(s+1)} &= \operatorname{argmin}_{A \in \mathcal{M}_{\widehat{\Theta}}} L_\rho(A, B^{(s)}, \lambda^{(s)}, \xi^{(s)}, y^{(s)}, U^{(s)}) && (3.15)\\
B^{(s+1)} &= \operatorname{argmin}_{B \in \mathcal{M}_{\widehat{\Theta}}} L_\rho(A^{(s+1)}, B, \lambda^{(s)}, \xi^{(s)}, y^{(s)}, U^{(s)}) && (3.16)\\
\lambda^{(s+1)} &= \operatorname{argmin}_{\lambda} L_\rho(A^{(s+1)}, B^{(s+1)}, \lambda, \xi^{(s)}, y^{(s)}, U^{(s)}) && (3.17)\\
\xi^{(s+1)} &= \operatorname{argmin}_{\xi \succeq 0} L_\rho(A^{(s+1)}, B^{(s+1)}, \lambda^{(s+1)}, \xi, y^{(s)}, U^{(s)}) && (3.18)\\
y_{ijk}^{(s+1)} &= y_{ijk}^{(s)} + (|B_{ij}^{(s+1)}| + \xi_{ijk}^{(s+1)} - \tau\lambda_{ik}^{(s+1)} - \tau\mathbb{I}(j \neq k) + \tau\lambda_{jk}^{(s+1)}) && (3.19)\\
U^{(s+1)} &= U^{(s)} + (A^{(s+1)} - B^{(s+1)}) && (3.20)
\end{aligned}
$$

Note the updates in $\xi$, $y$ and $U$ are straightforward. The updates in $A$ and $B$ are now carried out in a much smaller parameter space (compared with the entire collection of $p \times p$ matrices), and thus the computational cost is much reduced. Specifically, let $I_j$ denote the potential nonzero positions for the $j$th row of A, then $A_{I_j}$ is updated by solving the following minimization problem:

$$
\min_{A_{I_j}} \frac{1}{2} \sum_{i=1}^{n} (x_{ij} - \sum_{k \neq j, k \in I_j} x_{ik} A_{jk})^2 + \frac{\rho}{2} ||A_{I_j} - B_{I_j}^{(s)} + U_{I_j}^{(s)}||^2.
$$

Equivalently, $A_{I_j}$ is the solution for:

$$
(X_{I_j}^T X_{I_j} + \rho I) A_{I_j} = X_{I_j}^T X_j + \rho(B_{I_j}^{(s)} - U_{I_j}^{(s)}).
$$

Similarly for $B$. Note that at each iteration, $A^{(s+1)}$ is obtained by rows, that is, $A^{(s+1)}$ is obtained by solving $p$ separate equations. Therefore, it implies that mis-specification in $I_j$ (due to errors in $\widehat{E}_\Theta$) will not affect much the updates in other rows of $A$.

The update in $\lambda$ is a little involved, and we provide more details.

**Algorithm for updating $\lambda$.** When updating $\lambda$ while holding the rest of the parameters fixed, we aim to minimize the following objective function (after omitting a constant scalar):

$$
f(\lambda) = \sum_k \sum_{i \neq j} \left(|B_{ij}| w_{ij} + \tau(1 - w_{ij}) + \xi_{ijk} - \tau\lambda_{ik} - \tau\mathbb{I}(j \neq k) + \tau\lambda_{jk} + y_{ijk}\right)^2.
$$

The first-order condition is then given by:

$$2\tau(Q\lambda + W) = 0,$$

where

$$Q = (2\tau)\begin{pmatrix} p-1 & -1 & \cdots & -1 \\ -1 & p-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & p-1 \end{pmatrix}, \tag{3.21}$$

and the $(i,k)$ entry of $W$ is given as follows, for $i,k = 1,\cdots,p$:

$$W_{ik} = \sum_{j\neq i}(|B_{ji}|w_{ji} - |B_{ij}|w_{ij}) - \tau\sum_{j\neq i}(w_{ji} - w_{ij}) + \sum_{j\neq i}(\xi_{jik} - \xi_{ijk}) + \sum_{j\neq i}(y_{jik} - y_{ijk})$$
$$- \tau\left(\sum_{j\neq i}\mathbb{I}(i\neq k) - \sum_{j\neq i}\mathbb{I}(j\neq k)\right).$$

If $Q$ were invertible, the minimizer for $f(\lambda)$ would be:

$$\lambda^* = -Q^{-1}W.$$

However, the $Q$ matrix given in (3.21) is of rank $p-1$ and thus not invertible. Hence minimizing $f(\lambda)$ w.r.t. $\lambda$ potentially has infinite number of optimizers. Note that if we use the generalized inverse of $Q$ for $Q^{-1}$, then during the updating iterations, the value of $\lambda$ may explode (observed in simulation studies).

Now among the infinitely many possible optimizers, we propose to use the one that has the smallest Frobenius norm. Specifically, we consider the following optimization problem when updating $\lambda$:

$$\min_{\lambda} \quad \widetilde{f}(\lambda) \equiv f(\lambda) + \rho\cdot\mathrm{tr}(\lambda^T\lambda), \qquad \rho > 0. \tag{3.22}$$

Denote the solution to (3.22) by $\widetilde{\lambda}(\rho)$, then the desired optimizer $\lambda^*$ is given by:

$$\lambda^* = \lim_{\rho\downarrow 0}\widetilde{\lambda}(\rho).$$

For completeness, we derive the solution for $\lambda^*$ in the following.

Note the first order condition for optimizing (3.22) is given by:

$$\nabla \widetilde{f}(\lambda) = (2\tau Q + 2\rho \mathrm{I})\lambda + (2\tau)W \overset{\text{set}}{=} 0,$$

i.e.,

$$\left(Q + \frac{\rho}{\tau}\mathrm{I}\right)\lambda = -W. \tag{3.23}$$

Given the special form of $Q$, we see that

$$Q = (2\tau)\left(p\mathrm{I} - ee^T\right), \text{ where } e = (1, \cdots, 1)^T \in \mathbb{R}^{p\times 1}.$$

Then the LHS of (3.23) is given by:

$$\left(2\tau p + \frac{\rho}{\tau}\right)\mathrm{I} - (2\tau)ee^T,$$

and its inverse is given by:

$$\begin{aligned}\left(2\tau p + \frac{\rho}{\tau}\right)^{-1}\left[\mathrm{I} - \frac{2\tau}{2\tau p + \rho/\tau}ee^T\right]^{-1} &= \left(2\tau p + \frac{\rho}{\tau}\right)^{-1}\left[\mathrm{I} + \frac{2\tau^2/(2\tau^2 p + \rho)}{1 + 2\tau^2/(2\tau^2 p + \rho)\|e\|^2}ee^T\right] \\ &= \frac{\tau}{2\tau^2 p + \rho}\left[\mathrm{I} + \frac{2\tau^2}{4\tau^2 p + \rho}ee^T\right],\end{aligned}$$

where the first equality comes from the following identity:

$$\left(\mathrm{I} + cVV^T\right)^{-1} = \mathrm{I} - \frac{c}{1 + c\|V\|^2}VV^T \qquad \text{for some vector } V.$$

Therefore, the minimizer of (3.22) is given by:

$$\widetilde{\lambda}(\rho) = -\frac{\tau}{2\tau^2 p + \rho}\left[\mathrm{I} + \frac{2\tau^2}{4\tau^2 p + \rho}ee^T\right]W.$$

Now let $\rho \to 0$, we have:

$$\lambda^* = -\frac{1}{2\tau p}\left[\mathrm{I} + \frac{1}{2p}ee^T\right]W.$$

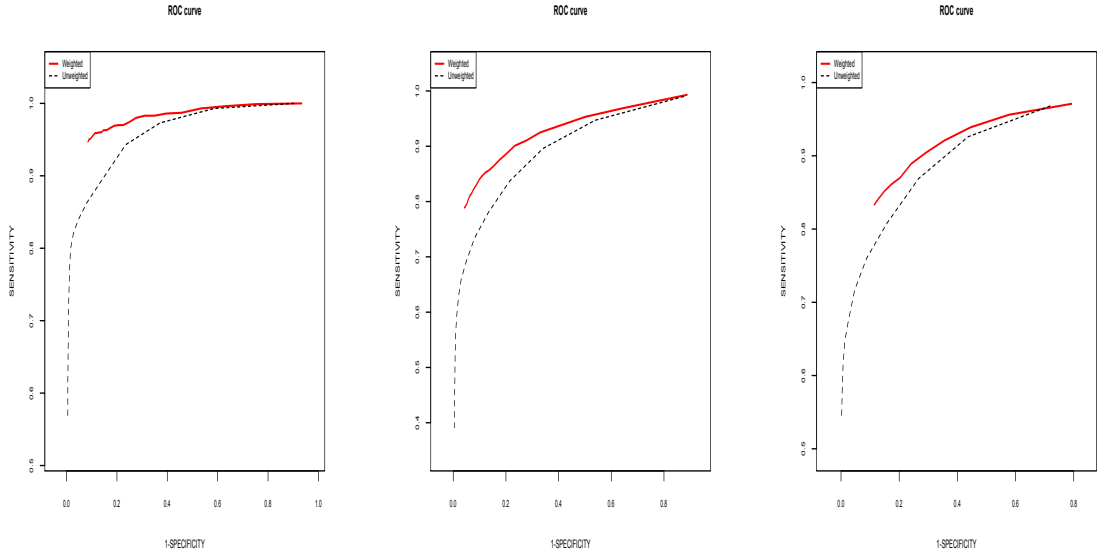
## 3.3 Simulation studies

In this section, we investigate the performance of the proposed method and compare it with the one-step approach, which optimizes (3.13) on the entire space of $p \times p$ matrices.

We consider DAGs with $p$ =30, 50, 100, and 200 nodes, and the edges come from an Erdos-Renyi random graph. Specifically, we generate the graph according to $P(A_{ij} \neq 0) =$

$0.02, i > j$. Values of the nonzero entries in $A$ are generated according to Unif$(-0.25, -1) \cup$ Unif$(0.25, 1)$. The data are then generated according to (3.1) with $\sigma^2 = 1$, and $n = 100$. All results are averages over 50 replications.

We first look at the results for step 1 of the proposed method. Figure 3.1 shows the ROC curves in identifying the edges in the moral graph of $A$, i.e. $\Theta$. As one can see, the weighted graphical lasso performs significantly better than the standard graphical lasso in recovering the moral graph structure. As expected, as $p$ increases, the performaces of both weighted graphical lasso and standard graphical lasso deteriorate. However, note that the graph structure in $A$ is our primary interest, rather than the moral graph. What we truly care about is whether edges in $A$ can be retained in $\widehat{\Theta}$ so that they remain in the parameter space in the second step of the proposed method and so that there is a chance to identify them in the second step. Specifically, we are concerned if $A_{ij} \neq 0$, then whether the corresponding $\widehat{\Theta}_{ij}$ is nonzero. The results are summarized in Figure 3.2, where the sensitivity is computed using $A$ as the baseline, rather than $\Theta$. Note that for the four settings ($p$ =30,50,100,200) we have considered, the ROC curves almost overlap, and maintain a sensitivity close to 1 for a wide range of specificity values. This is a desired result as it implies that though some of the edges in the moral graph may be missed by the weighted graphical lasso, most edges in $A$ are kept in $\widehat{\Theta}$ and they will be used to define the parameter space for the second step of the proposed method.




(a) $p = 30, n = 100$ (b) $p = 50, n = 100$ (c) $p = 100, n = 100$

Figure 3.1: ROC curves for $\widehat{\Theta}$ in identifying edges in the moral graph of $A$. Solid red curves correspond to the weighted graphical lasso, and the dashed black curves correspond to the standard graphical lasso.

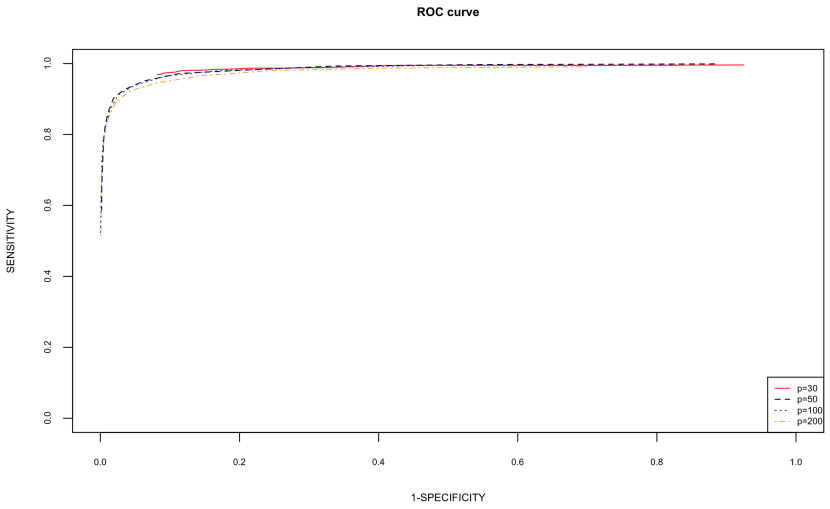



Figure 3.2: ROC curves for $\widehat{\Theta}$ when using $A$ as the baseline. $n = 100$, $p$ =30, 50, 100, 200.

Table 3.1 summarizes the final results in terms identifying edges in $A$ and also estimation accuracy for $A$. Again, we evaluate the graph identification performance using sensitivity and specificity while the estimation accuracy using the Frobenius norm. We

also recorded the computational time, which includes both step 1 and step 2 for the proposed method. As one can see, as expected, in terms of the computational cost, the two-step method is much faster than the one-step method, and the difference is more dramatic when $p$ increases. Further, the two-step method also performs better than the one-step method in terms of identifying the edges in $A$. This is probably due to the bias-variance trade-off, as the two-step method first reduces the size of the parameter space (without losing much bias) so that one only needs to search for the model within a much restricted parameter space in the second step (resulting in much reduced variance).

Table 3.1: Simulation results based on 50 replications

| $(n, p)$ | | Two-step | One-step |
|---|---|---|---|
| $(100, 30)$ | Specificity | 0.90(0.010) | 0.85(0.010) |
| | Sensitivity | 0.90(0.080) | 0.90(0.076) |
| | Frobenius loss | 1.71(0.091) | 1.67(0.087) |
| | Computational time | 80s | 205s |
| $(100, 50)$ | Specificity | 0.94(0.006) | 0.86 (0.005) |
| | Sensitivity | 0.95(0.0259) | 0.95(0.053) |
| | Frobenius loss | 2.48(0.082) | 2.49(0.062) |
| | Computational time | 350s | 610s |
| $(100, 100)$ | Specificity | 0.95(0.003) | 0.80(0.003) |
| | Sensitivity | 0.90(0.029) | 0.83(0.024) |
| | Frobenius loss | 5.43(0.072) | 5.56(0.057) |
| | Computational time | 3335s | 29205s |

## 3.4 Data example

In this section, we apply the proposed method to a stock return data and also compare the performance with the one-step method.

The data are collected from S&P500 via http://finance.yahoo.com, over the period of 2013-01-01 to 2014-12-31 with 458 consecutive trading days. We collect daily returns of 10 stocks over this period, specifically, YHOO (Yahoo), APPL (Apple), GOOGL (Google), IBM (IBM), QCOM (Qualcomm), T (AT&T), VZ(Verizon), GM (General Motors), LUV (Southwest Airlines), and AME (Ametek). In addition, we also collected the daily returns of the S&P500, so we can compute the daily excess return of each stock. The excess return of a financial asset is the return that exceeds a particular benchmark or index with similar

level of risk. In this case, we use the S&P500 index as the benchmark and the regression coefficient between long term stock return and long term S&P500 index return as the level of risk. Specifically, the excess return of stock $i$ at time $t$ can be calculated as follows:

$$exr_{i,t} = r_{i,t} - \beta_i \cdot r_{m,t}$$

where $r_{i,t}$ is the daily return of stock $i$ at time $t$, $r_{m,t}$ is the daily return of S&P500 at time $t$, and $\beta_i$ is the regression coefficient if we regress the long term return of stock $i$ on the long term S&P500 index return. Note the excess rate of return is essentially the return of a stock after removing the effect of the market.

We then apply both the one-step and two-step methods to the excess returns to investigate the "intrinsic" dependence structures of these stocks. The results are shown in Figure 3.3. As one can see, the two-step method identifies few edges than the one-step method, which is in accordance with the results in simulation studies, i.e. the two-step method tends to enjoy higher specificity than the one-step method in simulation studies. It can also be seen that edges identified by the two-step method are essentially connecting companies within the same sector, e.g. Yahoo and Google, Verizon and AT&T. The one-step method did not discover the pair of Verizon and AT&T, while at the same time, some of the connections identified by the one-step method are a little dubious, for example, Google and GM, Yahoo and GM, GM and Southeast Airlines. Both methods identified the pair of Southwest Airlines and Ametek; though both companies belong to the sector of public transportation, it is not clear whether they are closely related.

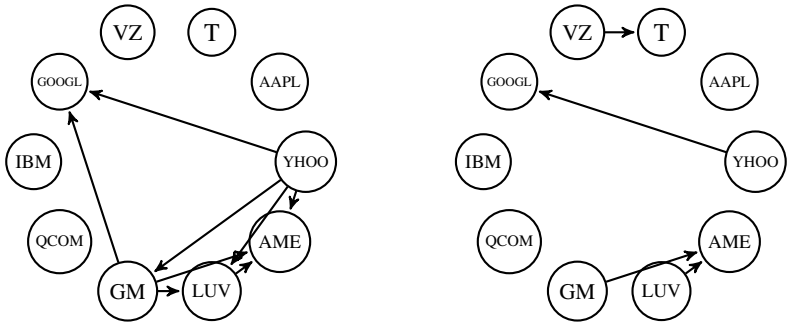



Figure 3.3: Estimated dependence structures among 10 stocks using the one-step (left) and two-step (right) methods

## 3.5 Summary

The Directed Acyclic Graph (DAG) is a commonly used tool to encode the causal relationship between random variables. Estimation of the DAG structure is often a challenging problem as the computational complexity scales exponentially in the graph size when the total ordering of the DAG is unknown. To reduce the computational cost, and also with the aim of improving the estimation accuracy via the bias-variance trade-off, we have proposed a two-step approach for estimating the DAG in this chapter, when data are generated from a linear structural equation model. In the first step, we infer the moral graph of the DAG via estimation of the inverse covariance matrix, which reduces the space that one would search for the DAG. In the second step, we apply a penalized likelihood method for estimating the DAG to the reduced space. Numerical results indicate that the proposed method compares favorably with the one-step method in terms of both computational cost and estimation accuracy.

# CHAPTER 4

# Estimating Cointegrated Vectors with Structured Sparsity

## 4.1 Introduction

Cointegration is a statistical tool for analyzing multiple time series data and it has been widely used in econometrics and macroeconomic analysis. The pioneering concept of cointegration was initially proposed in Granger (1981) with the specification and analysis focusing primarily in the spectral domain. The idea was later formalized by Engle and Granger (1987), who established the connection between the autoregressive and the error correction representations of the cointegrated system. The authors proposed to estimate the cointegrated vector with a two-stage estimator under the regression framework, with estimation and discussion focusing on the cointegration of two univariate series. Later on, Johansen (1988) considered to estimate the cointegrated vector in a multivariate setting through maximizing the profile likelihood function. Asymptotic properties of the estimators have also been discussed, as well as a procedure for testing the number of cointegrating vectors based on the likelihood ratio test statistic. Another test with a similar hypothesis yet based on the unit-root residuals was proposed by Phillips and Ouliaris (1990). All these work consider the cointegrated vector being dense, i.e. the cointegration series involve all coordinates of the original series.

Much more recently, Wilms and Croux (2016) considered the sparse cointegration, in which some of the entries in the cointegrating vectors are exactly zero, so that each cointegrated series is formulated by a linear combination of only a few coordinates of the original multivariate series. The authors showed that sparse cointegration leads to better forecasting performance, yet the cointegrated series themselves are never of primary concern.

In this chapter, we consider a similar setting in which the cointegrating vectors are sparse. However, instead of simply assuming the cointegrating vectors are sparse, we assume the cointegrating vectors are simultaneously group sparse and elementwise sparse,

which lays one of the major differences between our model setup and that in Wilms and Croux (2016).

Such a "mixed-sparsity" assumption is not arbitrary and has an important association with how the cointegrating vectors are obtained thus the formulation of the cointegrated series. It is well-known that if the original multivariate system is cointegrated, then the coefficient matrix has a low-rank representation and can be decomposed into two components, one of which encodes the cointegrating vectors and the other encodes the "speed" of cointegration. The decomposition, however, is not uniquely identifiable. With the sole assumption that the cointegrated vectors are elementwise sparse as in Wilms and Croux (2016), there may be some other sparse (or non-sparse) representations of the cointegrating vectors which yield similar or even superior forecasting performances. Consequently, it becomes uninformative to place any attention on the cointegrated series. The limitation can be partially addressed by the additional assumption that the cointegrated vectors are also group sparse. By properly placing the group-sparse assumption, we are able to obtain cointegrating vectors whose group-sparsity pattern is invariant to linear transformations, albeit the non-uniqueness in identifying the cointegrating vectors. In other words, we are able to eliminate an invariant set of the coordinates of the original series based on the obtained cointegrated series, no matter what the ultimate sparse cointegrating vectors are from the low-rank decomposition. We retain the elementwise sparsity assumption as in Wilms and Croux (2016) to encourage further sparse representation of the cointegrating vectors. For example, as in one of the data applications we will see later, suppose one is interested in constructing portfolios of stocks based on their cointegration relations, then from the perspective of reducing the transaction cost and improving the portfolio stability, it would be desirable to shrink the large basket of stocks to a smaller subset such that stocks that are marginally related are removed in the first place, and only a few stocks are involved for each cointegrated series.

The rest of the chapter is organized as follows. In Section 4.2, we present the formal problem formulation and introduce the method which leads to the cointegrated vectors with our desired sparsity pattern. In Section 4.3, we evaluate the performance of the proposed method using simulation studies, with a focus on the error of the estimated coefficient matrix and the recovery of the cointegrated space. In Section 4.4, we apply the proposed method to data examples, one being stock data and the other being treasury yield data. Section 4.5 summarizes the chapter.

## 4.2 Problem formulation and the proposed method

We start the presentation with a general $p$-dimensional vector autoregressive (VAR) process. Consider a multivariate process $X_t \in \mathbb{R}^p$ that evolves according to a VAR$(d)$ model, i.e.

$$X_t = \mu_t + \Phi_1 X_{t-1} + \Phi_2 X_{t-2} + \cdots + \Phi_d X_{t-d} + \epsilon_t, \tag{4.1}$$

where $\mu_t \in \mathbb{R}^p$ is the vector of intercept terms, $\Phi_1, \ldots, \Phi_d$ are $p \times p$ transition matrices, and $\epsilon_t \in \mathbb{R}^p$ is an innovation process satisfying

$$\mathbb{E}(\epsilon_t) = 0, \qquad \mathbb{E}(\epsilon_t \epsilon_t^T) = \Sigma_\epsilon, \qquad \mathbb{E}(\epsilon_t \epsilon_s^T) = 0 \ \text{ for } \ t \neq s.$$

The characteristic polynomial $\mathcal{A}(z)$ of $\{X_t\}$ is then defined as

$$\mathcal{A}(z) := I - \Phi_1 z - \cdots - \Phi_d z^d.$$

If the roots of $|\mathcal{A}(z)| = 0$ lie outside the unit circle, $\{X_t\}$ is unit-root stationary or an $I(0)$ process, and $\{X_t\}$ is called an $I(d)$ process if its $d$th-order difference process $\{\Delta^d X_t\}$ is unit-root stationary. Here we assume $\{X_t\}$ is at most an $I(1)$ process, that is, $\Delta X_t$ is stationary if $X_t$ is not. Specifically, we follow the definition in Lütkepohl (2005) and refer to $\{X_t\}$ as cointegrated if there are linear combinations given in the form of $\beta^T X_t$ that are $I(0)$, where $\beta \in \mathbb{R}^{p \times r}$, and the $r$ cointegrated series are given by $\beta_{\cdot i}^T X_t$, where $\beta_{\cdot i} \in \mathbb{R}^p$ denotes the $i$th column of $\beta$ and is a cointegrating vector.

