Title
Estimation of Graphical Models: Convex Formulations and Algorithms

Author
Li, Jinchao

Publication Date
2015

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Estimation of Graphical Models: Convex Formulations and Algorithms

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Jinchao Li

2015

Abstract of the Dissertation

Estimation of Graphical Models: Convex Formulations and Algorithms

by

Jinchao Li

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2015

Professor Lieven Vandenberghe, Chair

A Gaussian graphical model is a graph representation of conditional independence relations among Gaussian random variables. A fundamental problem in the estimation of Gaussian graphical models is the selection of the graph topology given relatively small amounts of data. This problem is often solved via ℓ_1-regularized maximum likelihood estimation, for which many large-scale convex optimization algorithms have been developed. In this thesis, we consider several extensions of Gaussian graphical models and develop fast algorithms based on convex optimization methods.

As a first extension, we consider the restricted sparse inverse covariance selection problem where the set of zero entries of the inverse covariance matrix is partially known and an ℓ_1-norm penalization is applied to the remaining entries. The proximal Newton method is an attractive algorithm for this problem since the key computations in the algorithm, which include the evaluation of gradient and Hessian of the log-likelihood function, can be implemented efficiently with sparse chordal matrix techniques. We analyze the convergence of the inexact proximal Newton method for the penalized maximum likelihood problem. The convergence analysis applies to a wider class of problems with a self-concordant term in the objective. The numerical results indicate that the method can reach a high accuracy,

even with inexact computation of the proximal Newton steps.

As a second extension, we consider Gaussian graphical models for time series, with focus on the estimation of multiple time series graphical models with similar graph structures or identical graph structure but different edge coefficients. We formulate a joint estimation method for estimating multiple time series graphical models simultaneously, with a group penalty on the edge coefficients for different models. We apply the Douglas-Rachford algorithm to solve the estimation problem for the joint model, and provide model selection methods for choosing parameters. Both synthetic and real data (fMRI brain activity and international stock markets) examples are provided to demonstrate the advantage of the joint estimation method.

The last extension is the generalization of Gaussian graphical models for time series to latent variables. We illustrate the effect of latent variables on the conditional independence structure, and describe a Gaussian graphical model for time series with latent variables. The Douglas-Rachford method is applied to this problem. Simulations with synthetic data demonstrate how the method recovers the graph topology.

The dissertation of Jinchao Li is approved.

Yao Kung

Adnan Youssef Darwiche

Lieven Vandenberghe, Committee Chair

University of California, Los Angeles

2015

To my parents.

Table of Contents

List of Figures

List of Tables

Acknowledgments

This dissertation concludes my Ph.D. research at UCLA, and it indicates the end of my graduate study journey. This would not have been possible without the support of many people.

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Lieven Vandenberghe, for the guidance and mentorship throughout my graduate life. His dedication to work, insightful mind, and conscientious attitude towards work have all influenced me a lot towards rigorous thinking and personal development. I am extremely grateful and so honored to have the opportunity to work and learn from him. This thesis would never have been completed without his guidance. Numerous conversations with him have provided me with not only inspiration on research, but also the pathway towards a professional and successful person, which I would appreciate for my whole life. Besides, his kindness and patience also have enabled me to explore my interest in other fields. For all of these, I cannot thank him enough.

I would also like to thank my Ph.D. committee members Professors Vwani Roychowdhury, Kung Yao, and Adnan Darwiche. I appreciate your precious time to review my work and valuable comments. In particular I want to thank Professor Vwani Roychowdhury for his help of initializing my Ph.D. study at UCLA. I am grateful for his help and guidance.

In addition to my committee members, I wish to thank Dr. Martin S. Andersen for Chompack software. Without Chompack, the work on restricted covariance selection could not have come to fruition. I am also thankful to my colleagues in the same group: Yifan Sun, Daniel O'Connor, Hsiao-Han Chao, Rong Rong, and Cameron Gunn. I appreciate the moments and activities shared together in the past few years.

Last but certainly not least, I have to thank my parents for their support throughout all stages of education in my life. My graduate study journey could not have been possible without their encouragement. I dedicate this thesis to my parents, for their unconditional love, support and encouragement.

Vita

2009	B.S. (Electrical Engineering), Zhejiang University, Hangzhou, Zhejiang, China.
2010	M.S. (Electrical Engineering), UCLA, Los Angeles, California.
2015	M.S. (Statistics), UCLA, Los Angeles, California.

CHAPTER 1

Introduction

A graphical model is a graph representation of the relation among a set of random variables, where nodes are used to represent the variables and edges are used to represent the relations among them. As graphical models have the advantage of combining probability theory with graph theory, they have become a popular statistical tool in areas including machine learning and statistics [FHT01, Bis06, KF09, Mur12]. Based on whether the edges are directed or not, graphical models can be classified as directed graphical models and undirected graphical models. A typical type of directed graphical models is the Bayesian network, where the joint distribution of the nodes is factorized via a directed acyclic graph (DAG) [Dar09]. It can facilitate the combination of prior knowledge and data, and is widely used to learn causal relationships of the variables. It also can readily handle incomplete data and is an effective method to deal with data over-fitting [Hec95] in Bayesian network learning.

Undirected graphical models, compared with Bayesian networks, are undirected and can be cyclic. Despite of the lack of directionality, the potential cyclic feature makes undirected graphical models capable of tackling problems that Bayesian networks cannot address. Thus, they have received wide applications in different areas including machine learning, computer vision, and statistics [RH05]. Throughout the thesis, we focus on undirected graphical models. The background of undirected graphical models will be provided for the purpose of making our topic easier to understand, but only essentials parts are included. For

more details, we refer readers to [Lau96, Jor99, Jor04, WJ08, KF09].

1.1 Static Gaussian graphical models

Among undirected graphical models, one simple example is a Gaussian graphical model (GGM). Nodes in Gaussian graphical models are used to represent components of an n-dimensional multivariate Gaussian random variable $x \sim N(0, \Sigma)$. An edge exists between node x_i and node x_j if and only if x_i and x_j are conditionally dependent, conditioned on all the other variables in the graph.

1.1.1 Covariance selection

In Gaussian graphical models, x_i and x_j are conditionally independent given all the remaining components, if and only if $(\Sigma^{-1})_{ij} = 0$. Therefore, the topology of the graph is equivalent to the zero pattern of the inverse covariance matrix Σ^{-1} (also known as *precision* or *concentration* matrix). Based on the availability of the graph topology as a priori knowledge, we consider three estimation problems: graphical model estimation with a given topology, without a given topology, and for the cases when the topology is partially known.

Covariance selection with a given pattern Given data samples following a Gaussian distribution $N(0, \Sigma)$, the inverse covariance matrix can be estimated by maximum likelihood estimation (MLE),

$$\text{maximize} \qquad \frac{N}{2} \log\det \Sigma^{-1} - \frac{N}{2} \mathbf{tr}(C\Sigma^{-1}), \tag{1.1}$$

where C is the sample covariance matrix. Problem (1.1) has a trivial solution $\Sigma^{-1} = C^{-1}$ if C is nonsingular. However, C^{-1} is not a good estimate for Σ^{-1} if the sample size is small. Even worse, C can be singular. A better estimation can be obtained if the conditional independence structure E ($(i,j) \notin E$ if they are

conditionally independent) is given as a priori knowledge, whereby the estimation problem can be formulated as:

$$\begin{array}{ll} \text{minimize} & -\log\det X + \mathbf{tr}(CX) \\ \text{subject to} & X_{ij} = 0, \quad (i,j) \notin E, \end{array} \tag{1.2}$$

where we have made the variable substitution $X = \Sigma^{-1}$ to make problem (1.2) convex. This is the well-known problem in Gaussian graphical models named *covariance selection* [Dem72]. Problem (1.2) has a closed form solution for certain sparsity patterns, but in general needs to be solved using iterative optimization algorithms.

ℓ_1-norm penalized sparse covariance selection The covariance selection problem has received wide applications, but the sparsity pattern is usually unknown. To address this issue, one approach is to iterate through all possible patterns by exhaustive searches and select the best pattern based on model selection criteria. But this method is computationally intractable, especially for high-dimensional data. A better and more widely used approach is to add an ℓ_1-norm penalty in the objective function [FHT07, YL07, BEd08], *i.e.*, to solve

$$\text{minimize} \quad -\log\det X + \mathbf{tr}(CX) + \lambda \sum_{i \neq j} |X_{ij}|. \tag{1.3}$$

The ℓ_1-norm penalty in (1.3) can be interpreted as imposing a Laplace distribution prior on X, and by imposing the ℓ_1-norm penalty, sparsity is introduced in X, thus yielding a sparse inverse covariance matrix. The ℓ_1-norm penalized problem (1.3) is also well suitable for high dimensional data when the number of variables n is greater than the sample size N. Problem (1.3) is not easy to solve using classical methods due to two reasons. First, the ℓ_1-norm penalty in the objective function is non-differentiable, so methods that require the evaluation of gradient and Hessian can not be applied directly to this problem. Second, the optimization variable is an $n \times n$ matrix. For a graph with n vertices, we need to solve

a problem with n^2 optimization variables. If interior-point methods are applied to this problem, $O(n^6)$ operations are needed, which is too costly for large scale problems. Therefore, this problem has received wide interest from researchers and many algorithms have been proposed in recent years, most of which are based on block coordinate descent methods or first-order splitting methods. Some other algorithms have been proposed to speed up the computation by exploiting the graph structure [DVR08, MH12, HDRB12a, HSDR14]. For readers interested in more details, we refer to [FHT07, BGd08, DGK08, SR09, Lu09, Lu10, LT10, WST10].

Restricted covariance selection In some scenarios, the zero pattern is partially known, and we want to penalize the remaining entries with an ℓ_1-norm penalty to introduce sparsity on the unknown part of the zero pattern. In those cases, we can formulate the problem by combining (1.2) and (1.3),

$$\begin{array}{ll} \text{minimize} & -\log\det X + \mathbf{tr}(CX) + \lambda \sum_{i\neq j} |X_{ij}| \\ \text{subject to} & X_{ij} = 0, \quad (i,j) \notin E. \end{array} \tag{1.4}$$

We refer to (1.4) as *restricted* covariance selection. This problem has been analyzed by only a few authors [LT10, WST10], and known algorithms are limited to relatively small problems (a few thousands variables). As one contribution of this thesis, we reformulate (1.4) by extending the nonzero pattern E to a chordal structure E' by chordal embedding [DVR08], and penalize the edges in $E'\backslash E$ with an indicator function. The proximal Newton method is an attractive algorithm for the reformulated problem, since the key computations involved are the evaluations of gradient and Hessian of the log-likelihood function, which can computed efficiently with algorithms for chordal sparse matrices [VA14, AV15]. More details will be provided in chapter 3.

1.1.2 Covariance selection with latent variables

For many applications throughout science and engineering, one may not have access to observations of all the relevant variables. That is, some variables are hidden or latent. However, according to the definition of conditional independence, whether node x_i and node x_j are conditionally independent is based on the information of all the other variables. If some important variables are missing, the estimation result will be inaccurate. The effect of latent variables on Gaussian graphical models has been analyzed by Chandrasekharan et al. in [CPW10]. To be specific, let x_{o} denote the observable variables and x_{h} denote the hidden variables. Then the covariance matrix of full variables $x_{\mathrm{o}} \cup x_{\mathrm{h}}$ can be denoted as

$$\Sigma = \begin{bmatrix} \Sigma_{\mathrm{oo}} & \Sigma_{\mathrm{oh}} \\ \Sigma_{\mathrm{ho}} & \Sigma_{\mathrm{hh}} \end{bmatrix},$$

and the corresponding precision matrix can be denoted as

$$\Sigma^{-1} = K = \begin{bmatrix} K_{\mathrm{oo}} & K_{\mathrm{oh}} \\ K_{\mathrm{ho}} & K_{\mathrm{hh}} \end{bmatrix}.$$

Using Schur complement, we can get

$$\Sigma_{\mathrm{oo}}^{-1} = K_{\mathrm{oo}} - K_{\mathrm{oh}} K_{\mathrm{hh}}^{-1} K_{\mathrm{ho}}.$$

In this problem, we are interested in estimating K_{oo}, but due to the lack of observations of latent variables, the estimation result is $\Sigma_{\mathrm{oo}}^{-1}$. If we define $S = K_{\mathrm{oo}}$, and $L = K_{\mathrm{oh}} K_{\mathrm{hh}}^{-1} K_{\mathrm{ho}}$, then S is sparse if the corresponding graphical model is sparse, and L is low rank if the number of latent variables is small compared to the number of observable variables. With this variable substitution, the estimation problem can be formulated as:

$$\begin{array}{ll} \text{minimize} & -\log\det(S-L) + \mathbf{tr}\left(C(S-L)\right) + \gamma\|S\|_1 + \lambda\,\mathbf{tr}(L) \\ \text{subject to} & S - L \succ 0, \quad L \succeq 0. \end{array}$$

This model has been extended to autoregressive (AR) time series [ZS14]. The details will be covered in chapter 5, where fast first-order algorithms for the time series latent model will also be discussed.

1.1.3 Joint Gaussian graphical models

In some applications, we are interested in estimating multiple graphical models simultaneously, where the models share certain similarities with each other, but each one has their own uniqueness. For instance, in the area of brain connectivity network estimation [Fri11, QHLC14], the brain connectivity networks are similar for people sharing common features like demographic and health status, but vary based on each individual's own status. In this scenario, estimating the brain network as a single one fails to exploit the fundamental differences among the graphs, while estimating them separately may overlook the commonality of the networks. Thus, a joint graphical model is proposed to estimating multiple graphical models with distinct but related conditions altogether. Two classes of approaches have been discussed to encourage similarity among graphs, *i.e.*, an *edge based* approach and a *node based* approach. An *edge based* approach [KSAX10, GLMZ11, ZW12, DWW14] assumes that the similarity cross graphs is independent for each edge, and a *node based* approach [MCH$^+$12, TLM$^+$14, MLF$^+$14] assumes that the similarities and differences among the graphs are introduced by nodes.

Let us assume there are K distinct graphs, where each one has $N^{(k)}$ observations. If we use $X^{(k)}$ to denote the inverse covariance matrix for condition k and use $C^{(k)}$ to denote the empirical covariance matrix for condition k, the log-likelihood takes the form

$$\ell\left(X^{(1)},\ldots,X^{(K)}\right)=\frac{1}{2}\sum_{k=1}^{K}N^{(k)}\left(\log\det X^{(k)}-\mathbf{tr}(C^{(k)}X^{(k)})\right).$$

Then the penalized maximum log-likelihood problem can be formulated as

$$\underset{X^{(1)},\ldots,X^{(K)}}{\text{minimize}} \quad -\ell\left(X^{(1)}\ \ldots, X^{(K)}\right) + \gamma \sum_{k=1}^{K} \sum_{i \neq j} |X_{ij}^{(k)}| + h(X^{(1)}, \ldots, X^{(K)}),$$

where the first penalty is used to penalize the off-diagonal elements of K precision matrices, and the penalty function $h(X^{(1)}, \ldots, X^{(K)})$ is taken to encourage shared characteristics among $X^{(k)}, k = 1, \ldots, K$. Two commonly used penalty terms for the edge based approach are the *fused graphical lasso* (FGL) and the *group graphical lasso* (GGL).

Fused graphical lasso The fused lasso penalty [TSR+05] takes the difference between pairs of graphs and combines them in the following function:

$$h(X^{(1)}, \ldots, X^{(K)}) = \lambda \sum_{k \neq k'} \sum_{i,j} |X_{ij}^{(k)} - X_{ij}^{(k')}|,$$

where λ is a nonnegative tuning parameter. When λ is large, many edge coefficients will be identical across graphs. Therefore, FGL penalizes the difference across classes aggressively, and it encourages not only structure similarity, but also similar edge values.

Group graphical lasso The group lasso penalty [YL07] penalizes the edges at the same position across graphs altogether while the penalization of edges at different positions are conducted independently. For edges at the same position in different graphs, the ℓ_2-norm penalty is used. This can be characterized as

$$h(X^{(1)}, \ldots, X^{(K)}) = \lambda \sum_{i,j} \sqrt{\sum_{k=1}^{K} (X_{ij}^{(k)})^2}.$$

Compared with FGL, GGL penalizes the edges less aggressively, and focuses on introducing similar structures instead of similar edge values.

1.2 Gaussian graphical models for time series

The Gaussian graphical model can be generalized to stationary Gaussian time series to explain the relationships between different sequences. As in the static Gaussian graphical model, each node denotes one component of time series variables, and each absent edge means the corresponding two variables are conditionally independent.

To extend graphical models, we first need to consider the definition of conditional independence for time series. In the definition given in [Bri01, Dah00], conditional independence is based on the whole temporal sequence of all the other random variables. For a multivariate Gaussian time series sequence $x(t)$, whether $x_i(t)$ and $x_j(t)$ are conditionally independent is based on the correlation of these two components after removing the linear effects of the rest of the time series. In other words, $x_i(t)$ and $x_j(t)$ are conditionally independent if $\epsilon_i(t)$ and $\epsilon_j(t)$ are independent, where $\epsilon_i(t)$ and $\epsilon_i(t)$ mean optimal linear prediction residuals using the remaining components except i and j. It has been shown in [Bri01, Dah00] that two components $x_i(t)$ and $x_j(t)$ are conditionally independent if and only if

$$\left(S(\omega)^{-1}\right)_{ij} = 0, \tag{1.5}$$

for all ω, where $S(\omega)$ means the spectral density. Therefore, the structure of the graph can be estimated by analyzing the zero pattern in the inverse spectral matrix. Based on (1.5), nonparametric methods have been used to analyze Gaussian graphical models for time series in [BJ04]. A more common approach is based on parametric methods. Let us consider a Gaussian time series $x(t)$ following the autoregressive (AR) model,

$$x(t) = -\sum_{k=1}^{p} A_k x(t-k) + w(t), \tag{1.6}$$

where $x(t) \in \mathbf{R}^n$, $w(t) \sim N(0, \Sigma)$ is Gaussian white noise, and p is the order of the autoregressive process. In [SDV10], it has been shown that (1.5) is equivalent

to

$$\left(\sum_{l=0}^{p-k} A_l^T \Sigma^{-1} A_{l+k}\right)_{ij} = 0, \quad k = 0, \ldots, p.$$

Thus, the conditional independence constraints can be expressed in terms of Σ and $A_k, k = 1, \ldots, p$. If we apply penalized conditional maximum likelihood estimation to (1.6) [SDV10, SV10], we can obtain

$$\text{minimize} \quad -2 \log \det B_0 + \mathbf{tr}(CB^T B) + h(B^T B), \tag{1.7}$$

where h is a penalty function for introducing sparsity for the conditional independence structure and

$$B = \begin{bmatrix} B_0 & B_1 & \cdots & B_p \end{bmatrix} = \begin{bmatrix} \Sigma^{-1/2} & \Sigma^{-1/2} A_1 & \cdots & \Sigma^{-1/2} A_p \end{bmatrix}.$$

If we substitute $X = B^T B$, (1.7) becomes a convex problem. The relaxation is exact if the empirical sample matrix C is block Toeplitz. In this thesis, we consider two extensions of (1.7): joint graphical models for time series and latent graphical models for time series. The details will be discussed in chapter 4 and chapter 5.

1.3 Outline of the thesis and contributions

In chapter 2, we provide background on convex optimization algorithms that will be used throughout the thesis. The basics of monotone operators, proximal operators, duality and optimality condition will be provided. We also describe first-order splitting methods including Douglas-Rachford method applied to the primal form, and primal-dual form.

In chapter 3, we consider the cases when the zero pattern is partially known. We start the chapter by introducing Gaussian graphical models and extend the model to the restricted covariance selection problem. Then, we provide background on chordal graphs, and reformulate the restricted covariance selection

problem via chordal extension. Then we describe the inexact proximal Newton algorithm for self-concordant functions, and provide convergence analysis with inexact proximal Newton steps. Lastly, we apply the inexact proximal Newton algorithm to the restricted covariance selection problem.

In chapter 4, we extend the work of joint Gaussian graphical models to time series, where a group penalty is utilized to force edge similarity among different graphs. We apply Spingarn's method to this model and analyze model selection methods based on synthetic data. Experiments based on fMRI brain scanning datasets and international stock markets datasets are provided to demonstrate the performance of joint model.

In chapter 5, we consider the scenarios when latent variables exist in the time series graphical model. We discuss the impact of latent variables on the connectivity of the graph, and provide an extension from latent static graphical models to time series graphical models. The Douglas-Rachford method is applied to this model. Experiments based on synthetic data are provided to illustrate the effectiveness of the model.

CHAPTER 2

Optimization Algorithms

In this chapter, we review first-order splitting algorithms for solving general convex optimization problems with the form:

$$\text{minimize} \quad f(x) + g(Ax), \tag{2.1}$$

where both f and g are closed convex functions. We focus on methods that are suitable for problems of this form when f and g have inexpensive proximal operators (see section 2.2) and A is a structured matrix. We are interested in problems in form (2.1) for two reasons. For one reason, all problems discussed in this thesis, and many problems in statistics, machine learning and signal processsing can be formulated as in (2.1) with simple f, g and A. Another reason is that first-order splitting methods for (2.1) have been extensively studied in recent years. In most problems, $f(x)$ is usually defined to introduce data fidelity such as the mean squares error term in lasso type problems or the log-likelihood function in maximum likelihood estimation problems [Tib96, FHT08, Tib96, Zou06]. $g(x)$ is usually a regularization term such as $\ell_1, \ell_2, \ell_\infty$ norms or combinations. For complicated problems where there is no direct inexpensive operator such as the group lasso penalty [YL06, FHT10, MvdGB08] or the elastic net penalty [ZH05], auxiliary variables can be introduced to transform the problems to (2.1).

We are interested in first-order splitting algorithms for several reasons. First, the solution is not required to reach high accuracy for many large scale problems, so the convergence rate of first-order algorithms is good enough. Second, the computation cost of each iteration in first-order algorithms is cheap since it does not

need to evaluate Hessians, which is expensive especially for large scale problems. Most importantly, splitting methods can be applied to most problems since f and g are not required to be differentiable. For the cases where f or g does not have simple proximal operators, auxiliary variables can be introduced to split problems into a sequence of sub-problems with simple proximal operators.

The rest of this chapter is organized as follows. In section 2.1, we review the fundamentals of duality and optimality conditions. In section 2.2, we provide the background of monotone operators and proximal operators. Then, in section 2.3, we discuss first order splitting methods with emphasis on the Douglas-Rachford splitting method. We also provide examples of proximal operators needed in the thesis, and refer readers to [BPC+11, PB13] for more details.

2.1 Duality and optimality conditions

In general, splitting methods can be applied to the primal form, dual form, and optimality conditions (primal-dual form) of (2.1). Therefore, in this section we give a review of duality and optimality conditions.

Subgradient and subdifferential g is a *subgradient* of a convex function f at $x \in \mathbf{dom}\, f$ if and only if

$$f(y) \geq f(x) + g^T(y - x), \quad \forall y \in \mathbf{dom}\, f.$$

The *subdifferential* $\partial f(x)$ of f at x is defined as the set of all subgradients:

$$\partial f(x) = \{g | f(y) \geq f(x) + g^T(y - x), \forall y \in \mathbf{dom}\, f\}.$$

The set $\partial f(x)$ is a closed convex set [Roc70], which can be verified directly based on the definition of closed convex set.

Conjugate function The conjugate of function f is defined as

$$f^*(y) = \sup_{x \in \mathbf{dom}\, f} \left(y^T x - f(x) \right).$$

For a closed and convex function f, $y \in \partial f(x)$ is equivalent to $x \in \partial f^*(y)$ [Roc70].

Duals and optimality conditions By taking the sub-differential, the optimality condition of (2.1) can be expressed as

$$0 \in \partial f(x) + A^T \partial g(Ax). \tag{2.2}$$

In order to obtain the dual form and optimality condition, we first reformulate (2.1) as

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax = y. \end{array} \tag{2.3}$$

The Lagrangian of (2.3) is

$$L(x, y, z) = f(x) + g(y) + z^T (Ax - y).$$

By minimizing the Lagrangian over (x, y), we can obtain the dual function

$$\begin{array}{rl} \inf\limits_{x,y} L(x, y, z) & = -\sup\limits_{x,y} \left((-A^T z)^T x - f(x) + z^T y - g(y) \right) \\ & = -f^*(-A^T z) - g^*(z), \end{array} \tag{2.4}$$

where the infimum is achieved under the condition

$$\begin{array}{rcl} -A^T z & \in & \partial f(x) \\ z \in \partial g(y) & \Leftrightarrow & y \in g^*(z). \end{array} \tag{2.5}$$

Thus the dual problem can be formulated as

$$\text{maximize} \quad -f^*(-A^T z) - g^*(z), \tag{2.6}$$

with the corresponding optimality condition as

$$0 \in -A \partial f^*(-A^T z) + \partial g^*(z). \tag{2.7}$$

The constraint $Ax = y$ and (2.5) form the primal-dual optimality conditions:

$$0 \in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix}. \tag{2.8}$$

2.2 Monotone operators and proximal operators

In this section, we provide the background of monotone operators and proximal operators. This is useful for introducing the Douglas-Rachford splitting algorithm in section 2.3.

2.2.1 Monotone operators

For a multivalued operator F that maps $x \in \mathbf{R}^n$ to a set $F(x) \subseteq \mathbf{R}^n$, the domain is defined as $\mathbf{dom}\, F = \{x \in \mathbf{R}^n | F(x) \neq \emptyset\}$, and the graph is defined as $\mathbf{gr}(F) = \{(x, y) \in \mathbf{R}^n \times \mathbf{R}^n | x \in \mathbf{dom}\, F, y \in F(x)\}$. The operator F is monotone if and only if

$$(y - \hat{y})^T (x - \hat{x}) \geq 0, \quad \forall x, y \in \mathbf{dom}\, F, y \in F(x), \hat{y} \in F(\hat{x}).$$

F is a maximal monotone operator if and only if its graph is not a strict subset of the graph of another monotone operator. One example of monotone operators is the gradient ∇f for a differentiable convex function f. This can be easily verified as follows. Based on the convexity of f,

$$f(x) \geq f(y) + \nabla f(y)^T (x - y) \quad \text{and} \quad f(y) \geq f(x) + \nabla f(x)^T (y - x).$$

Combining these two inequalities gives

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq 0, \quad \forall x, y \in \mathbf{dom}\, f.$$

Therefore, we can see $\nabla f(x)$ is a monotone operator.

Resolvent The *resolvent* of the operator F is defined as $(I + \lambda F)^{-1}$. If F is a maximal monotone operator, the resolvent $(I + tF)^{-1}(x)$ is single valued mapping defined at all points.

2.2.2 Proximal operators

The proximal operator of a convex function f is defined as

$$\text{prox}_{tf}(x) = \underset{u}{\text{argmin}} \left(f(u) + \frac{1}{2t}\|u - x\|_2^2 \right). \tag{2.9}$$

$\text{prox}_{tf}(x)$ and the subdifferential operator ∂f are related as

$$\text{prox}_{tf} = (I + t\partial f)^{-1}.$$

Thus, we can see that proximal operator $\text{prox}_{tf}(x)$ is the resolvent of the subdifferential operator ∂f with parameter $t > 0$, and the resolvent is a single valued mapping. Some properties of proximal operators are listed below.

Separable sum If f is separable across multiple variables,

$$f(x) = \sum_{i=1}^{n} f_i(x_i),$$

where $x = (x_1, \ldots, x_n)$. Then

$$\text{prox}_{tf}(x) = \begin{bmatrix} \text{prox}_{tf_1}(x_1) \\ \vdots \\ \text{prox}_{tf_n}(x_n) \end{bmatrix}.$$

Moreau decomposition For $t > 0$,

$$x = \text{prox}_{tf}(x) + t\text{prox}_{t^{-1}f^*}(x/t).$$

This rule, known as the Moreau decomposition, shows that the proximal operator of f^* can be computed as easily as the proximal operator of f [PB13]. This means that, if one of the two proximal operators f^* and f is easy to evaluate, we can always evaluate the other one efficiently.

Some examples of f

- Indicator functions. Suppose $f(x)$ is an indicator function of a closed set C: $f(x) = 0$ if $x \in C$, infinity otherwise. Then $\text{prox}_{tf}(x)$ is the Euclidean projection of x on C, which we denote as $P_C(x)$.

  Euclidean ball $C = \{x \mid \|x\|_2 \leq 1\}$:

$$P_C(x) = \begin{cases} x/\|x\|_2, & \|x\|_2 > 1 \\ x, & \|x\|_2 \leq 1. \end{cases}$$

  ℓ_1-norm ball $C = \{x \mid \|x\|_1 \leq 1\}$:

$$P_C(x)_i = \begin{cases} x_i - t, & x_i > t \\ 0, & -t \leq x_i \leq t \\ x_i + t, & x_i < -t, \end{cases}$$

  where $t = 0$ if $\|x\|_1 \leq 1$; otherwise t is the solution of the equation

$$\sum_{i=1}^{n} \max\{|x_i| - t, 0\} = 1.$$

  Positive semidefinite cone $C = \mathbf{S}_+^n$:

$$P_C(X) = \sum_{i=1}^{n} \max\{0, \lambda_i\} q_i q_i^T,$$

  where $\lambda_i, i = 1, \ldots, n$, are the eigenvalues from the eigenvalue decomposition $X = \sum_{i=1}^{n} \lambda_i q_i q_i^T$.

  Affine set $Ax = y$:

$$P_C\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} I \\ A \end{bmatrix} (I + A^T A)^{-1} \begin{bmatrix} I \\ A \end{bmatrix}^T \begin{bmatrix} x \\ y \end{bmatrix}$$

- ℓ_1-norm. For the case $f(x) = \|x\|_1$, prox_f is a soft-thresholding operator:

$$\text{prox}_{tf}(x)_i = \begin{cases} x_i - t, & x_i > t \\ 0, & -t \leq x_i \leq t \\ x_i + t, & x_i < -t. \end{cases}$$

- General norms. For a general norm $f = \|x\|$, the conjugate of f is an indicator function of a unit dual norm ball:

$$f^*(y) = \begin{cases} 0, & \|y\|_* \leq 1 \\ +\infty, & \|y\|_* > 1. \end{cases}$$

By applying Moreau decomposition,

$$\begin{aligned} \text{prox}_{tf}(x) &= x - t\text{prox}_{t^{-1}f^*}(x/t) \\ &= x - tP_C(x/t), \end{aligned}$$

where $P_C(x)$ means projection onto the dual norm ball $C = \{x \mid \|x\|_* \leq 1\}$. If $f(x) = \|x\|_2$, $C = \{x \mid \|x\|_2 \leq 1\}$. If $f(x) = \|x\|_\infty$, $C = \{x \mid \|x\|_1 \leq 1\}$.

- $-\log\det X$. The proximal operator aims to solve

$$\underset{X^+}{\text{minimize}} \quad -\log\det X^+ + \frac{1}{2t}\|X^+ - X\|_F^2.$$

By taking the derivative over X^+, it is equivalent to solving

$$-(X^+)^{-1} + \frac{1}{t}(X^+ - X) = 0. \tag{2.10}$$

As we see X can be decomposed as $X = Q\Lambda Q^T$ by eigenvalue decomposition, with $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and $Q^TQ = QQ^T = I$. Multiplying (2.10) by Q^T on the left and by Q on the right gives

$$\tilde{X}^+ - t(\tilde{X}^+)^{-1} = \Lambda,$$

where $\tilde{X}^+ = Q^TX^+Q$. The corresponding solution can be constructed as

$$\tilde{X}^+_{ii} = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4t}}{2}.$$

Thus the proximal operator can be formulated as $\text{prox}_{tf}(X) = Q\tilde{X}^+Q^T$.

2.3 First-order splitting methods

A closed convex function f is minimized by x^* if $x^* \in \partial f$. This is equivalent to $x^* = (I + t\partial f)^{-1} = \text{prox}_{tf}(x^*)$. If f has a simple proximal operator, the *proximal point method* can be used for minimizing f,

$$x^+ = \text{prox}_{tf}(x).$$

This method is of interest if the proximal operator evaluations are much easier than minimizing f directly. For composite problems with the form $f(x) + g(Ax)$, the proximal operator for the full term is expensive, while each of f and g has an inexpensive proximal operator. This type of problems can be solved by splitting methods, where in each iteration, the proximal operators prox_{tf} and prox_{tg} can be evaluated independently based on a splitting scheme. One well-known splitting method is the Douglas-Rachford splitting method [EB92, CP07]. That is, if we aim to obtain x satisfying $0 \in F(x)$ for an operator F, the Douglas-Rachford splitting method first splits $F(x) = A(x) + B(x)$, where A and B are two maximal monotone operators with inexpensive resolvent evaluations, and then follows the iterations:

$$\begin{aligned} x^+ &= (I + tB)^{-1}(z) \\ y^+ &= (I + tA)^{-1}(2x^+ - z) \\ z^+ &= z + y^+ - x^+. \end{aligned} \tag{2.11}$$

Based on the method of applying the Douglas-Rachford algorithm, we will discuss Spingarn's method in section 2.13 and the primal-dual Douglas-Rachford algorithm in section 2.3.2.

2.3.1 Spingarn's method

Problem (2.1) can be formulated as

$$\text{minimize} \quad f(x) + g(y) + \delta_{\mathcal{V}}(x, y), \tag{2.12}$$

where $\delta_{\mathcal{V}}$ is an indicator function defined as

$$\delta_{\mathcal{V}}(x,y) = \begin{cases} 0, & (x,y) \in \mathcal{V} \\ +\infty, & \text{otherwise,} \end{cases}$$

and $\mathcal{V} = \{(x,y) \mid Ax = y\}$. By applying the Douglas-Rachford splitting scheme (2.11) to (2.12), the algorithm can be formulated as

$$\begin{aligned} x &= \operatorname{prox}_{tf}(u) \\ y &= \operatorname{prox}_{tg}(v) \\ \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} &= \begin{bmatrix} I \\ A \end{bmatrix} (I + A^T A)^{-1} \begin{bmatrix} I \\ A \end{bmatrix}^T \begin{bmatrix} 2x - u \\ 2y - v \end{bmatrix} \\ u^+ &= u + \hat{x} - x \\ v^+ &= v + \hat{y} - y. \end{aligned} \tag{2.13}$$

The third equality in (2.13) is obtained by evaluating the proximal operator of the indicator function $\delta_{\mathcal{V}}(x,y)$, which can be characterized by $P_{\mathcal{V}}(2x-u, 2y-v)$, a projection of $(2x-u, 2y-v)$ onto the subspace $\mathcal{V}$. This update scheme is also known as Spingarn's method. This method is efficient if prox_{tf}, prox_{tg}, and $(I + A^T A)^{-1}$ can be evaluated efficiently.

Residuals When the algorithm (2.13) converges, if we use the notations $x = \operatorname{prox}_{tf}(u)$, $y = \operatorname{prox}_{tg}(v)$, $\tilde{x} = \frac{1}{t}(u - x)$ and $\tilde{y} = \frac{1}{t}(v - y)$, then they satisfy the optimality conditions of (2.12),

$$(x,y) \in \mathcal{V}, \quad (\tilde{x}, \tilde{y}) \in \mathcal{V}^{\perp}, \quad \text{and} \quad (\tilde{x}, \tilde{y}) \in (\partial f(x), \partial g(y)),$$

where $\mathcal{V}^{\perp}$ is the complement of subspace $\mathcal{V}$. Therefore for iteration k, the primal residual can be defined as

$$r_{\mathrm{p}}^{(k)} = P_{\mathcal{V}}\left(x^{(k)}, y^{(k)}\right) - \left(x^{(k)}, y^{(k)}\right)$$

and dual residual can be defined as

$$r_{\mathrm{d}}^{(k)} = -P_{\mathcal{V}}\left(\hat{x}^{(k)}, \hat{y}^{(k)}\right).$$

With this definition of primal and dual residuals, for iteration k,

$$\left(x^{(k)}, y^{(k)}\right)+r_{\mathrm{p}}^{(k)} \in \mathcal{V}, \quad \left(\hat{x}^{(k)}, \hat{y}^{(k)}\right)+r_{\mathrm{d}}^{(k)} \in \mathcal{V}^{\perp}, \quad \left(\hat{x}^{(k)}, \hat{y}^{(k)}\right) \in \left(\partial f(x^{(k)}), \partial g(y^{(k)})\right). \tag{2.14}$$

Moreover, we also define the primal relative residual as

$$\frac{\|r_{\mathrm{p}}^{(k)}\|_2}{\max\{1, \| \left(x^{(k)}, y^{(k)}\right) \|_2\}},$$

and the dual relative residual as

$$\frac{\|r_{\mathrm{d}}^{(k)}\|_2}{\max\{1, \| \left(\hat{x}^{(k)}, \hat{y}^{(k)}\right) \|_2\}}.$$

2.3.2 Primal-dual method

If we apply the Douglas-Rachford splitting scheme (2.11) to the optimality condition (2.8), we can formulate the primal-dual Douglas-Rachford algorithm as:

$$\begin{aligned}
x &= \mathrm{prox}_{tf}(u) \\
z &= \mathrm{prox}_{tg^*}(v) \\
&= v - t\mathrm{prox}_{g/t}(v/t) \\
\begin{bmatrix} w \\ y \end{bmatrix} &= \begin{bmatrix} I & tA^T \\ -tA & I \end{bmatrix}^{-1} \begin{bmatrix} 2x-u \\ 2z-v \end{bmatrix} \\
u^+ &= u + w - x \\
v^+ &= v + y - z.
\end{aligned}$$

The inverse in step 3 can be written as

$$\begin{aligned}
\begin{bmatrix} I & tA^T \\ -tA & I \end{bmatrix}^{-1} &= \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} I \\ tA \end{bmatrix} (I + t^2 A^T A)^{-1} \begin{bmatrix} I \\ -tA \end{bmatrix}^T \\
&= \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -tA^T \\ I \end{bmatrix} (I + t^2 A A^T)^{-1} \begin{bmatrix} tA^T \\ I \end{bmatrix}^T.
\end{aligned}$$

The primal-dual Douglas-Rachford algorithm is efficient if prox_{tf}, $\mathrm{prox}_{g/t}$, and $(I + t^2 A^T A)^{-1}$ (or $(I + t^2 A A^T)^{-1}$) can be evaluated efficiently.

Residual The primal and dual residuals are defined as

$$\begin{bmatrix} r_{\mathrm{d}} \\ r_{\mathrm{p}} \end{bmatrix} = \frac{1}{t} \begin{bmatrix} I & tA^T \\ -tA & I \end{bmatrix} \begin{bmatrix} x - w \\ z - y \end{bmatrix}.$$

To see this, we have

$$\begin{aligned} \begin{bmatrix} r_{\mathrm{d}} \\ r_{\mathrm{p}} \end{bmatrix} &= \frac{1}{t} \begin{bmatrix} I & tA^T \\ -tA & I \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} - \frac{1}{t} \begin{bmatrix} 2x - u \\ 2z - v \end{bmatrix} \\ &= \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} - \frac{1}{t} \begin{bmatrix} x - u \\ z - v \end{bmatrix} \\ &\in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix}. \end{aligned}$$

Thus, if $r_{\mathrm{d}} = 0$ and $r_{\mathrm{p}} = 0$, the optimality condition is satisfied.

CHAPTER 3

Gaussian Graphical Models

The Gaussian graphical model (GGM) [Lau96, Dar09, KF09], also referred to as Gaussian Markov random field (GMRF), is a representation of conditional independence relations of multivariate Gaussian random variables $x \sim N(0, \Sigma)$ as an undirected graph. To be specific, in a Gaussian graphical model, nodes are used to represent random variables, and edges are used to represent conditional dependence between variables. There is no edge between vertices x_i and x_j if and only if x_i and x_j are conditionally independent given the other variables. This can be characterized by the inverse covariance matrix Σ^{-1}: x_i and x_j are conditionally independent if and only if $(\Sigma^{-1})_{ij} = 0$. Therefore identifying the graph topology is equivalent to identifying the sparsity pattern in Σ^{-1}.