The model in (4.1) can be equivalently written in the form of a vector error correction model (VECM), i.e.

$$\Delta X_t = \mu_t + \Pi X_{t-1} + \Phi_1^* \Delta X_{t-1} + \cdots + \Phi_{d-1}^* \Delta X_{t-d+1} + \epsilon_t,$$

where the correspondence between the parameters of the VECM and those of the VAR model is given by

$$\Phi_j^* = -\sum_{i=j+1}^{d} \Phi_i, \qquad \text{and} \qquad \Pi = \Phi_1 + \cdots + \Phi_d - I = -\mathcal{A}(1). \tag{4.2}$$

With the VECM representation, the rank of $\Pi$ determines the number of cointegrated series of $X_t$. There are two extreme scenarios: if $\text{rank}(\Pi) = 0$, then $\Pi = 0$ and $X_t$ is not conintegrated; if $\text{rank}(\Pi) = p$, then $X_t$ is an $I(0)$ process and can be studied directly from the VAR representation. However, if $0 < \text{rank}(\Pi) = r < p$, we can write $\Pi = \alpha\beta^T$ for

some $\alpha, \beta \in \mathbb{R}^{p\times r}$. Then $X_t$ is cointegrated with $r$ linearly independent cointegrated series given by $\beta^T X_t$, and these $r$ linear combinations are unit-root stationary. The rest of the chapter focuses on this non-trivial case, i.e., $0 < \text{rank}(\Pi) < p$.

Let $W_t = \beta^T X_t$ be the $r$ linearly independent cointegrated series with $\beta = [\beta_1, \ldots, \beta_r]$ being the cointegration vectors. In many real-world applications, each cointegrated series does not necessarily involve all coordinates of $X_t$, and further some coordinates are not expected to appear in any of the cointegrated series. In other words, the cointegrating matrix $\beta$ possesses the structure of being simultaneously elementwise sparse and group (row-wise) sparse.

We are interested in investigating how to estimate such a $\beta$ with the designated sparsity pattern from a snapshot of the random process that constitutes our sample. Without loss of generality and for easiness of presentation, we consider the special case $d = 2$ and $\mu_t = 0$, and the VECM is simplified to

$$\Delta X_t = \Pi X_{t-1} + \Phi_1^* \Delta X_{t-1} + \epsilon_t. \tag{4.3}$$

The estimation procedure can be easily generalized to the situation when $d > 2$.

### 4.2.1 Estimation

In this subsection, we formulate an optimization problem with appropriate regularization terms, whose solution is the estimated cointegrating matrix with the designated sparse structure. We first provide an outline of the proposed method. Specifically, by writing $\Pi = \alpha\beta^T$ and profiling out the other parameters (i.e. $\Phi_1, \alpha, \Sigma_\epsilon$), we obtain the profile likelihood with $\beta$ being the sole parameter. Without the assumption of sparsity or the presence of penalty terms, the minimizer of the negative profile likelihood $\check{\beta}$ can be obtained by solving a generalized eigenvalue problem and then extracting the eigenvectors. With the presence of penalty terms, we solve the same generalized eigenvalue problem, but aim to obtain simultaneously group sparse and elementwise-sparse eigenvectors. These sparse eigenvectors can be obtained from a procedure analogous to the sparse PCA (Zou et al., 2006) upon transformations.

We start the derivation by obtaining the cointegrating vectors without any sparsity constraint (Johansen, 1988). Given centered data $\{x_0, \ldots, x_n\}$, which is a snapshot of the underlying process $\{X_t\}$, we denote

$$\Delta\mathcal{X}_{[t_1:t_2]} = \begin{bmatrix} \Delta x_{t_1} & \cdots & \Delta x_{t_2} \end{bmatrix}^T \qquad \text{and} \qquad \mathcal{X}_{[t_1:t_2]} = \begin{bmatrix} x_{t_1} & \cdots & x_{t_2} \end{bmatrix}^T.$$

Under the Gaussianity assumption that $\epsilon_t \sim \mathcal{N}(0, \Sigma_\epsilon)$ and write $\Pi = \alpha\beta^T$, the log-likelihood function based on VECM is given by

$$\ell(\alpha, \beta, \Sigma_\epsilon) = -\frac{n}{2}\log|\Sigma_\epsilon| - \frac{1}{2}\sum_{t=2}^{n}(\Delta x_t - \alpha\beta^T x_{t-1} - \Phi_1^*\Delta x_{t-1})^T\Sigma_\epsilon^{-1}(\Delta x_t - \alpha\beta^T x_{t-1} - \Phi_1^*\Delta x_{t-1}) + \text{constant} \tag{4.4}$$

After first profiling out $\Phi_1^*$ by partial regression, and then followed by $\alpha$ and $\Sigma_\epsilon$, the unconstrained estimated cointegrating matrix $\check{\beta}$ maximizes the profile log-likelihood and is the solution to the following minimization problem:

$$\begin{aligned}\check{\beta} :&= \underset{\beta\in\mathbb{R}^{p\times r}}{\operatorname{argmin}} \left\{\det\left[S_{00} - S_{01}\beta(\beta^T S_{11}\beta)^{-1}\beta^T S_{10}\right]\right\} \\ &= \underset{\beta}{\operatorname{argmin}} \left\{|\beta^T S_{11}\beta - \beta^T S_{10}S_{00}^{-1}S_{01}\beta| \cdot |\beta^T S_{11}\beta|^{-1}\right\},\end{aligned} \tag{4.5}$$

where

$$S_{00} = \frac{1}{n-1}R_0^T R_0, \qquad S_{01} = \frac{1}{n-1}R_0^T R_1, \qquad \text{and} \qquad S_{11} = \frac{1}{n-1}R_1^T R_1, \tag{4.6}$$

are sample covariances based on partial regression residuals $R_0$ and $R_1$, obtained by respectively regressing $\Delta X_t$ and $X_{t-1}$ on $\Delta X_{t-1}$. Note the equality in (4.5) comes from the matrix identity

$$\det\left(\begin{bmatrix} S_{00} & S_{01}\beta \\ \beta^T S_{10} & \beta^T S_{11}\beta \end{bmatrix}\right) = |S_{00}|\cdot|\beta^T S_{11}\beta - \beta^T S_{10}S_{00}^{-1}S_{01}\beta| = |\beta^T S_{11}\beta|\cdot|S_{00} - S_{01}\beta(\beta^T S_{11}\beta)^{-1}\beta^T S_{10}|,$$

followed by viewing $|S_{00}|$ as a constant since it does not involve $\beta$. The solution to (4.5) are the first $r$ columns of $Q$, where $Q$ are eigenvectors satisfying

$$S_{11}Q\Lambda = S_{10}S_{00}^{-1}S_{01}Q, \qquad \text{subject to} \qquad Q^T S_{11}Q = I, \tag{4.7}$$

where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_p)$, and these $\lambda$'s are non-decreasing solutions to the generalized eigen-equation[1]

$$|\lambda S_{11} - S_{10}S_{00}^{-1}S_{01}| = 0. \tag{4.8}$$

Now we elaborate on how to obtain $Q$ satisfying (4.7) that in essence are the eigenvectors of the generalized eigenvalue problem, as it is closely related to our later formulation of the optimization problem. Decompose $S_{11}$ as $S_{11} = LL^T$, where $L = Q_{S_{11}}\Lambda_{S_{11}}^{1/2}$, with $Q_{S_{11}}$ being the eigenvectors of $S_{11}$ and $\Lambda_{S_{11}}$ being the diagonal matrix formed by the cor-

[1]Note $|\psi^T(M_1 - M_2)\psi||\psi^T M_1\psi|^{-1}$ can be minimized by solving the generalized eigen-equation $|\lambda M_1 - M_2| = 0$.

responding eigenvalues. Then $L^{-1}S_{11}(L)^{-T} = I$ holds by the construction of $L$. Now for (4.8), let $Q_M$ be the eigenvectors corresponding to eigenvalues $\lambda$ that are solutions to the following equation:

$$|\lambda I - L^{-1}S_{10}S_{00}^{-1}S_{01}(L)^{-T}| = 0, \tag{4.9}$$

such that $Q_M$ satisfies

$$L^{-1}S_{10}S_{00}^{-1}S_{01}(L)^{-T} = Q_M \Lambda Q_M^T, \text{ or equivalently,} Q_M^T \left(L^{-1}S_{10}S_{00}^{-1}S_{01}(L)^{-T}\right) Q_M = \Lambda.$$

By letting $Q := (L)^{-T}Q_M$, $Q$ satisfies

$$Q^T \left(S_{10}S_{00}^{-1}S_{01}\right) Q = \Lambda \qquad \text{and} \qquad Q^T S_{11} Q = I,$$

hence also satisfies $S_{11}Q\Lambda = S_{10}S_{00}^{-1}S_{01}Q$. Moreover, the $\lambda$'s which solve (4.9) are also solutions to (4.8). In other words, such $Q$ is the solution to (4.7) and contains our desired eigenvectors corresponding to the generalized eigenvalue problem: it is obtained by first solving the usual eigenvalue problem based on $\left[L^{-1}S_{10}S_{00}^{-1}S_{01}(L)^{-T}\right]$ as denoted in (4.9), and then transforming the obtained eigenvectors $Q_M$ by left-multiplying $(L)^{-T}$.

With a pre-specified $r$, the above procedure, by extracting the first $r$ columns of $Q$ that are eigenvectors corresponding to the generalized eigenvalue problem in (4.7) and (4.8), offers $r$ cointegrating vectors without any sparsity structure. This implies that if a special structure of the cointegrating matrix is desired, it can be achieved by incorporating regularization terms that induce such a structure in solving the same eigenvalue problem. Specifically, denote

$$M := L^{-1}S_{10}S_{00}^{-1}S_{01}L^{-T}. \tag{4.10}$$

Then combining the above derivations and applying the sparse PCA idea in Zou et al. (2006), we consider the following criterion that encourages both group sparse and elementwise sparse structures in $\widehat{\beta}$:

$$\begin{aligned}(\widehat{A}, \widehat{B}) = \underset{A\in\mathbb{R}^{p\times r}, B\in\mathbb{R}^{p\times r}}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{n} \|z_i - AB^T z_i\|^2 + \lambda \sum_{j=1}^{r} \|B_{\cdot j}\|_2^2 + \rho \sum_{i=1}^{p}\sum_{j=1}^{r} \left|\left[L^{-T}B\right]_{ij}\right| + \gamma \sum_{i=1}^{p} ||\left[L^{-T}B\right]_{i\cdot}||_2 \Big\}, \\ \text{subject to} \quad A^T A = I_{r\times r},\end{aligned} \tag{4.11}$$

where $Z = [z_1^T, \ldots, z_n^T] \in \mathbb{R}^{n\times p}$ $(n \geq p)$ comes from the decomposition $M = Z^T Z$, and the desired sparse cointegrating vectors $\widehat{\beta} = [\widehat{\beta}_1, \ldots, \widehat{\beta}_r]$ are given by $\widehat{\beta} = L^{-T}B$. Note that the decomposition $M = Z^T Z$ comes analogously from the self-constrained formulation of principal component analysis and may not be unique. We will later show in the appendix

that the specific decomposition does not make a difference as only $M$ is involved in the optimization procedure.

Once again, note that based on the previous derivation, the unconstrained cointegrating vectors $\check{\beta}$ consist of the first $r$ columns of $Q$, which is the solution to the generalized eigenvalue problem (4.7) and can be obtained by solving the usual eigenvalue problem based on $M$ and transforming the obtained eigenvectors $Q_M$ by left-multiplying $(L)^{-T}$ (see equation (4.9)). This suggests that a sparse cointegrating matrix $\widehat{\beta}$ can be obtained if we are able to obtain sparse eigenvectors corresponding to the generalized eigenvalue problem, or equivalently, the corresponding sparsity structure has been taken into consideration in obtaining the eigenvectors of $M$. Zou et al. (2006) showed that the columns of $\widehat{B}$, which is the solution to the following optimization problem, are proportional to the first $r$ eigenvectors of $M$:

$$\begin{aligned}(\widehat{A}, \widehat{B}) &= \underset{A,B}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{n} ||z_i - AB^T z_i||^2 + \lambda \sum_{j=1}^{r} ||b_j||^2 \Big\} \\ \text{subject to} \quad & A^T A = I_{r\times r},\end{aligned}$$

where $Z$ again comes from the decomposition of $M = Z^T Z$, $A_{p\times r} = [a_1, \ldots, a_r]$, and $B = [b_1, \ldots, b_r]$ with $\widehat{b}_j \propto Q_{M,j}$, for $j = 1, \ldots, r$. To induce the sparsity in $\widehat{b}_j$'s, Zou et al. (2006) applied the $\ell_1$-norm penalty on columns of $B$ and formulate the following optimization problem:

$$\begin{aligned}(\widehat{A}, \widehat{B}) = \underset{A,B}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{n} ||z_i - AB^T z_i||^2 + \lambda \sum_{j=1}^{r} ||b_j||^2 + \sum_{j=1}^{p} \rho_j ||b_j||_1 \Big\} \\ \text{subject to } A^T A = I_{r\times r}.\end{aligned}$$

In our setting, we assume the cointegrating vectors, which are given by $L^{-T}B$, are simultaneously group sparse and elementwise sparse, where the group sparsity is imposed on the rows of the transformed matrix. Therefore, instead of directly imposing such constraints on the eigenvectors $B$ as in Zou et al. (2006), we apply the corresponding regularization terms on the transformed eigenvector $L^{-T}B$. Thus there are three regularization terms in the optimization problem (4.11): the first term $\lambda \sum_{j=1}^{r} \|B_{\cdot j}\|^2$ comes from the self-constrained formulation in principal component analysis, and the choice of $\lambda$ will not affect the final solution (Zou et al., 2006); the second term induces the elementwise sparsity; and similarly, the third term induces the group sparsity by penalizing the sum of the $\ell_2$ norm of the rows of $L^{-T}B$, corresponding to a group lasso penalty.

Before delving into the algorithm that solves (4.11), we provide further insight on the

group sparse assumption on the cointegrating matrix $\beta$. First we note that the decomposition $\Pi = \alpha\beta^T$ is not unique: for an arbitrary invertible matrix $U \in \mathbb{R}^{r\times r}$, let $\widetilde{\alpha} := \alpha U$, $\widetilde{\beta} := \beta U^{-T}$, then $\Pi = \widetilde{\alpha}\widetilde{\beta}^T$. Therefore, the cointegrating matrix can also be represented by $\widetilde{\beta}$, with the corresponding cointegrated series given by $\widetilde{\beta}^T X_t$. However, $\beta$ and $\widetilde{\beta}$ span the same space. In other words, we are not able to identify a unique cointegrating matrix, but the cointegration space is uniquely identifiable. The imposed group sparse structure on the cointegration matrix $\beta$ is invariant to such transformations: for $U \in \mathbb{R}^{r\times r}$ and $\widetilde{\beta} = \beta U^{-T}$, if the $i$th row of $\beta$ is identically zero, then the $i$th row of $\widetilde{\beta}$ will also be identically zero. Consequently, the above outlined procedure offers some cointegrating matrix $\widehat{\beta}$ that equips the corresponding cointegrating vectors with the following property: each cointegrated series only involves a few coordinates of the original series and some coordinates of the original series do not show up in any of the cointegrated series.

**Solving for the sparse cointegrating matrix.** Now we specify the algorithm for optimizing (4.11). The solution to (4.11) can be obtained by iteratively solving for $A$ and $B$, while holding the other one fixed. Specifically, let $Q = L^{-T}B$, and since $\sum_{i=1}^{n} \|z_i - AB^T z_i\|^2 = \|Z - ZBA^T\|_F^2$, the objective function in (4.11) can be rewritten as

$$\mathcal{L}(A, Q) = \|Z - ZL^T QA'\|_F^2 + \lambda \sum_{j=1}^{r} \|(L^T Q)_{\cdot j}\|_2^2 + \rho \sum_{i=1}^{p}\sum_{j=1}^{r} |Q_{ij}| + \gamma \sum_{i=1}^{p} ||Q_{i\cdot}||_2 \quad (4.12)$$

Hence, we can equivalently solve the optimization by iteratively updating $A$ and $Q$, and ultimately obtain the desired $B$ by applying the transformation on $Q$ via $B = L^T\widehat{Q}$. Algorithm 2 outlines the alternate update between $A$ and $Q$. The exact update of $Q$ further involves a coordinate descent algorithm and is given in the appendix.

Overall, we summarize the computational procedure for obtaining the cointegrating vectors with the designated sparsity pattern in Algorithm 3.

### 4.2.2 Tuning parameter selection.

The exact solution to (4.12) relies on the input values of the tuning parameters.

There are two tuning parameters in the proposed criterion, $\rho$ and $\gamma$. Since the data under consideration is time series, cross-validation becomes infeasible, whereas for information based selection criterion, the calculation of the degrees of freedom is complicated under our proposed framework and may not have closed-form formulae. Here we consider choosing tuning parameters based on the average prediction error on the validation set, using a rolling window strategy. Specifically, let $T_r$ denote the length of each rolling window, $T_v$ denote

**Algorithm 2:** Algorithm for obtaining $(\widehat{A}, \widehat{Q})$ from (4.12).

**Input**: Input matrices $M$ and $L$, tuning parmaeters $\lambda$, $\rho$ and $\gamma$.

1 **Initialization.** Initialize $A$ by setting $\widehat{A}^{(0)}$ to be the first $r$ eigenvectors of $M$.

2 **while** *not converged* **do**

3 For fixed $\widehat{A}^{(m)}$, update $Q$ by:

$$\widehat{Q}^{(m)} = \underset{Q}{\operatorname{argmin}} \left\{ \|Z - ZL^TQ(\widehat{A}^{(m)})^T\|_F^2 + \lambda \cdot \text{trace}(Q^TLL^TQ) + \rho \sum_{i=1}^{p}\sum_{j=1}^{r} |Q_{ij}| + \gamma \sum_{i=1}^{p} ||Q_{i\cdot}||_2 \right\} \tag{4.13}$$

For fixed $\widehat{Q}^{(m)}$, compute the SVD of $\left[ML^T\widehat{Q}^{(m)}\right] = UDV^T$ and update $\widehat{A}$ by $\widehat{A}^{(m+1)} = UV^T$

4 **end**

5 **Normalization.** Let $\widehat{B}^{\infty} = L^T\widehat{Q} = [\widehat{b}_1, \ldots, \widehat{b}_r]$, then let $\check{B} = [\check{b}_1, \ldots, \check{b}_r]$ where $\check{b}_1 = \widehat{b}_1/\|\widehat{b}_1\|_2$.

6 Let $\widehat{\beta} = (L^T)^{-1}\check{B}$, i.e., $\widehat{\beta} = \widehat{Q}\text{diag}(1/\|\widehat{b}_1\|_2, \ldots, 1/\|\widehat{b}_r\|_2)$.

**Output**: Sparse cointegrating matrix $\widehat{\beta}$.

the length of the validation set, and $\Delta_{\text{jump}}$ denote the fixed jump size. The total number of rolling windows is then

$$\left\lfloor \frac{T - (T_r + T_v)}{\Delta_{\text{jump}}} \right\rfloor, \qquad \text{where } T \text{ is the length of the entire available sample.}$$

Then for each rolling window $w_i$, we fit the model to the data using data points within the rolling wondow $w_i$, that is, data points with indices $(1 + (w_i - 1)\Delta_{\text{jump}}), \ldots, (T_r + (w_i - 1)\Delta_{\text{jump}})$, obtain parameter estimates based on Algorithm 3, and then do one-step-ahead forecast on the validation set $(T_r + 1 + (w_i - 1)\Delta_{\text{jump}}), \ldots, (T_r + T_v + (w_i - 1)\Delta_{\text{jump}})$. The tuning parameters $(\rho, \gamma)$ are then selected based on the pair that gives the minimum relative mean square forecasting error, averaged over all rolling windows.

## 4.3 Simulation studies

In this section, we evaluate the performance of the proposed method using synthetic data. We consider two settings, where the underlying processes $\{X_t\}$ all evolve according to a VAR$(1)$ model, whose VECM representation is given by

$$\Delta X_t = \Pi X_{t-1} + \epsilon,$$

**Algorithm 3:** Obtaining sparse cointegrating vectors.

**Model:** $\Delta X_t = \Pi X_{t-1} + \Phi_1^* \Delta X_{t-1} + \epsilon_t$.

**Input**: Observed data $X$, tuning parameters $\rho$ and $\gamma$, $\lambda$ is fixed at a small postive value.

1. Obtain partial regression residuals $R_0$, $R_1$ and the residual sample covariances $S_{00}$, $S_{01}$ and $S_{11}$.

2. Calculate the matrix on which a usual eigenvalue problem is considered:

$$M = L^{-1} S_{10} S_{00}^{-1} S_{01} L^{-T}, \qquad \text{where} \qquad S_{11} = LL^T.$$

3. Solve the optimization problem (4.11), by alternately updating $A$ and $Q$ according to Algorithm 2.

4. Obtain the sparse cointegrating vector $\widehat{\beta}$ based on the convergent solution.

**Output**: Simultaneously group sparse and elementwise sparse cointegrating vector $\widehat{\beta}$.

with the corresponding VAR representation being $X_t = (I + \Pi) X_{t-1} + \epsilon$.

It is worth pointing out the choice of model parameters for a cointegrated system can not be arbitrary. Roughly speaking, in order to construct a cointegrated series from a multivariate $I(1)$ system, restrictions on some matrix-valued polynomial involving interactions among the coefficient matrices are required. We discuss some of the restrictions in the appendix. For example, under a VAR$(1)$ setup, the restriction translates to

$$\rho_{\max}(I + \Pi) < 1, \qquad \text{where } \rho_{\max}(\cdot) \text{ denotes the spectral radius.}$$

**S1. Cointegrated series in non-overlapping pairs.** In this setting, we consider a cointegrated system in which each cointegrated series is a linear combination of non-overlapping pairs of the original series.

We illustrate the generating mechanism on a simpler model in which only two series are involved. To generate two cointegrated series $\{\xi_t\}$ and $\{\eta_t\}$ with the cointegrating vector being $(1, \beta)$, we can sequentially generate data at each time point according to

$$\begin{aligned} \xi_t &= \xi_{t-1} + \alpha_\xi (\xi_{t-1} + \beta \eta_{t-1}) + \epsilon_t^\xi \\ \eta_t &= \eta_{t-1} + \alpha_\eta (\xi_{t-1} + \beta \eta_{t-1}) + \epsilon_t^\eta \end{aligned}$$

where $(\alpha_\xi, \alpha_\eta, \beta)$ satisfies $|1 + \alpha_\xi + \alpha_\eta \beta| < 1$ (see Tsay, 2005). This is a 2-dimensional

case for a VAR$(1)$ model with the following VECM:

$$\Delta X_t = \begin{bmatrix} \alpha_\xi \\ \alpha_\eta \end{bmatrix} \begin{bmatrix} 1 & \beta \end{bmatrix} X_{t-1} + \epsilon_t.$$

Now we consider a $p$-dimensional cointegrated system $X_t$ in which there are $r$ cointegrated series ($r \leq p/2$) (hence a total of $2r$ series are involved in the cointegrated relations). With non-overlapping pairs of series, the above procedure can be carried out in parallel. Specifically, we proceed as follows. For each cointegrated series $k = 1, \ldots, r$:

1) Uniformly choose $\alpha_{k,1}$ and $\alpha_{k,2}$ from $(-0.9, -0.85, -0.8, \ldots, -0.1, 0.1, 0.15, \ldots, 0.9)$, which determine the speed of cointegration of the two series in the cointegrated series indexed by $k$.

2) Uniformly choose the sum $\gamma_k := 1+\alpha_{k,1}+\alpha_{k,2}\beta_k$ from $(-0.9, -0.8, \ldots, -0.1, 0.1, \ldots, 0.9)$, hence automatically we have $|\gamma_k| < 1$.

3) Solve for $\beta_k = \frac{\gamma_k - 1 - \alpha_{k,1}}{\alpha_{k,2}}$, with $(1, \beta_k)$ being the effective entries of the cointegrating vector for the $k$th cointegrated series.

We fill $\alpha_{k,1}$, $\alpha_{k,2}$ and $\beta_k$ into $\alpha \in \mathbb{R}^{p\times r}$ and $\beta \in \mathbb{R}^{p\times r}$, which later form $\Pi := \alpha\beta^T$. Specifically, for $\alpha$, its $(2k-1, k)$ and $(2k, k)$ entries are respectively filled with $\alpha_{k,1}$ and $\alpha_{k,2}$ for $k = 1, \ldots, r$, and the rest entries are set zero. Similarly for $\beta$, its $(2k-1, k)$ entry is 1 and its $(2k, k)$ entry is $\beta_k$ with the rest being zero. Finally, the sequence is generated according to

$$X_t = (I + \Pi)X_{t-1} + \epsilon_t,$$

and we control the magnitude of $\Sigma_\epsilon$ to obtain the desired level of signal-to-noise ratio.

Note that for a cointegrated system $X_t$ generated in this way, there are $r$ cointegrated series, with the $k$th cointegrating vector being

$$(\underbrace{0, \ldots, 0}_{2(k-1)}, 1, \beta_k, \underbrace{0, \ldots, 0}_{(p-2k)})^T,$$

and the corresponding cointegrated series being $X_{t,(2k-1)} + \beta_k X_{t,(2k)}$. Note that only the first $2r$ coordinates are involved in the cointegrated series, and the rest $(p-2r)$ coordinates are stand-alone univariate random walks. Also note that the above mentioned restrictions are satisfied since we require $|1 + \alpha_{k,1} + \alpha_{k,2}\beta_k| < 1$ for all $k = 1, \ldots, r$, and together with the block structure of $\alpha$ and $\beta$, this ensures that the coefficient matrix conforms with the restriction.

**S2. Cointegrated series with an identical set of series involved.** In this setting, we consider a cointegrated system in which all cointegrated series have an identical set of coordinates of the original series involved. In other words, the matrix of cointegrating matrix is group sparse.

As we have pointed out at the beginning of this section, the coefficient matrix needs to satisfy certain restrictions so that the multivariate system can even have any cointegrated relations. With a pre-specified model dimension $p$ and the number of cointegrated series $r$, we generate the model parameters as follows:

1) Randomly generate each entry of $\alpha \in \mathbb{R}^{p\times r}$ from $\mathsf{Unif}[(-0.21,-0.19)\cup(0.19,0.21)]$.

2) Let $s_{\mathcal{G}}$ denote the group sparsity level of $\beta \in \mathbb{R}^{p\times r}$, then $p(1-s_{\mathcal{G}})$ rows of $\beta$ are set to zero at random. The rest of the entries are nonzero and randomly generated from $\mathsf{Unif}[(-2.1,-1.9)\cup(1.9,2.1)]$.

3) Check if the restriction $\rho_{\max}(I+\alpha\beta^T) < 1$ is satisfied. If $\rho_{\max} < 1$, proceed with the above generated $\alpha$ and $\beta$ to generate data according to $X_t = (I+\alpha\beta^T)X_{t-1}+\epsilon_t$; otherwise, repeat steps 1 and 2 until the constraint on the spectral radius is satisfied.

Empirically, for **S2**, the required $\alpha$ and $\beta$ can be generated using a while loop until the restriction is satisfied, which does not take long. For all settings, we consider the following two types of structure for $\Sigma_\epsilon$:

a) $\Sigma_\epsilon = \sigma^2\mathrm{I}$, that is, each coordinate of $\epsilon_t$ is independently identically distributed (iid);

b) $\Sigma_{\epsilon,ij} = (\sigma^2)\rho^{|i-j|}$ for some $0 < |\rho| < 1$, that is, the correlation has an AR$(1)$-type decay.

In terms of evaluation, as previously mentioned, the cointegrating vectors are not uniquely identifiable, but the cointegration space is. Hence, instead of comparing the estimated cointegrating vectors to the data generating cointegrating vector $\beta^\star$, we measure the principal angle between the space spanned by columns of $\beta^\star$ and that by columns of the estimates $\widehat{\beta}$. Empirically, the angle $\theta$ (denoted as Space.Err) is calculated using $\theta = \arccos(d_{\max})$, where $d_{\max}$ is the largest singular value of $Q_{\widehat{\beta}}^T Q_{\beta^\star}$, with $Q_{\widehat{\beta}}$ and $Q_{\beta^\star}$ coming from the QR decomposition of $\widehat{\beta}$ and $\beta^\star$, respectively. Additionally, we measure the relative error of the estimated $\Pi$ matrix, given by

$$\text{Pi.Err} = \frac{\|\widehat{\Pi}-\Pi^\star\|_F}{\|\Pi^\star\|_F}.$$

The results are summarized in Tables 4.1 - 4.3. In general, the proposed method works reasonably well in both estimating $\Pi$ and identifying the space spanned by columns of

$\beta$. As expected, as the signal to noise ratio increases, the relative estimation error tends to decrease. The performance of the proposed method in identifying the space spanned by $\beta$ does not depend much on the signal to noise ratio though. Further, as the sparsity of the underlying model increases, the performance of the proposed method also tends to improve. The difference between performances under the two types of error covariance, i.e. $\Sigma_\epsilon = \sigma^2 \mathrm{I}$ vs $\Sigma_\epsilon = (\sigma^2 0.5^{|i-j|})$ is not much.

Table 4.1: Simulation results for S1 with $p = 100$, $r = 20$ over 50 replications

| | SNR | 0.75 | 1.25 | 1.6 | 2.0 |
|---|---|---|---|---|---|
| $\Sigma_\epsilon = \sigma^2 \mathrm{I}$ | Pi.Err | 0.20(0.047) | 0.22(0.026) | 0.13(0.010) | 0.12(0.008) |
| | Space.Err | 0.006(0.0019) | 0.005(0.0016) | 0.002(0.0006) | 0.0008(0.00035) |
| $\Sigma_\epsilon = \sigma^2(0.5^{\|i-j\|})$ | Pi.Err | 0.22(0.009) | 0.25(0.026) | 0.16(0.009) | 0.14(0.0091) |
| | Space.Err | 0.008(0.0022) | 0.005(0.0017) | 0.003(0.001) | 0.001(0.0005) |

Table 4.2: Simulation results for S2 with $p = 20$, $r = 5$, $\|\beta_{\cdot,j}\|_0 = 6$, over 50 replications

| | SNR | 1.5 | 2.5 | 3.5 | 4.5 |
|---|---|---|---|---|---|
| $\Sigma_\epsilon = \sigma^2 \mathrm{I}$ | Pi.Err | 0.92(0.10) | 0.49(0.201) | 0.23(0.10) | 0.15(0.108) |
| | Space.Err | 0.10(0.09) | 0.03(0.012) | 0.03(0.016) | 0.05(0.031) |
| $\Sigma_\epsilon = \sigma^2(0.5^{\|i-j\|})$ | Pi.Err | 0.86(0.13) | 0.40(0.158) | 0.20(0.12) | 0.10(0.054) |
| | Space.Err | 0.09(0.09) | 0.04(0.023) | 0.03(0.021) | 0.04(0.025) |

Table 4.3: Simulation results for S2 with $p = 30$, $r = 5$, $\|\beta_{\cdot,j}\|_0 = 4$, over 50 replications

| | SNR | 1.5 | 2.5 | 4.0 | 4.5 |
|---|---|---|---|---|---|
| $\Sigma_\epsilon = \sigma^2 \mathrm{I}$ | Pi.Err | 0.34(0.14) | 0.17(0.10) | 0.57(0.18) | 0.45(0.15) |
| | Space.Err | 0.03(0.019) | 0.02(0.015) | 0.02(0.017) | 0.02(0.018) |
| $\Sigma_\epsilon = \sigma^2(0.5^{\|i-j\|})$ | Pi.Err | 0.38(0.17) | 0.17(0.11) | 0.61(0.20) | 0.50(0.20) |
| | Space.Err | 0.04(0.020) | 0.02(0.019) | 0.03(0.021) | 0.03(0.019) |

## 4.4 Data examples

In this section, we apply the proposed method to two data examples that are potentially useful in constructing portfolios in the financial market. The first example was briefly mentioned in the introduction section and we seek cointegrated series among a set of stock prices, while in the second example, we seek cointegrated series among treasury yields of different maturities, as several earlier work have indicated that the term structure of U.S. treasury bills can be modeled as a cointegrated system (Bradley and Lumpkin, 1992, Hall

et al., 1992, Zhang, 1993). For both examples, we assume the cointegrating vectors are simultaneously elementwise sparse and group sparse.

For both examples, we assume the original series follow a VAR$(1)$ model, whose VECM form is

$$\Delta X_t = \alpha\beta^T X_{t-1} + \epsilon_t, \tag{4.14}$$

or equivalently, $X_t = (I+\alpha\beta^T)X_{t-1}+\epsilon_t$, if written in the VAR form. We split the collected data into training and testing sets, obtain the cointegrating vector(s) on the training set, and then apply the corresponding transformation to series on the testing sets and check whether the cointegrated series on the testing set is stationary, via the augmented Dickey-Fuller test (ADF) tests. Specifically, for training data, we select tuning parameters $\rho$ and $\gamma$ based on the selection procedure described in Section 4.2.2, then fit the model to the entire training set with the selected tuning parameters, and obtain the cointegrating vectors.

### 4.4.1 Financial sector stock data

In this example, we use daily price data of stocks within the financial sector (based on GICS) that are among S&P 100 index constituents, as of March 2017. This gives us a total number of 14 stocks. We seek 2 cointegrating vectors among these stocks prices such that the cointegrated series are stationary. The time span under consideration is January 1, 2014 to December 31, 2016. The training set contains data ranging from January 1, 2014 to December 28, 2015, and the testing set contains data from January 1, 2016 to December 31, 2016. Note that the original stock price series are $I(1)$, as their differenced series are roughly $I(0)$ (via the ADF test).

Table 4.4 shows the correlation matrix for the log-returns of these stocks. In general, these stocks have exhibited a large degree of comovement in terms of their log-returns, with the smallest correlation detected between ALL and AXP (0.37) and the largest between GS and MS (0.88).

As previously mentioned, after we obtain the cointegrating vectors from the training data and the corresponding cointegrated series, we apply the same linear transformation on the testing data, so that the cointegrated series on the testing set are also available. Then we use the ADF test to test the stationarity of the cointegrated series on both the training set and the testing set. Table 4.5 shows the $p$-values of the test results, with the null hypothesis being that the series is stationary. Both tests fail to reject the null hypothesis (especially on the testing set) indicating that the identified series are stationary.

Table 4.6 shows the estimated cointegrating vectors, and the detected cointegrated time series are shown in Figure 4.1 and Figure 4.2 respectively. As one can see, the identified

Table 4.4: Correlation matrix for log-returns of financial sector stocks

| | AIG | ALL | AXP | BAC | BK | BLK | C | COF | GS | JPM | MET | MS | USB | WFC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AIG | 1.00 | | | | | | | | | | | | | |
| ALL | 0.56 | 1.00 | | | | | | | | | | | | |
| AXP | 0.51 | 0.37 | 1.00 | | | | | | | | | | | |
| BAC | 0.70 | 0.47 | 0.53 | 1.00 | | | | | | | | | | |
| BK | 0.69 | 0.50 | 0.56 | 0.79 | 1.00 | | | | | | | | | |
| BLK | 0.69 | 0.53 | 0.52 | 0.69 | 0.72 | 1.00 | | | | | | | | |
| C | 0.75 | 0.52 | 0.54 | 0.88 | 0.79 | 0.74 | 1.00 | | | | | | | |
| COF | 0.62 | 0.46 | 0.57 | 0.70 | 0.69 | 0.67 | 0.73 | 1.00 | | | | | | |
| GS | 0.72 | 0.54 | 0.55 | 0.81 | 0.77 | 0.74 | 0.83 | 0.71 | 1.00 | | | | | |
| JPM | 0.76 | 0.56 | 0.54 | 0.85 | 0.80 | 0.72 | 0.87 | 0.72 | 0.85 | 1.00 | | | | |
| MET | 0.76 | 0.49 | 0.51 | 0.78 | 0.75 | 0.70 | 0.79 | 0.67 | 0.76 | 0.79 | 1.00 | | | |
| MS | 0.72 | 0.51 | 0.55 | 0.83 | 0.78 | 0.74 | 0.85 | 0.69 | 0.88 | 0.84 | 0.77 | 1.00 | | |
| USB | 0.72 | 0.57 | 0.56 | 0.81 | 0.80 | 0.74 | 0.82 | 0.73 | 0.79 | 0.85 | 0.76 | 0.79 | 1.00 | |
| WFC | 0.71 | 0.56 | 0.53 | 0.79 | 0.76 | 0.71 | 0.78 | 0.71 | 0.78 | 0.83 | 0.73 | 0.77 | 0.85 | 1.00 |

Table 4.5: $p$-value from the ADF test on identified cointegrated series

| | Training set | Testing Set |
|---|---|---|
| Series 1 | 0.99 | 0.82 |
| Series 2 | 0.99 | 0.98 |

cointegrated series exhibit a much higher degree of stationarity on the training set than on the testing set, though the detected series on the testing set fail to reject the null hypothesis that they are stationary via the ADF test. Note the estimation is based on the implicit assumption that the dynamic of the underlying series is unchanged over time, which is hardly satisfied for real data.

Table 4.6: Estimated cointegrating vectors for financial sector stocks

| | AIG | ALL | AXP | BAC | BK | BLK | C | COF | GS | JPM | MET | MS | USB | WFC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CV 1 | 0.08 | -0.19 | 0.07 | 0.00 | -0.14 | 0.07 | -0.29 | -0.61 | 0.00 | 0.00 | 0.70 | 0.47 | -1.10 | 0.65 |
| CV 2 | -0.13 | -0.39 | 0.14 | 0.00 | -0.20 | 0.04 | 0.57 | 0.30 | -0.26 | 0.16 | -0.79 | 0.78 | 0.00 | 0.69 |

Figure 4.1: Estimated cointegrated series 1 for financial stocks

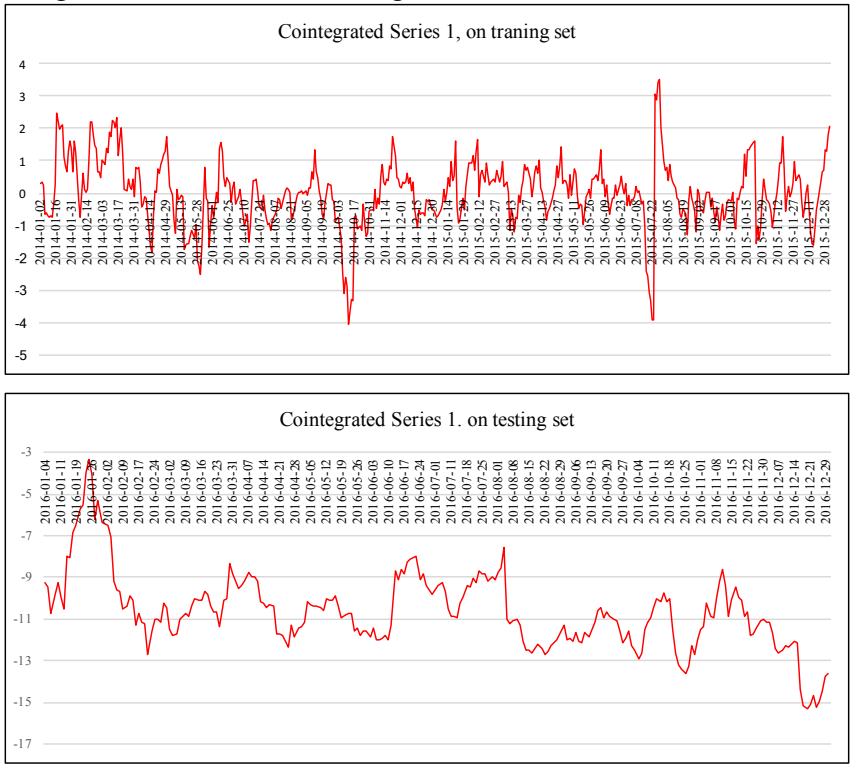
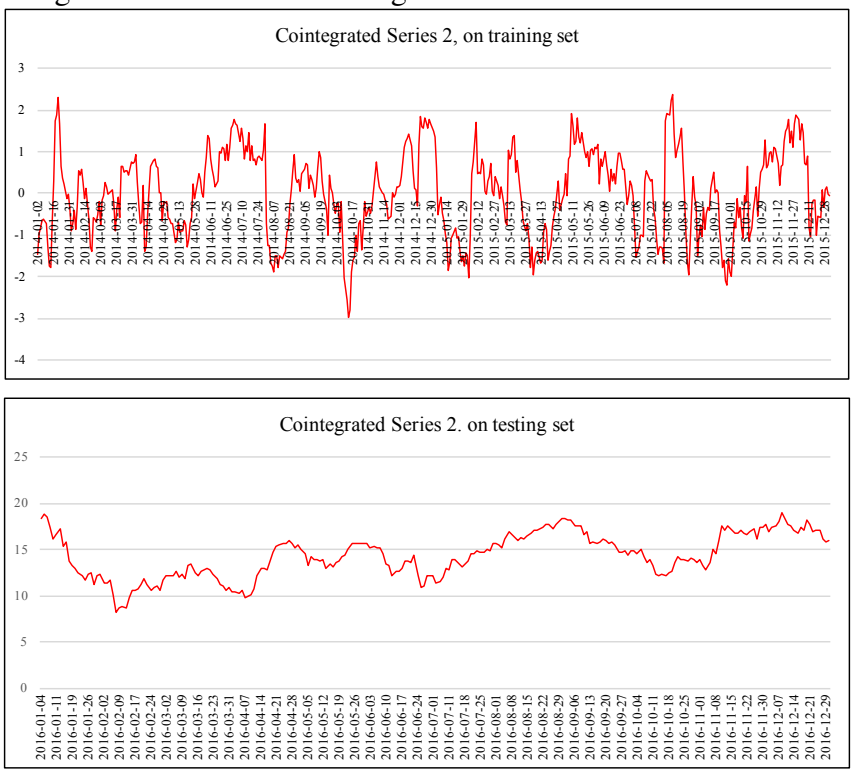



Figure 4.2: Estimated cointegrated series 2 for financial stocks



### 4.4.2 Treasury yield data

In this example, we use monthly data of treasury yields of different maturities, and consider a total number of 7 series with the maturities being 1yr, 2yrs, 3yrs, 5yrs, 7yrs, 10yrs, and 20yrs respectively. We seek 1 cointegrated relation among these series such that the cointegrated series is stationary. The time span under consideration is January 1995 to December 2016. We use the data ranging from 1995 to 2014 as the training data, and the data ranging from 2015 to 2016 as the testing data.

In principle, as the cointegrated series represents the long-run relationship among the interest rates, it is reasonable to expect the existence of more than one cointegrated series, as the long-run equilibrium may not be unique. With 7 given series, we could expect as many as 6 cointegrated series, given the special inter-linkage among these rates. However, as indicated by Bradley and Lumpkin (1992), some of the cointegrated series might be sensitive and unstable. Hence in their work, they focused on reporting one cointegrated series, with the 7yr rate as the dependent variable. On the other hand, although there are available testing procedures for testing the number of cointegrated relations, they are sensitive to regime change (Hall et al., 1992). Given the time span under consideration for our collected data, there has been one significant regime change after the 2008 subprime mortgage crisis, as a result of the Federal Reserve's accommodative monetary policy such as quantitative easing, and a minor regime change around 2002 after the "Dotcom Bubble". For the above mentioned reason, we do not conduct any test to determine the number of cointegrated relations. We set $r = 1$ and focus on presenting the resulting cointegrating vector. Empirically, we have tried with larger values of $r$, however, many of the detected cointegrated series have rejected the null hypothesis of the ADF test against their non-stationary alternatives.

Figure 4.3 shows the treasury yields of different maturities. As one can see, the original series are non-stationary, but always move toward the same direction.

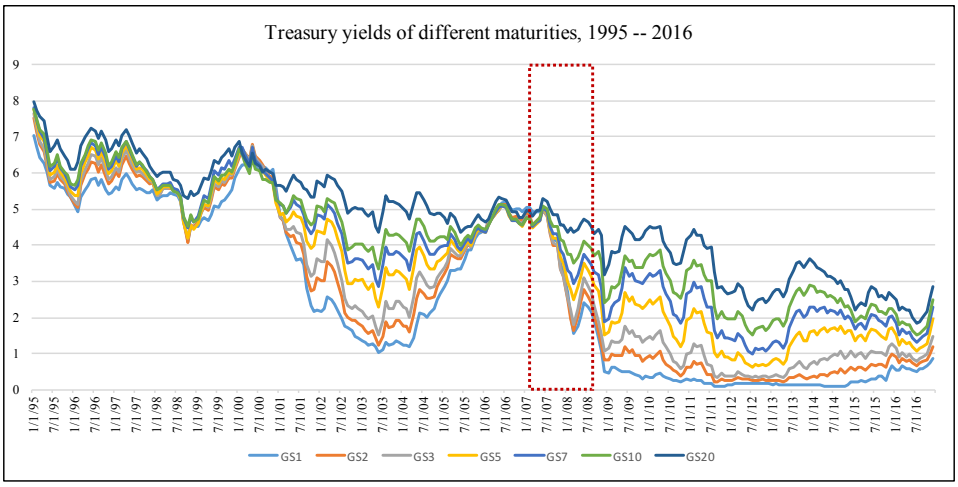


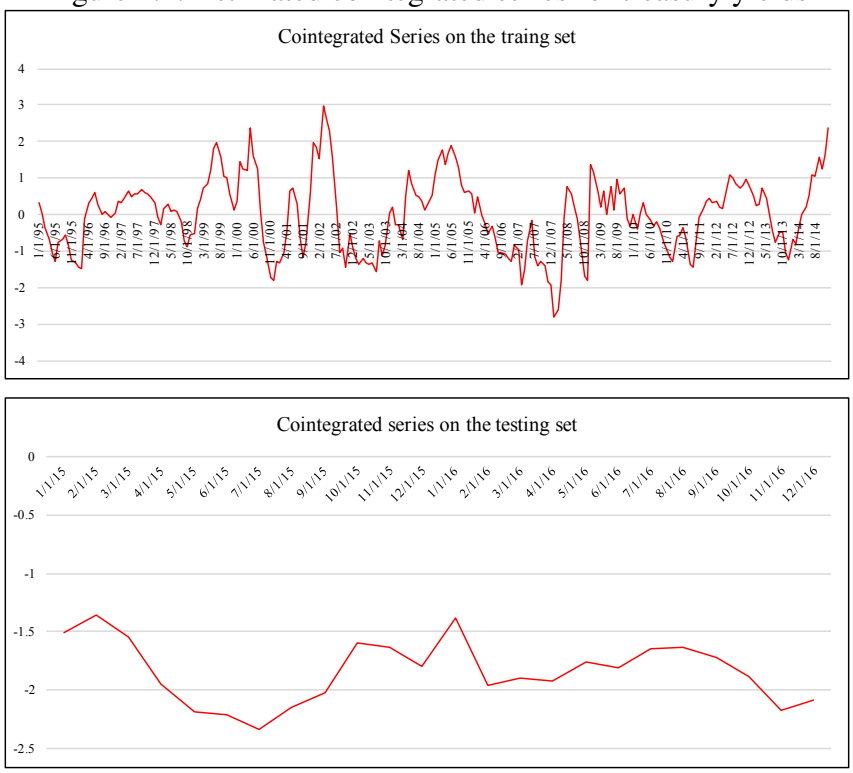

Figure 4.3: Treasury yields of different maturities over time

The estimated cointegrating vector is shown in Table 4.7, which contains the coefficients corresponding to treasury notes/bonds with different maturities in the cointegrated series formed by their linear combination. We can see that treasury yields of similar maturities have similar magnitudes in coefficients: (1yr vs. 2yrs), (3yrs vs. 5yrs), and (7yrs vs. 10yrs); yet the estimated coefficients have opposite signs. This is also consistent with the belief that there is some long-run equilibrium among these rates, and the divergence of rates will not be persistent due to the arbitrage activities in the secondary market. The cointegrated series on both the training and the testing sets fail to reject the null hypothesis of stationarity, with $p$-values being 0.98 and 0.97 respectively. Figure 4.4 shows the time series plots for the cointegrated series on both the training and testing sets.