Gaussian graphical models have wide applications in various areas including computer graphics, computer vision and machine learning [RH05]. They also serve as a useful tool for analyzing complex biological systems, information extraction, communication networks, among others [BB01, Bis06, FHT01]. As an important statistical tool, Gaussian graphical models have been studied extensively over the past decade, with focus on modeling methods and efficient algorithms [MB06, FHT08, HDRB12b]. We will review them in detail in section 3.1.

In the existing literature about Gaussian graphical models, most research has focused on estimating Gaussian graphical models with a given sparsity pattern [Dem72], or with an ℓ_1-norm penalty in the objective function [FHT01, MB06, FHT08]. In this chapter, we focus on problems where the sparsity pattern is

partially known, and we aim to estimate the sparse graphical model given this prior knowledge. This special problem has been analyzed only in a few papers [Lu10, WST10, LT10], and we refer to this problem as *restricted* Gaussian graphical model throughout the thesis. The restricted Gaussian graphical model is interesting for two reasons. First, given the partial knowledge of the sparsity pattern, the number of parameters is reduced and therefore the estimation accuracy is improved. Second, fast algorithms can be developed by taking advantage of the graph structure. For example, if the nonzero pattern is close to a block-diagonal pattern, a divide-and-conquer procedure can be applied to facilitate the estimation [HDRB12b].

Among different types of sparsity patterns, we are particularly interested in chordal structure. This is because chordal sparse matrix computation techniques enable us to evaluate sparse Cholesky decomposition, projected matrix inverse efficiently. For the log-determinant term $\log\det X$, the gradient, Hessian and inverse Hessian can also be evaluated efficiently [ADV13]. Most importantly, general patterns can also enjoy the fast computations of chordal sparse matrix via chordal extension [DVR08].

The proximal Newton method [WST10, HSDR14] is attractive for estimating restricted Gaussian graphical models. This is because the most expensive computations involved are the evaluations of projected gradient and Hessian, and they can be computed efficiently with chordal sparse matrix techniques. More details will be provided in section 3.3.

The rest of this chapter is organized as follows. In section 3.1, we review the definition of conditional independence and estimation methods for Gaussian graphical models. In section 3.2, we provide the background of chordal structure and introduce a restricted Gaussian graphical model via chordal embedding. In section 3.3, we introduce the proximal Newton algorithm [LSS12, LSS14, ST15], extend the algorithm to self-concordant functions with inexact proximal Newton

steps, and provide the corresponding convergence analysis. Finally, we provide simulation results for the restricted Gaussian graphical model using the inexact proximal Newton algorithm.

3.1 Gaussian graphical models

In this section, we will present the definition of conditional independence, and introduce the restricted Gaussian graphical model.

3.1.1 Conditional independence

For an n-dimensional random variable x, the components x_i and x_j are conditionally independent if and only if

$$p(x_i, x_j|x_k) = p(x_i|x_k)p(x_j|x_k), \quad \text{for } k \neq i, j.$$

If x follows a Gaussian distribution $N(0, \Sigma)$, the conditional independence relation can be characterized by the inverse covariance matrix Σ^{-1} (also known as *precision* matrix or *concentration* matrix) [Dem72]:

$$\left(\Sigma^{-1}\right)_{ij} = 0, \quad \text{iff } x_i \text{ and } x_j \text{ are conditionally independent.} \tag{3.1}$$

To see this, if we define the entry (i, j) of the precision matrix Σ^{-1} as σ_{ij}, the conditional probability follows

$$p(x_i, x_j|x_k, k \neq i, j) \propto \frac{1}{(2\pi)^{n/2}(\det \Sigma)^{1/2}} \exp\{\sigma_{ii}x_i^2 + \sigma_{jj}x_j^2 + 2\sigma_{ij}x_ix_j\}. \tag{3.2}$$

If $\sigma_{ij} = 0$, we can derive $p(x_i, x_j|x_k) = p(x_i|x_k)p(x_j|x_k)$ from (3.2), and thus x_i and x_j are conditionally independent.

The conditional independence property of Gaussian graphical models can also be obtained from the perspective of Schur complement. First let y define (x_i, x_j),

and z define the remaining components, then Σ can be expressed as

$$\Sigma = P \begin{bmatrix} \Sigma_{yy} & \Sigma_{yz} \\ \Sigma_{yz}^T & \Sigma_{zz} \end{bmatrix} P^T,$$

where P is a permutation matrix that shifts components x_i and x_j to the top-left corner. The inverse of Σ can be written as

$$\Sigma^{-1} = P \begin{bmatrix} (\Sigma_{yy} - \Sigma_{yz}\Sigma_{zz}^{-1}\Sigma_{yz}^T)^{-1} & * \\ * & * \end{bmatrix} P^T = P \begin{bmatrix} (\Sigma_{y|z})^{-1} & * \\ * & * \end{bmatrix} P^T,$$

where in the second equality we have used the covariance property of conditional distribution for Gaussian random variables $\Sigma_{y|z} = \Sigma_{yy} - \Sigma_{yz}\Sigma_{zz}^{-1}\Sigma_{yz}^T$. If x_i and x_j are conditionally independent, $\Sigma_{y|z}$ is diagonal. This result can be expressed as in (3.1) in terms of Σ^{-1}.

3.1.2 Covariance selection

Given N independent and identically distributed (*i.i.d.*) samples $\{x_1, \ldots, x_N\}$ from an n-dimensional multivariate Gaussian distribution $N(0, \Sigma)$, the likelihood can be expressed as

$$p(x; \Sigma) = \frac{1}{\left((2\pi)^n \det \Sigma\right)^{N/2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{N} x_i^T \Sigma^{-1} x_i\right). \tag{3.3}$$

In order to estimate the unknown variable Σ, we provide two approaches: maximum likelihood estimation (MLE) and minimization of Kullback-Leibler divergence.

Maximum likelihood estimation The log-likelihood function of (3.3) is

$$\log p(x; \Sigma) = -\frac{N}{2}\log\det\Sigma - \frac{1}{2}\sum_{i=1}^{N} x_i^T \Sigma^{-1} x_i - \frac{N}{2} n \log(2\pi). \tag{3.4}$$

The maximum likelihood estimation (MLE) problem is formulated as

$$\text{maximize} \qquad \frac{N}{2}\log\det\Sigma^{-1} - \frac{N}{2}\,\mathbf{tr}(C\Sigma^{-1}), \tag{3.5}$$

where $C = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T$ is the sample covariance matrix. From (3.5), we can see that if C is nonsingular, C^{-1} is the maximum likelihood estimate of the inverse covariance matrix.

Kullback-Leibler divergence An alternative approach is by minimizing the Kullback-Leibler divergence $D_{\mathrm{KL}}(p \parallel q)$ of $p(x) \sim N(0, C)$ from $q(x) \sim N(0, \Sigma)$. The Kullback-Leibler divergence can be formulated as

$$\begin{aligned} D_{\mathrm{KL}}(p \parallel q) &= \int p(x) \log \frac{p(x)}{q(x)} dx \\ &= \int p(x) \left[\frac{1}{2} \log \frac{(\det \Sigma)}{(\det C)} - \frac{1}{2} x^T C^{-1} x + \frac{1}{2} x^T \Sigma^{-1} x \right] dx \\ &= \frac{1}{2} \log \frac{(\det \Sigma)}{(\det C)} - \frac{1}{2} \mathbf{tr}\{E[xx^T] C^{-1}\} + \frac{1}{2} E[xx^T \Sigma^{-1}] \\ &= \frac{1}{2} \left[\log \frac{\det \Sigma}{\det C} - n + \mathbf{tr}\left(C\Sigma^{-1}\right) \right]. \end{aligned} \tag{3.6}$$

By minimizing (3.6), we can see the problem turns into (3.5).

In practice, one may not use C^{-1} as the estimate for Σ^{-1}. For one reason, C is required to be non-singular, and MLE is not a robust estimator of Σ^{-1}. Another reason is that one may want to impose some prior structural conditions on Σ^{-1} (3.1), and we only need to focus on structural nonzero entries. With the prior knowledge of the sparsity pattern, the number of parameters to be estimated is reduced, and thus the estimation accuracy can be improved. This problem is often referred to as *covariance selection* [Dem72], which can be formulated as:

$$\begin{aligned} &\text{minimize} && \mathbf{tr}(C\Sigma^{-1}) + \log\det\Sigma \\ &\text{subject to} && (\Sigma^{-1})_{ij} = 0, \quad \text{for } (i,j) \in \bar{E}, \end{aligned} \tag{3.7}$$

where the sets

$$\begin{aligned} E &\subseteq \{(i,j) \mid i,j \in \{1,2,\ldots,n\},\, i > j\}, \\ \bar{E} &= \{(i,j) \mid i,j \in \{1,2,\ldots,n\},\, i > j\} \setminus E \end{aligned}$$

are a subset of the off-diagonal index pairs and its complement. We refer to the set E, which contains the positions of the possibly nonzero entries in Σ^{-1}, as the

sparsity pattern of Σ^{-1}. Problem (3.7) includes an implicit constraint that the variable Σ is positive definite. Dempster observed that the problem is convex if $X = \Sigma^{-1}$ is used as the optimization variable. After this change of variables, the covariance selection problem can be written as a convex optimization problem

$$\text{minimize} \quad \mathbf{tr}(CX) - \log\det X + \psi(X), \tag{3.8}$$

where ψ is the indicator function of the sparsity pattern:

$$\psi(X) = \sum_{(i,j)\in\bar{E}} \delta(X_{ij}), \qquad \delta(u) = \begin{cases} 0 & u = 0 \\ \infty & u \neq 0. \end{cases} \tag{3.9}$$

With structurally restricted zeros, the number of free variables are reduced and thus the estimation variance can be reduced.

3.1.3 ℓ_1-norm penalized covariance selection

In most applications, the sparsity pattern is unknown. One approach is to enumerate all possible sparsity patterns, and use information-theoretic criteria such as Akaike or Bayes information criteria (AIC or BIC) to select the model. Unfortunately, this approach is computationally intractable for high-dimensional data. To solve this issue, extensive research has been conducted over the past decade. Meinshausen et al. [MB06] proposed the method of neighbor selection by fitting a lasso model for each variable with other variables as predictors. A more popular approach is to add an ℓ_1-norm penalty to the log-likelihood objective, *i.e.*, to solve (3.8) with

$$\psi(X) = \lambda \sum_{i>j} |X_{ij}|. \tag{3.10}$$

Theoretically, this approach is reasonable since the graph structure Σ^{-1} is sparse for most applications, especially for applications with high-dimensional data ($n \gg N$) where only a few components are conditionally independent.

Problem (3.7) with the penalty in (3.10) has received extensive interest in recent years, and many optimization algorithms have been proposed. Among

different approaches, one type of algorithms is based on block coordinate descent (BCD) method. The main idea is to optimize one row or column of the matrix sequentially while fixing the remaining entries for each iteration. For example, Friedman et al. [FHT07] and Banerjee et al. [BGd08] applied the BCD scheme to the dual problem of (3.7), and solved the subproblem by a coordinate decent method (*glasso*) and an interior-point method (COVSEL) respectively. Instead of working on the dual problem, Scheinberg and Rish [SR09] proposed a greedy coordinate ascent method (SINCO) applied to the primal problem directly. In addition to BCD based approaches, another type of algorithms is based on first-order methods. d'Aspremont [dBG08] applied Nesterov's optimal first-order method [Nes05]; Yuan [Yua09], Scheinberg et al. [SMG10] and Goldfarb et al. [GMS13] applied the scheme of alternating direction method; all of these methods are applied to the primal problem. As an alternative, Duchi et al. [DGK08] solved the dual problem using a projected gradient method, and Lu [Lu09, Lu10] using Nesterov's optimal first-order method [Nes05]. Another major type of approaches is based on interior point methods with fast computation of (inexact) Newton steps [LT10, WST10, HDRS11, HSDR14]. Some other authors also seek efficient methods by decomposing the problem based on the sparsity structure [DVR08, MH12, HDRB12a].

3.1.4 Restricted sparse inverse covariance selection

For some applications, the structural pattern is partially given, and we aim to estimate the sparse graph structure based on the given constraints. This is equivalent to the combination of functions (3.9) and (3.10),

$$\psi(X) = \sum_{(i,j)\in\bar{E}} \delta(X_{ij}) + \lambda \sum_{(i,j)\in E} |X_{ij}|. \tag{3.11}$$

With the choice of ψ in (3.8), the off-diagonal entries of X indexed by $\bar{E}$ are constrained to be zero; the remaining entries are penalized by an ℓ_1-norm penalty.

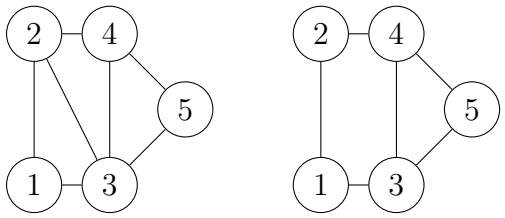
The constraints on the entries in $\bar{E}$ then represent the prior information about the sparsity pattern. For example, if the random variable contains consecutive values of a vector autoregressive process with lag r, then the inverse covariance matrix is block-banded with half-bandwidth r. Incorporating prior information of this kind reduces the number of parameters to be estimated in the maximum-likelihood problem, and hence the number of samples needed for a good estimate. We will refer to problem (3.8) with the penalty function (3.11) as *restricted* sparse inverse covariance selection. This problem has been analyzed in [Lu10, WST10, LT10]. However, they did not exploit the fast computation feature of nonzero structure, and their algorithms can not be applied for very large scale problems (for synthetic examples with $n = 1000$, the simulation takes more than one hour to converge). In the rest of this chapter, we will review background of chordal graphs, provide the proximal Newton method for self-concordant functions with time complexity analysis, and show that fast computation methods for chordal structure can be applied to *restricted* sparse inverse covariance selection to improve the time complexity.

3.2 Chordal sparsity patterns

In this section, we describe important properties about chordal structure. For readers who wish to know more details about chordal structure, we refer to [BP93, DVR08, ADV13, VA14].

3.2.1 Chordal graph

Let $G = (V, E)$ be a connected undirected graph where V is the set of vertices and E is the set of edges. A symmetric matrix X can be represented by a graph $G = (V, E)$, where an edge exists between two vertices v_i and v_j if and only if $\{i, j\} \notin E$. A graph G is *chordal* if every cycle of four or more vertices has a



(a) Chordal (b) Nonchordal

Figure 3.1: The graph (a) is a chordal graph because all cycles of length four or greater have a chord. The graph (b) is a nonchordal graph because there is a cycle of length four (1-2-3-4) without a chord.

chord, where a chord means an edge that joins two nonconsecutive vertices of the cycle (as shown in Fig. 3.1). We refer to the corresponding matrix as a *chordal matrix*.

A perfect elimination ordering (PEO) [DR83] is an ordering $v_1, \ldots, v_n$ of the vertices of the graph such that, if we eliminate the vertices in the order of PEO and add edges to all following vertices that are adjacent to the current eliminated vertex, and follow this procedure in the ordering at each elimination step, then a PEO generates no extra edge in the process, a process called *elimination game*. If a graph is chordal, it has a perfect elimination ordering, and it can be efficiently computed by the maximum cardinality search (MCS) algorithm [TY84, BP93] in linear time. For this reason, chordal graphs are also known as *perfect elimination graphs*.

Cholesky factorization For a sparse positive definite matrix X, the Cholesky factorization $PXP^T = LL^T$ (P is a permutation matrix and L is lower triangular) can be implemented by screening the sparsity patterns of $L + L^T$ (symbolic Cholesky factorization), and computing the corresponding nonzero entries of the Cholesky factor L. In general, the process of Cholesky factorization of a sparse positive definite matrix generates some fill-in (*i.e.*, added edges) in the graph of

X. However, if the matrix X has a chordal sparsity pattern, it can be factored with zero fill-in following the perfect elimination ordering, *i.e.*, $L + L^T$ has the same sparsity pattern as PXP^T. Therefore, we only need to screen the sparsity pattern of PXP^T instead of $L + L^T$ [BP93]. For the non-chordal case, we add fill-in to the graph to make it chordal by computing a symbolic Cholesky factorization (implicitly) following the order of *elimination tree* (a tree where the parent of node k is the row index j of the first nonzero below the diagonal of column k of X, if all nodes are labeled from 1 to n), a process called *chordal embedding*. The amount of fill-in generally depends on the ordering of the nodes, and with different orderings we can obtain different chordal embeddings.

One important algorithm to calculate the entry values in the Cholesky factor L is the *multifrontal method*, which is a recursion on the elimination tree. The performance of the multifrontal method can be improved by combining the vertices into supernodes and applying block eliminations for the corresponding columns. The supernodes are closely related to cliques in the graph, where a clique is a subset of nodes in G such that all pairs of vertices are adjacent. A maximal clique of G is a clique that cannot be extended by including one more adjacent vertex, and thus it is not a subset of another larger clique. For the chordal matrix X, if we use J_i to denote the union of index i and the row indices of the nonzero entries below the diagonal in column i of L, then it is shown that J_i is a clique [BP93]. Based on the elimination tree, we can construct the *supernodal elimination tree* by grouping together columns with the same nonzero structure into cliques J_i's, and each clique can be treated as a dense matrix for computation. The supernodal elimination tree is also referred to as clique tree or junction tree [Dar09]. The sparse matrix, elimination tree and clique for band matrix are shown in Fig. 3.2. Similar to the multifrontal algorithm, the supernodal multifrontal algorithm is a recursion on the clique tree. More details can be found in [DVR08].

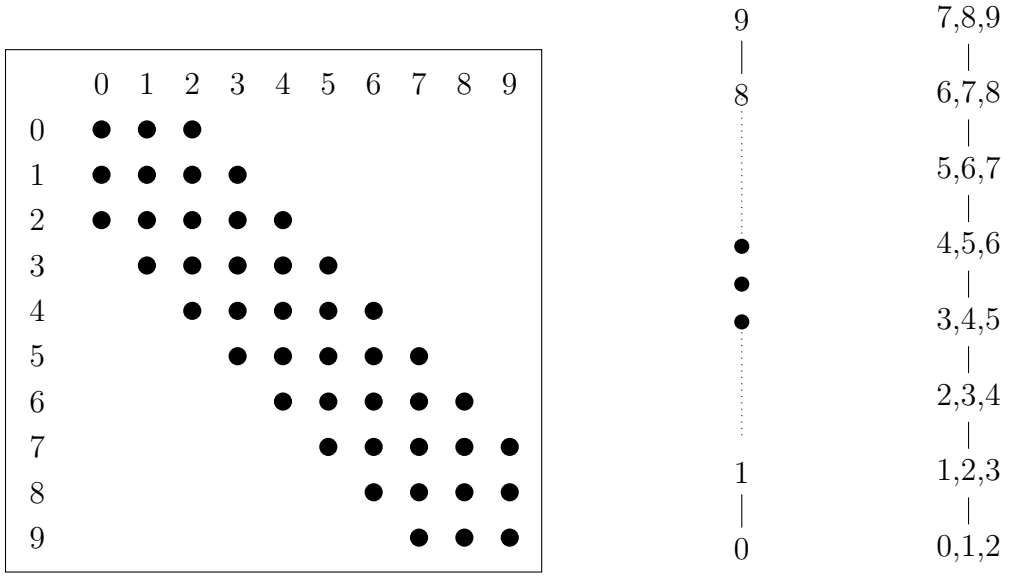


Figure 3.2: Left: Sparsity pattern of a 10×10 band matrix with bandwidth 5. Middle: Elimination tree of the band matrix. Right: Clique tree of the band matrix.

3.2.2 Computations of log-determinant function

For a matrix $X \in \mathbf{S}_E^n$, where $\mathbf{S}_E^n$ denotes the space of $n \times n$ symmetric matrices with sparsity pattern E, chordal sparse matrix computations can be applied to evaluate $\log\det X$ and its derivatives. To evaluate $\log\det X$ at a given $X \succ 0$, we compute a sparse Cholesky factorization

$$X = P^T L L^T P.$$

Adding the logarithms of the diagonal elements of L gives $\phi(X) = -2\sum_i \log L_{ii}$. Given the Cholesky factorization, the gradient and Hessian can also be computed by algorithms that are closely related to the multifrontal algorithm for sparse Cholesky factorization and use similar recursions on an elimination tree or supernodal elimination tree [ADV13, VA14]. The gradient of ϕ, as a function from $\mathbf{S}_E^n$ to $\mathbf{R}$, is given by

$$\nabla\phi(X) = \mathrm{P}_E(C - X^{-1}),$$

where P_E denotes projection on $\mathbf{S}^n_E$. Computing the gradient therefore requires computing the entries of X^{-1} on the diagonal and in positions $(i,j) \in E$, but not any of the other entries. For a chordal pattern, this projected inverse can be computed by a recursion on the elimination tree. The Hessian $\mathcal{H}_X$ of ϕ at $X \in \mathbf{dom}\,\phi$ is a linear mapping from $\mathbf{S}^n_E$ to $\mathbf{S}^n_E$ defined by

$$\mathcal{H}_X(V) = \nabla^2\phi(X)[V] = \frac{d}{d\alpha}\nabla\phi(X+\alpha V)\Big|_{\alpha=0} = \mathrm{P}_E(X^{-1}VX^{-1}V).$$

For a chordal pattern E, the evaluations of $\mathcal{H}_X(V)$ or $\mathcal{H}_X^{-1}(V)$ can be computed by two recursions on the elimination tree. The complexity of each of these operations is roughly the same as the cost of a sparse Cholesky factorization with sparsity pattern E. We refer the interested reader to [VA14] for details and historical background on these techniques.

3.2.3 Restricted inverse covariance selection via chordal extension

We now apply the technique of chordal sparse matrix computations to the restricted covariance selection problem, and reformulate (3.8) with $\phi(X)$ defined in (3.11) as follows. We first compute a *triangulation* or *chordal extension* E' of the sparsity pattern E, *i.e.*, a sparsity pattern E' that contains E and is also *chordal* [VA14]. Instead of optimizing over $X \in \mathbf{S}^n_E$, as in (3.8), we can then restrict X, without loss of generality, to $\mathbf{S}^n_{E'}$. Thus the problem can be written equivalently as

$$\text{minimize} \quad \phi(X) + \psi(X) \tag{3.12}$$

with a *sparse* matrix variable $X \in \mathbf{S}^n_{E'}$, and functions $\phi, \psi : \mathbf{S}^n_{E'} \to \mathbf{R}$ defined as

$$\phi(X) = \mathbf{tr}(CX) - \log\det X, \qquad \psi(X) = \sum_{(i,j)\in E'\setminus E} \delta(X_{ij}) + \gamma \sum_{(i,j)\in E} |X_{ij}|.$$

As mentioned we define $\mathbf{dom}\,\phi = \{X \in \mathbf{S}^n_{E'} \mid X \succ 0\}$. The second term ψ is separable and its proximal operator reduces to simple component-wise operations (soft-thresholding for entries in positions $(i,j) \in E$; substituting zero for entries

in positions $(i,j) \in E' \setminus E$). The first term ϕ is self-concordant [NN94] with $X \in \mathbf{S}^n_{E'}$. This makes the proximal Newton method [OONR12, ST13, BNO15] an attractive algorithm for (3.12), since the key computation in the algorithm is involved with the evaluation of $\phi(X)$, $\nabla\phi(X)$, and $\nabla^2\phi(X)[V]$, which can be evaluated efficiently with specialized algorithms for chordal structure. In section 3.3, we will present the proximal Newton method for self-concordant functions, and provide the convergence analysis for inexact Newton steps.

3.3 Proximal Newton method[1]

The *proximal Newton algorithm* is a method for solving composite optimization problems

$$\text{minimize} \quad f(x) = g(x) + h(x) \tag{3.13}$$

with g convex and twice continuously differentiable, and h convex and possibly non-differentiable. Problem (3.12) is a composite convex optimization problem that can be expressed as (3.13) if we represent the matrices X as vectors x of length $|E'| + n$. At each iteration of the proximal Newton algorithm, an update $x := x + \alpha v(x)$ is made, where α is a positive step size and $v(x)$ is the *proximal Newton step* at x, defined as

$$v(x) = \operatorname*{argmin}_v \left(g(x) + \nabla g(x)^T v + \frac{1}{2} v^T \nabla^2 g(x) v + h(x+v) \right). \tag{3.14}$$

The vector $x + v(x)$ minimizes an approximation

$$\hat{f}_x(y) = g(x) + \nabla g(x)^T (y-x) + \frac{1}{2}(y-x)^T \nabla^2 g(x)(y-x) + h(y) \tag{3.15}$$

of the cost function f, obtained by replacing g with a second-order approximation around x. For this reason the algorithm is also called a *successive quadratic approximation method* [BNO15]. When h is zero, the proximal Newton step is

[1] This section is from our paper *Inexact proximal Newton methods for self-concordant functions* by Jinchao Li, Martin S. Andersen, and Lieven Vandenberghe.

$v(x) = -\nabla^2 g(x)^{-1} \nabla g(x)$ and the proximal Newton method reduces to the standard Newton method for minimizing $g(x)$.

The proximal Newton method and some of its variants have recently been studied for applications in statistics and machine learning, in which $h(x)$ is an ℓ_1-norm penalty, added to a differentiable objective to promote sparsity in the solution [HSDR11, OONR12, LSS14, BNO15, TDKC15]. This approach is motivated by the fact that the optimization problem in (3.14) is a lasso problem (minimization of a convex quadratic function plus an ℓ_1-norm) that can be solved by efficient iterative algorithms. More generally, the proximal Newton method is interesting when h has an inexpensive proximal operator, so the subproblem in (3.14) can be solved by proximal gradient methods.

With exact steps $v(x)$, the proximal Newton algorithm is known to have the same excellent convergence properties as the Newton method for smooth unconstrained minimization: fast local convergence, and global convergence from any starting point if a proper step size selection is used. These convergence properties are discussed in [LSS14] under the assumptions that g is strongly convex with a Lipschitz continuous gradient, and in [TDKC15] for self-concordant functions g.

In practice, it is expensive to compute the proximal Newton step accurately, since $v(x)$ is found by minimizing (3.15) numerically. This is a fundamental difference with the standard Newton method. It is therefore important to understand the convergence of the proximal Newton method with inexact steps [OONR12, ST13, BNO15]. Lee, Sun, and Saunders [LSS14, page 1428] propose the following criterion for accepting an approximation v of (3.14). A vector v is accepted as an approximate proximal Newton step at x if it satisfies

$$\|\hat{F}_{x,t}(x+v)\| \leq \eta_{\mathrm{f}} \|F_t(x)\| \tag{3.16}$$

where $t \leq 1/\lambda_{\max}(\nabla^2 g(x))$, and F_t, $\hat{F}_{x,t}$ are the *gradient mappings* [Nes04, section 2.2.3] of the cost function f and its local approximation $\hat{f}_x$, respectively. The

forcing term η_{f} in (3.16) can be adjusted adaptively to obtain superlinear local convergence. Byrd, Nocedal, and Oztoprak [BNO15] use a similar condition, but also impose the condition

$$\hat{f}_x(x+v) - f(x) \leq \beta \left(\nabla g(x)^T v + h(x+v) - h(x) \right)$$

with $\beta \in (0, 1/2)$ and show that this ensures global convergence when g is strongly convex with a Lipschitz continuous gradient. The papers [BNO15, LSS14, TDKC15] also analyze variable metric or quasi-Newton methods, in which approximate Hessians are used in the approximation (3.15).

In this section, we extend the results of [TDKC15] for the (exact) proximal Newton method for self-concordant functions g to the proximal Newton method with inexact steps. In the algorithms we analyze, the condition (3.16) is replaced by the following criterion: a step v is accepted as an approximation of $v(x)$ if a residual

$$r \in \nabla g(x) + \nabla^2 g(x) v + \partial h(x+v),$$

in the optimality conditions for (3.14) is known that satisfies the inequality

$$\|\nabla^2 g(x)^{-1/2} r\| \leq (1-\theta) \|\nabla^2 g(x)^{1/2} v\|.$$

We show that if g is self-concordant, then the inexact proximal Newton method converges globally if a damped stepsize or backtracking line search is used. The $1-\theta$ plays a role similar to the forcing term η_{f} in (3.16). We show that the local convergence is quadratic if $\theta = 1$, linear if θ constant and less than one, and superlinear if θ approaches one as the algorithm converges.

The composite optimization problem (3.13) with self-concordant functions g has important applications in machine learning [TDKC15]. The proximal Newton method that we develop in subsections 3.3.1–3.3.3 is motivated by the application to sparse inverse covariance selection. In this problem, the smooth component g is self-concordant, but it is not strongly convex and its gradient is not Lipschitz

continuous on its entire domain. Moreover, in the large sparse setting for the restricted sparse inverse covariance selection problem, matrix-vector products with the Hessian $\nabla^2 g(x)v$ or the inverse Hessian $\nabla^2 g(x)^{-1}w$ can be computed quite efficiently, at roughly the same cost as the gradient $\nabla g(x)$. These properties make it possible to compute a sufficiently accurate approximate Newton step by applying a proximal gradient method to minimize (3.15).

In this section, we first review the definition and key properties of self-concordant functions, and present a theorem that provides bounds on the optimum of (3.13) in terms of the magnitude of inexact proximal Newton steps. Then we discuss the proximal Newton method with a damped step size and a backtracking line search, respectively, and give global and local convergence results that account for inexactness of the search directions. In section 3.4 we discuss the application to covariance selection and present some numerical results.

3.3.1 Proximal Newton step for self-concordant functions

We consider unconstrained optimization problems of the form (3.13) with $g : \mathbf{R}^n \to \mathbf{R}$ self-concordant and $h : \mathbf{R}^n \to \mathbf{R}$ closed, convex, and possibly non-differentiable. We assume the problem is feasible ($\mathbf{dom}\, f = \mathbf{dom}\, g \cap \mathbf{dom}\, h \neq \emptyset$). This implies that the sum $f = g + h$ is a closed function (see, for example, [HUL93, page 158]).

3.3.1.1 Self-concordance

Specifically, we make the following assumptions about g.

- g is closed, convex, with open domain.
- g is three times continuously differentiable with $\nabla^2 g(x)$ positive definite on $\mathbf{dom}\, g$.

- The Hessian of g satisfies the matrix inequality

$$\frac{d}{d\alpha}\nabla^2 g(x+\alpha v)\bigg|_{\alpha=0} \preceq 2\|v\|_x \nabla^2 g(x) \tag{3.17}$$

  for all $x \in \mathbf{dom}\, g$ and all v, where $\|v\|_x = (v^T\nabla^2 g(x)v)^{1/2}$. (The inequality $A \preceq B$ means $B - A$ is positive semidefinite.)

These properties characterize self-concordant functions as defined by Renegar [Ren01] and Nesterov [Nes04]. They define a subclass of the self-concordant functions introduced in [NN94]: in Nesterov and Nemirovski's book, closed self-concordant functions are called *strongly* self-concordant, self-concordant functions with nonsingular Hessians are called *nondegenerate*, and the fundamental inequality (3.17) includes a scaling parameter a that we take to be one. Nesterov [Nes04, page 181] refers to self-concordant functions with $a = 1$ as *standard* self-concordant functions.

For future reference, we list the properties of self-concordant functions that will be used in the thesis.

- *Bounds on Hessian* [NN94, theorem 2.1.1]. If $x, y \in \mathbf{dom}\, g$ and $\|y-x\|_x < 1$, then

$$(1 - \|y-x\|_x)^2\nabla^2 g(x) \preceq \nabla^2 g(y) \preceq \frac{1}{(1-\|y-x\|_x)^2}\nabla^2 g(x). \tag{3.18}$$

- *Bounds on gradient* [Nes12, lemma 1]. If $x, y \in \mathbf{dom}\, g$ and $\|y-x\|_x < 1$, then

$$\|\nabla g(y) - \nabla g(x) - \nabla^2 g(x)(y-x)\|_{x*} \leq \frac{\|y-x\|_x^2}{1-\|y-x\|_x}. \tag{3.19}$$

  Here $\|v\|_{x*} = (v^T\nabla^2 g(x)^{-1}v)^{1/2}$ denotes the dual norm of $\|\cdot\|_x$.

- *Bounds on function value* [Nes04, theorems 4.1.7 and 4.1.8]. If $x, y \in \mathbf{dom}\, g$, then

$$\omega(\|y-x\|_x) \leq g(y) - g(x) - \nabla g(x)^T(y-x) \leq \omega^*(\|y-x\|_x), \tag{3.20}$$

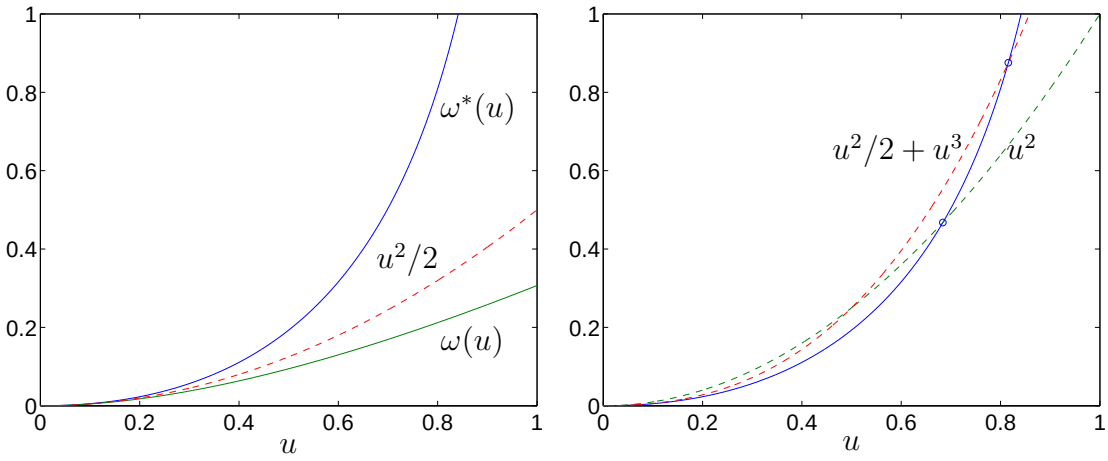


Figure 3.3: *Left.* The functions $\omega(u) = u - \log(1+u)$ and $\omega^*(u) = -u - \log(1-u)$. *Right.* The function $\omega^*(u)$ in solid line, with two upper bounds $\omega^*(u) \leq u^2$ for $u \leq 0.68$ and $\omega^*(u) \leq u^2/2 + u^3$ for $u \leq 0.81$.

where ω and ω^* denote the functions

$$\omega(u) = u - \log(1+u), \qquad \omega^*(u) = -u - \log(1-u).$$

The left-hand inequality in (3.20) holds for all $x, y \in \mathbf{dom}\, g$. The right-hand inequality holds for all $x, y \in \mathbf{dom}\, g$ with $\|y - x\|_x < 1$. Note that ω and ω^* are Fenchel conjugates (Legendre transforms). In particular, we will use the fact that

$$\inf_v \left(\omega(v) - uv\right) = -\omega^*(u), \qquad \inf_u \left(\omega^*(u) - uv\right) = -\omega(v). \tag{3.21}$$

Figure 3.3 shows the two functions and illustrates the inequalities $\omega(u) \leq u^2/2 \leq \omega^*(u)$ and

$$\omega^*(u) \leq u^2/2 + u^3 \quad \text{for } u \in [0, 0.81), \quad \omega^*(u) \leq u^2 \quad \text{for } u \in [0, 0.68). \tag{3.22}$$

A useful lower bound on $\omega(u)$ is

$$\omega(u) \geq \frac{u^2}{2(1+u)} \quad \text{for } u \geq 0. \tag{3.23}$$

- *Dikin ellipsoid theorem* [NN94, theorem 2.1.1.b]. The (open) Dikin ellipsoid at $x \in \mathbf{dom}\, g$ is defined as

$$\mathcal{E}_x = \{y \mid \|y - x\|_x < 1\}.$$

  The upper bound in (3.20) implies that $\mathcal{E}_x \subset \mathbf{dom}\, g$.

3.3.1.2 Scaled proximal operator

The proximal operator of a closed convex function h is defined as

$$\text{prox}_h(y) = \underset{u}{\text{argmin}} \left(h(u) + \frac{1}{2}\|u - y\|^2 \right),$$

where $\|\cdot\|$ denotes the Euclidean norm. It can be shown that the proximal operator $\text{prox}_h(y)$ is uniquely defined for all y [Mor65].

With every $x \in \mathbf{dom}\, g$ we can associate a *scaled proximal operator* $\text{prox}_{h,x}$, defined in a similar way as the standard proximal operator, but using the local quadratic norm $\|v\|_x = (v^T \nabla^2 g(x) v)^{1/2}$ instead of the Euclidean norm:

$$\text{prox}_{h,x}(y) = \underset{u}{\text{argmin}} \left(h(u) + \frac{1}{2}\|u - y\|_x^2 \right). \tag{3.24}$$

This scaled proximal operator can be expressed in terms of the standard (unscaled) proximal operator of the function $\tilde{h}(y) = h(\nabla^2 g(x)^{-1/2} y)$:

$$\text{prox}_{h,x}(y) = \nabla^2 g(x)^{1/2} \text{prox}_{\tilde{h}}(\nabla^2 g(x)^{1/2} y).$$

It can be shown (directly from the definition (3.24) or by reduction to the unscaled proximal operator) that $u = \text{prox}_{h,x}(y)$ exists and is unique for all $x \in \mathbf{dom}\, g$ and all y, and that it is the unique solution of the monotone inclusion problem

$$0 \in \partial h(u) + \nabla^2 g(x)(u - y). \tag{3.25}$$

As an immediate consequence we note that if $x^\star$ minimizes $f(x)$, *i.e.*, $0 \in \nabla g(x^\star) + \partial h(x^\star)$, then

$$x^\star = \text{prox}_{h,x}\left(x^\star - \nabla^2 g(x)^{-1} \nabla g(x^\star) \right) \tag{3.26}$$

for all $x \in \mathbf{dom}\, g$. Conversely, if $x^\star$ satisfies (3.26) for some $x \in \mathbf{dom}\, g$, then $x^\star$ minimizes f.

3.3.1.3 Proximal Newton step

The *proximal Newton step* $v(x)$ at x is defined as

$$\begin{aligned} v(x) &= \mathrm{prox}_{h,x}\left(x - \nabla^2 g(x)^{-1}\nabla g(x)\right) - x \\ &= \operatorname*{argmin}_u \left(g(x) + \nabla g(x)^T v + \frac{1}{2} v^T \nabla^2 g(x) v + h(x+v) \right). \end{aligned}$$

From the second expression, or from the first expression and (3.25), we see that $v(x)$ is characterized by the condition

$$0 \in \nabla g(x) + \nabla^2 g(x) v(x) + \partial h(x + v(x)), \tag{3.27}$$

and that x is optimal if and only if $v(x) = 0$.

The magnitude $\|v(x)\|_x$ of the Newton step in the local norm $\|\cdot\|_x$ plays an important role in the analysis of Newton's method for minimizing self-concordant functions (*i.e.*, problem (3.13) with $h(x) = 0$) [NN94, Nes04]. In [NN94] $\|v(x)\|_x$ is called the *Newton decrement* of f at x.

When $h(x)$ is nonzero, it is generally not possible to compute $v(x)$ very accurately, and it is important to allow for inexact proximal Newton steps. In the algorithms discussed in the next subsections, the following criterion will be used for accepting a vector v as an inexact proximal Newton step at x: there exists an r such that

$$r \in \nabla g(x) + \nabla^2 g(x) v + \partial h(x + v), \qquad \|r\|_{x*} \leq (1-\theta)\|v\|_x, \tag{3.28}$$

where $\theta \in (0, 1]$ is an algorithm parameter. We can interpret $1 - \theta$ as a bound on the relative error in the conditions (3.27) that characterize the exact proximal Newton step. With $\theta = 1$, the condition requires $r = 0$ and therefore $v = v(x)$, the exact proximal Newton step.