Table 4.7: Estimated cointegrating vector for treasury yields with constant maturity

| Maturity | 1yr | 2yr | 3yr | 5yr | 7yr | 10yr | 20yr |
|---|---|---|---|---|---|---|---|
| Estimated coefficient | -6.37 | 7.31 | 1.81 | -2.24 | 0 | 0 | -1.28 |

Figure 4.4: Estimated cointegrated series for treasury yields




## 4.5 Summary.

In this chapter, we have considered a cointegrated VAR system, from which we are able to obtain cointegrating vectors that are simultaneously group and elementwise sparse. Specifically, the group sparse pattern is invariant to linear transformations, enabling us to remove a subset of the coordinates of the original series from the cointegration space, regardless of the exact estimates of the cointegrating vector. The optimization criterion whose objective function incorporates proper regularization terms, has been formulated based on solving a generalized eigenvalue problem, which is a surrogate of maximizing the profile likelihood function. We have applied the proposed method to two real world data examples and obtained interpretable results.

# CHAPTER 5

# Sparse Rank Support Vector Machines

## 5.1 Introduction

In this chapter, we consider the learning to rank problem. Learning to rank has attracted significant interests in recent years, with broad applications in areas such as web search, recommender systems and information retrieval; for detailed introduction on the topic, we refer the readers to Liu et al. (2009). In standard ranking problems, we are given a set of "queries" $q_1, \ldots, q_n$, where $n$ denotes the number of queries, and for each $q_i, i = 1, \ldots, n$, there are a set of associated "documents" $x_{i1}, \ldots, x_{im}$, where each $x_{ik}, k = 1, \ldots, m$ is a vector containing $p$ features (which usually depend on the query $q_i$), i.e. $x_{ik} = (x_{ik1}, \ldots, x_{ikp}) = (x_{ikj})_{j=1}^{p}$, and together with the query-document features, there are also associated ratings (rankings) $r_{ik} \in \mathbb{R}$, indicating how much the document $x_{ik}$ is "relevant" for query $q_i$, with large value of $r_{ik}$ being more relevant and small value of $r_{ik}$ being less relevant. Note that for notational simplicity (and without loss of generality), we assume the number of documents $m$ is the same for different queries, but in general, this may not be the case.

The goal is to learn a ranking function $\widehat{f}(x)$, such that when a new query $q^*$ and the associated query-documents $x_1^*, \ldots, x_m^*$ are given, one can use the ranking function to produce scores $\widehat{f}(x_1^*), \ldots, \widehat{f}(x_m^*)$, such that these scores match well with the underlying ranking of these documents, in terms of their relevance to the query. In this chapter, we focus on the linear ranking function, i.e. $f(x) = w^T x$.

The support vector machine (Cortes and Vapnik, 1995) has been a popular tool for classification problems in both the machine learning and statistics communities, and it has been extended to address the learning to rank problem with great success (Joachims, 2002). Specifically, in the learning to rank setting, the rank SVM considers the following

optimization criterion:

$$\min_{w,\xi_{ikk'}} \quad \sum_{i=1}^{n} \sum_{r_{ik}>r_{ik'}} \xi_{ikk'} + \lambda \sum_{j} w_j^2 \tag{5.1}$$

$$\text{subject to} \quad w^T x_{ik} - w^T x_{ik'} \geq 1 - \xi_{ikk'} \ \text{ for } r_{ik} > r_{ik'}, \tag{5.2}$$

$$\xi_{ikk'} \geq 0, \tag{5.3}$$

where $\lambda$ is a tuning parameter. The criterion implies that for query $q_i$, if the rank of document $k$ is higher than that of document $k'$, i.e. $r_{ik} > r_{ik'}$, then the score $w^T x_{ik}$ should be larger than the score $w^T x_{ik'}$, for a "margin" of at least $(1 - \xi_{ikk'})$, $\xi_{ikk'} \geq 0$, and the summation of $\xi_{ikk'}$ over all document pairs (within the same query) should be as small as possible.

Note (5.1) can be written in an equivalent "loss + penalty" form, i.e.

$$\min_{w} \sum_{i} \sum_{r_{ik}>r_{ik'}} (1 - w^T x_{ik} + w^T x_{ik'})_+ + \lambda \sum_{j=1} w_j^2, \tag{5.4}$$

where the loss $(1 - w^T x_{ik} + w^T x_{ik'})_+$ is called the hinge loss and the penalty is called the ridge penalty. The hinge loss can be considered as a convex relaxation of the indicator function $\mathbb{I}(w^T x_{ik} < w^T x_{ik'})$, which directly compares the two scores $w^T x_{ik}$ and $w^T x_{ik'}$. The idea of penalizing by the sum-of-squares of the parameters is also used in neural networks, where it is known as weight decay. The ridge penalty shrinks the fitted coefficients towards zero. It is well known that this shrinkage has the effect of controlling the variances of $\widehat{w}$, hence possibly improves the fitted model's prediction accuracy, especially when there are many highly correlated features. So from a statistical function estimation point of view, the ridge penalty could possibly explain the success of the rank SVM. On the other hand, computational learning theory has associated the good performance of the rank SVM to its margin maximizing property (Joachims, 2006), a property of the hinge loss.

Note the standard rank SVM uses all features when learning the ranking function. This could be undesirable, especially in the high-dimensional setting, as many features may not be relevant for ranking, and keeping them in the ranking function will introduce unnecessary noise when it comes to prediction. In this chapter, we propose to extend the standard rank SVM model to the high-dimensional setting so that irrelevant features can be removed from the ranking function. Specifically, the rest of the chapter is organized as follows. In Section 2, we describe two sparse rank SVM models and develop efficient algorithms for solving them. In Section 3 and 4, we demonstrate the performances of the proposed methods using simulation studies and a real-world stock selection problem. Section 5 concludes

the chapter.

## 5.2 Sparse rank support vector machines

In this section, we consider two versions of sparse rank SVMs, the $\ell_1$-norm rank SVM and the elastic-net rank SVM. Specifically, in the $\ell_1$-norm rank SVM, we propose to replace the $\ell_2$-norm penalty in the standard rank SVM with the $\ell_1$-norm penalty Tibshirani (1996), i.e.

$$\min_{w} \sum_{i} \sum_{r_{ik}>r_{ik'}} (1 - w^T x_{ik} + w^T x_{ik'})_+ + \lambda \sum_{j} |w_j|, \tag{5.5}$$

where $\lambda$ is a tuning parameter, and when $\lambda$ is large enough, the solution $\hat{w}$ is sparse. It is well-known that the $\ell_1$-norm penalty has two major limitations: 1) the number of selected features is upper bounded by the sample size. Therefore, when the number of relevant features exceeds the sample size, for example in the high-dimensional setting, the $\ell_1$-norm penalty can only discover a portion of the relevant features; 2) for highly correlated and relevant features, the $\ell_1$-norm penalty tends to pick only one or a few of them. To address these two limitation, we also propose to use the elastic-net penalty (Zou and Hastie (2005)) for the rank SVM, i.e.

$$\min_{w} \sum_{i} \sum_{r_{ik}>r_{ik'}} (1 - w^T x_{ik} + w^T x_{ik'})_+ + \lambda_1 \sum_{j} |w_j| + \lambda_2 \sum_{j} w_j^2, \tag{5.6}$$

where both $\lambda_1$ and $\lambda_2$ are tuning parameters. Note the only difference between (5.5) and (5.6) is that in the $\ell_1$-norm rank SVM, we penalize the $\ell_1$-norm of the coefficient vector $w$, while in the elastic-net rank SVM, we penalize both the $\ell_1$-norm and the $\ell_2$-norm penalties of $w$, and by doing so, the elastic-net rank SVM enjoys several benefits: 1) similar to the $\ell_1$-norm rank SVM, it automatically selects features; 2) the number of selected features is no longer upper bounded by the sample size; 3) when features are highly correlated, they tend to be selected or removed together.

Since (5.5) is a special case of (5.6) with $\lambda_1 = 0$, for most of the discussion in the rest of the chapter, we focus on (5.6).

### 5.2.1 Algorithm

To solve the elastic-net rank SVM, note that (5.6) can be transformed into a quadratic programming problem, i.e.

$$\min_{w_j^+, w_j^-, \xi_{ikk'}} \quad \sum_i \sum_{r_{ik} > r_{ik'}} \xi_{ikk'} + \lambda_1 \sum_j (w_j^+ + w_j^-) + \lambda_2 \sum_j (w_j^+ + w_j^-)^2 \tag{5.7}$$

$$\text{subject to} \quad (w^+ - w^-)^T x_{ik} - (w^+ - w^-)^T x_{ik'} \geq 1 - \xi_{ikk'} \text{ for } r_{ik} > r_{ik'}, \tag{5.8}$$

$$\xi_{ikk'} \geq 0, \;\; w_j^+ \geq 0, \;\; w_j^- \geq 0, \tag{5.9}$$

where $w_j^+$ and $w_j^-$ are positive and negative parts of $w_j$ respectively, and at most one of them is non-zero. One can solve (5.7) - (5.9) using standard software packages, but the computational cost tends to be high as a generic quadratic programming algorithm would not take into account the special structure of the elastic-net rank SVM. In this subsection, we develop an efficient algorithm for solving the elastic-net rank SVM based on the bundle method and the order statistics tree data structure.

Note the criterion in (5.6) can be written as:

$$J(w) = L(w) + P(w), \tag{5.10}$$

where

$$L(w) = \sum_i \sum_{r_{ik} > r_{ik'}} (1 - w^T x_{ik} + w^T x_{ik'})_+, \tag{5.11}$$

and

$$P(w) = \lambda_1 \sum_{j=1}^{p} |w_j| + \lambda_2 \sum_{j=1} w_j^2. \tag{5.12}$$

The bundle method is a general approach for solving optimization problems of format (5.10), where $L(w)$ and $P(w)$ are convex and non-negative, and it is especially efficient when $P(w)$ is quadratic in $w$, such as $P(w) = \lambda w^T w$ (Le et al., 2008, Teo et al., 2010). The bundle method provides an iterative algorithm; the essence of the algorithm is to construct a piecewise linear lower bound approximation of $L(w)$ at each iteration and solve an approximate optimization problem of (5.10). Specifically, suppose at iteration $t$, the "optimizer" from the previous iteration is $w^{(t-1)}$, then using the first order Taylor expansion and convexity of $L(w)$, we have

$$L(w) \;\geq\; L(w^{(t-1)}) + \nabla L(w^{(t-1)})^T (w - w^{(t-1)}) \tag{5.13}$$

$$\;=\; a_t^T w + b_t, \tag{5.14}$$

where $a_t = \nabla L(w^{(t-1)})$ is any sub-gradient of $L(w)$ at $w^{(t-1)}$, $b_t = L(w^{(t-1)}) - a_t^T w^{(t-1)}$, and $a_t^T w + b_t = 0$ is often referred as a cutting plane. Then the lower bound of $L(w)$ is constructed as

$$L_t(w) = \max_{s=1,\ldots,t} (a_s^T w + b_s), \tag{5.15}$$

where we pool all cutting planes from earlier iterations to construct a (hopefully tight) lower bound of $L(w)$. Further, $w^{(t)}$ can be updated via the following optimization problem:

$$w^{(t)} = \arg\min_w L_t(w) + P(w). \tag{5.16}$$

Note if $P(w)$ is quadratic in $w$, since $L_t(w)$ is piecewise linear (with $t$ pieces), (5.16) can be formulated as a quadratic programming problem, and it can be solved efficiently if $t$ is not large (e.g. on the order of 10 or 100) (Le et al., 2008)

In the setting of the elastic-net rank SVM, recall $P(w)$, which contains both $\ell_1$-norm and $\ell_2$-norm of $w$. If we solve (5.16) directly using quadratic programming, the $\ell_1$-norm penalty $\|w\|_1$ will add $p$ linear constraints to the optimization problem, and if $p$ is large, these extra linear constrains will render the computation inefficient. To address this difficulty, we consider a quadratic approximation of $\|w\|_1$ and use that to replace the $\ell_1$-norm penalty (Fan and Li, 2001); specifically, we define:

$$P_t(w) = \lambda_1 \sum_j \frac{w_j^2}{2|w_j^{(t-1)}|} + \lambda_2 \sum_j^p w_j^2, \tag{5.17}$$

and propose to update $w_{(t)}$ using the following criterion:

$$w^{(t)} = \arg\min_w L_t(w) + P_t(w). \tag{5.18}$$

Note now $L_t(w)$ is piecewise linear in $w$ and $P_t(w)$ is quadratic, thus (5.18) can be solved efficiently. Overall, the algorithm is presented in Algorithm 4.

Finally, note that Algorithm 4 involves the computation of $L_t(w^{(t)})$ in each iteration, and direct computation of $L(w)$ requires going through all pairs of "documents" associated with a "query", which would lead to the $O(nm^2)$ computational complexity and is inefficient. Here we first simplify the formula for $L(w)$, as well as its sub-gradient, then propose to use the order statistics tree to reduce the computational cost. Specifically, note $L(w)$ (5.11) can also be written as follows:

$$L(w) = \sum_i \sum_k \left[ (c_{ik}^+(w) - c_{ik}^-(w)) w^T x_{ik} + c_{ik}^+(w) \right], \tag{5.19}$$

**Algorithm 4:** A bundle algorithm for the elastic-net rank SVM

**Require:** $w^{(0)}$, $\epsilon \geq 0$
$t \leftarrow 0$
**while** $\epsilon_t > \epsilon$ **do**
  $t \leftarrow t + 1$
  $a_t \leftarrow$ a sub-gradient of $L(w)$ at $w^{(t-1)}$
  $b_t \leftarrow L(w^{(t-1)}) - a_t^T w^{(t-1)}$
  Update $L_t(w)$ by adding the new cutting plane $a_t^T w + b_t = 0$
  Update $P_t(w)$ according to (5.17)
  $w_{(t)} \leftarrow \text{argmin}_w \, L_t(w) + P_t(w)$
  $\epsilon_t \leftarrow J(w^{(t)}) - J_t(w^{(t)})$
**end while**
**Output:** $w^{(t)}$

where

$$c_{ik}^{+}(w) = \text{Cardinality } \{k' : (r_{ik} > r_{ik'}) \text{ and } (w^T(x_{ik} - x_{ik'} < 1)\}, \tag{5.20}$$

$$c_{ik}^{-}(w) = \text{Cardinality } \{k' : (r_{ik} < r_{ik'}) \text{ and } (w^T(x_{ik'} - x_{ik} < 1)\}. \tag{5.21}$$

Similarly, a subgradient of $L(w)$ can be written as

$$\nabla L(w) = \sum_i \sum_k (c_{ik}^{+}(w) - c_{ik}^{-}(w)) x_{ik}. \tag{5.22}$$

Thus, efficient computation of $L(w)$ and $\nabla L(w)$ reduces to efficient counting of $c_{ik}^{+}(w)$ and $c_{ik}^{-}(w)$. It turns out that the order statistics tree data structure can be used for that purpose. We refer readers to Cormen et al. (2001) and Airola et al. (2011) for details and only summarize the major results here.

For a binary search tree, a node $u$ contains a real valued key$(u)$. The height of a binary search tree is the length of the path from the root node to the lowest leaf node, and the size of a (sub-)tree is the number of elements it contains. The order statistics tree is a binary search tree with the following properties:

- The search property: let $u_1$ be a node in a binary search tree. If $u_2$ is a node in the left subtree of $u_1$, then key$(u_2) \leq$ key$(u_1)$. If $u_2$ is a node in the right subtree of $u_1$, then key$(u_2) \geq$ key$(u_1)$.
- Balance: the height of the tree is $O(log(m))$ after insertions and deletions, where $m$ is the number of nodes in the tree.

**Algorithm 5:** Counting $c_{ik}^{+}(w)$ and $c_{ik}^{-}(w)$ for "query" $i$

**Input:** $(r_{i1}, x_{i1}), \ldots, (r_{im}, x_{im})$, $w$
**Output:** $c_{ik}^{+}(w)$ and $c_{ik}^{-}(w)$ for $k = 1, \ldots, m$
Sort $w^T x_{ik}$ in order: $w^T x_{i\pi(1)} \leq \cdots \leq w^T x_{i\pi(m)}$
Initialize both $c_i^{+}$ and $c_i^{-}$ length $m$ vectors of zeros
Initialize $T$ as an empty tree
Initialize $k = 1$
**for** $k' = 1, 2, \ldots, m$ **do**
  **while** $k \leq m$ and $w^T x_{i\pi(k)} - w^T x_{i\pi(k')} < 1$ **do**
    Insert $r_{i\pi(k)}$ into the tree $T$
    $k = k + 1$
  **end while**
  $c_{i\pi(k')}^{+}(w) =$ count-larger$(T, r_{i\pi(k')})$
**end for**
Initialize $T$ as an empty tree
Initialize $k = m$
**for** $k' = m, m-1, \ldots, 1$ **do**
  **while** $k \geq 1$ and $w^T x_{i\pi(k')} - w^T x_{i\pi(k)} < 1$ **do**
    Insert $r_{i\pi(k)}$ into the tree $T$
    $k = k - 1$
  **end while**
  $c_{i\pi(k')}^{-}(w) =$ count-smaller$(T, r_{i\pi(k')})$
**end for**

- The size information is stored for each node:

$$\text{size}(u) = \text{size}(\text{left}(u)) + \text{size}(\text{right}(u)) + 1$$

This data structure allows efficient counting of $c_{ik}^{+}(w)$ and $c_{ik}^{-}(w)$ for each "query" $i$. Specifically, the algorithm proceeds as Algorithm 5 (Airola et al., 2011), where the computational cost is $O(nm\log(m))$, rather than $O(nm^2)$.

### 5.2.2 Tuning parameter selection

The elastic-net rank SVM contains two tuning parameters, one for the $\ell_1$-norm penalty, the other for the $\ell_2$-norm penalty. Similar as in the usual supervised learning setting, e.g. regression or classification, they can be selected using validation or cross-validation. Specifically, for a given pair of $(\lambda_1, \lambda_2)$, one first obtains the estimate of $w$, denoted as $\widehat{w}(\lambda_1, \lambda_2)$, then the performance of $\widehat{w}(\lambda_1, \lambda_2)$ can be evaluated on a separate validation set based on

its ranking accuracy, which can be defined as

$$\sum_i \sum_{r_{ik} > r_{ik'}} \mathbb{I}(\widehat{w}^T x_{ik} > \widehat{w}^T x_{ik'}), \tag{5.23}$$

where the first summation is over the validation set, and one can select the tuning parameters that maximize the ranking accuracy on the validation set. Similarly for cross-validation.

## 5.3 Simulation studies

In this section, we evaluate the performance of the proposed elastic-net rank SVM on synthetic datasets and compare with the standard rank SVM and the $\ell_1$-norm rank SVM.

We consider the setting where the ratings (rankings) are generated according to a linear model, i.e.

$$r_{ik} = \sum_{j=1}^{p} w_j x_{ikj} + \epsilon_{ik}, \quad k = 1, \ldots, m; i = 1, \ldots, n; \tag{5.24}$$

where $n$ is the number of queries, $m$ is the number of instances per query, and $p$ is the number of features. We set the first five elements of $w$ as 1 and the rest as 0. For each query $i$, the feature vectors $x_{ik}, k = 1, \ldots, m$ are generated independently from the multivariate normal distribution with mean 0 and covariance $\Sigma$, i.e. $\mathcal{N}(0, \Sigma)$, where $\Sigma$ will be specified later. The random noise $\epsilon_{ik}, k = 1, \ldots, m$ are generated independently from the distribution $\mathcal{N}(0, \sigma^2)$.

Regarding $\Sigma$, we consider three different structures:

- Identity: features are independent of each other, specifically, $\Sigma = I_{p \times p}$
- Equal correlation: features are correlated, and all pairwise correlations are the same, specifically,

$$\Sigma = \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \cdots & \cdots & \cdots & \cdots \\ \rho & \cdots & \rho & 1 \end{bmatrix},$$

  where we set $\rho = 0.4$ in simulation studies.
- AR1 autocorrelation: features are correlated, and pairwise correlations decay expo-

nentially with respect to the index difference, specifically,

$$\Sigma_{jj'} = \rho^{|j-j'|},$$

where we set $\rho = 0.6$ in simulation studies.

In each simulation study, we split the $n$ queries into the training part and the validation part. We use the training part to estimate $w$ and use the validation part to select the tuning parameters. When using the validation set to select tuning parameters, we compute an empirical rank accuracy (ERA), i.e.

$$\texttt{ERA} = \frac{\sum_{i \in \text{val}} \sum_{k,k'} \mathbb{I}(\widehat{w}^T x_{ik} > \widehat{w}^T x_{ik'}, r_{ik} > r_{ik'})}{n_{\text{val}} \cdot m(m-1)}. \tag{5.25}$$

To evaluate the performance of the final estimate $\widehat{w}$, we consider two measures. The first measure is referred as the Angle, and it is defined as

$$\texttt{Angle} = \arccos\left(\frac{w^T \widehat{w}}{\|w\| \cdot \|\widehat{w}\|}\right), \tag{5.26}$$

which measures the angle between the two vectors, $w$ and $\widehat{w}$. The second measure is referred as the rank accuracy (RA); it is computed on a separate independent test set (also of size $n$ queries) and defined as

$$\texttt{RA} = \frac{\sum_{i=1}^{n} \sum_{k,k'} \mathbb{I}(\widehat{w}^T x_{ik} > \widehat{w}^T x_{ik'}, w^T x_{ik} > w^T x_{ik'})}{n \cdot m(m-1)}. \tag{5.27}$$

### 5.3.1 Effects of $n$ and $m$

We first investigate the effects of $n$ (number of queries) and $m$ (number of instances per query). We fix $p = 100$, and for each structure of the covariance $\Sigma$, we set a value for $\sigma^2$ such that the signal-to-noise ratio is equal to 1. We considered two cases: $n = 200, m = 50$ and $n = 100, m = 100$; in both cases we have $n \times m = 10000$ instances.