The next theorem shows that if v satisfies (3.28) for some r, and $\|v\|_x$ is sufficiently small, then x is close to optimal for (3.13). The theorem is an extension of theorem 4.1.11 in [Nes04], which characterizes the distance to the minimum of a self-concordant function in terms of the norm $\|v(x)\|_x$ of the Newton step when $\|v(x)\|_x < 1$.

Theorem 1. *Suppose $x \in \boldsymbol{dom}\, g$, $x+v \in \boldsymbol{dom}\, h$, and v and r satisfy (3.28) with $\theta \in (0,1]$. If*

$$\|v\|_x < \frac{1}{2-\theta}. \tag{3.29}$$

then the following properties hold.

- *f is bounded below and*

$$\inf_y f(y) \geq f(x+v) + \theta\|v\|_x^2 - \omega^*(\|v\|_x) - \omega^*((2-\theta)\|v\|_x). \tag{3.30}$$

- *The sublevel set $\mathcal{S}_x = \{y \mid f(y) \leq f(x+v)\}$ is bounded: $\mathcal{S}_x \subseteq \{y \mid \|y-x\|_x \leq \hat{\rho}\}$ where $\hat{\rho}$ is the positive root of the nonlinear equation*

$$\omega(\rho) - \rho(2-\theta)\|v\|_x = \max\{0, \omega^*(\|v\|_x) - \theta\|v\|_x^2\} \tag{3.31}$$

  if $\|v\|_x > 0$, and $\hat{\rho} = 0$ if $\|v\|_x = 0$.

- *f has a unique minimizer $x^\star$ and $\|x - x^\star\|_x \leq \hat{\rho}$.*

Proof. We first note that, by the Dikin ellipsoid theorem, $x + v \in \mathbf{dom}\, g$, since $\|v\|_x < 1$. Therefore $x + v \in \mathbf{dom}\, f = \mathbf{dom}\, g \cap \mathbf{dom}\, h$, and the right-hand side of (3.30) and the sublevel set $\mathcal{S}_x$ are well defined.

To show (3.30) we consider an arbitrary $y \in \mathbf{dom}\, f$. We combine the lower bound on $g(y)$ from (3.20) and the upper bound on $g(x+v)$ from (3.20), to get

$$\begin{aligned} g(y) &\geq g(x) + \nabla g(x)^T(y-x) + \omega(\|y-x\|_x) \\ &\geq g(x+v) + \nabla g(x)^T(y-x-v) - \omega^*(\|v\|_x) + \omega(\|y-x\|_x). \end{aligned}$$

A lower bound on $h(y)$ follows from from the subgradient in (3.28):

$$h(y) \geq h(x+v) + (r - \nabla g(x) - \nabla^2 g(x)v)^T(y - x - v).$$

Adding the lower bounds on $g(y)$ and $h(y)$ gives a lower bound on $f(y)$:

$$\begin{aligned} & f(y) - f(x+v) \\ & \quad \geq \quad (r - \nabla^2 g(x)v)^T(y-x) - r^T v + \|v\|_x^2 - \omega^*(\|v\|_x) + \omega(\|y-x\|_x) \\ & \quad \geq \quad (r - \nabla^2 g(x)v)^T(y-x) - \|r\|_{x*}\|v\|_x + \|v\|_x^2 - \omega^*(\|v\|_x) + \omega(\|y-x\|_x) \\ & \quad \geq \quad (r - \nabla^2 g(x)v)^T(y-x) + \theta\|v\|_x^2 - \omega^*(\|v\|_x) + \omega(\|y-x\|_x). \end{aligned} \tag{3.32}$$

Next, we find a lower bound for the right-hand side of (3.32). We express y as $y = x + tw$ with $\|w\|_x = 1$ and $t \geq 0$ and write (3.32) as

$$f(x+tw) \geq f(x+v) + t(r - \nabla^2 g(x)v)^T w + \theta\|v\|_x^2 - \omega^*(\|v\|_x) + \omega(t).$$

We first consider the minimum of the right-hand side over w. Using the Cauchy-Schwarz inequality, the triangle inequality, and the condition (3.28) we get

$$\begin{aligned} f(x+tw) \quad & \geq \quad f(x+v) - t\|r - \nabla^2 g(x)v\|_{x*} + \theta\|v\|_x^2 - \omega^*(\|v\|_x) + \omega(t) \\ & \geq \quad f(x+v) - t(\|r\|_{x*} + \|v\|_x) + \theta\|v\|_x^2 - \omega^*(\|v\|_x) + \omega(t) \\ & \geq \quad f(x+v) - t(2-\theta)\|v\|_x + \theta\|v\|_x^2 - \omega^*(\|v\|_x) + \omega(t). \end{aligned} \tag{3.33}$$

The lower bound (3.30) now follows if we use the conjugacy relation (3.21) to minimize the right-hand side of (3.33) over t.

To show the bound on the sublevel set, we note that (3.33) implies that $f(x + tw) > f(x+v)$ when

$$\omega(t) - t(2-\theta)\|v\|_x > \omega^*(\|v\|_x) - \theta\|v\|_x^2.$$

When $v = 0$, this holds for any $t > 0$. For nonzero v, it holds if t is greater than the positive root of the nonlinear equation (3.31).

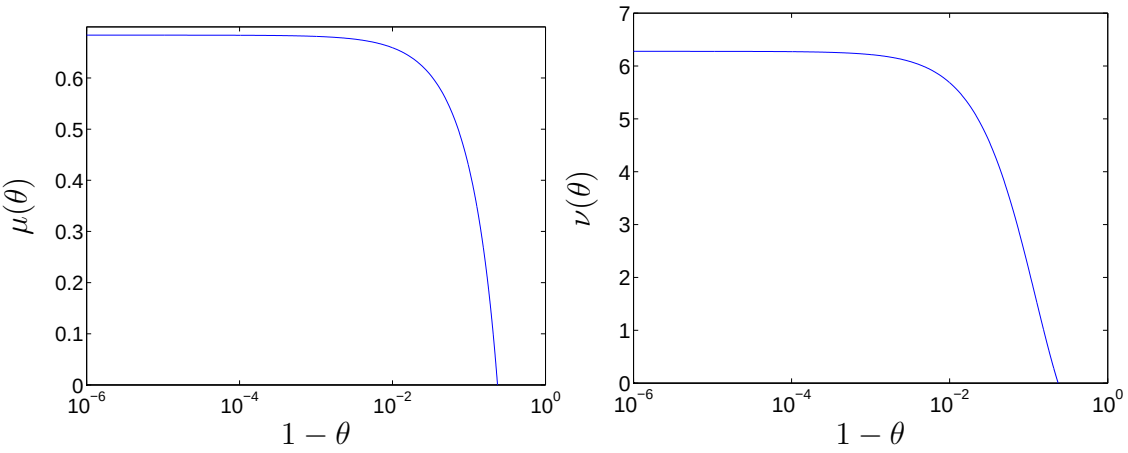


Figure 3.4: *Left.* $\mu(\theta)$ is the solution u of the nonlinear equation $\omega^*((2-\theta)u) = \theta u^2$ for $3-\sqrt{5} \leq \theta \leq 1$. We have $\mu(1) = 0.68$ and $\mu(3-\sqrt{5}) = 0$. *Right.* The function $\nu(\theta)$ defined in (3.36). We have $\nu(1) = 6.28$ and $\nu(3-\sqrt{5}) = 0$.

Finally, since f is a closed function, it attains its minimum if the sublevel sets are bounded (by the Weierstrass theorem [Ber09, page 119]). Since f is also strictly convex (the sum of a strictly convex function g and a convex function h), the minimizer is unique. □

The bounds on $f(x^\star)$ and $\|x-x^\star\|_x$ in theorem 1 can be simplified by restricting $\|v\|_x$ to a smaller interval than allowed by (3.29). We mentioned in section 3.3.1.1, that $\omega^*(u) \approx u^2/2$ for small u and $\omega^*(u) \leq u^2$ for $u \in [0, 0.68]$. More generally, for each $\theta \in (3-\sqrt{5}, 1] = (0.764, 1]$ there exists a positive $\mu(\theta)$ such that

$$\omega^*((2-\theta)u) \leq \theta u^2 \quad \text{for } u \in [0, \mu(\theta)] \tag{3.34}$$

(see figure 3.4). If $\theta \in (3-\sqrt{5}, 1]$, we can use the inequality (3.34) to simplify the lower bound (3.30) as follows: if $\|v\|_x \leq \mu(\theta)$, then

$$\begin{aligned} \inf_y f(y) &\geq f(x+v) + \theta\|v\|_x^2 - 2\omega^*((2-\theta)\|v\|_x) \\ &\geq f(x+v) - \theta\|v\|_x^2. \end{aligned} \tag{3.35}$$

Hence, for sufficiently small $\|v\|_x$, the quantity $\theta\|v\|_x^2$ gives an upper bound on

$f(x+v) - \inf_y f(y)$.

We can also derive a simple upper bound on $\hat{\rho}$. For $0 < \|v\|_x \leq \mu(\theta)$ and $\theta \in (3-\sqrt{5}, 1]$, the right-hand side of (3.31) is zero because of (3.34), and $\hat{\rho}$ is the positive root of the equation

$$\log(1+\rho) = \rho(1 - (2-\theta)\|v\|_x).$$

In other words, $\hat{\rho} = \phi^{-1}(1-(2-\theta)\|v\|_x)$ where $\phi(t) = \log(1+t)/t$. Since ϕ^{-1} is a convex function and $\phi^{-1}(1) = 0$, Jensen's inequality gives

$$\hat{\rho} \leq \left(1 - \frac{\|v\|_x}{\mu(\theta)}\right)\phi^{-1}(1) + \frac{\|v\|_x}{\mu(\theta)}\phi^{-1}(1-(2-\theta)\mu(\theta)) = \frac{\nu(\theta)}{\mu(\theta)}\|v\|_x$$

where

$$\nu(\theta) = \phi^{-1}\left(1-(2-\theta)\mu(\theta)\right). \tag{3.36}$$

This function is shown in figure 3.4. It follows that when $\|v(x)\|_x \leq \mu(\theta)$, the sublevel set $\mathcal{S}_x$ is bounded by a ball with radius $(\nu(\theta)/\mu(\theta))\|v(x)\|_x$ around x. In particular,

$$\|x - x^\star\|_x \leq \frac{\nu(\theta)}{\mu(\theta)}\|v\|_x. \tag{3.37}$$

For $\theta = 1$ and $v = v(x)$, the bounds (3.35) and (3.37) are

$$\inf_y f(y) \geq f(x+v(x)) - \|v(x)\|_x^2, \qquad \|x - x^\star\|_x \leq 9.18\,\|v(x)\|_x, \tag{3.38}$$

and these are valid if $\|v(x)\|_x \leq 0.68$. In the following subsection we will be interested in values of θ close to one, and it will be useful to note that $\mu(\theta) = 1/4$ for $\theta = 0.84$. In particular, if $\theta \geq 0.84$, then the bound (3.35) holds for $\|v\|_x \leq 1/4$.

3.3.2 Damped proximal Newton method

Now we analyze the following version of the proximal Newton method with inexact proximal Newton steps.

The exit condition guarantees that $f(x+v) - \inf_y f(y) \leq \delta$. This follows from the fact that (3.35) holds if $\theta \geq 0.84$ and $\|v\|_x \leq 1/4$, as we saw at the end of

Algorithm 1 Proximal Newton algorithm with damped stepsize

Input: A starting point $x \in \mathbf{dom}\, g$ and three parameters $\theta_{\min} \in [0.9, 1]$, $\eta \in (0, 1/4]$, and $\delta \in (0, 1)$.

Repeat:

1. Compute a step v that satisfies (3.28) for some r and $\theta \geq \theta_{\min}$.
2. If $\|v\|_x \leq 0.25$ and $\theta\|v\|_x^2 \leq \delta$, return $x + v$.
3. Otherwise, set $x := x + \alpha v$ with

$$\alpha = \frac{\theta}{1 + \theta\|v\|_x} \quad \text{if } \|v\|_x \geq \eta, \qquad \alpha = 1 \quad \text{otherwise.}$$

the previous section. The lower bound $\theta_{\min} \geq 0.9$ is imposed only to simplify this stopping criterion. Alternatively, one can take any $\theta_{\min} \in (0, 1]$ and use (3.30) to bound $f(x + v) - \inf_y f(y)$.

Note that the starting point x is not required to be in $\mathbf{dom}\, h$. However, the Dikin ellipsoid theorem guarantees that $x \in \mathbf{dom}\, f$ after the first iteration.

3.3.2.1 Local convergence

The following theorem extends a quadratic convergence result for Newton's method applied to a self-concordant function [Nes04, theorem 4.1.14]. A related result is [TDKC15, theorem 7] on the local convergence of the exact proximal Newton method with self-concordant g. For $\theta = 1$, theorem 2 gives an improvement over [TDKC15, theorem 7], which requires the condition $\|v(x)\|_x < 1 - 1/\sqrt{2}$; see also [TDKC15, remark 10]. Theorem 2 further generalizes these results by allowing inexact proximal Newton steps.

Theorem 2 (Unit steps)**.** *Suppose $x \in \mathbf{dom}\, g$, $x + v \in \mathbf{dom}\, h$, $\|v\|_x < 1$, and (3.28) is satisfied for some r and $\theta \in (0, 1]$. Define $x^+ = x + v$. Suppose*

$x^+ + v^+ \in \boldsymbol{dom}\, h$ *and*

$$r^+ \in \nabla g(x^+) + \nabla^2 g(x^+)v^+ + \partial h(x^+ + v^+), \qquad \|r^+\|_{x^+ *} \leq (1-\theta^+)\|v^+\|_{x^+}$$

holds for some r^+ *and* $\theta^+ \in (0,1]$. *Then*

$$\|v^+\|_{x^+} \leq \frac{\|v\|_x}{\theta^+(1-\|v\|_x)}\left(1-\theta+\frac{\|v\|_x}{1-\|v\|_x}\right).$$

If $\|v\|_x \leq 1-1/\sqrt{2} = 0.293$, *we have the simpler bound*

$$\|v^+\|_{x^+} \leq \frac{\sqrt{2}\|v\|_x}{\theta^+}\left(1-\theta+\sqrt{2}\|v\|_x\right). \tag{3.39}$$

Proof. We first note that $x^+ = x + v \in \mathbf{dom}\, g$ as a consequence of the Dikin ellipsoid theorem. Define

$$w = r - \nabla g(x) - \nabla^2 g(x)v, \qquad w^+ = r^+ - \nabla g(x^+) - \nabla^2 g(x^+)v^+.$$

We have $w \in \partial h(x+v)$ and $w^+ \in \partial h(x^+ + v^+)$, by definition of r and r^+. Monotonicity of the subdifferential ∂h implies that

$$(w^+ - w)^T v^+ = (w^+ - w)^T (x^+ + v^+ - x - v) \geq 0.$$

This observation is used in the first inequality of the following derivation:

$$\begin{aligned}
\|v^+\|_{x^+} &\leq \|v^+ + \nabla^2 g(x^+)^{-1}(w^+ - w)\|_{x^+} \\
&\leq \|\nabla^2 g(x^+)^{-1}(r^+ - \nabla g(x^+) - w)\|_{x^+} \\
&= \|r^+ - \nabla g(x^+) - w\|_{x^+ *} \\
&\leq \|r^+\|_{x^+ *} + \|\nabla g(x^+) + w\|_{x^+ *} \\
&= \|r^+\|_{x^+ *} + \|r + \nabla g(x^+) - \nabla g(x) - \nabla^2 g(x)v\|_{x^+ *} \\
&\leq (1-\theta^+)\|v^+\|_{x^+} + \|r\|_{x^+ *} + \|\nabla g(x^+) - \nabla g(x) - \nabla^2 g(x)v\|_{x^+ *} \\
\theta^+\|v^+\|_{x^+} &\leq \frac{1}{1-\|v\|_x}\left(\|r\|_{x*} + \|\nabla g(x+v) - \nabla g(x) - \nabla^2 g(x)v\|_{x*}\right) \\
&\leq \frac{\|v\|_x}{1-\|v\|_x}\left(1-\theta+\frac{\|v\|_x}{1-\|v\|_x}\right).
\end{aligned}$$

On the second line we use the definition of w^+, and on the fifth line the definition of w. Line 7 follows from (3.18), which implies that

$$\|z\|_{x+v,*}^2 = z^T\nabla^2 g(x+v)^{-1}z \leq \frac{1}{(1-\|v\|_x)^2} z^T\nabla^2 g(x)^{-1}z = \frac{\|z\|_{x*}^2}{(1-\|v\|_x)^2}.$$

The last step follows from (3.19). □

Theorem 2 can be used to establish local convergence of algorithm 1.

Exact proximal Newton method. Suppose the starting point x satisfies $\|v\|_x < \eta$ and we take $\theta_{\min} = 1$, so $v = v(x)$. The inequality (3.39) reduces to

$$\|v(x^+)\|_{x^+} \leq 2\|v(x)\|_x^2 \tag{3.40}$$

and, since $\eta \leq 1/4$, we have $\|v(x^+)\|_{x^+} < \eta$. All subsequent iterates therefore satisfy $\|v(x)\|_x < \eta$. It then follows from (3.40) that after k iterations

$$2\|v(x)\|_x \leq (2\eta)^{2^k} \leq \left(\frac{1}{2}\right)^{2^k}.$$

This shows that algorithm 1 converges quadratically when started at a point with $\|v(x)\|_x < \eta$. Since $\|v(x)\|_x^2 \leq (1/2)^{2^{k+1}}$, the exit condition $\|v\|_x^2 \leq \delta$ is satisfied after less than $\log_2 \log_2(1/\delta)$ iterations.

Inexact proximal Newton method. Suppose the starting point x satisfies $\|v\|_x < \eta$ and we take θ constant. From (3.39),

$$\begin{aligned}
\|v^+\|_{x^+} &\leq \sqrt{2}\left(\frac{1+\sqrt{2}\eta}{\theta} - 1\right)\|v\|_x \\
&\leq \sqrt{2}\left(\frac{1+\sqrt{2}/4}{0.9} - 1\right)\|v\|_x \\
&= 0.713\,\|v\|_x.
\end{aligned}$$

Therefore $\|v\|_x$ converges to zero linearly. If we let $\theta \to 1$, then the inequality (3.39) shows superlinear convergence.

3.3.2.2 Global convergence

The next theorem is an extension of a global convergence result for the standard damped Newton method for self-concordant functions [Nes04, theorem 4.1.12]. When $\theta = 1$, the result is identical to [TDKC15, theorem 6].

Theorem 3 (Damped steps)**.** *Suppose $x \in \mathbf{dom}\, f$, $x + v \in \mathbf{dom}\, h$, and (3.28) is satisfied for some r and $\theta \in (0, 1]$. If $\alpha = \theta/(1 + \theta\|v\|_x)$, then*

$$f(x + \alpha v) \leq f(x) - \omega(\theta\|v\|_x).$$

Proof. First note that $\alpha\|v\|_x < 1$. Hence $x + \alpha v \in \mathbf{dom}\, f$ as a consequence of the Dikin ellipsoid theorem. To show the upper bound on $f(x+\alpha v)$ we apply the upper bound (3.20) with $y = x + \alpha v$:

$$g(x + \alpha v) \leq g(x) + \alpha\nabla g(x)^T v + \omega^*(\alpha\|v\|_x).$$

An upper bound on $h(x+\alpha v)$ follows from Jensen's inequality and the subgradient of h at $x + v$ from (3.28):

$$\begin{aligned} h(x + \alpha v) &\leq h(x) + \alpha(h(x + v) - h(x)) \\ &\leq h(x) + \alpha(r - \nabla g(x) - \nabla^2 g(x)v)^T v \\ &= h(x) + \alpha(r - \nabla g(x))^T v - \alpha\|v\|_x^2. \end{aligned}$$

Adding the upper bounds on g and h gives

$$\begin{aligned} f(x + \alpha v) &\leq f(x) + \alpha(r^T v - \|v\|_x^2) + \omega^*(\alpha\|v\|_x) \\ &\leq f(x) + \alpha(\|r\|_{x*}\|v\|_x - \|v\|_x^2) + \omega^*(\alpha\|v\|_x) \\ &\leq f(x) - \alpha\theta\|v\|_x^2 + \omega^*(\alpha\|v\|_x). \end{aligned} \tag{3.41}$$

This bound holds when $\alpha\|v\|_x < 1$. The right-hand side is minimized at $\alpha = \theta/(1 + \theta\|v\|_x)$, with minimum value $f(x) - \omega(\theta\|v\|_x)$. □

Theorem 3 implies that if $\|v\|_x \geq \eta$ in algorithm 1, then

$$f(x + \alpha v) \leq f(x) - \omega(\theta\eta),$$

so the cost function is decreased by at least a positive amount $\omega(\theta\eta)$. If the function is bounded below, we must reach $\|v\|_x < \eta$ after a finite number of iterations. Hence algorithm 1 converges from any starting point if the problem is bounded below.

3.3.3 Proximal Newton method with backtracking line search

As pointed out in [LSS14] the proximal Newton algorithm is readily modified to include a backtracking line search of the type used in [BV04, chapter 9]. We will analyze the following algorithm and use it in the experiments of section 3.4.

Algorithm 2 Proximal Newton algorithm with line search

Input: A starting point $x \in \mathbf{dom}\, f$, and parameters $\theta_{\min} \in (0, 1]$, $\beta \in (0, 1)$, and $\gamma \in (0, \theta_{\min}/2)$.

Repeat:

1. Compute a step v that satisfies (3.28) for some r and $\theta \geq \theta_{\min}$.
2. If $\|v\|_x$ is sufficiently small, return $x + v$.
3. Otherwise, set $x := x + \alpha v(x)$ where α is the largest number in $\{1, \beta, \beta^2, \beta^3, \ldots\}$ for which

$$x + \alpha v \in \mathbf{dom}\, f, \qquad f(x + \alpha v) \leq f(x) - \alpha\gamma\theta\|v\|_x^2. \tag{3.42}$$

To formulate a rigorous stopping condition that guarantees a bound on $f(x + v) - \inf_y f(y)$ one can use the inequality (3.30) in theorem 1, which is valid for any $\theta \in (0, 1)$, or the simpler inequality (3.35), which assumes $\theta > 0.764$.

We refer to the condition (3.42) as the *condition of sufficient decrease*. Note that the starting point of algorithm 2 is required to be in $\mathbf{dom}\, f$, so the right-

hand side in the condition of sufficient decrease is well defined in the first iteration. Alternatively, one can start at $x \in \mathbf{dom}\, g$ and use a damped Newton step in the first iteration.

The following observation extends a result for the standard Newton method with backtracking line search applied to self-concordant functions [BV04, section 9.6.4].

Theorem 4. *The stepsize selected by the backtracking line search satisfies*

$$\frac{\beta\theta}{1+\theta\|v\|_x} < \alpha \leq 1.$$

A unit stepsize is selected if $\|v\|_x \leq \theta(1-\gamma) - 1/2$.

Proof. We first note that the step size $\hat{\alpha} = \theta/(1+\theta\|v\|_x)$ satisfies the condition of sufficient decrease. This can be seen from the upper bound (3.41):

$$\begin{aligned} f(x+\hat{\alpha}v) &\leq f(x) - \hat{\alpha}\theta\|v\|_x^2 + \omega^*(\hat{\alpha}\|v\|_x) \\ &= f(x) - \omega(\theta\|v\|_x) \\ &\leq f(x) - \frac{\theta^2\|v\|_x^2}{2(1+\theta\|v\|_x)} \\ &= f(x) - \hat{\alpha}\theta\|v\|_x^2/2 \\ &\leq f(x) - \hat{\alpha}\gamma\|v\|_x^2. \end{aligned}$$

Line 3 follows from the inequality (3.23). The last step follows because $\gamma \leq \theta/2$. Since $\hat{\alpha}$ satisfies the condition of sufficient decrease, the stepsize α selected by the line search can not be less than or equal to

$$\beta\hat{\alpha} = \frac{\beta\theta}{1+\theta\|v\|_x}.$$

For the second part of the theorem, note that if $\|v\|_x \leq \theta(1-\gamma) - 1/2$ then,

again using (3.41),

$$\begin{aligned} f(x+v) &\leq f(x) - \theta\|v\|_x^2 + \omega^*(\|v\|_x) \\ &\leq f(x) - \theta\|v\|_x^2 + \frac{1}{2}\|v\|_x^2 + \|v\|_x^3 \\ &= f(x) - (\theta - 1/2 - \|v\|_x)\|v\|_x^2 \\ &\leq f(x) - \gamma\theta\|v\|_x^2. \end{aligned}$$

Line 2 follows from the first inequality in (3.22). □

Theorem 4 can be combined with the analysis of section 3.3.2 to show that algorithm 2 has the same convergence properties as algorithm 1. Choose any positive η. If $\|v\|_x > \eta$, the condition of sufficient decrease and the lower bound on the stepsize from theorem 4 guarantees

$$\begin{aligned} f(x+\alpha v) &\leq f(x) - \alpha\gamma\theta\|v\|_x^2 \\ &\leq f(x) - \beta\gamma\frac{\theta^2\|v\|_x^2}{1+\theta\|v\|_x} \\ &\leq f(x) - \beta\gamma\frac{\theta_{\min}^2\eta^2}{1+\theta_{\min}\eta}. \end{aligned}$$

(The last step follows from monotonicity of the function $t^2/(1+t)$.) If the problem is bounded below, the algorithm reaches a stopping condition $\|v\|_x \leq \eta$, for any positive η, after a finite number of iterations.

Moreover, if we choose $\theta_{\min} > 1/2$ and $\gamma < 1 - 1/(2\theta_{\min})$ then theorem 4 guarantees that for sufficiently small $\|v\|_x$, a unit stepsize is chosen and the local convergence results of section 3.3.2.1 apply.

3.4 Numerical examples

In this section we present some results for the proximal Newton method applied to (3.12). We use the Python packages CHOMPACK [AV15] and CVXOPT [ADV15] for the sparse matrix computations (evaluation of ϕ and its gradient,

Hessian, and inverse Hessian). The main purpose of the experiments is to compare the convergence properties with the theoretical results in subsections 3.3.2–3.3.3. Our implementation is not optimized, because it requires several conversions between different sparse matrix formats. Moreover the proximal Newton algorithm itself, and some key functions of CHOMPACK (such as the symbolic factorization), are implemented in Python and would be faster when implemented directly in C. This must be kept in mind when comparing the computation times for different parameter values in the experiments.

In section 3.4.1, we describe the FISTA algorithm for solving the subproblem of computing the Newton step. In section 3.4.2, we demonstrate the convergence rate of proximal Newton method via experiments using synthetic data with band patterns. In section 3.4.3, we give the convergence rate of proximal Newton method based on saprsity patterns from University of Florida matrix collections.

3.4.1 Subproblem

In the experiments a basic version of the FISTA algorithm [BT09] was used to minimize the function (3.15) in the subproblems. At iteration k of FISTA, a new estimate v^k of the solution of the subproblem is computed, by making a proximal gradient update

$$v^k = \mathrm{prox}_{th}\left(x + w - t(\nabla g(x) + \nabla^2 g(x)w)\right) - x$$

where w is the previous value v^{k-1} plus an an extrapolation term,

$$w = v^{k-1} + \frac{k-2}{k+1}\left(v^{k-1} - v^{k-2}\right).$$

From the definition of the proximal operator prox_{th}, the following relation between these variables holds:

$$\frac{1}{t}(w - v^k) \in \nabla g(x) + \nabla^2 g(x)w + \partial h(x + v^k).$$

This shows that the vector

$$r = \frac{1}{t}(w - v^k) + \nabla^2 g(x)(v^k - w) = (\frac{1}{t}I - \nabla^2 g(x))(w - v^k)$$

satisfies $r \in \nabla g(x) + \nabla^2 g(x)v^k + \partial h(x + v^k)$. In our implementation, r was used in the condition $\|r\|_{x*} \leq (1 - \theta)\|v^k\|_x$ to determine whether to accept v^k as an inexact proximal Newton step v.

To select the FISTA stepsize t, we used the simple backtracking strategy suggested in [BT09]. More sophisticated variants of FISTA, such as N83 in the TFOCS package [BCG11], or methods that use different strategies for selecting t [SGB14], are likely to lead to substantial improvements over our results. We also note that several first-order methods could be used as alternatives to FISTA, including the coordinate descent method [HSDR11] and the orthant-based method [BNO15].

3.4.2 Band patterns

In the first experiment we use a band pattern E of size $n = 1000$ with half-bandwidth 20. Band patterns are chordal, so $E' = E$ in this experiment. To generate a sample covariance matrix C we first create a sparse matrix Σ^{-1} as follows. We randomly select 80% of the entries within the band E, and set them to zero. For the remaining entries in E, we randomly generate values following a normal distribution $N(0, 1)$. A multiple of the identity is added to the matrix Σ^{-1} if it is not positive definite. We then generate $N = 10n$ samples from the distribution $N(0, \Sigma)$ and form the sample covariance matrix C. The regularization parameter in (3.12) was set to $\lambda = 0.02$.

Figure 3.5 shows the convergence of algorithm 2 with different, constant values of the parameter θ, and backtracking parameters $\gamma = 0.01$, $\beta = 1/2$. The first figure confirms the conclusions about the effect of θ in the theoretical analysis of subsection 3.3.3. It also shows that the proximal Newton method can reach a high

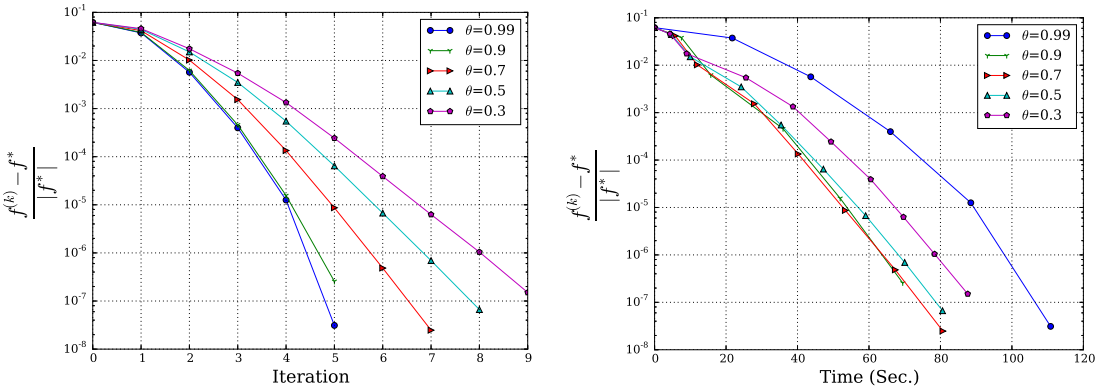


Figure 3.5: Convergence of the proximal Newton method in the first experiment, for different values of θ.

accuracy, even with very inaccurate solutions of the subproblems (low values of θ). The second figure shows the convergence versus elapsed time (on a machine with a 2.5GHz Intel Core i7 processor). The plots suggest there is a value of θ that gives the fastest convergence. Although the best value of θ and the overall solution times are likely to be quite different in a more optimized implementation of the algorithm, the figure shows the benefits that can be expected from improvements in the algorithm for the subproblem, and from strategies for adapting θ during the algorithm, as suggested in [LSS14]. Compared with algorithms in [Lu10, WST10, LT10], although they did not use band patterns for experiments, the results show that our algorithm is at least one order faster than their algorithms for problems at the same scale.

3.4.3 Sparsity patterns from University of Florida collection

In the second experiment we use three patterns from the UF collection [DH11]. Table 3.1 gives the dimension and the number of nonzeros $2|E|+n$ for each pattern, and the number of nonzeros in a chordal extension (the second and third patterns are chordal, so $E = E'$). We generate a sample covariance matrix as in the first experiment. We first generate a sparse matrix $\Sigma^{-1} \in \mathbf{S}^n_E$. A randomly selected

Name	n	nnz	nnz after extension
1138_bus	1138	4054	5392
Chem97ZtZ	2541	7361	7361
mhd4800b	4800	27520	27520

Table 3.1: Three sparsity patterns from the University of Florida collection.

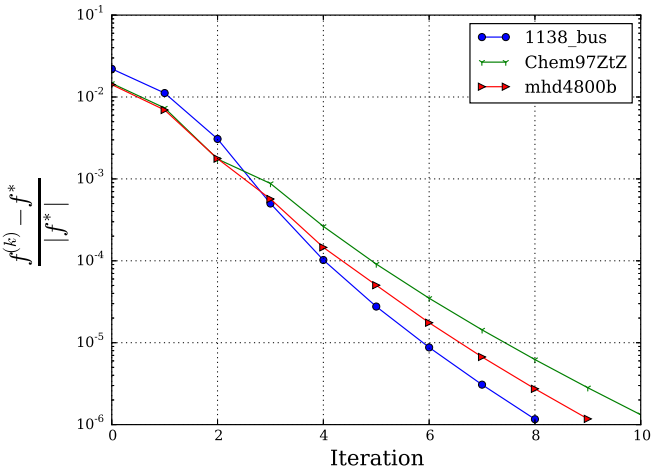


Figure 3.6: Convergence of the proximal Newton method for the three test problems in the second experiment.

subset of 30% of the entries in E are set to zero. The values of the remaining entries in E are chosen from $N(0,1)$. A multiple of the identity is added to make the matrix positive definite. We then use Σ to generate $N = 10n$ samples and form the sample covariance C.

Figure 3.6 shows the convergence of algorithm 2 for the three problems. We use $\theta = 0.5$, $\gamma = 0.01$, and $\beta = 1/2$. Even though the dimensions of the three problems are quite different, the method converges in roughly the same, small number of iterations, as is typical for the standard Newton method.

3.5 Conclusion

In this chapter, we presented the restricted sparse covariance selection problem, in which we imposed prior constraints on the sparsity pattern of the inverse covariance matrix. As a general sparsity pattern can be extended to chordal structure by chordal embedding, the constraints on the extended nonzero chordal pattern are imposed while penalizing the fill-in with an indicator function explicitly in the objective function. The log-determinant term in the cost function of this problem is self-concordant, and efficient methods exist for evaluating the matrix-vector products with its Hessian and inverse Hessian needed in the proximal Newton method.

We also presented an analysis of the proximal Newton method for minimizing a sum of a self-concordant function and a function with an inexpensive proximal mapping. The analysis extends results from [TDKC15] by taking into account inexactness of the computation of the proximal Newton steps. The conclusions are similar to the results reached in [LSS14, BNO15] under different assumptions on the smooth component of the cost function.

In the numerical examples, we applied the proximal Newton method to the restricted sparse covariance selection problem with inexact Newton steps calculated by FISTA algorithm. Preliminary numerical results indicate that the method can reach a high accuracy, even with inexact computation of the proximal Newton steps. The most important questions for further research concern the choice of algorithm for solving the subproblems, and the formulation of good strategies for adaptive control of the accuracy with which the subproblems are solved.

CHAPTER 4

Joint Graphical models of autoregressive time series

In chapter 3, we have discussed algorithms for estimating a single Gaussian graphical model. In many applications it is useful to estimate a collection of models, since the data collected are often obtained from several categories with some similarity across the categories, while each one has their uniqueness. For example, in the estimation of brain connectivity network [Fri11, QHLC14], the newtorks vary for different subjects due to individual differences, but are expected to be more similar if the corresponding subjects share many common features. Another example is gene expression measurements for lung cancer patients and brain cancer patients [MCH$^+$12]. There are some substantial commonality shared between these two groups such as the tumor-specific pathways, while the gene regulatory networks differ from each other due to different etiologies of these two diseases. For all these applications, estimating the models as one ignores the categorical differences, while estimating them separately overlooks the similarity between different structures. Therefore, in recent years, there has been a line of research [KSAX10, GLMZ11, ZW12, DWW14] focusing on estimating multiple Gaussian graphical models simultaneously, and we refer to this problem as a *joint Gaussian graphical model* throughout this chapter.

In this chapter, we analyze the joint Gaussian graphical model for autoregressive time series (joint GGM-AR). Analogous to the extension from single Gaussian graphical models to joint Gaussian graphical models, the joint Gaussian graphical

model for autoregressive time series is directly extended from the Gaussian graphical model for autoregressive time series (GGM-AR) analyzed in [SDV10, SV10]. This extension is crucial to analyze time series data with multiple categories, e.g., to analyze stock markets relations using composite indexes in different time periods, or to measure brain activity networks using fMRI scanning data from different subjects.

This chapter is organized as follows. In section 4.1 we give a review of Gaussian graphical models for time series. In section 4.2.2, based on joint static Gaussian graphical models, we propose joint Gaussian graphical models for autoregressive time series, and discuss two possible choices of cross graph penalties. In section 4.3, we present the Douglas-Rachford algorithm used for solving joint Gaussian graphical models for autoregressive time series. In section 4.4, we present our model selection methods for choosing parameters in the model. Last, in section 4.5, we present some experiment results based on synthetic data. We also discuss its applications to international stock markets analysis and fMRI brain network analysis.

4.1 Gaussian graphical models for time series

Gaussian graphical models for time series have been analyzed using both nonparametric and parametric methods. An example of nonparametric methods is the algorithm proposed by Bach and Jordan to forecast stationary Gaussian time series based on the spectral density matrix [BJ04]. More literature focus on parameteric methods [Dah00, SDV10, SV10, ALW13], where autoregressive (AR) graphical models or autoregressive moving-average (ARMA) models are used. In this chapter we discuss parametric methods for AR models, extending the results in [SDV10, SV10]. For a p-order AR Gaussian process, the sequence can be

formulated as

$$x(t) = -\sum_{k=1}^{p} A_k x(t-k) + w(t), \tag{4.1}$$

where $x(t) \in \mathbf{R}^n$ and $w(t)$ is Gaussian white noise with zero mean and covariance Σ. Given N samples $x(0), \ldots, x(N-1)$ from (4.1), our goal is to estimate Σ and A_k, $k = 0, \ldots, p$. By variable substitution, (4.1) can be reformulated as

$$B_0 x(t) = -\sum_{k=1}^{p} B_k x(t-k) + v(t), \tag{4.2}$$

with $v(t) \sim N(0, I)$, and $B_0 = \Sigma^{-1/2}$, $B_k = \Sigma^{-1/2} A_k$ for $k = 1, \ldots, p$.

4.1.1 Conditional independence

In Gaussian graphical models, components x_i and x_j are conditionally independent, given the other components of $x(t)$, if and only if $(\Sigma^{-1})_{ij} = 0$. A similar property holds for Gaussian time series.

Let us first assume that $x(t)$ is an n-dimensional Gaussian time series sequence. We define $x_i(t), x_j(t)$ as the i-th element and j-th element of $x(t)$ respectively, and $x_\alpha(t)$ as the $n-2$ dimensional vector excluding the elements $x_i(t)$ and $x_j(t)$. Suppose the whole temporal sequence of $x_\alpha(t)$ is used to predict $x_i(t)$ and $x_j(t)$, then the linear estimation residuals for $x_i(t)$ and $x_j(t)$ can be defined as

$$\begin{bmatrix} \epsilon_i(t) \\ \epsilon_j(t) \end{bmatrix} = \begin{bmatrix} x_i(t) \\ x_j(t) \end{bmatrix} - \sum_{k=-\infty}^{\infty} A_k^{\text{opt}}(t-k) x_\alpha(k),$$

where A_k^{opt} is the optimal $2 \times (n-2)$ coefficient matrix for optimal linear estimation. Then $x_i(t)$ and $x_j(t)$ are conditionally independent if and only if $\text{cov}\,(\epsilon_i(t), \epsilon_j(t)) = 0$. In the frequency domain, this relation can be represented by the power spectrum $S(\omega)$, which is defined as

$$S(\omega) = \sum_{k=-\infty}^{+\infty} R_k e^{-jk\omega}, \text{ where } R_k = \mathbf{E} x(t+k) x(t)^T.$$

It has been shown in [Bri01, Dah00] that $x_i(t)$ and $x_j(t)$ are conditionally independent if and only if

$$\left(S(\omega)^{-1}\right)_{ij} = 0, \quad \text{for all } \omega. \tag{4.3}$$

For an autoregressive process (4.2), the inverse spectrum can be expressed as

$$S(\omega)^{-1} = Y_0 + \sum_{k=1}^{p} \left(Y_k e^{-\mathrm{i}k\omega} + Y_k^T e^{\mathrm{i}k\omega}\right), \tag{4.4}$$

where

$$Y_k = \sum_{l=0}^{p-k} B_l^T B_{l+k}, \quad k = 0, \ldots, p. \tag{4.5}$$

Therefore, the conditional independence (4.3) can be written as

$$\left(\sum_{l=0}^{p-k} B_l^T B_{l+k}\right)_{ij} = 0, \quad k = 0, \ldots, p. \tag{4.6}$$

4.1.2 Estimation for Gaussian autoregressive time series

We provide three approaches to estimate Σ and A_k, $k = 1, \ldots, p$, in Gaussian graphical models for time series [SDV10, BJR11].