We also investigated the effect the split of the $n$ queries into training and validation parts. Specifically, we considered three possible proportions of the validation set, 0.2, 0.4, and 0.6. For example, when $n = 200$ and the proportion is set to 0.2, the 200 queries are randomly split into two sets, one with 160 queries (training) and the other with 40 queries (validation), and so on. The idea is that when the training set is large but the validation set is small, the estimated $\widehat{w}$ is relatively stable but the selection of the tuning parameters will be unstable, vice versa, and we wish to investigate how the size of the training set and the

size of the validation set trade off with each other.

Table 5.3.1 summarizes the results. First, as one can see, the rank accuracies are more than 90% in almost all cases, which demonstrate the validity of the methods. Comparing $n = 100, m = 100$ and $n = 200, m = 50$, they give similar performances in terms of both rank accuracy and angle estimation. This is not surprising as the rank SVM (when $n = 1$) is essentially fitting a rank regression model, and the convergence rate of rank regression is $\sqrt{m}$, thus as long as $n \times m$ does not change, the performance are similar. In terms of the training and the validation split, it seems the performance does not depend much on the ratio of the two, at least within the range we have considered; even when the validation part is as large as 60% of the dataset, the performance of the final estimate did not degrade much. Further, we observe that the performance of standard rank SVM is consistently worse than that of the $\ell_1$-norm rank SVM (as the underlying true model is sparse), and $\ell_1$-norm rank SVM does not perform as well as the elastic-net rank SVM, under all three correlation structures, with the advantage of the elastic-net rank SVM being more prominent when the features are correlated.

Table 5.1: Simulation results under 3 correlation structures. We set $p = 100$ and $\sigma^2$'s are set such that the signal-to-noise ratio is equal to 1. Three methods are compared, the standard rank SVM, the $\ell_1$-norm rank SVM and the elastic-net rank SVM. All results are averages over 50 replications.

| Correlation Structure | Method | Validation Proportion | n=200, m=50 | | n=100, m=100 | |
|---|---|---|---|---|---|---|
| | | | Angle | RA | Angle | RA |
| Identity | $\ell_1$ | 0.2 | 0.213 (0.038) | 0.932 (0.025) | 0.210 (0.017) | 0.934 (0.011) |
| | | 0.4 | 0.210 (0.029) | 0.933 (0.019) | 0.195 (0.022) | 0.938 (0.014) |
| | | 0.6 | 0.231 (0.044) | 0.926 (0.028) | 0.199 (0.024) | 0.937 (0.016) |
| | $\ell_2$ | 0.2 | 0.271 (0.020) | 0.914 (0.012) | 0.273 (0.020) | 0.912 (0.013) |
| | | 0.4 | 0.274 (0.022) | 0.913 (0.014) | 0.262 (0.012) | 0.916 (0.008) |
| | | 0.6 | 0.270 (0.020) | 0.914 (0.012) | 0.265 (0.012) | 0.915 (0.008) |
| | E-net | 0.2 | 0.154 (0.022) | 0.951 (0.014) | 0.125 (0.029) | 0.960 (0.019) |
| | | 0.4 | 0.141 (0.027) | 0.955 (0.017) | 0.116 (0.025) | 0.963 (0.016) |
| | | 0.6 | 0.143 (0.027) | 0.954 (0.017) | 0.112 (0.022) | 0.964 (0.014) |
| Equal | $\ell_1$ | 0.2 | 0.340 (0.068) | 0.925 (0.027) | 0.341 (0.077) | 0.923 (0.037) |
| | | 0.4 | 0.341 (0.067) | 0.925 (0.029) | 0.341 (0.070) | 0.926 (0.027) |
| | | 0.6 | 0.350 (0.087) | 0.916 (0.046) | 0.318 (0.071) | 0.928 (0.028) |
| | $\ell_2$ | 0.2 | 0.493 (0.030) | 0.894 (0.015) | 0.490 (0.032) | 0.892 (0.015) |
| | | 0.4 | 0.496 (0.030) | 0.893 (0.013) | 0.485 (0.030) | 0.896 (0.016) |
| | | 0.6 | 0.495 (0.026) | 0.895 (0.011) | 0.486 (0.030) | 0.895 (0.014) |
| | E-net | 0.2 | 0.272 (0.073) | 0.941 (0.029) | 0.263 (0.065) | 0.941 (0.025) |
| | | 0.4 | 0.261 (0.064) | 0.943 (0.025) | 0.242 (0.058) | 0.943 (0.030) |
| | | 0.6 | 0.238 (0.060) | 0.945 (0.033) | 0.209 (0.035) | 0.951 (0.014) |
| AR1 | $\ell_1$ | 0.2 | 0.339 (0.076) | 0.923 (0.031) | 0.297 (0.110) | 0.934 (0.052) |
| | | 0.4 | 0.316 (0.054) | 0.929 (0.022) | 0.307 (0.085) | 0.931 (0.043) |
| | | 0.6 | 0.314 (0.048) | 0.923(0.027) | 0.286 (0.078) | 0.939 (0.029) |
| | $\ell_2$ | 0.2 | 0.427 (0.047) | 0.901 (0.019) | 0.441 (0.022) | 0.898 (0.008) |
| | | 0.4 | 0.409 (0.035) | 0.906 (0.014) | 0.409 (0.016) | 0.904 (0.009) |
| | | 0.6 | 0.407 (0.025) | 0.905 (0.009) | 0.409 (0.022) | 0.904 (0.009) |
| | E-net | 0.2 | 0.201 (0.060) | 0.951 (0.030) | 0.196 (0.059) | 0.956 (0.030) |
| | | 0.4 | 0.165 (0.043) | 0.960 (0.023) | 0.200 (0.056) | 0.953 (0.032) |
| | | 0.6 | 0.140 (0.023) | 0.967 (0.011) | 0.170 (0.043) | 0.963 (0.020) |

### 5.3.2 Effects of $p$ and signal-to-noise ratio

Next we investigate the effects of $p$ (number of features) and the signal-to-noise ratio. We fix $n = 100, m = 100$, and the validation proportion is set to 0.4. We considered two cases for $p$: $p = 50$ and $p = 200$, and three levels of the signal-to-noise ratio, 0.25, 0.5 and 1.

Table 5.2 summarizes results. It is not surprising to see that as the signal-to-noise ratio decreases, the performances of all method degrade. As $p$ increases (the number of nonzero $w_j$'s is fixed), the performances of the $\ell_1$-norm rank SVM and the elastic-net rank SVM do not change much, while that of the standard rank SVM degrades significantly. Similar as in the previous simulation study, overall we also observe that the elastic-net rank SVM performs better than the $\ell_1$-norm rank SVM and the $\ell_1$-norm rank SVM performs better than the standard rank SVM under all three correlation structures.

Table 5.2: Simulation result under 3 correlation structures. We fix $n = 100, m = 100$. Three methods are compared, the standard rank SVM, the $\ell_1$-norm rank SVM and the elastic-net rank SVM. All results are averages over 50 replications.

| Correlation Structure | Method | SNR | p=50 | | p=200 | |
|---|---|---|---|---|---|---|
| | | | Angle | RA | Angle | RA |
| Identity | $\ell_1$ | 1 | 0.177 (0.028) | 0.943 (0.019) | 0.216 (0.032) | 0.931 (0.021) |
| | | 0.5 | 0.299 (0.041) | 0.905 (0.026) | 0.310 (0.079) | 0.901 (0.051) |
| | | 0.25 | 0.520 (0.064) | 0.834 (0.040) | 0.563 (0.099) | 0.821 (0.063) |
| | $\ell_2$ | 1 | 0.187 (0.010) | 0.940 (0.007) | 0.363 (0.009) | 0.884 (0.006) |
| | | 0.5 | 0.369 (0.046) | 0.881 (0.029) | 0.641 (0.020) | 0.796 (0.012) |
| | | 0.25 | 0.647 (0.034) | 0.793 (0.022) | 0.977 (0.030) | 0.691 (0.018) |
| | E-net | 1 | 0.123 (0.035) | 0.960 (0.023) | 0.112 (0.021) | 0.964 (0.014) |
| | | 0.5 | 0.246 (0.044) | 0.921 (0.028) | 0.204 (0.046) | 0.935 (0.029) |
| | | 0.25 | 0.466 (0.089) | 0.851 (0.058) | 0.440 (0.061) | 0.860 (0.038) |
| Equal | $\ell_1$ | 1 | 0.277 (0.025) | 0.939 (0.022) | 0.336 (0.051) | 0.923 (0.024) |
| | | 0.5 | 0.550 (0.053) | 0.868 (0.062) | 0.507 (0.091) | 0.882 (0.042) |
| | | 0.25 | 0.789 (0.063) | 0.792 (0.086) | 1.059 (0.220) | 0.731 (0.137) |
| | $\ell_2$ | 1 | 0.350 (0.025) | 0.921 (0.012) | 0.636(0.027) | 0.869 (0.011) |
| | | 0.5 | 0.651 (0.053) | 0.863 (0.021) | 0.953 (0.054) | 0.796 (0.031) |
| | | 0.25 | 0.909 (0.063) | 0.794 (0.073) | 1.222 (0.035) | 0.686 (0.063) |
| | E-net | 1 | 0.188 (0.061) | 0.958 (0.026) | 0.256 (0.072) | 0.939 (0.042) |
| | | 0.5 | 0.423 (0.117) | 0.902 (0.071) | 0.408 (0.072) | 0.912 (0.030) |
| | | 0.25 | 0.809 (0.155) | 0.830 (0.066) | 0.870 (0.184) | 0.832 (0.053) |
| AR1 | $\ell_1$ | 1 | 0.313 (0.070) | 0.931 (0.028) | 0.284 (0.044) | 0.929 (0.027) |
| | | 0.5 | 0.493 (0.088) | 0.867 (0.072) | 0.487 (0.081) | 0.881 (0.044) |
| | | 0.25 | 0.673 (0.165) | 0.827 (0.073) | 0.669 (0.231) | 0.828 (0.139) |
| | $\ell_2$ | 1 | 0.301 (0.029) | 0.933 (0.015) | 0.552 (0.018) | 0.871 (0.008) |
| | | 0.5 | 0.506 (0.063) | 0.874 (0.027) | 0.824 (0.057) | 0.794 (0.030) |
| | | 0.25 | 0.851 (0.120) | 0.778 (0.062) | 1.146 (0.061) | 0.680 (0.043) |
| | E-net | 1 | 0.191 (0.031) | 0.959 (0.015) | 0.151 (0.032) | 0.966 (0.013) |
| | | 0.5 | 0.283 (0.049) | 0.926 (0.036) | 0.307 (0.104) | 0.922 (0.071) |
| | | 0.25 | 0.541 (0.146) | 0.865 (0.074) | 0.630 (0.121) | 0.839 (0.079) |

## 5.4 Data example

In this section, we apply the proposed method to a real-world stock selection problem. Specifically, we treat each trading day as a "query", different stocks in a trading day as "documents". The rating (or ranking) of a "document" is the return of the stock on the next trading day. For each trading day, features of a stock are summary statistics of the stock. In particular, we considered 21 features, falling into different categories, such as the correlation between the stock and the market (Beta), the asset change of the company (Value), the price change of the stock (Momentum), the market value of the company (Size), profit and sales of the company (Growth), which have all been considered as important factors impacting the price of a stock.

We obtained daily returns and the 21 daily features of 2958 A-shares stocks listed in the Shanghai Stock Exchange (SHSE) and Shenzhen Stock Exchange (SZSE) in China from January 2013 to August 2017. We wish to build a ranking model, such that when given new features of these stocks, we can use the model to rank them and the rankings are in accordance with the returns of these stocks on the next trading day. To evaluate the performance of the ranking model, we used a strategy that is commonly used in the financial industry when selecting stocks: on each trading day, we long the top 100 stocks based on the rankings given by the ranking model and short the bottom 100, then we compute the returns of these long-short holdings on the next trading day; in this way, we obtain a curve of cumulative returns over time, and it can tell us how well the ranking model selects stocks.

Figures 5.1-5.4 plot the cumulative return curves under this long-short strategy when using individual features to rank the stocks. As one can see, some features already work pretty well by themselves, while some others do not work so well; different features may also work well at different time periods. For example, for a time period, the momentum features may work better than the value features, while at other time periods, the situation is reversed. Further, Figure 5.5 shows the pairwise correlations between the 21 features' daily returns (not cumulative returns) when this long-short strategy is used for each individual feature. As one can see, some features' daily returns are highly (positively) correlated (especially when they come from the same category), some features' returns are weakly correlated, and there are also features that are strongly negatively correlated, indicating they work at different time periods.

In order to evaluate and compare the performances of the three methods, i.e. the standard rank SVM, the $\ell_1$-norm rank SVM, and the elastic-net rank SVM, we divide the data into training (1/1/2013-12/31/2014), validation (1/1/2015-12/31/2015), and testing (1/1/2016-8/3/2017) sets. We use the validation set to choose tuning parameters. The

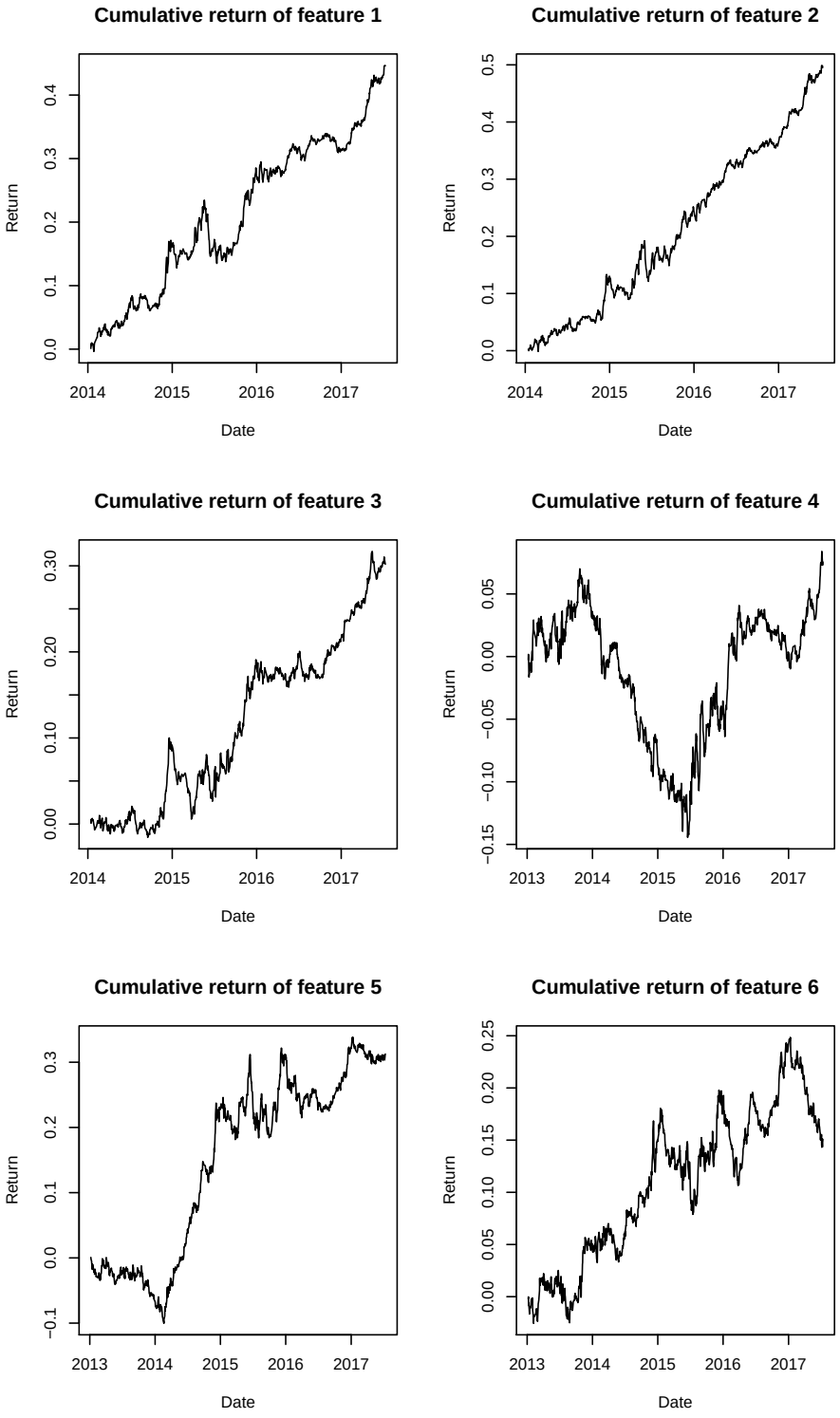



Figure 5.1: Cumulative return curves when long top 100 stocks and short bottom 100 stocks using individual features (features 1-6).

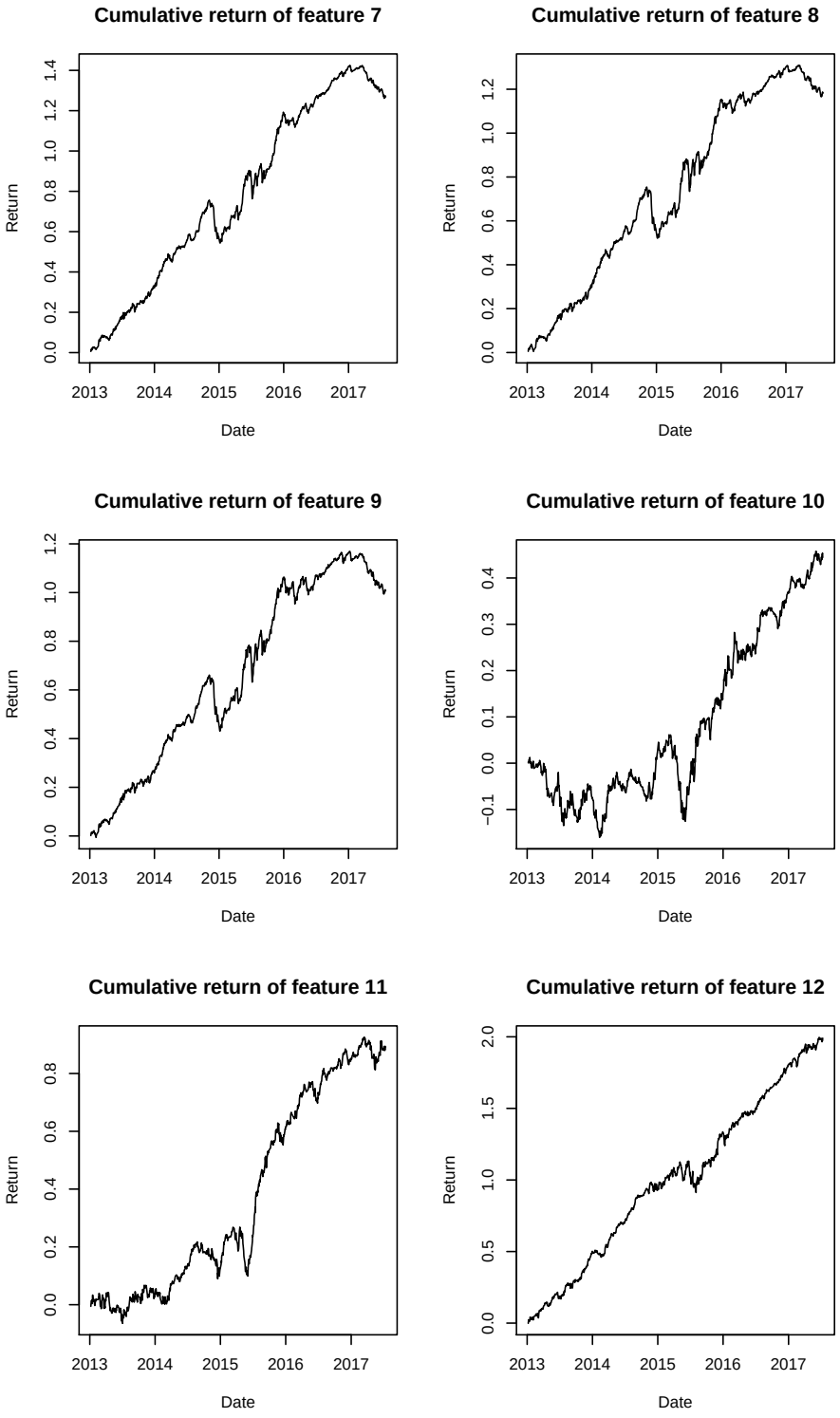



Figure 5.2: Cumulative return curves when long top 100 stocks and short bottom 100 stocks using individual features (features 7-12).

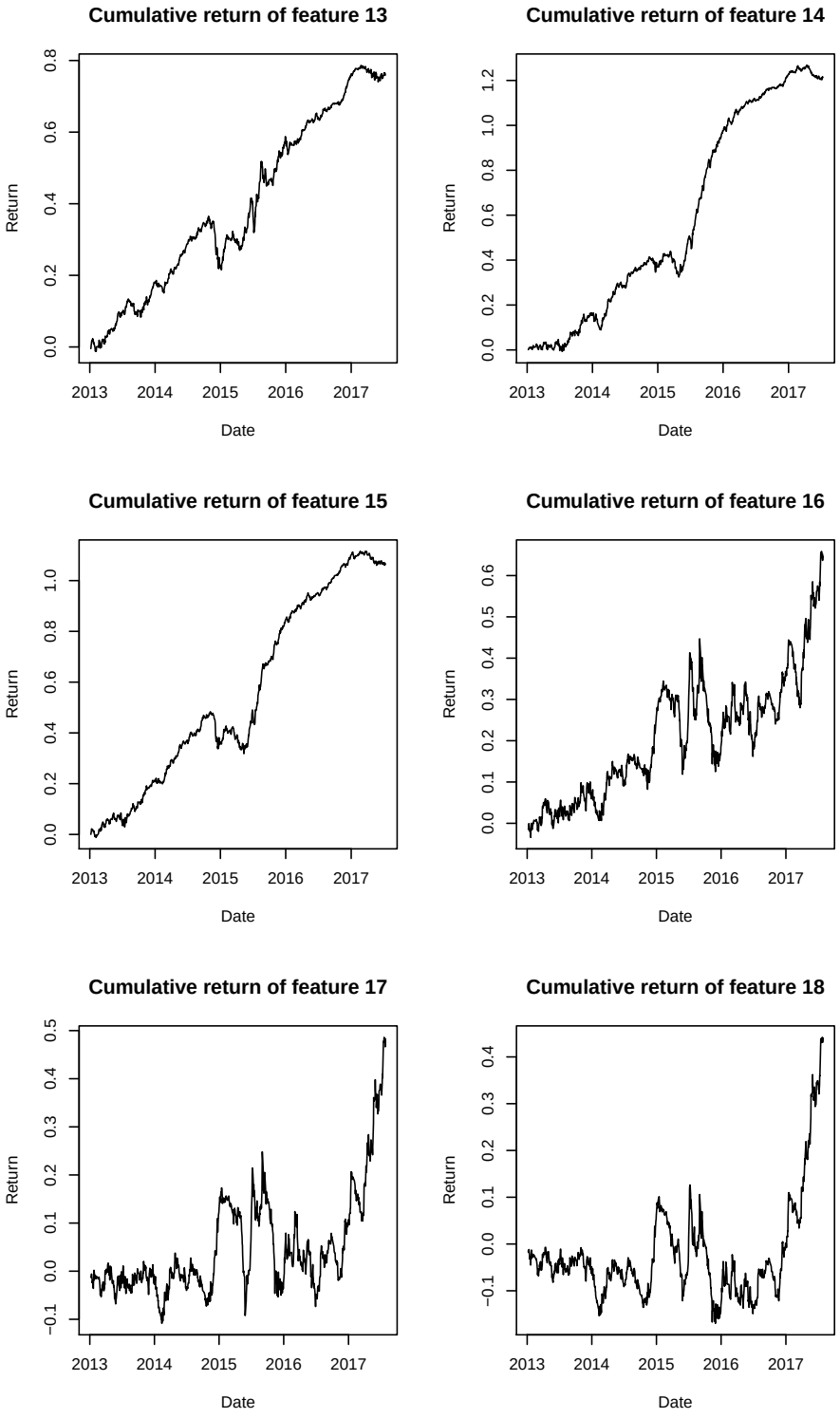



Figure 5.3: Cumulative return curves when long top 100 stocks and short bottom 100 stocks using individual features (features 13-18).