Least square linear prediction Given a multivariate time series sequence $x(t)$, the optimal linear prediction problem can be written as

$$\hat{x}(t) = -\sum_{k=1}^{p} A_k x(t-k),$$

where the prediction of $x(t)$ is based on past values $x(t-1), \ldots, x(t-p)$. The prediction error is characterized by

$$\epsilon(t) \;=\; x(t) - \hat{x}(t) \;=\; x(t) + \sum_{k=1}^{p} A_k x(t-k).$$

The optimal linear prediction coefficients A_k can be obtained by minimizing $\mathbf{E}\|\epsilon(t)\|_2^2$, or equivalently

$$\text{minimize} \quad \mathbf{tr}\left(A\mathcal{T}(R)A^T\right),$$

where the notations are defined as follows:

- $A = \begin{bmatrix} I & A_1 & \cdots & A_p \end{bmatrix}$, $R = \begin{bmatrix} R_0 & R_1 & \cdots & R_p \end{bmatrix}$.
- $\mathbf{M}^{n,p}$ representing sets of matrices X satisfying

  $X = \begin{bmatrix} X_0 & X_1 & \cdots & X_p \end{bmatrix}$, where $X_0 \in \mathbf{S}^n$, and $X_k \in \mathbf{R}^{n \times n}$ for $k = 1, \ldots, p$.
- $\mathcal{T}$ is a block-Toeplitz operator $\mathcal{T} : M^{n,p} \to \mathbf{S}^{n(p+1)}$ defined as

$$\mathcal{T}(S_0, S_1, \ldots, S_p) = \begin{bmatrix} S_0 & S_1^T & \cdots & S_p^T \\ S_1 & S_0 & \cdots & S_{p-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ S_p & S_{p-1} & \cdots & S_0 \end{bmatrix}. \tag{4.7}$$

In practice, $\mathcal{T}(R)$ is replaced by the sample covariance C, which can be obtained from the *windowed* estimate or the *non-windowed* estimate.

- For the non-windowed estimate, suppose we have observations $x(1), \ldots, x(N)$, then

$$C = \frac{1}{N-p} H H^T,$$

  where

$$H = \begin{bmatrix} x(p+1) & x(p+2) & \cdots & x(N) \\ x(p) & x(p+1) & \cdots & x(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ x(1) & x(2) & \cdots & x(N-p) \end{bmatrix}.$$

- For the windowed estimate,

$$C = \frac{1}{N} H H^T,$$

  where

$$H = \begin{bmatrix} x(1) & x(2) & \cdots & x(p+1) & \cdots & x(N) & 0 & \cdots & 0 \\ 0 & x(1) & \cdots & x(p) & \cdots & x(N-1) & x(N) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x(1) & \cdots & x(N-p) & x(N-p+1) & \cdots & x(N) \end{bmatrix}.$$

By replacing $\mathcal{T}(R)$ with C, the estimation problem reduces to

$$\text{minimize} \quad \mathbf{tr}\left(ACA^T\right). \tag{4.8}$$

The optimality conditions of (4.8) can be expressed as

$$\begin{bmatrix} C_{00} & C_{01} & \cdots & C_{pp} \\ C_{10} & C_{11} & \cdots & C_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ C_{p0} & C_{p1} & \cdots & C_{pp} \end{bmatrix} \begin{bmatrix} I \\ A_1^T \\ \vdots \\ A_p^T \end{bmatrix} = \begin{bmatrix} \hat{\Sigma} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{4.9}$$

where $\hat{\Sigma} = ACA^T$. Comparing the non-windowed estimate and the windowed estimate, the non-windowed estimate is slightly more accurate when N is small [SM97]. On the other hand, the windowed estimate has important theoretical and practical properties. If the windowed estimate is used, C is block-Toeplitz, and the solution of (4.9) always provide a stable model. Furthermore, when C is block-Toeplitz, the equations in (4.9) has the same form as Yule-Walker equations, and can be solved efficiently using classical methods like Levinson-Durbin recursion [SM97].

Conditional maximum likelihood Given the observations $x(1), \ldots, x(N)$, the conditional likelihood of an autoregressive process (4.1) can be expressed as

$$\begin{aligned} &\frac{1}{\left((2\pi)^n \det \Sigma\right)^{(N-p)/2}} \exp\left(-\frac{1}{2}\sum_{t=p+1}^{N} \mathbf{x}(t)^T A^T \Sigma^{-1} A \mathbf{x}(t)\right) \\ &= \left(\frac{\det B_0}{(2\pi)^{n/2}}\right)^{N-p} \exp\left(-\frac{1}{2}\sum_{t=p+1}^{N} \mathbf{x}(t)^T B^T B \mathbf{x}(t)\right), \end{aligned} \tag{4.10}$$

where $\mathbf{x}(t)$ is a $(p+1)n$ vector defined by

$$\mathbf{x}(t) = \left(x(t), x(t-1), \ldots, x(t-p)\right),$$

and

$$A = \begin{bmatrix} I & A_1 & \cdots & A_p \end{bmatrix}, \quad B = \begin{bmatrix} B_0 & B_1 & \cdots & B_p \end{bmatrix}.$$

Taking the logarithm of (4.10), we can obtain the log-likelihood function as

$$\mathcal{L}(B) = \frac{N-p}{2}\left(2\log\det B_0 - \mathbf{tr}(CB^TB)\right). \tag{4.11}$$

where C follows the non-windowed estimate. The conditional maximum likelihood estimation can be formulated as

$$\text{minimize} \quad -2\log\det B_0 + \mathbf{tr}(CB^TB). \tag{4.12}$$

The optimality conditions of (4.12) are the same as (4.9). Thus the conditional maximum likelihood estimation is equivalent to least square estimation with non-windowed estimate C.

Maximum entropy estimation The maximum entropy problem introduced by [Bur75, ALW13] is formulated as:

$$\begin{array}{ll} \text{maximize} & \frac{1}{2\pi}\int_{-\pi}^{\pi}\log\det S(\omega)d\omega \\ \text{subject to} & \frac{1}{2\pi}\int_{-\pi}^{\pi}S(\omega)e^{jk\omega}d\omega = \bar{R}_k, \quad k=0,\ldots,p. \end{array} \tag{4.13}$$

The matrices $\bar{R}_k$ are given as the empirical covariance matrices following $\bar{R}_k = \frac{1}{N}\sum_{t=1}^{N-k}x(t+k)x(t)^T$. If we change the sign of the objective, the Lagrangian can be formulated as

$$-\frac{1}{2\pi}\int_{-\pi}^{\pi}\log\det S(\omega)d\omega + \mathbf{tr}(Z_0(R_0-\bar{R}_0)) + 2\sum_{k=1}^{p}\mathbf{tr}\left(Z_k^T(R_k-\bar{R}_k)\right).$$

Differentiating with respect to R_k gives

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}S^{-1}(\omega)e^{jk\omega}d\omega = Z_k, \quad k=0,\ldots,p. \tag{4.14}$$

It is easy to see that (4.4) is the inverse transform of (4.14), and therefore $Y_k = Z_k$ for $0 \le k \le p$. For the following part, notation Y_k is used instead of Z_k. Based on (4.4), we denote $Y(\omega) = S(\omega)^{-1}$, so the dual of (4.13) can be formulated as

$$\text{minimize} \quad -\frac{1}{2\pi}\int_{-\pi}^{\pi}\log\det Y(\omega)d\omega + \mathbf{tr}(Y_0^T\bar{R}_0) + 2\sum_{k=1}^{p}\mathbf{tr}(Y_k^T\bar{R}_k) - n.$$

By using Jensen's formula [Ahl79, page 207], we have

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\log\det Y(\omega)d\omega = \log\det(B_0^T B_0).$$

Furthermore, by using the relation (4.5),

$$\mathbf{tr}(Y_0^T\bar{R}_0) + 2\sum_{k=1}^{p}\mathbf{tr}(Y_k^T\bar{R}_k) = \mathbf{tr}(\mathcal{T}(\bar{R})B^TB) = \mathbf{tr}(CB^TB),$$

where C is the windowed estimate. Then the dual problem can be reduced to

$$\text{minimize} \quad -2\log\det B_0 + \mathbf{tr}(CB^TB). \tag{4.15}$$

Problem (4.15) has the same optimality conditions as in (4.9). Therefore, all the three estimation methods have the same form of optimality conditions, only with slightly different definitions for C.

4.1.3 Penalized estimation for Gaussian time series graphical models

By combining the estimation problem (4.15) with the conditional independence constraints (4.6), we can be formulate the constrained Gaussian time series graphical models as

$$\begin{aligned} &\text{minimize} && -2\log\det B_0 + \mathbf{tr}(CB^TB) \\ &\text{subject to} && \big(\mathcal{D}_k(B^TB)\big)_{ij} = 0, \quad k = 0,\ldots,p, \quad (i,j)\notin E, \end{aligned} \tag{4.16}$$

where $\mathcal{D}$ is the adjoint operator of $\mathcal{T}$, and is defined as

$$\mathcal{D}(X) = (\mathcal{D}_0(X), \mathcal{D}_1(X), \ldots, \mathcal{D}_p(X)),$$

with

$$\begin{cases} \mathcal{D}_0(X) = X_{00} + X_{1,1} + \cdots + X_{p,p}, & k = 0 \\ \mathcal{D}_k(X) = 2(X_{k0} + X_{k+1,1} + \cdots + X_{p,p-k}), & k = 1,\ldots,p. \end{cases} \tag{4.17}$$

By introducing an indicator function $\tilde{h} : \mathbf{S}^n \times \mathbf{R}^{n\times n} \times \cdots \times \mathbf{R}^{n\times n} \to \mathbf{R}$,

$$\tilde{h}(Y)_{k,ij} = \begin{cases} 0, & (i,j)\notin E, \quad k = 0,\ldots,p \\ +\infty, & \text{otherwise}, \end{cases}$$

problem (4.16) can be written as

$$\text{minimize} \quad -2\log\det B_0 + \mathbf{tr}(CB^TB) + \tilde{h}\left(\mathcal{D}(B^TB)\right). \tag{4.18}$$

The indicator function $\tilde{h}$ can be extended to other penalty functions. If $\tilde{h}$ is chosen as a lasso penalty, $\tilde{h}$ serves to promote sparsity in the topology. For example, one choice of $\tilde{h}$ is

$$\tilde{h}(Y) = \gamma \sum_{i>j} \max_{k=0,\ldots,p} \{|Y_{k,ij}|, |Y_{k,ji}|\}. \tag{4.19}$$

Problem (4.18) is non-convex because of the quadratic term B^TB, even though the penalty function $\tilde{h}$ is convex. If we replace B^TB with a variable $X \in \mathbf{S}^{n(p+1)}$, problem (4.18) is relaxed to a convex form:

$$\begin{array}{ll} \text{minimize} & -\log\det X_{00} + \mathbf{tr}(CX) + \tilde{h}\left(\mathcal{D}(X)\right) \\ \text{subject to} & X \succeq 0, \end{array} \tag{4.20}$$

where $X_{l,(l+k)}$ denotes sub-block $(l, l+k)$ of X. It has been shown in [SV10] that the relaxation is exact if C is block-Toeplitz (e.g., C is the windowed estimate). We will show this result in section 4.1.4 using duality theory. Since C is a block-Toeplitz matrix, $\mathbf{tr}(CX)$ can be expressed as

$$\mathbf{tr}\left(CX\right) = \mathbf{tr}(\bar{C}^T\mathcal{D}(X)),$$

where $\bar{C} = \begin{bmatrix} C_0 & C_1 & \cdots & C_p \end{bmatrix}$. To make the notations simple, we define

$$h(Y) = \mathbf{tr}(\bar{C}^TY) + \tilde{h}\left(Y\right), \tag{4.21}$$

so $h(\mathcal{D}(X)) = \mathbf{tr}(CX) + \tilde{h}\left(\mathcal{D}(X)\right)$. With this new notation, (4.20) can be formulated as

$$\begin{array}{ll} \text{minimize} & -\log\det X_{00} + h(\mathcal{D}(X)) \\ \text{subject to} & X \succeq 0. \end{array} \tag{4.22}$$

4.1.4 Optimality conditions

In this section we prove that the solution of (4.22) has rank n. Assume Slater's condition holds for (4.22). X is optimal if it is feasible, *i.e.*,

$$X \succeq 0, \qquad X_{00} \succ 0, \qquad \mathcal{D}(X) \in \mathbf{dom}\, h,$$

and there exists a $Z \in \partial h(\mathcal{D}(X))$ such that

$$\begin{bmatrix} X_{00}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \preceq \mathcal{T}(Z), \qquad \left(\mathcal{T}(Z) - \begin{bmatrix} X_{00}^{-1} & 0 \\ 0 & 0 \end{bmatrix}\right) X = 0. \tag{4.23}$$

The first condition implies $\mathcal{T}(Z) \succ 0$ [SDV10]. This can be verified by induction on p. Firstly, if $p = 0$, it is trivial that $\mathcal{T}(Z) \succ 0$. Secondly, suppose this property holds for $p-1$, *i.e.*, the leading $np \times np$ submatrix $\mathcal{T}(Z)$ is positive definite, then by exploiting the Toeplitz structure, $\mathcal{T}(Z)$ can be partitioned as

$$\mathcal{T}(Z) = \begin{bmatrix} Z_0 & U^T \\ U & V \end{bmatrix},$$

where $V \succ 0$. The Schur complement of V in $\mathcal{T}(Z)$ is

$$Z_0 - U^T V^{-1} U \succeq X_{00}^{-1} \succ 0. \tag{4.24}$$

Given with the condition $V \succ 0$, (4.24) shows $\mathcal{T}(Z) \succ 0$. Then, it is trivial to see $\mathcal{T}(Z) - \begin{bmatrix} X_{00}^{-1} & 0 \\ 0 & 0 \end{bmatrix}$ has rank np. Therefore the second condition of (4.23) implies $\mathbf{rank}(X) = n$ since X is at least rank n. Given the optimal Z, we can find X from

$$\begin{bmatrix} Z_0 & Z_1^T & \cdots & Z_p^T \\ Z_1 & Z_0 & \cdots & Z_{p-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ Z_p & Z_{p-1} & \cdots & Z_0 \end{bmatrix} \begin{bmatrix} X_{00} \\ X_{10} \\ \vdots \\ X_{p0} \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{4.25}$$

If we define $B_k = -X_{00}^{-1}X_{0k}$ and $\Sigma = X_{00}^{-1}$, then

$$\begin{bmatrix} Z_0 & Z_1^T & \cdots & Z_{p-1}^T \\ Z_1 & Z_0 & \cdots & Z_{p-2}^T \\ \vdots & \vdots & \ddots & \vdots \\ Z_{p-1} & Z_{p-2} & \cdots & Z_0 \end{bmatrix} \begin{bmatrix} B_1^T \\ B_2^T \\ \vdots \\ B_p^T \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_p \end{bmatrix} \tag{4.26}$$

and

$$\Sigma = Z_0 - \begin{bmatrix} Z_1^T & Z_2^T & \cdots & Z_p^T \end{bmatrix} \begin{bmatrix} B_1^T \\ B_2^T \\ \vdots \\ B_p^T \end{bmatrix}. \tag{4.27}$$

We can obtain the analytical solution easily for Σ and B_k by solving (4.26) and (4.27). Also, we can see (4.25) has the same form as the Yule-Walker equations. Classical methods like Levinson-Durbin recursion can be used to solve (4.25) [SM97].

4.1.5 Dual problem

To derive the dual, first we rewrite problem (4.20) as

$$\begin{array}{ll} \text{minimize} & -\log\det X_{00} + h(Y) \\ \text{subject to} & \mathcal{D}(X) = Y \\ & X \succeq 0. \end{array} \tag{4.28}$$

Then the dual of (4.28) can be formulated as

$$\begin{array}{ll} \text{maximize} & \log\det U - h^*(Z) + n \\ \text{subject to} & \begin{bmatrix} U & 0 \\ 0 & 0 \end{bmatrix} \preceq \mathcal{T}(Z), \end{array}$$

where h^* is the conjugate of h,

$$h^*(y) = \sup_{x \in \mathbf{dom}\, h} \left(y^T x - f(x) \right).$$

4.2 Joint Gaussian graphical models

In this section, we first review the current work on joint static Gaussian graphical models, and then provide the extensions to time series.

4.2.1 Joint static Gaussian graphical models

Joint static Gaussian graphical models are the methods to jointly estimate multiple graphical models of related but different distributions [KSAX10, GLMZ11, ZW12, DWW14]. Assume there are K distinct Gaussian graphical models with the distributions $N(0, \Sigma^{(k)})$ for $k = 1, \ldots, K$, and each model has similar structures or the same structure but different coefficients in $(\Sigma^{(k)})^{-1}$. $N^{(k)}$ observations are obtained from each model. If we use $X^{(k)}$ to represent the precision matrix and $C^{(k)}$ to represent the empirical covariance matrix for graph k, the log-likelihood is

$$\ell\left(X^{(1)}, \ldots, X^{(K)}\right) = \frac{1}{2} \sum_{k=1}^{K} N^{(k)} \left(\log\det X^{(k)} - \mathbf{tr}(C^{(k)} X^{(k)})\right). \tag{4.29}$$

In order to estimate $X^{(k)}$ for different models, one approach is to estimate them independently. However, this approach overlooks the structural similarity among different models. Moreover, when the estimations are taken separately, the number of data samples for each estimation is decreased, so that the estimation accuracy is also decreased. If the data samples are limited, this approach is subject to be over-fitting. A better approach is to estimate the collection of graphical models simultaneously with a cross graph penalty used to promote the common structure across graphs. This approach is referred to as a *joint Gaussian graphical model* (Joint GGM)[MLF+14, DWW14, GLMZ11, QHLC14], which can be formulated as:

$$\underset{X^{(1)},\ldots,X^{(K)}}{\text{minimize}} \quad -\ell\left(X^{(1)}, \ldots, X^{(K)}\right) + \gamma \sum_{k=1}^{K} \sum_{i \neq j} |X_{ij}^{(k)}| + h(X^{(1)}, \ldots, X^{(K)}), \tag{4.30}$$

where the first regularization term is taken to penalize off-diagonal elements of the precision matrices, and the penalty function $h(X^{(1)}, \ldots, X^{(K)})$ is taken to encourage shared characteristics among different models. Two classes of penalty terms have been proposed, an *edge based* penalty [GLMZ11, DWW14] and a *node based* penalty [TLM$^+$14, MLF$^+$14]. The node based penalty assumes that the similarities and differences between graphical models are driven by individual nodes, so the connectivity patterns of one node to all the other nodes are shared among those K graphs. As for the edge based penalty [GLMZ11, ZW12, MCH$^+$12, DWW14], it assumes that the connectivity of each individual edge is shared among all the graphs. In this thesis, we focus on the edge based penalty, for which two different penalty functions are considered.