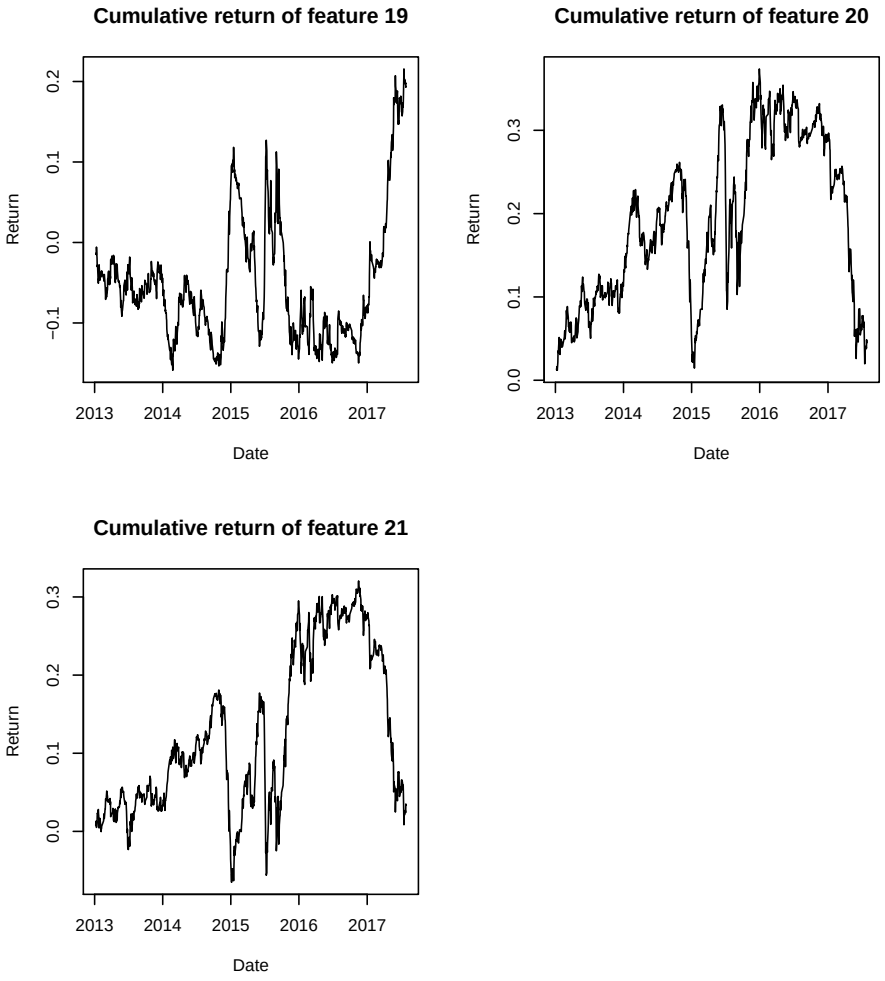



Figure 5.4: Cumulative return curves when long top 100 stocks and short bottom 100 stocks using individual features (features 19-21).

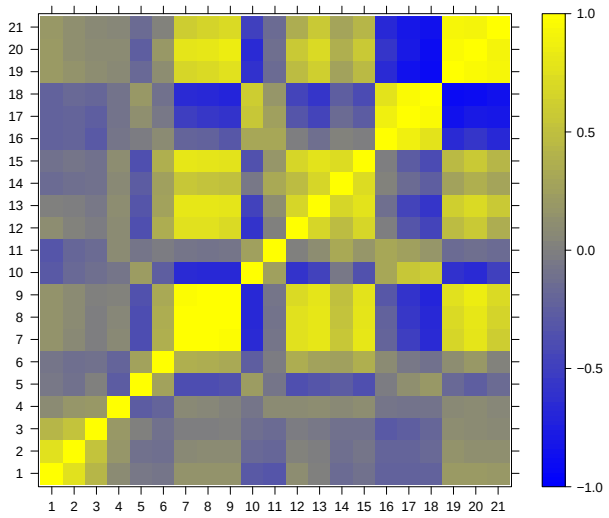



Figure 5.5: Correlations between 21 features' daily returns when the long-short strategy is used for each individual feature

Table 5.3: Cumulative returns and Sharpe ratios (SR) of the three methods in training, validation and testing periods

| | Training | | Validation | | Testing | |
|---|---|---|---|---|---|---|
| | Return | SR | Return | SR | Return | SR |
| $\ell_1$ | 0.698 | 3.63 | 0.958 | 6.14 | 0.328 | 2.26 |
| $\ell_2$ | 0.752 | 3.68 | 0.873 | 5.14 | 0.204 | 0.95 |
| E-net | 0.820 | 4.00 | 1.070 | 6.36 | 0.405 | 2.18 |

final models are then applied to the testing set to compute the cumulative returns under the long-short strategy.

The results are plotted in Figure 5.6 and summarized in Table 5.3. As one can see, all three methods perform reasonably well. Over the period of five years, the long-short strategy makes more than 200% profit with little drawback. Overall, the elastic-net rank SVM performs the best with higher returns than the standard rank SVM and the $\ell_1$-norm rank SVM. Table 5.3 summarizes the cumulative return and the Sharpe ratio for the training, validation and testing periods separately. Again, as one can see, in testing period, the elastic-net rank SVM achieved the highest return and Sharpe ratio.

## 5.5 Summary

In this chapter, we focus on the learning to rank problem with sparse feature selection. In particular, we have extended the standard rank SVM method to the sparse setting, by

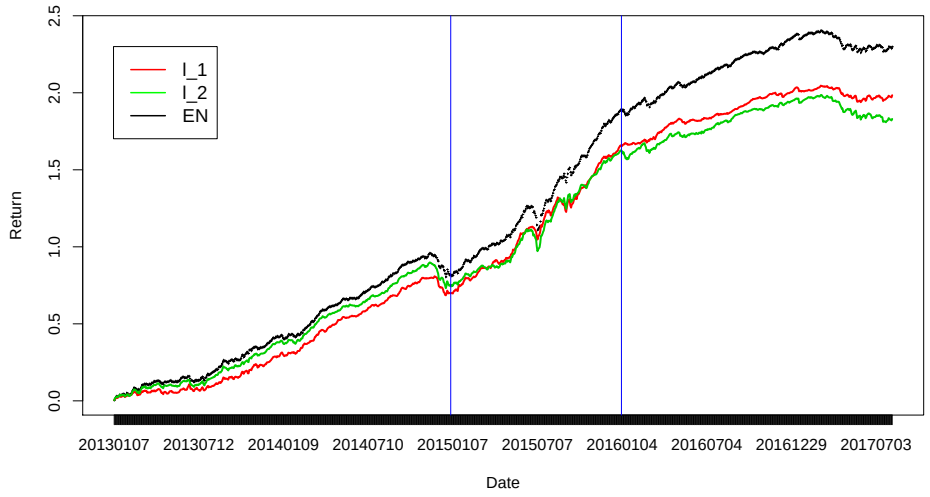



Figure 5.6: Cumulative return curves when long top 100 stocks and short bottom 100 stocks using the fitted model to rank stocks. The two vertical lines indicate the separation of training, validation and testing sets.

applying the lasso and elastic-net penalties. We employed the bundle method and the order statistics tree data structure to reduce the computational complexity. Numerical results indicate that the proposed method works well in both simulation studies and a real-world stock selection problem.

# CHAPTER 6

# Future Work

Looking into future, we list some potential directions that one may continue to pursue.

In Chapter 2, as mentioned at the end of Section 2.3.4, using the maximum likelihood framework to pool information between variables does not help in estimating the mean. Nevertheless, if methods other than the maximum likelihood framework are used, significant improvements may be achievable. For instance, in the single observation setting with independent variables, it is known that the James-Stein estimator (James and Stein, 1961, Stein, 1956) can improve the estimation after introducing pooling between variables. It will be interesting to know if such estimators can be defined by incorporating the general covariance information in Gaussian graphical model problems. Another direction of future work is to allow both the mean and the covariance matrix to vary across observations while being cohesive according to a network structure.

In Chapter 3, since the final estimation is restricted to the moral graph identified in step 1, which is encoded by $\widehat{\Theta}$, it is crucial to understand how the estimation accuracy in step 2 is affected if the moral graph is mis-specified. Among various possible mis-specifications of the moral graph, we are particularly interested in cases where the estimated moral graph in step 1 is not a super-graph of the true moral graph corresponding to the DAG. In other words, we are concerned with cases in which some true edges are missed from the very beginning of step 2, due to them not being identified in the estimated moral graph in step 1. It would be interesting to quantify how such type of mis-specifications affect the accuracy of the estimation in step 2.

# BIBLIOGRAPHY


Odd O Aalen, Kjetil Røysland, Jon Michael Gran, and Bruno Ledergerber. Causality, mediation and time: a dynamic viewpoint. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 175(4):831–861, 2012.

Antti Airola, Tapio Pahikkala, and Tapio Salakoski. Training linear ranking svms in linearithmic time using red–black trees. *Pattern Recognition Letters*, 32(9):1328–1336, 2011.

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

Norbert Binkiewicz, Joshua T Vogelstein, and Karl Rohe. Covariate assisted spectral clustering. *arXiv preprint arXiv:1411.2158*, 2014.

Michael G Bradley and Stephen A Lumpkin. The treasury yield curve as a cointegrated system. *Journal of Financial and Quantitative Analysis*, 27(03):449–463, 1992.

Andries E Brouwer and Willem H Haemers. *Spectra of graphs*. Springer Science & Business Media, 2011.

T Tony Cai, Hongzhe Li, Weidong Liu, Jichun Xie, et al. Covariate-adjusted precision matrix estimation with an application in genetical genomics. *Biometrika*, 100(1):139–156, 2013.

Michael B Cohen, Rasmus Kyng, Gary L Miller, Jakub W Pachocki, Richard Peng, Anup B Rao, and Shen Chen Xu. Solving sdd linear systems in nearly m log 1/2 n time. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 343–352. ACM, 2014.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Introduction to algorithms second edition, 2001.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

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

Alexandre d'Aspremont, Onureena Banerjee, and Laurent El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30 (1):56–66, 2008.

Robert F Engle and Clive WJ Granger. Co-integration and error correction: representation, estimation, and testing. *Econometrica: Journal of the Econometric Society*, pages 251–276, 1987.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

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

Clive WJ Granger. Some properties of time series data and their use in econometric model specification. *Journal of econometrics*, 16(1):121–130, 1981.

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, page asq060, 2011.

Anthony D Hall, Heather M Anderson, and Clive WJ Granger. A cointegration analysis of treasury bill yields. *The Review of Economics and Statistics*, pages 116–126, 1992.

David Hallac, Jure Leskovec, and Stephen Boyd. Network lasso: Clustering and optimization in large graphs. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 387–396. ACM, 2015.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Timothy R Hughes, Matthew J Marton, Allan R Jones, Christopher J Roberts, Roland Stoughton, Christopher D Armour, Holly A Bennett, Ernest Coffey, Hongyue Dai, Yudong D He, et al. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, 2000.

W. James and Charles Stein. Estimation with quadratic loss. In *Proc. 4th Berkeley Sympos. Math. Statist. and Prob., Vol. I*, pages 361–379. Univ. California Press, Berkeley, Calif., 1961.

Pengsheng Ji and Jiashun Jin. Coauthorship and citation networks for statisticians. *arXiv preprint arXiv:1410.2840*, 2014.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006.

Søren Johansen. Statistical analysis of cointegration vectors. *Journal of Economic Dynamics and Control*, 12(2):231–254, 1988.

Markus Kalisch and Peter Buhlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

Steffen L Lauritzen. *Graphical models*. Oxford University Press, 1996.

Quoc V Le, Alex J Smola, and Svn Vishwanathan. Bundle methods for machine learning. In *Advances in neural information processing systems*, pages 1377–1384, 2008.

Lung-fei Lee. Identification and estimation of econometric models with group interactions, contextual factors and fixed effects. *Journal of Econometrics*, 140(2):333–374, 2007.

Wonyul Lee and Yufeng Liu. Simultaneous multiple response regression and inverse covariance matrix estimation via penalized gaussian maximum likelihood. *Journal of multivariate analysis*, 111:241–255, 2012.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

Tianxi Li, Elizaveta Levina, and Ji Zhu. Prediction models for network-linked data. *arXiv preprint arXiv:1602.01192*, 2016.

Yanming Li, Bin Nan, and Ji Zhu. Multivariate sparse group lasso for the multivariate multiple linear regression with an arbitrary group structure. *Biometrics*, 71(2):354–363, 2015.

Jiahe Lin, Sumanta Basu, Moulinath Banerjee, and George Michailidis. Penalized maximum likelihood estimation of multi-layered gaussian graphical models. *J. Mach. Learn. Res.*, 17(1):5097–5147, January 2016. ISSN 1532-4435.

Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

Po-Ling Loh and Peter Buhlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *Journal of Machine Learning Research*, 15(1):3065–3105, 2014.

Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

Charles F Manski. Identification of endogenous social effects: The reflection problem. *The Review of Economic Studies*, 60(3):531–542, 1993.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, pages 1436–1462, 2006.

Nicolai Meinshausen and Peter Buhlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

Karthik Mohan, Palma London, Maryam Fazel, Daniela M Witten, and Su-In Lee. Node-based learning of multiple gaussian graphical models. *Journal of Machine Learning Research*, 15(1):445–488, 2014.

Jonas Peters and Peter Bühlmann. Identifiability of gaussian structural equation models with same error variances. Technical report, 2012.

Peter CB Phillips and Sam Ouliaris. Asymptotic properties of residual based tests for cointegration. *Econometrica: Journal of the Econometric Society*, pages 165–193, 1990.

Bogdan Raducanu and Fadi Dornaika. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 45(6):2432–2444, 2012.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, Bin Yu, et al. High-dimensional covariance estimation by minimizing 1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Adam J Rothman, Peter J Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

Adam J Rothman, Elizaveta Levina, and Ji Zhu. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19(4):947–962, 2010.

Xiaotong Shen, Wei Pan, and Yunzhang Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232, 2012.

Ali Shojaie and George Michailidis. Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika*, 97(3):519–538, 2010.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*, volume 81. MIT press, 2000.

Charles Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley symposium on mathematical statistics and probability*, volume 1, pages 197–206, 1956.

Minh Tang, Daniel L Sussman, and Carey E Priebe. Universally consistent vertex classification for latent positions graphs. *The Annals of Statistics*, 41(3):1406–1430, 2013.

Choon Hui Teo, SVN Vishwanthan, Alex J Smola, and Quoc V Le. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11(Jan):311–365, 2010.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Robert Tibshirani. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385–395, 1997.

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

Ruey S Tsay. *Analysis of financial time series*, volume 543. John Wiley & Sons, 2005.

Sara Van de Geer, Peter Bühlmann, et al. L1-penalized maximum likelihood for sparse directed acyclic graphs. *Annals of Statistics*, 41(2):536–567, 2013.

Elif Vural and Christine Guillemot. Out-of-sample generalizations for supervised manifold learning for classification. *IEEE Transactions on Image Processing*, 25(3):1410–1424, 2016.

Martin J Wainwright. Sharp thresholds for high-dimensional and noisy recovery of sparsity using l1-constrained quadratic programming. *IEEE Transactions on Information Theory*, 2009.

Ines Wilms and Christophe Croux. Forecasting using sparse cointegration. *International Journal of Forecasting*, 32(4):1256–1267, 2016.

Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining*, pages 1151–1156. IEEE, 2013.

Wankou Yang, Changyin Sun, and Lei Zhang. A multi-manifold discriminant analysis method for image feature extraction. *Pattern Recognition*, 44(8):1649–1657, 2011.

Jianxin Yin and Hongzhe Li. A sparse conditional gaussian graphical model for analysis of genetical genomics data. *The annals of applied statistics*, 5(4):2630, 2011.

Jianxin Yin and Hongzhe Li. Adjusting for high-dimensional covariates in sparse precision matrix estimation by $\ell 1$-penalization. *Journal of multivariate analysis*, 116:365–381, 2013.

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

Yiping Yuan, Xiaotong Shen, Wei Pan, and Zizhuo Wang. Constrained likelihood for reconstructing a directed acyclic gaussian graph. *To Be Published*, 2014.

Hua Zhang. Treasury yield curves and cointegration. *Applied Economics*, 25(3):361–367, 1993.

Shuheng Zhou, John Lafferty, and Larry Wasserman. Time varying undirected graphs. *Machine Learning*, 80(2-3):295–319, 2010.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

# APPENDIX A

# Proofs of the Main Results in Chapter 2

## A.1 Proof

We first introduce two additional matrix norms for theoretical discussion. Let $||M||_\infty$ denote its elementwise maximum, which is $||M||_\infty = \max_{ij} |M_{ij}|$, $||M||_{1,1}$ be its columnwise maximum norm, i.e. $||M||_{1,1} = \max_j ||M_{\cdot j}||_1$, and let $||M||_{\infty,\infty}$ denote its rowwise maximum norm, i.e. $||M||_{\infty,\infty} = \max_i ||M_{i\cdot}||_1$.

The following lemma summarizes a few concentration inequalities that we will use.

**Lemma 1** (Concentration of norm of general multivariate Gaussian)**.** *For a Gaussian random vector $x \sim \mathcal{N}(0, \Sigma)$ where $\Sigma \in \mathbb{R}^{p\times p}$, we have*

$$\mathbb{P}(|||x||_2 - \sqrt{\text{tr}(\Sigma)}| > t) \leq 2\exp(-c\frac{t^2}{\phi_{\max}(\Sigma)})$$

*for some constant $c > 0$ and $\phi_{\max}(\Sigma)$ is the largest eigenvalue of $\Sigma$. Furthermore, we have*

$$\mathbb{P}(|||x||_2^2 - \text{tr}(\Sigma)| > t) \leq 2\exp(-c\frac{t}{\phi_{\max}(\Sigma)}).$$

*For $||x||_1$, we have*

$$\mathbb{P}(|||x||_1 - \frac{2}{\pi}\sum_{i=1}^{p}\sqrt{\Sigma_{ii}}| > t) \leq 2\exp(-c\frac{t^2}{p\phi_{\max}(\Sigma)})$$

*Proof of Lemma 1.* The first one is directly from the fact of concentration of Lipchitz function of sub-gaussian random vectors. The second one is true from the definition of sub-exponential random variables. □

*Proof of Proposition 1.* Since the node degrees on a lattice network are almost the same, we expect $\mathcal{L}_s$ to be very close to $\mathcal{L}_n$. In the following proof, we will treat $\mathcal{L}_s$ as $\mathcal{L}_n$. By basic theories of graph spectrum (Brouwer and Haemers, 2011), one can show that the eigenvalues of such networks are (up to a constant scaling)

$$2 - \cos(\frac{i\pi}{\sqrt{n}}) - \cos(\frac{j\pi}{\sqrt{n}}) = 2\sin^2(\frac{i\pi}{2\sqrt{n}}) + 2\sin^2(\frac{j\pi}{2\sqrt{n}}), i, j \in [\sqrt{n}].$$

Set $m$ to be the smallest integer such that $t(m) \geq n^{-1/3}$, then the rest eigenvalues for $n > n - m$ must satisfy

$$2\sin^2(\frac{i\pi}{2\sqrt{n}}) + 2\sin^2(\frac{j\pi}{2\sqrt{n}}) \leq n^{-1/3},$$

which for large enough $n$ is approximately equivalent to

$$\frac{\pi^2}{2}(\frac{i^2}{n} + \frac{j^2}{n}) \leq n^{-1/3}. \tag{A.1}$$

This requires $(i, j)$ to be in the ball with radius $\sqrt{\frac{2}{\pi^2}n^{2/3}}$. The proportion of such pairs is approximately the ratio between the area of the 1/4 ball and that of the square of $[0, \sqrt{n}] \times [0, \sqrt{n}]$, which is $\frac{n^{2/3}}{2\pi n}$. Thus we have

$$m \leq c\frac{n^{2/3}}{2\pi}$$

for some constant $c$. We now prove the second argument. Assume all the inequalities in Assumption 1 hold with equal signs. From (A.1), we have

$$\begin{aligned}
\sum_{i<n} \frac{1}{\tau_i^2} &= \sum_{1\leq i,j\leq\sqrt{n}} \frac{1}{(2\sin^2(\frac{i\pi}{2\sqrt{n}}) + 2\sin^2(\frac{j\pi}{2\sqrt{n}}))^2} \\
&= n \sum_{1\leq i,j\leq\sqrt{n}} \frac{1}{(2\sin^2(\frac{i\pi}{2\sqrt{n}}) + 2\sin^2(\frac{j\pi}{2\sqrt{n}}))^2} \frac{1}{\sqrt{n}} \frac{1}{\sqrt{n}} \\
&\geq n \sum_{1\leq i,j\leq\sqrt{n}} \frac{1}{(2\frac{\pi^2 i^2}{4n} + 2\frac{\pi^2 j^2}{4n})^2} \frac{1}{\sqrt{n}} \frac{1}{\sqrt{n}} \\
&\approx \frac{4n}{\pi^2} \int_{\frac{\pi}{2\sqrt{n}}\leq x,y\leq 1} \frac{1}{(2x^2 + 2y^2)^2} dxdy \\
&\geq \frac{4n}{\pi^2} \int_{(\frac{\sqrt{2}\pi}{2\sqrt{n}})^2\leq x^2+y^2\leq 1} \frac{1}{4(x^2 + y^2)^2} dxdy \\
&= \frac{n}{\pi}(\frac{2n}{\pi^2} - 1),
\end{aligned}$$

in which the integration approximation error is in a much lower order and will not change the magnitude. Therefore, we have

$$\sum_{i<n} \beta_i^2 = n^{-\frac{2(1+\delta)}{3}} \sum_{i<n} \frac{1}{\tau_i^2} \geq c' n^{\frac{4-2\delta}{3}} \geq n$$

for $\delta < 1/2$. Further, when $\delta < 1/2$, there must exist $\beta$ satisfying the two requirements of

Assumption 1 simultaneously since

$$||\mu - P_{\mathbf{1}}\mu||_2^2 = \sum_{i<n} \beta_i^2.$$

□

For analysis, we first write problem (2.2) into an equivalent one so we can have an easier form for theoretical derivation. Recall that $\mathcal{L}_s = U\Lambda U^T$ is the eigen-decomposition of $\mathcal{L}_s$ and for any $\mu \in \mathbb{R}^n$, we can write $\mu = U\beta$ where $\beta \in \mathbb{R}^n$. Substituting $\mu$ in (2.2), we obtain an equivalent problem in the form of weighted ridge regression, i.e.

$$\min ||y - U\beta||_2^2 + \alpha \sum_i \tau_i \beta_i^2, \tag{A.2}$$

where $y = x_{\cdot j}$ for each $j$ when we solve for the $j$th variable and the corresponding $U\hat{\beta}$ gives $\hat{\mu}_{\cdot j}$.

We first present a lemma regarding the mean estimation property for a single variable mean vector $\mu$. It gives the estimation bound for a univariate Laplacian smoothing which can be of independent interest, and its proof will be the major component when we establish the error bounds of estimated mean matrix $\hat{M}$.

**Lemma 2** (Single Laplacian smoothing bound)**.** *Under Assumptions 1-3, let* $\alpha = n^{\frac{1+\delta}{3}}$*, the estimated* $\hat{\mu}$ *in each of the problem of* (A.2) *satisfies*

$$||\hat{\beta} - \beta^*||_\infty \leq 1 + c_G\sqrt{4\sigma^2 \log(n - m_G)} n^{-\frac{1+\delta}{3}} \sqrt{m_G} + \sqrt{4\sigma^2 \log m_G} \tag{A.3}$$

*with probability at least* $1 - \exp(-\log(n - m_G)) - \exp(-\log m_G)$*, and*

$$||\hat{\beta} - \beta^*||_1 \leq (1 + 2\sigma)[(n - m_G)\sqrt{m_G} n^{-\frac{1+\delta}{3}} + m_G] \tag{A.4}$$

*with probability at least* $1 - \exp(-(n - m_G)) - \exp(-m_G)$*, and*

$$||\hat{\beta} - \beta^*||_2 \leq \sqrt{(1 + 4\sigma^2)(c_G^2 m_G n^{\frac{1-2\delta}{3}} + 1)} \tag{A.5}$$

*with probability at least* $1 - \exp(-(n - m_G)) - \exp(-m_G)$*. In particular, for each* $\hat{\mu}_{\cdot j}$ *which is the estimate of the* $j$*th variable, we can replace* $\sigma^2$ *by* $\Sigma^*_{jj}$*.*

*Proof of Lemma 2.* We can directly write out the solution of (A.2) as

$$\hat{\beta} = (I + \alpha\Lambda)^{-1}\beta^* + (I + \alpha\Lambda)^{-1}U^T\epsilon = (I + \alpha\Lambda)^{-1}\beta^* + (I + \alpha\Lambda)^{-1}\tilde{\epsilon},$$

where $\tilde{\epsilon} \sim \mathcal{N}(0, \sigma^2 I)$. Therefore we have

$$||\hat{\beta} - \beta^*||_\infty \leq ||(\frac{\alpha\tau_i}{1 + \alpha\tau_i}\beta_i^*)_{i=1}^n||_\infty + ||(\frac{1}{1 + \alpha\tau_i}\tilde{\epsilon}_i)_{i=1}^n||_\infty := ||\mathcal{I}||_\infty + ||\mathcal{I}\mathcal{I}||_\infty,$$

$$\begin{aligned}
\text{where } ||\mathcal{I}||_\infty &\le \max_{i<n} \frac{\alpha}{1+\alpha\tau_i} \max_{i<n} |\tau_i \beta_i^*| \\
&\le \frac{\alpha}{1+\alpha\tau_{n-1}} n^{-\frac{1+\delta}{3}} \text{ (by Assumption 1 )} \\
&\le \alpha n^{-\frac{1+\delta}{3}} = 1. \qquad \text{(A.6)}
\end{aligned}$$

On the other hand, we can decompose $\mathcal{I}\mathcal{I}$ into two parts: the first $n - m_G$ elements and the remaining $m_G$ elements. Then for the first $n - m_G$ elements, we have

$$\begin{aligned}
||\mathcal{I}\mathcal{I}_{1:n-m_G}||_\infty &\le \max_{i\le n-m_G} \frac{1}{1+\alpha\tau_i} \max_{i\le n-m_G} |\tilde{\epsilon}_i| = \frac{1}{1+\alpha t(m_G)} \max_{i\le n-m_G} |\tilde{\epsilon}_i| \text{ (by Assumption 2 )} \\
&\le \frac{1}{1+t(m_G)n^{\frac{1+\delta}{3}}} \max_{i\le n-m_G} |\tilde{\epsilon}_i| \le \frac{\sqrt{4\sigma^2 \log(n-m_G)}}{t(m_G)n^{\frac{1+\delta}{3}}} \\
&\le c_G \sqrt{4\sigma^2 \log(n-m_G)} n^{-\frac{1+\delta}{3}} \sqrt{m_G}, \qquad \text{(A.7)}
\end{aligned}$$

with probability at least $1 - \exp(-\log(n - m_G))$. For the rest part, with probability at least $1 - \exp(-\log(m_G))$, we have

$$||\mathcal{I}\mathcal{I}_{n-m_G+1:n}||_\infty \le \sqrt{4\sigma^2 \log m_G}. \qquad \text{(A.8)}$$

This completes the proof for (A.3).

For the $L_1$ norm, we have

$$\begin{aligned}
||\mathcal{I}||_1 = \sum_i \frac{\alpha\tau_i|\beta_i|}{1+\alpha\tau_i} &\le \sum_{i\le n-m_G} |\beta_i| + \sum_{i>n-m_G} \frac{\alpha\tau_i|\beta_i|}{1+\alpha\tau_i} \\
&\le \frac{n-m_G}{t(m_G)} n^{-\frac{1+\delta}{3}} + \sum_{i>n-m_G} \frac{\alpha}{1+\alpha\tau_{n-1}} n^{-\frac{1+\delta}{3}} \\
&\le \frac{n-m_G}{t(m_G)} n^{-\frac{1+\delta}{3}} + \sum_{i>n-m_G} \alpha n^{-\frac{1+\delta}{3}} \\
&= \frac{n-m_G}{t(m_G)} n^{-\frac{1+\delta}{3}} + \sum_{i>n-m_G} 1 \\
&= c_G(n-m_G)\sqrt{m_G} n^{-\frac{1+\delta}{3}} + m_G. \qquad \text{(A.9)}
\end{aligned}$$

$$
\begin{aligned}
||\mathcal{I}\mathcal{I}||_1 &= \sum_{i\le n-m_G} \frac{1}{1+\alpha\tau_i}|\tilde{\epsilon}_i| + \sum_{i>n-m_G} \frac{1}{1+\alpha\tau_i}|\tilde{\epsilon}_i| \\
&\le \sum_{i\le n-m_G} \frac{1}{1+t(m_G)n^{\frac{1+\delta}{3}}}|\tilde{\epsilon}_i| + \sum_{i>n-m_G} |\tilde{\epsilon}_i| \\
&\le \frac{2\sigma(n-m_G)}{t(m_G)n^{\frac{1+\delta}{3}}} + 2\sigma m_G \qquad \text{(A.10)} \\
&\le 2c_G\sigma(n-m_G)\sqrt{m_G}n^{-\frac{1+\delta}{3}} + 2\sigma m_G \qquad \text{(A.11)}
\end{aligned}
$$

with probability at least $1-\exp(-(n-m_G))-\exp(-m_G)$.

For the $L_2$ norm, we have

$$
\begin{aligned}
||\mathcal{I}||_2^2 &= \sum_i \frac{\alpha^2\tau_i^2|\beta_i|^2}{(1+\alpha\tau_i)^2} \le \sum_{i\le n-m_G} |\beta_i|^2 + \sum_{i>n-m_G} \frac{\alpha^2\tau_i^2|\beta_i|^2}{(1+\alpha\tau_i)^2} \\
&\le \frac{n-m_G}{t(m_G)^2}n^{-\frac{2(1+\delta)}{3}} + \sum_{i>n-m_G} (\frac{\alpha}{1+\alpha\tau_{n-1}})^2 n^{-\frac{2(1+\delta)}{3}} \\
&\le \frac{n-m_G}{t(m_G)^2}n^{-\frac{2(1+\delta)}{3}} + \sum_{i>n-m_G} \alpha^2 n^{-\frac{2(1+\delta)}{3}} \\
&= \frac{n-m_G}{t(m_G)^2}n^{-\frac{2(1+\delta)}{3}} + \sum_{i>n-m_G} 1 \\
&= c_G^2(n-m_G)m_G n^{-\frac{2(1+\delta)}{3}} + m_G. \qquad \text{(A.12)}
\end{aligned}
$$

$$
\begin{aligned}
||\mathcal{I}\mathcal{I}||_2^2 &= \sum_{i\le n-m_G} (\frac{1}{1+\alpha\tau_i})^2|\tilde{\epsilon}_i|^2 + \sum_{i>n-m_G} (\frac{1}{1+\alpha\tau_i})^2|\tilde{\epsilon}_i|^2 \\
&\le \sum_{i\le n-m_G} \frac{1}{t(m_G)^2}n^{-\frac{2(1+\delta)}{3}}|\tilde{\epsilon}_i|^2 + \sum_{i>n-m_G} |\tilde{\epsilon}_i|^2 \\
&\le 4c_G^2\sigma^2(n-m_G)m_G n^{-\frac{2(1+\delta)}{3}} + 4\sigma^2 m_G \qquad \text{(A.13)}
\end{aligned}
$$

with probability at least $1-\exp(-(n-m_G))-\exp(-m_G)$ by Bernstein inequality. Combining (A.12) and (A.13) completes the proof.

□

Now we proceed to obtain the error bound across $p$ columns. We can still represent each column of $M^*$ by basis expansion over $U$, which results in an expansion coefficient matrix $B^* = (\beta^*_{\cdot 1}, \beta^*_{\cdot 2}, \cdots, \beta^*_{\cdot p})$ such that $M^* = UB^*$. Denote the estimate of (2.2) to be $\hat{M}$ and define $\hat{B}$ to be the corresponding eigen-expansion coefficient matrix on $U$, such that

$$\hat{B} = U^T\hat{M}.$$

Note $\hat{B}$ can be seen as an estimate of $B^*$. We first state the estimation error bound of $\hat{B}$ in Lemma 3 and then the estimation error bound of $\hat{M}$ is a direct result.

**Lemma 3.** *Under Assumptions 2, 3 and 4, we have*

$$||\hat{B} - B^*||_\infty \leq C\sigma[(c_G\sqrt{\log pn}\sqrt{m_G}n^{-\frac{1+\delta}{3}}) \vee \sqrt{\log(pm_G)}] \tag{A.14}$$

*with probability at least* $1-\exp(-c\log(p(n-m_G)))-\exp(-c'\log(pm_G))$ *for some constant* $C, c'$ *and* $c$*. Moreover, we have*

$$||\hat{B} - B^*||_{1,1} \leq (2\sqrt{2}\sigma+1)[c_G\sqrt{m_G}n^{\frac{2-\delta}{3}} + \sqrt{\log p}m_G] \tag{A.15}$$

*with probability at least* $1-\exp(-cn)-\exp(-Cm_G\log p)$*. In Frobenius norm, we have*

$$||\hat{B} - B^*||_F \leq \sqrt{(1+4\sigma^2)(c_G^2 m_G n^{\frac{1-2\delta}{3}}+1)p} \tag{A.16}$$

*with probability at least* $1-\exp(-p(n-m_G))-\exp(-pm_G)$.

*Proof of Lemma 3.* We first check the elementwise maximum norm. The bound of (A.6) is deterministic so it still holds for all columns of $B$. For each column, the bound of (A.7) needs to be scaled up by $\sqrt{\log p}$ and changed to be

$$c_G\sqrt{4\sigma^2\log p(n-m_G)}n^{-\frac{1+\delta}{3}}\sqrt{m_G},$$

which holds with probability at least $1-\exp(-c\log(p(n-m_G)))$. Finally, the term of (A.8) can be bounded by $\sqrt{C\sigma^2\log(pm_G)}$ with probability at least $1-\exp(-c'\log(pm_G))$ for some constant $C'$, by the Gaussian property. Combining the three parts leads to (A.14) as

$$\begin{aligned}||\hat{B}-B||_\infty &\leq 1+\sqrt{4\sigma^2\log p(n-m_G)}n^{-\frac{1+\delta}{3}}\sqrt{m_G}+\sqrt{C'\sigma^2\log(pm_G)}\\ &\leq C\sigma[(c_G\sqrt{\log p(n-m_G)}n^{-\frac{1+\delta}{3}}\sqrt{m_G})\vee\sqrt{\log(pm_G)}].\end{aligned}$$

For the column-wise maximum norm, note that the bound of (A.9) still holds across all columns, as it is deterministic. The first half of (A.10) is true for all columns with probability at least $1-p\exp(-(n-m_G))$. Since we assume $p < c\log(n)$, this can be bounded below by $1-\exp(-c'n)$ for another constant $c'$. Finally, the second half of (A.10) now has to be scaled up by $\sqrt{2\log p}$ since is has to be controlled across $p$ columns, with probability at least $1-\exp(-Cm_G\log p)$. Combining these three terms we obtain

$$\begin{aligned}||\hat{B}-B||_{1,1} &\leq c_G(n-m_G)\sqrt{m_G}n^{-\frac{1+\delta}{3}}+m_G+2\sigma c_G(n-m_G)\sqrt{m_G}n^{-\frac{1+\delta}{3}}+2\sigma\sqrt{2\log p}m_G\\ &\leq (2\sqrt{2}\sigma+1)[c_G(n-m_G)\sqrt{m_G}n^{-\frac{1+\delta}{3}}+\sqrt{\log p}m_G].\end{aligned}$$

For the Euclidean norm, the deterministic part is the same as we sum across $p$ columns. The random noise level of the $np$ Gaussian random variables $\tilde{\epsilon}_{ij}$ is controlled by first summing across $p$ columns for each row, as they are correlated. Then we further sum up over rows. The magnitude is controlled by Lemma 1. This completes the proof.

□

*Proof of Theorem 1.*

$$||\hat{M} - M||_\infty = ||U(\hat{B} - B)||_\infty \leq \min(||U||_\infty ||\hat{B} - B||_{1,1}, ||U||_{\infty,\infty} ||\hat{B} - B||_\infty).$$

The corollary is a direct result of (A.15) and the fact that $U$ is an orthogonal matrix and $||U||_\infty \leq 1$.

The Frobenius norm error is more straightforward since $||\hat{M} - M^*||_F = ||U(\hat{B} - B^*)||_F = ||\hat{B} - B^*||_F$ as $U$ is an orthonormal matrix.

□

Now we proceed to prove Theorem 2. Let

$$\hat{S} = \frac{1}{n}(X - \hat{M})^T(X - \hat{M})$$

$$S = \frac{1}{n}E^T E$$

We first need a concentration inequality about $\hat{S}$ around $S$ to incorporate the noises introduced in the estimation of $M$, which we will show in the following lemma.

**Lemma 4.** *Under the condition of Theorem 1, we have*

$$||\hat{S} - S||_\infty \leq C \max\Big(\sqrt{\log pn} m_G n^{-\frac{2+2\delta}{3}}, \sqrt{\log pn}\sqrt{\log p} m_G^{3/2} n^{-\frac{4+\delta}{3}}, \sqrt{\log pn}\sqrt{m_G} n^{-\frac{1+\delta}{3}}, \sqrt{\log pn}\sqrt{\log p}\frac{m_G}{n}\Big)$$

*with probability at least* $1 - \exp(-c\log(p(n - m_G))) - \exp(-c'\log(pm_G))$ *for some constant* $c, c'$ *and* $C$ *that only depend on* $N_G, c_G$ *and* $\sigma$.

*Proof of Lemma 4.*

$$\begin{aligned}
\hat{S} - S &= \frac{1}{n}(UB^* + E - U\hat{B})^T(UB^* + E - U\hat{B}) - \frac{1}{n}E^T E \\
&= \frac{1}{n}[(\hat{B} - B^*)^T(\hat{B} - B^*) - E^T U((\hat{B} - B^*)) - ((\hat{B} - B^*))^T U^T E + E^T E] - \frac{1}{n}E^T E \\
&= \frac{1}{n}(\hat{B} - B^*)^T(\hat{B} - B^*) - \frac{1}{n}E^T U(\hat{B} - B^*) - \frac{1}{n}(\hat{B} - B^*)^T U^T E. \qquad \text{(A.17)}
\end{aligned}$$

Due to Lemma 3, we have

$$
\begin{aligned}
||\frac{1}{n}(\hat{B}-B^*)^T(\hat{B}-B^*)||_\infty &\leq \frac{1}{n}||\hat{B}-B^*||_\infty||\hat{B}-B^*||_{1,1} \\
&\leq \frac{C}{n}[(\sqrt{\log pn}\sqrt{m_G}n^{-\frac{1+\delta}{3}}) \vee \sqrt{\log(pm_G)}][\sqrt{m_G}n^{\frac{2-\delta}{3}}+\sqrt{\log p}m_G] \\
&= \frac{C}{n}\max\Big([\sqrt{\log pn}m_G n^{\frac{1-2\delta}{3}}+\sqrt{\log pn}\sqrt{\log p}m_G^{3/2}n^{-\frac{1+\delta}{3}}], \\
&\qquad [\sqrt{\log pm_G}\sqrt{m_G}n^{\frac{2-\delta}{3}}+\sqrt{\log pm_G}\sqrt{\log p}m_G]\Big) \\
&\leq C'\max\Big(\sqrt{\log pn}m_G n^{-\frac{2+2\delta}{3}}, \sqrt{\log pn}\sqrt{\log p}m_G^{3/2}n^{-\frac{4+\delta}{3}}, \\
&\qquad \sqrt{\log pm_G}\sqrt{m_G}n^{-\frac{1+\delta}{3}}, \sqrt{\log pm_G}\sqrt{\log p}\frac{m_G}{n}\Big)
\end{aligned} \tag{A.18}
$$

with probability at least $1-\exp(-c\log(p(n-m_G)))-\exp(-c'\log(pm_G))$.

On the other hand, note that $U^TE=(U_{\cdot}.E_{\cdot j})_{i,j=1}^n$ and $||U_{i\cdot}||_2=1$, thus $(U^TE)_{ij}\sim\mathcal{N}(0,\sigma^2)$. As a result, we have

$$
||U^TE||_\infty \leq \sqrt{2\sigma^2\log(np)}
$$

with probability at least $1-\exp(-c\log(np))$. Therefore the second term and the third term in (A.17) satisfy

$$
\begin{aligned}
||\frac{1}{n}E^TU(\hat{B}-B^*)||_\infty &\leq \frac{1}{n}||U^TE||_\infty||\hat{B}-B^*||_{1,1} \\
&\leq C''\frac{1}{n}\sqrt{\log(np)}[\sqrt{m_G}n^{\frac{2-\delta}{3}}+\sqrt{\log p}m_G] \\
&= C''[\sqrt{\log np}\sqrt{m_G}n^{-\frac{1+\delta}{3}}+\sqrt{\log np}\sqrt{\log p}\frac{m_G}{n}]
\end{aligned} \tag{A.19}
$$

with probability at least $1-\exp(-c\log(p(n-m_G))-\exp(-c'\log(pm_G))-\exp(-c''\log(np))$. Note that both terms in the summation dominate the last two terms in (A.18). Thus substituting (A.18) and (A.19) into (A.17) leads to

$$
\begin{aligned}
||\hat{S}-S||_\infty \leq C'''\max\Big(&\sqrt{\log pn}m_G n^{-\frac{2+2\delta}{3}}, \sqrt{\log pn}\sqrt{\log p}m_G^{3/2}n^{-\frac{4+\delta}{3}}, \\
&\sqrt{\log pn}\sqrt{m_G}n^{-\frac{1+\delta}{3}}, \sqrt{\log pn}\sqrt{\log p}\frac{m_G}{n}\Big)
\end{aligned}
$$

with probability at least $1-\exp(-c\log(p(n-m_G)))-\exp(-c'\log(pm_G))$ for some constant $C'''$, $c$ and $c'$.

□

The theoretical property of the glasso step (2.3) can be obtained in an almost identical way as in Ravikumar et al. (2011), with modifications made due to using $\hat{S}$ as the input instead of $S$. First, we have a modified concentration.

**Lemma 5.** *Assume we have*

$$||\hat{S} - S||_\infty \leq C\bar{\nu}(n, p, m_G)$$

*for some constant $C$ and function $\bar{\nu}(n, p, m_G)$ about $n, p, m_G$. Under Assumption 3, we have*

$$||\hat{S} - \Sigma||_\infty \leq C'\Big( \bar{\nu}(n, p, m_G) \vee \sqrt{\frac{2c \log p}{n}} \Big)$$

*with probability at least $1 - \exp(-c \log p)$ for some constants $C'$ and $c$ that only depend on $\sigma$.*

We now give the error bound for the estimation of (2.3), given an estimate of sample covariance matrix, denoted by $\hat{S}$. Specifically, the result shows that even if $\hat{S}$ is not a consistent estimate of the sample covariance matrix, we can still achieve vanishing errors and sparsistency as long as $\hat{S}$ is reasonably close to $S$.

**Proposition 4.** *Assume $\hat{S}$ satisfies*

$$||\hat{S} - \Sigma||_\infty \leq \nu(n, p) := C \max \Big( \sqrt{\log pn} m_G n^{-\frac{2+2\delta}{3}}, \sqrt{\log pn}\sqrt{\log p} m_G^{3/2} n^{-\frac{4+\delta}{3}}, \sqrt{\log pn}\sqrt{m_G} n^{-\frac{1+\delta}{3}}, \sqrt{\log pn}\sqrt{\log p}\frac{m_G}{n}, \sqrt{\frac{\log p}{n}} \Big) \tag{A.20}$$

*where $C, c$ are positive constants. Under Assumption 5, let $\hat{\Theta}$ be the estimate from minimizing (2.3) with $\lambda = \frac{8}{\rho}\nu(n, p)$. If $n$ is large enough so that*

$$\nu(n, p) < \frac{1}{6(1 + 8/\rho)\psi \max\{\kappa_{\Sigma^*}\kappa_{\Gamma^*}, (1 + 8/\rho)\kappa_{\Sigma^*}^3 \kappa_{\Gamma^*}^2\}},$$

*then we have*

1. *The estimate $\hat{\Theta}$ satisfies*

$$||\hat{\Theta} - \Theta^*||_\infty \leq 2(1 + 8/\rho)\kappa_{\Gamma^*}\nu(n, p).$$

2. *$S(\hat{\Theta}) \subset S(\Theta^*)$ and includes all edges $(i, j)$ such that*

$$\max_{(i,j) \in S_o(\Theta^*)} |\Theta^*_{ij}| > 2(1 + 8/\rho)\kappa_{\Gamma^*}\nu(n, p).$$

*Proof of Proposition 4.* The proof is almost the same as the proof of Theorem 1 in Ravikumar et al. (2011). In particular, we just need to show that the primal-dual witness construction succeeds under the assumption. The choice of $\lambda = \frac{8}{\rho}\nu(n, p)$ ensures $||\hat{S} - \Sigma^*||_\infty \leq \frac{\rho}{8}\lambda$. With the requirement on the sample size, the assumptions of Lemmas 5 and 6 in Ravikumar et al. (2011) hold and we can get that strict dual feasibility holds for the primal-dual witness, which shows the procedure succeeds. Then the first half of the conclusion is a

direct result of Lemma 6 in Ravikumar et al. (2011) and the second conclusion is true by construction of the primal-dual witness procedure.

□

Note that Proposition 4 is deterministic as it is purely based on a fixed $\hat{S}$ and (2.3). It is proved following the *primal-dual witness* strategy in Ravikumar et al. (2011). Using Lemma 4 and Lemma 5 and then applying Proposition 4 conditioning on combining Lemma 4 and Lemma 5 leads to Theorem 2.

*Proof of Theorem 2.* The first three statements are direct results of combining Lemma 4, Lemma 5 and Proposition 4 in appendix. The rest are also similarly shown in Ravikumar et al. (2011). Specifically, the fourth statement is true since there are $s + p$ elements in $\Theta^*$. The last two statements come from the fact that

$$||\hat{\Theta} - \Theta^*|| \leq ||\hat{\Theta} - \Theta^*||_F$$

and

$$||\hat{\Theta} - \Theta^*|| \leq ||\hat{\Theta} - \Theta^*||_{\infty,\infty} \leq \psi||\hat{\Theta} - \Theta^*||_\infty.$$

□

For the proof of Proposition 2, we need a few properties about Kronecker products. Recall that given two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, their Kronecker product is defined to be an $(mp) \times (nq)$ matrix such that

$$A \otimes B = \begin{pmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \cdots & A_{mn}B \end{pmatrix}.$$

Given the matrix $A$, define $\text{vec}(A)$ to be the column vector after stacking all columns of $A$ into one $\text{vec}(A) = (A_{\cdot 1}, A_{\cdot 2}, \cdots, A_{\cdot n})$. Some standard properties about Kronecker product and matrix multiplications include (assuming the dimensions of the matrices are well matched for the operations)

$$\text{vec}(AB) = (I_q \otimes A)\text{vec}(B), A \in \mathbb{R}^{n\times p}, B \in \mathbb{R}^{p\times q} \tag{A.21}$$

$$\text{vec}(B^T \otimes A)\text{vec}(C) = \text{vec}(ACB), A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{p\times q}, C \in \mathbb{R}^{n\times p} \tag{A.22}$$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{A.23}$$

$$\begin{aligned} \text{tr}(ABA^T) &= \text{vec}(A)^T(B \otimes I_n)\text{vec}(A) \\ &= \text{vec}(A^T)^T(I_n \otimes B)\text{vec}(A^T), A \in \mathbb{R}^{n\times p}, B \in \mathbb{R}^{p\times p}. \end{aligned} \tag{A.24}$$

*Proof of Proposition 2.* We will only prove (2.24), since the proof of (2.23) is the same with $\Theta^*$ replaced by $I_p$. The conclusion is actually a direct result from the quadratic solution

after vectoring all the matrices. Specifically, the objective function (2.22) can be written as

$$\begin{aligned}
&\text{tr}(\Theta^*(X-UB)^T(X-UB)) + \alpha\text{tr}(B^T\Lambda B)\\
&= \text{vec}(X-UB)^T(\Theta^*\otimes I_n)\text{vec}(X-UB) + \alpha\text{vec}(B)^T(I_p\otimes\Lambda)\text{vec}(B)\\
&= \text{vec}(UB)^T(\Theta^*\otimes I_n)\text{vec}(UB) - 2\text{vec}(UB)^T(\Theta^*\otimes I_n)\text{vec}(X) + \alpha\text{vec}(B)^T(I_p\otimes\Lambda)\text{vec}(B) + const\\
&= \text{vec}(B)(I_p\otimes U^T)(\Theta^*\otimes I_n)(I_p\otimes U)\text{vec}(B) - 2\text{vec}(X)^T(\Theta^*\otimes I_n)(I_p\otimes U)\text{vec}(B)\\
&\qquad + \alpha\text{vec}(B)^T(I_p\otimes\Lambda)\text{vec}(B) + const\\
&= \text{vec}(B)^T[(\Theta^*\otimes I_n) + \alpha(I_p\otimes\Lambda)]\text{vec}(B) - 2\text{vec}(X)^T(\Theta^*\otimes U)\text{vec}(B) + const.
\end{aligned}$$

The minimizer of the quadratic function satisfies

$$[(\Theta^*\otimes I_n) + \alpha(I_p\otimes\Lambda)]\text{vec}(B) = (\Theta^*\otimes U^T)\text{vec}(X).$$

Substituting the relation $X = UB^* + U$ into the estimating equation gives

$$\begin{aligned}
[(\Theta^*\otimes I_n) + \alpha(I_p\otimes\Lambda)]\text{vec}(B) &= (\Theta^*\otimes U^T)\text{vec}(UB^* + E)\\
&= (\Theta^*\otimes U^T)(I_p\otimes U)\text{vec}(B^*) + (\Theta^*\otimes U^T)\text{vec}(E)\\
&= (\Theta^*\otimes I_n)\text{vec}(B^*) + \text{vec}(U^TE\Theta^*).
\end{aligned}$$

This gives

$$(\Theta^*\otimes I_n)\text{vec}(W) + \alpha(I_p\otimes\Lambda)\text{vec}(W) = -\text{vec}(U^TE\Theta^*).$$

We can then use (A.22) again to get

$$\text{vec}(W\Theta^*) + \alpha\text{vec}(\Lambda W) = -\text{vec}(U^TE\Theta^*).$$

This is equivalent to (2.24) by noticing that $\dot{E} = -U^TE\Theta^*$ satisfies the requirement.

□

Finally, we give the proof for Theorem 3 based on Proposition 2.

*Proof of Theorem 3.* Directly by definition, we have

$$W_{4,ij} = \frac{1}{\Theta^*_{jj} + \alpha\tau_i}(\alpha\tau_i B^*_{ij} + \dot{E}_{ij})$$

and

$$W_{3,ij} = \frac{1}{1 + \alpha\tau_i}(\alpha\tau_i B^*_{ij} + \dot{E}_{ij}).$$

This indicates that for any $i, j$ and arbitrary $\alpha$,

$$\min(1, \min_j \Theta^*_{jj}) \le \frac{W_{3,ij}}{W_{4,ij}} = \frac{\Theta^*_{jj} + \alpha\tau_i}{1 + \alpha\tau_i} \le \max(1, \max_j \Theta^*_{jj}). \tag{A.25}$$

We next show that under the assumption of diagonal dominance of $\Theta^*$, even $W_2$ cannot be much better. For each $j = 1, 2, \cdots, p$, from (2.24) it can be seen that

$$W_2\Theta^*_{\cdot j} + \alpha W_{2,\cdot j} = (\Theta^*_{jj}I + \alpha\Lambda)W_{2,\cdot j} + \Theta^*_{jj}\sum_{i\neq j}\frac{\Theta^*_{ij}}{\Theta^*_{jj}}W_{2,\cdot i} = \alpha\Lambda B_{\cdot j} + \dot{E}_{\cdot j}$$

Therefore, we have

$$W_{2,\cdot j}+(\Theta^*_{jj}I+\alpha\Lambda)^{-1}\Theta^*_{jj}\sum_{i\neq j}\frac{\Theta^*_{ij}}{\Theta^*_{jj}}W_{2,\cdot i} = \alpha(\Theta^*_{jj}I+\alpha\Lambda)^{-1}\Lambda B_{\cdot j}+(\Theta^*_{jj}I+\alpha\Lambda)^{-1}\dot{E}_{\cdot j} = W_{4,\cdot j} \tag{A.26}$$

in which the last equation comes from (2.26). By the triangle inequality, (A.26) leads to

$$||W_{2,\cdot j}||_\infty \leq ||W_{4,\cdot j}||_\infty+||(\Theta^*_{jj}I+\alpha\Lambda)^{-1}\Theta^*_{jj}\sum_{i\neq j}\frac{\Theta^*_{ij}}{\Theta^*_{jj}}W_{2,\cdot i}||_\infty \leq ||W_{4,\cdot j}||_\infty+\sum_{i\neq j}\frac{|\Theta^*_{ij}|}{\Theta^*_{jj}}\max_i||W_{2,\cdot i}||_\infty. \tag{A.27}$$

Taking the maximum over $j$ on both sides, we have

$$||W_2||_\infty \leq ||W_4||_\infty + \rho||W_2||_\infty. \tag{A.28}$$

Similarly by using triangle inequality from the other direction, we can show that $||W_2||_\infty \geq ||W_4||_\infty - \rho||W_2||_\infty$. Thus

$$\frac{||W_4||_\infty}{||W_2||_\infty} \in (1-\rho, 1+\rho).$$

Combining this with (A.25), we obtain

$$(1-\rho)\min(1, \min_j \Theta^*_{jj}) \leq \frac{||W_3||_\infty}{||W_2||_\infty} \leq (1+\rho)\max(1, \max_j \Theta^*_{jj}).$$

Note that (A.27) holds for any vector norm. For example, if we take $L_1$ norm instead, it gives similar bound in $||\cdot||_{1,1}$.

□

## A.2 Simulation model on the lattice network

The observation network used in Section 2.4 is shown in Figure A.1. It is an $\sqrt{n}\times\sqrt{n}$ lattice network where $n = 400$. In nontrivial cohesion setting of the simulation study, we generate the values of $\mu^*_{\cdot j}$ as follows. Given the row index $q$ and column index $r$, we set the value at the intersection node $(q, r)$ by $qr/n$. After generating all values, we center and scale the values to make sure values of all nodes are between 0 and 1. Then the node index $(q, r)$ is transformed into $1, 2, \cdots, n$ and the corresponding values are assigned to $\mu^*_{i,j}, i = 1, 2, \cdots, n$. In this case, we have

$$||\mathcal{L}_s\mu^*_{\cdot j}||_2^2 = 0.97 \text{ and } ||\mu^*_{\cdot j}||_2^2 = 51.5.$$

while

$$||\mu^*_{\cdot j} - P_{\mathbf{1}}\mu^*_{\cdot j}||_2^2 = 21.1 \text{ and } ||P_{\mathbf{1}}\mu^*_{\cdot j}||_2^2 = 30.4.$$

Therefore it is a nontrivial cohesion configuration.

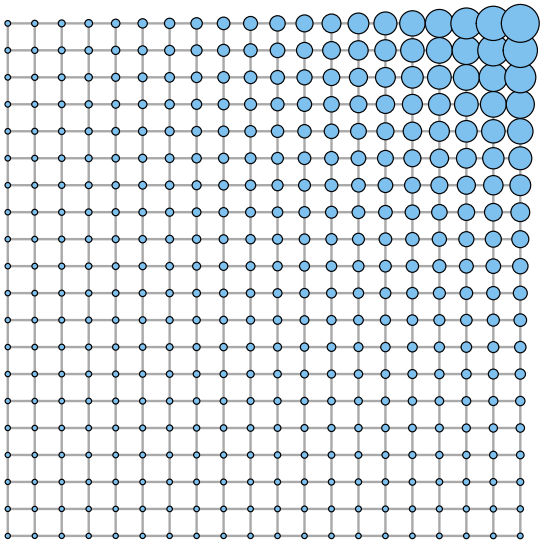


Figure A.1: The $20 \times 20$ grid network in simulation studies. The size of the node indicates the corresponding mean value in the nontrivial cohesion setting.

# APPENDIX B

# Proofs of the Main Results in Chapter 4

## B.1 Update $Q$.

In this section, we give the explicit algorithm for updating $Q$, coming from the alternate update between $A$ and $Q$ which solves the master optimization problem

$$\mathcal{L}(A,Q) = \|Z - ZL^TQA^T\|_F^2 + \lambda\sum_{j=1}^{r}\|(L'Q)_{\cdot j}\|_2^2 + \rho\sum_{i=1}^{p}\sum_{j=1}^{r}|Q_{ij}| + \gamma\sum_{i=1}^{p}||Q_{i\cdot}||_2,$$

and the optimization problem to be considered for updating $Q$ is given by (same as that in (4.13))

$$\widehat{Q}^{(m)} = \underset{Q}{\operatorname{argmin}}\left\{\|Z - ZL^TQ(\widehat{A}^{(m)})'\|_F^2 + \lambda\cdot\text{trace}(Q^TLL^TQ) + \rho\sum_{i=1}^{p}\sum_{j=1}^{r}|Q_{ij}| + \gamma\sum_{i=1}^{p}||Q_{i\cdot}||_2\right\}. \tag{B.1}$$

For fixed orthonormal $A$, let $A_\perp$ be an orthonormal matrix such that $[A : A_\perp]$ is a $p \times p$ orthonormal matrix, which gives

$$\|Z - XQA^T\|_F^2 = \|ZA_\perp\|_F^2 + \|ZA - XQ\|_F^2 = \|ZA_\perp\|_F^2 + \|Y - XQ\|_F^2,$$

where $X := ZL^T$, $Y := ZA$. The first term does not involve $Q$. Therefore, to solve for (B.1), it is equivalent to minimizing

$$f(Q) := \|Y - XQ\|_F^2 + \lambda\cdot\text{trace}(Q^TLL^TQ) + \rho\sum_{i=1}^{p}\sum_{j=1}^{r}|Q_{ij}| + \gamma\sum_{i=1}^{p}||Q_{i\cdot}||_2. \tag{B.2}$$

Consider the first two terms. Let

$$\begin{aligned}\|Y - XQ\|_F^2 + \lambda\cdot\text{trace}(Q^TLL^TQ) &= \text{trace}\left(Q'[\lambda LL^T + X^TX]Q - 2Q^TX^TY + Y^TY\right)\\ &= \text{trace}\left(Q^T\Gamma^T\Gamma Q - 2Q^T\Gamma^T\Gamma^{-T}X^TY + \text{constant}\right)\\ &= \|\Theta - \Gamma Q\|_F^2 + \text{constant},\end{aligned}$$

where

$$\Gamma = (\lambda S_{11} + X^T X)^{1/2} = (L(\lambda I + M)L^T)^{1/2} \qquad \text{and} \qquad \Theta = \Gamma^{-1} X^T Y = \Gamma^{-1} LMA.$$

From the above step, we see that the objective function involves neither $X$ nor $Y$, hence $Z$ is also excluded, which substantiates our earlier statement that the exact decomposition of $M$ will not enter into the optimization procedure. Now for a fixed orthonormal $A$, $\widehat{Q}$ can be equivalently updated by

$$\widehat{Q} = \underset{Q \in \mathbb{R}^{p \times r}}{\operatorname{argmin}} \left\{ \|\Theta - \Gamma Q\|_F^2 + \rho \sum_{i=1}^{p} \sum_{j=1}^{r} |Q_{ij}| + \gamma \sum_{i=1}^{p} ||Q_{i\cdot}||_2 \right\}, \tag{B.3}$$

which is similar to the sparse group lasso setup in Simon et al. (2013), but with the response being multivariate. Note that the lasso penalty has a universal regularization coefficient $\rho$, and the group lasso penalty has a universal regularization coefficient $\gamma$. Moreover, the group-sparse structure is non-overlapping. We use the mixed coordinate descent algorithm proposed by Li et al. (2015) to solve (B.3). To match with the scaling in Li et al. (2015), let $\gamma_n = \gamma/(2n)$ and $\rho_n = \rho/(2n)$. By Theorem 3.1 in Li et al. (2015), if

$$\sqrt{\sum_k (|S_{jk}|/n - \rho_n)_+^2} \leq \gamma_n,$$

then $\widehat{Q}_{j\cdot} = 0$, where $S_{jk} = \Gamma_{\cdot j}^T (\Theta - \Gamma Q_{j=0})_{\cdot k}$, $Q_{j=0}$ is identical to $Q$ except that its $j$th row is set to zero. The updating rule for $Q_{jk}$ is given in Algorithm 6. Specifically, if we only

impose a group-sparse structure on $\beta$, we can set $\rho = 0$ and proceed accordingly.

**Algorithm 6:** Updating rule for $\widehat{Q}$ given fixed $A$

**Input**: Fixed $A$, $\Gamma = (L(\lambda I + M)L^T)^{1/2}$, $\Theta = \Gamma^{-1}X^TY = \Gamma^{-1}LMA$, tuning parameters $\rho$ and $\gamma$, convergence tolerance $\epsilon$.

**Initialization.** Set $Q^{(0)} = 0$.

**while** $|||Q^{(m-1)} - Q^{(m-2)}|||_F > \epsilon$ **do**

  Set $S_{jk}^{(m-1)} = \Gamma_{\cdot j}^T(\Theta - \Gamma\widehat{Q}_{j=0}^{(m-1)})_{\cdot k}$, where $\widehat{Q}_{j=0}^{(m-1)}$ is identical to $\widehat{Q}^{(m-1)}$ except that its $j$th row is set to zero. Set $\widehat{Q}_{j,-k}^{(m-1)}$ to be identical to $\widehat{Q}_{j\cdot}^{(m-1)}$ except that its $k$th coordinate is set to zero. **foreach** $j = 1, \ldots, p$ **do**

    **foreach** $k = 1, \ldots, r$ **do**

      **if** $\|\widehat{Q}_{j,-k}^{(m-1)}\|_2 = 0$ **then**

        update $\widehat{Q}_{jk}$ by $\widehat{Q}_{jk}^{(m)} = \frac{\text{sgn}(S_{jk}^{(m-1)})\left(|S_{jk}^{(m-1)}| - n\gamma_n - n\rho_n\right)_+}{\|\Gamma_{\cdot j}\|_2^2}$;

      **else**

        update $\widehat{Q}_{jk}$ by $\widehat{Q}_{jk}^{(m)} = \frac{\text{sgn}(S_{jk}^{(m-1)})\left(|S_{jk}^{(m-1)}| - n\rho_n\right)_+}{\|\Gamma_{\cdot j}\|_2^2 + n\gamma_n/\|\widehat{Q}_{j\cdot}^{(m-1)}\|_2}$;

      **end**

    **end**

  **end**

**end**

**Output**: Updated $\widehat{Q}$.

## B.2 Characterization of a cointegrated VAR system

In this section, we briefly discuss the constraints on the parameters for a cointegrated VAR system. Without loss of generality, we assume $\{X_t\}$ is a mean-zero process. For $X_t = \Phi_1 X_{t-1} + \ldots \Phi_d X_{t-1} + \epsilon_t$ that is an $I(1)$ process, its characteristic polynomial is given by

$$\mathcal{A}(z) := I - \Phi_1 z - \cdots - \Phi_d z^d,$$

and satisfies the following:

(a) $|\mathcal{A}(z)| = |I_p - \Phi_1 z - \cdots - \Phi_d z^d| = (1 - \lambda_1 z)\ldots(1 - \lambda_d z) = 0$ for $z = 1$.

(b) All other roots that are not 1 are assumed to lie outside the unit circle.

Note that (a) corresponds to that $\Pi$ in the ECM representation being singular (see equation (4.2)), which is automatically satisfied if $\text{rank}(\Pi) = r < p$.

According to the Granger Representation Theorem, suppose

$$\Delta X_t = \alpha\beta^T X_{t-1} + \Phi_1^*\Delta X_{t-1} + \cdots + \Phi_d^*\Delta X_{t-1} + \epsilon_t,$$

where $\epsilon_t$ is white noise for $t = 1, 2 \ldots$. Define

$$C(z) := (1-z)I_p - \alpha\beta^T z - \sum_{i=1}^{d-1} \Phi_i^*(1-z)z^i,$$

and suppose the following conditions hold:

C.1 $\det C(z) = 0 \Rightarrow |z| > 1$ or $z = 1$.

C.2 The number of unit roots $z = 1$ is exactly $p - r$.

C.3 $\alpha$ and $\beta$ are both $p \times r$ matrices with $\text{rank}(\alpha) = \text{rank}(\beta) = r$.

Then $X_t$ has the representation

$$X_t = \Xi \sum_{i=1}^{t} \epsilon_i + \Xi^*(L)\epsilon_t + X_0^*,$$

where[1] $\Xi = \beta_\perp \left[\alpha_\perp^T \left(I_p - \sum_{i=1}^{d-1} \Phi_i^*\right) \beta_\perp\right]^{-1} \alpha_\perp^T$. Note $\Xi^*(L)u_t = \sum_{j=0}^{\infty} \Xi_j^* \epsilon_{t-j}$ is an $I(0)$ process and $X_0^*$ contains the initial values.

Note the rank of $\Xi$ is $(p-r)$, so under conditions (C.1) – (C.3), $X_t$ is driven by $(p-r)$ $I(1)$ components and $r$ $I(0)$ components. The first term is a $p$-dimensional random walk, and after multiplied by $\Xi$ which is of rank $(p-r)$, there are $(p-r)$ stochastic trends driving the system. $\Xi^*$ is determined by the model parameters. Specifically, define (see Lütkepohl, 2005) $\bar{\beta} := \beta(\beta^T\beta)^{-1} \in \mathbb{R}^{p\times r}$ and let

$$Q := \begin{bmatrix} \beta^T \\ \bar{\beta}_\perp^T \end{bmatrix} \in \mathbb{R}^{p\times p}, \qquad \text{so that} \quad Q^{-1} = \left[\bar{\beta} : \beta_\perp\right].$$

Further, let $\mathbf{\Phi}(z) := I_p - \sum_{i=1}^{d-1} \Phi_i^* z_i$, and let

$$B_*(z) := Q\left[\mathbf{\Phi}(z)\bar{\beta}(1-z) - \alpha z : \mathbf{\Phi}(z)\beta_\perp\right], \qquad B(z) = I_p - \sum_{i=1}^{d} B_i z^i := Q^{-1}B_*(z)Q,$$

and

$$\Theta(z) := B(z)^{-1} = \sum_{j=1}^{\infty} \Theta_j z^j,$$

which can be decomposed as

$$\Theta(z) = \Theta(1) + (1-z)\Theta^*(z) := \Theta(1) + \sum_{j=1}^{\infty} \Theta_j^* z^j(1-z),$$

[1]For a general matrix $M \in \mathbb{R}^{\times n}$ with $\text{rank}(M) = n$, we define $M_\perp$ to be its orthogonal complement, which is an $m \times (m-n)$ matrix with $\text{rank}(M_\perp) = m - n$ and $M^T M_\perp = 0$.

by letting $\Theta_0^* = \Theta_0 - \Theta(1)$ and $\Theta_i^* = -\sum_{j=i+1}^{\infty} \Theta_j$, for $i \geq 1$. Finally, $\Xi^*$ is given by

$$\Xi^*(z) = \Theta^*(z) + \bar{\beta}\beta^T B(z)^{-1}.$$

The Granger Representation Theorem outlines the restriction on the model parameters for a VAR system with a specified number of cointegration relations. Specifically, we consider two most relevant cases, with $d = 1$ and $d = 2$ respectively.

When $d = 1$, the VECM representation is given by:

$$\Delta X_t = \alpha\beta^T X_{t-1} + \epsilon_t.$$

Then $C(z) = (1 - z)I_p - \alpha\beta^T z = -(I + \alpha\beta^T)z + I_p$, and $\det C(z) = 0$ is required to have exactly $(p - r)$ unit roots, and $r$ roots that lie outside the unit circle. Equivalently, for the $(p - r)$ unit roots, we have

$$\det(I - (I + \alpha\beta^T)z) = 0,$$

which is automatically satisfied since $\text{rank}(\alpha\beta^T) = r$. For the $r$ roots that lie outside the unit circle, let $\lambda = 1/z$, then

$$\det(I - (I + \alpha\beta^T)z) = 0, \;\; |z| > 1 \quad \Leftrightarrow \quad \det(\lambda I - (I + \alpha\beta^T)) = 0, \;\; |\lambda| < 1.$$

This suggests that the eigenvalues of $I + \alpha\beta^T$ satisfy

$$\lambda_1 = \cdots = \lambda_{p-r} = 1, \qquad |\lambda_i| < 1 \;\text{ for } i = (p - r + 1), \ldots, p.$$

When $d = 2$, the VECM representation is given by:

$$\Delta X_t = \alpha\beta^T X_{t-1} + \Phi_1^* \Delta X_{t-1} + \epsilon_t.$$

Then

$$C(z) = (1 - z)I_p - \alpha\beta^T z - \Phi_1^*(1 - z)z = \Phi_1^* z^2 - (I + \alpha\beta^T + \Phi_1^*)z + I_p.$$

For $\det C(z) = 0$, it's required to have exactly $(p-r)$ unit roots, with the rest lying outside the unit circle. Again, the unit roots will be automatically satisfied given the low rank representation of $\alpha\beta^T$.