Fused graphical lasso (FGL) The *fused lasso penalty* [TSR$^+$05] penalizes the difference between all corresponding edges in the graphs. h can be formulated as

$$h(X^{(1)}, \ldots, X^{(K)}) = \lambda \sum_{k \neq k'} \sum_{i,j} |X_{ij}^{(k)} - X_{ij}^{(k')}|,$$

where λ is a nonnegative tuning parameter. When λ is large, more elements will be identical across graphs. This penalty penalizes the difference across graphs aggressively, and it encourages not only the structural similarity, but also similar edge values.

Group graphical lasso (GGL) The *group lasso penalty* [YL07] penalizes edges in the same position for all K graphs using ℓ_2-norm, and thus encourages a similar sparsity pattern across graphs. The penalty term can be formulated as

$$h(X^{(1)}, \ldots, X^{(K)}) = \lambda \sum_{i,j} \sqrt{\sum_{k=1}^{K} (X_{ij}^{(k)})^2}.$$

Compared with FGL, GGL only encourages a shared sparsity pattern, while FGL encourages shared edge values.

4.2.2 Joint Gaussian graphical model for autoregressive time series

In this section, we extend joint Gaussian graphical models to autoregressive time series. Suppose there are K similar Gaussian graphical models for autoregressive time series, we extend (4.18) to multiple graphs with an additional term introduced to promote a common structure across graphs. This estimation problem can be formulated as

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{K}\left(-2\log\det B_0^{(k)}+h_{\mathrm{i}}(Y^{(k)})\right)+h_{\mathrm{o}}(Y^{(1)},\ldots,Y^{(K)}) \\ \text{subject to} & Y^{(k)}=\mathcal{D}\left((B^{(k)})^TB^{(k)}\right), \qquad k=1,\ldots,K, \end{array} \tag{4.31}$$

where $h_{\mathrm{i}}(Y^{(k)})$ is introduced to promote sparsity within the k-th graphical model, and $h_{\mathrm{o}}(Y^{(1)},...,Y^{(K)})$ is introduced to promote a common structure across graphs. As analogous to joint static Gaussian graphical models, we have two choices of penalty function $h_{\mathrm{o}}(Y)$: the *fused Graphical lasso for autoregressive time series* (FGL-AR) and the *group Graphical lasso for autoregressive time series* (GGL-AR).

- FGL-AR:

$$h_{\mathrm{o}}(Y^{(1)},\ldots,Y^{(K)})=\lambda\sum_{k\neq k'}\sum_{l=0,\ldots,p}\sum_{i,j}|Y_{l,ij}^{(k)}-Y_{l,ij}^{(k')}|.$$

- GGL-AR:

$$h_{\mathrm{o}}(Y^{(1)},\ldots,Y^{(K)})=\lambda\sum_{l=0,\ldots,p}\sum_{i,j}\sqrt{\sum_{k=1}^{K}\left(Y_{l,ij}^{(k)}\right)^2}.$$

In this chapter, we focus on problems where similarity of edge structures is of more interest than edges values. Formulation (4.31) with the GGL-AR penalty is used in the remaining part of this chapter. Formulation (4.31) is non-convex due to the quadratic term $(B^{(k)})^TB^{(k)}$, even though $h_{\mathrm{i}}(Y^{(k)})$ and $h_{\mathrm{o}}(Y^{(1)},\ldots,Y^{(K)})$ are convex. As a convex relaxation, we make a change of variables $X^{(k)}=(B^{(k)})^TB^{(k)}$, so

$$Y^{(k)}=\mathcal{D}(X^{(k)}), \qquad k=1,\ldots,K. \tag{4.32}$$

Then (4.31) can be reformulated as:

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{K}\left(-\log\det X_{00}^{(k)}+h_{\mathrm{i}}(Y^{(k)})\right)+h_{\mathrm{o}}(Y^{(1)},\ldots,Y^{(K)}) \\ \text{subject to} & Y^{(k)}=\mathcal{D}\left(X^{(k)}\right) \\ & X^{(k)}\succeq 0, \quad k=1,\ldots,K. \end{array} \tag{4.33}$$

Assume Slater's condition holds, $X^{(k)}$ is optimal if it is feasible, *i.e.*,

$$X^{(k)}\succeq 0, \quad X_{00}^{(k)}\succ 0, \quad \mathcal{D}(X^{(k)})\in \mathbf{dom}\, h_{\mathrm{i}}, \quad \left(\mathcal{D}(X^{(1)}),\ldots,\mathcal{D}(X^{(K)})\right)\in \mathbf{dom}\, h_{\mathrm{o}}$$

and there exists $Z=(Z^{(1)},\ldots,Z^{(K)})$, with

$$Z^{(k)}\in \partial h_{\mathrm{i}}(\mathcal{D}(X^{(k)}))+\partial_{Y^{(k)}} h_{\mathrm{o}}\left(\mathcal{D}(X^{(1)}),\cdots,\mathcal{D}(X^{(K)})\right),$$

where $\partial_{Y^{(k)}} h_{\mathrm{o}}\left(Y^{(1)},\cdots,Y^{(K)}\right)$ is the subdifferential of h_{o} with respect to its k-th argument, such that

$$\begin{bmatrix} \left(X_{00}^{(k)}\right)^{-1} & 0 \\ 0 & 0 \end{bmatrix} \preceq \mathcal{T}(Z^{(k)}), \qquad \left(\mathcal{T}(Z^{(k)})-\begin{bmatrix} \left(X_{00}^{(k)}\right)^{-1} & 0 \\ 0 & 0 \end{bmatrix}\right) X^{(k)}=0. \tag{4.34}$$

Analogous to the analysis of single Gaussian graphical models for autoregressive time series (4.23), the first condition in (4.34) implies $\mathcal{T}(Z^{(k)})\succ 0$, and the second condition implies $\mathbf{rank}(X^{(k)})=n$. Thus, the relaxation $X^{(k)}=(B^{(k)})^T B^{(k)}$ is exact.

4.3 Algorithms

In this section, we apply the Douglas-Rachford method (Spingarn's method) introduced in chapter 2 section 2.3.1 to solve (4.31). First we reformulate (4.31) as:

$$\text{minimize} \quad \sum_{k=1}^{K} f(X^{(k)})+g(\mathcal{A}(X^{(1)}),\mathcal{A}(X^{(2)}),\cdots,\mathcal{A}(X^{(K)})). \tag{4.35}$$

- f is an indicator function of positive semidefinite cone $\mathbf{S}_{+}^{(p+1)n}$.

- $\mathcal{A} : \mathbf{S}^{(p+1)n} \to \mathbf{S}^n_{++} \times M^{n,p} \times M^{n,p}$ is a linear mapping:

$$\mathcal{A}(X^{(k)}) = \left(X^{(k)}_{00}, \mathcal{D}(X^{(k)}), \mathcal{D}(X^{(k)})\right).$$

- $g : (\mathbf{S}^n \times M^{n,p} \times M^{n,p}) \times \cdots \times (\mathbf{S}^n \times M^{n,p} \times M^{n,p}) \to \mathbf{R}$ is defined as

$$g(U, Y, Z) = \sum_{k=1}^{K} \left(-\log\det U^{(k)} + h_{\rm i}(Y^{(k)})\right) + h_{\rm o}(Z^{(1)}, \ldots, Z^{(K)}).$$

- $h_{\rm i}(Y^{(k)}) = \mathbf{tr}(\bar{C}^T Y^{(k)}) + \gamma \sum_{i>j} \max_{l=0,\ldots,p} \{|Y^{(k)}_{l,ij}|, |Y^{(k)}_{l,ji}|\}$.

- $h_{\rm o}(Z^{(1)}, \ldots, Z^{(K)}) = \lambda \sum_{l=0,\ldots,p} \sum_{i,j} \sqrt{\sum_{k=1}^{K} \left(Z^{(k)}_{l,ij}\right)^2}$.

The major operations involved with the Douglas-Rachford method are summarized as follows:

Proximal operator of f This is a projection onto the positive semidefinite cone of order $(p+1)n$.

Evaluation of $\left(I + t^2 \mathcal{A}^{\rm adj}\mathcal{A}\right)^{-1}$ The adjoint mapping of $\mathcal{A}$ is

$$\mathcal{A}^{\rm adj}(U, Y, Z) = \begin{bmatrix} U & 0 \\ 0 & 0 \end{bmatrix} + \mathcal{T}(Y) + \mathcal{T}(Z).$$

To evaluate $\left(I + t^2 \mathcal{A}^{\rm adj}\mathcal{A}\right)^{-1}$, we need to solve equations of the form (4.36),

$$X + t^2 \mathcal{A}^{\rm adj}\left(\mathcal{A}\left(X\right)\right) = B, \tag{4.36}$$

i.e.,

$$X + t^2 \begin{bmatrix} X_{00} & 0 \\ 0 & 0 \end{bmatrix} + 2t^2 \mathcal{T}(\mathcal{D}(X)) = B. \tag{4.37}$$

This is a linear equation problem, and the solution X can be computed by solving

$$\begin{aligned} X_{kk} + \tfrac{2t^2}{1+t^2}\mathcal{D}_0(X) &= \tfrac{1}{1+t^2} B_{kk}, \quad k = 0, \\ X_{kk} + 2t^2 \mathcal{D}_0(X) &= B_{kk}, \qquad k \geq 1. \end{aligned}$$

Proximal operator of g The proximal operator of g can be evaluated separately for $-\sum_{k=1}^{K} \log\det U^{(k)}$, $\sum_{k=1}^{K} h_{\mathrm{i}}(Y^{(k)})$, and $h_{\mathrm{o}}(Z^{(1)}, \ldots, Z^{(K)})$.

- The proximal operator of $-\log\det U^{(k)}$ can be computed by eigenvalue decomposition.
- The proximal operator of $h_{\mathrm{i}}(Y^{(k)})$ can be computed by projection onto an ℓ_1-norm ball plus Moreau decomposition.
- The proximal operator of $h_{\mathrm{o}}(Z^{(1)}, \ldots, Z^{(K)})$ can be computed by projection onto an ℓ_2-norm ball plus Moreau decomposition.

The details of evaluating these proximal operators can be found in chapter 2 section 2.2.2.

4.4 Model selections

The graphical models we obtain from estimation are affected by the tuning parameters $\{\lambda_1, \lambda_2\}$. If the estimation is used to aid data analysis and hypothesis testing, the choice of tuning parameters is usually guided by practical considerations such as model stability and interpretability. For instance, when we iterate though a set of tuning parameters, if some edge remains in the graph with increasing tuning parameters, and this infers that the connectivity is strong. For applications where there is no practical clue, we prefer using model selection methods to choose the tuning parameters. Two model selection approaches are included in this chapter. The first approach is based on information theoretical criteria such as *Akaike Information Criteria* (AIC) and *Bayes Information Criteria* (BIC) [BA02], which are defined as follows:

$$\begin{aligned} \mathrm{AIC} &= -2\mathcal{L} + 2k_e, \\ \mathrm{BIC} &= -2\mathcal{L} + k_e \log N, \end{aligned} \tag{4.38}$$

where $\mathcal{L}$ is the log-likelihood of the ML estimate, N is the total number of samples $\sum_{k=1}^{K} N^{(k)}$, and k_e is the effective number of parameters.

Analogous to (4.11), the log-likelihood $\mathcal{L}$ of K autoregressive processes is given by

$$\mathcal{L} = \sum_{k=1}^{K} \frac{N^{(k)} - p}{2} \left(\log\det X_{00}^{(k)} - \mathbf{tr}(C^{(k)} X^{(k)}) \right), \tag{4.39}$$

where $X^{(k)}$ is the optimal solution for each graphical model, and

$$k_e = \frac{n(n+1)K}{2} - \sum_{k=1}^{K} |\mathcal{V}^{(k)}| + p \sum_{k=1}^{K} (n^2 - 2|\mathcal{V}^{(k)}|), \tag{4.40}$$

where $|\mathcal{V}^{(k)}|$ is the total number of conditionally independent pairs of variables for graphical model k.

Another approach is using M-fold cross validation. We randomly split the dataset into M equal sized subsets. Of those subsets, a single subset is used for validation and the remaining subsets are used for training. The cross-validation process is repeated so that each subset is used exactly once for validation. The cross validation result is produced by averaging the results from all the repetitions. In particular, first we select a validation subset, and use the remaining $M-1$ subsets for training to obtain the estimation result $\hat{X}$. Then, we substitute $\hat{X}$ to the validation subset, and calculate the log-likelihood. This procedure repeats for M times, and for each time a different subset is used for validation. Finally, we sum up those K log-likelihood results and use this result as the cross validation score for the current set of parameters. The set of parameters corresponding to the minimum negative log-likelihood is chosen as the model selection result. The formula for cross validation can be expressed as in (4.41):

$$\mathrm{CV}(\lambda_1, \lambda_2) = \sum_{d=1}^{D} \sum_{k=1}^{K} \left(-\log\det \hat{X}_{00}^{(k,-d)} + \mathbf{tr}(C^{(k,d)} \hat{X}^{(k,-d)}) \right) \tag{4.41}$$

where $C^{(k,d)}$ is the empirical covariance matrix for graphical model k using data subset d, $\hat{X}^{(k,-d)}$ is the estimation result for graphical model k excluding data subset d. An experiment of model selections will be provided in section 4.5.1.

4.5 Numerical experiments

In this section, we compare model selection methods via synthetic experiments. We also provide both synthetic data and real data examples to demonstrate the performance of joint Gaussian graphical models for autoregressive time series.

4.5.1 Model selections

To test model selection methods, we set the testing environment as follows:

- Synthetic data generation. We set $K = 3, n = 100, p = 1$ and generate three graphs with the same pattern but different coefficient values for each A_k. To generate the data, we first create a sparse symmetric pattern matrix E. In the lower triangular part of E, 99% entries are randomly selected to be zeros. For the coefficients in the AR process, A_0 is chosen to be the identity matrix, and A_k is chosen to follow the sparsity pattern E for $k = 1, \ldots, p$. For each non-zero entry in A_k, we set the value as 0.5 or -0.5 with equal probability. The covariance Σ is set as the identity matrix. This procedure is repeated until we obtain a stable AR process. We then generate $N^{(k)}$ samples from the AR process with $N^{(k)} = 0.5n, n, 2n, 4n, 8n, 12n, 16n$, respectively.

- Model parameters. We test all different combinations of parameters from $\gamma = 0.005, 0.01$ and $\lambda = 0.0005, 0.001, 0.002, 0.005, 0.01, 0.02$.

- Number of folds in cross validation. $M = 5$.

- Measurement score. F_1 score is used as a reference for the comparison of model selection methods. It is defined as

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}},$$

  where FP denotes False Positive, TP denotes True Positive, FN denotes False Negative, recall is defined as TP/(TP+FN), and precision is defined

as TP/(TP+FP). These values (TP, FP, FN) are calculated by comparing the estimated topology with the true topology, where nonzero entries are labeled as positive and zeros are labeled as negative. The graph topology is obtained by comparing the ℓ_∞-norms of the entries of partial coherence $R(\omega)$ [Bri01, Dah00], $\rho_{ij} = \sup_\omega |R(\omega)_{ij}|$, with a given threshold (0.1 is used in our experiment), where the partial coherence is defined as

$$R(\omega) = \mathbf{diag}(S(\omega)^{-1})^{-1/2} S(\omega)^{-1} \mathbf{diag}(S(\omega)^{-1})^{-1/2}.$$

The experimental results of model selection methods and F_1 score are shown in Fig. 4.5.1. It shows that BIC and cross validation are effective as both the minimums of these two curves match the maximum of F_1 score. BIC tends to choose simpler models than AIC theoretically, and it also works better than AIC in the experiment. Cross validation is the best one, but it is computationally expensive. In practice, if the problem size is small, cross validation is preferred. If computational time is a big concern, BIC can be used as an alternative.

4.5.2 Small examples of synthetic data

- **Setting 1.** We use the same setting ($K = 3, n = 100, p = 1$) and the same A_k as in the model selection experiments.
- **Setting 2.** We set $K = 3, n = 100, p = 1$. In order to generate three graphs with similar patterns, first we generate the shared pattern of A_k using the same procedure as in setting 1 (or model selection experiments). The nonzero density in the shared pattern is 1%. Then for each individual graph, we randomly generate additional 0.2% nonzero entries, and each entry value is set to be equal to 0.5 or -0.5 with the same probability.

Fig. 4.2, Fig. 4.3 and Fig. 4.4 demonstrate the estimation results with different numbers of samples. For each estimate, the optimal parameters are chosen by

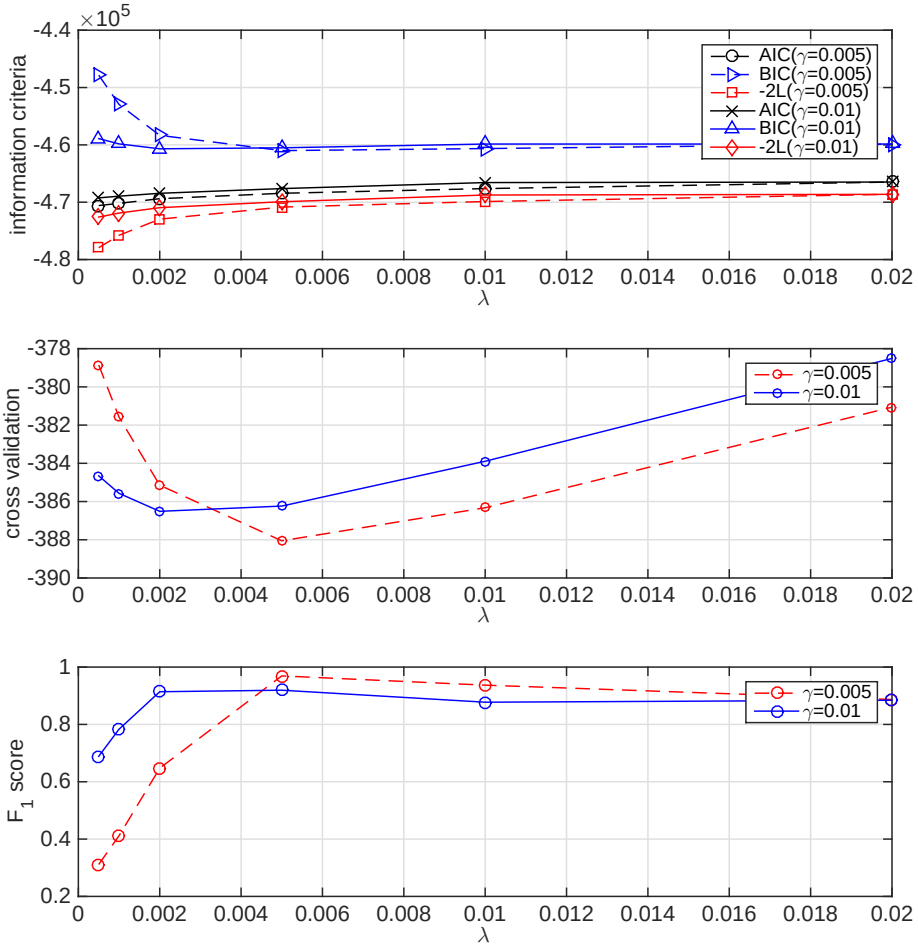


Figure 4.1: Model Selection using synthetic data. The top figure shows the comparison among AIC,BIC, and negative log-likelihood for different γ and λ. The middle figure shows the curve for cross validation. The bottom figure provide the F_1 score as a ground truth.

scanning through different combinations of parameters and comparing the estimation results with the true pattern. Also note that in this experiment, only edges with the partial correlation value greater than 0.1 are considered as positive, while the rest of them are considered as negative. In these figures, we can see that for the same sample size, the group Gaussian graphical model for autoregressive time series performs better than separate Gaussian graphical models for autoregressive time series in terms of TPR, FPR and F_1 score. When the sample size increases, the difference of estimation accuracy decreases until the sample size is big enough that all methods can reach perfect recovery. Fig. 4.2 also demonstrates the performance of the non-regularized model in terms of F_1 score. Compared with other estimations, the non-regularized model has poor recovery.

Fig. 4.5 shows the convergence rate of the group Gaussian graphical model for autoregressive time series using Spingarn's method. The primal residual and dual residual are defined in chapter 2 section 2.3.1. From Fig. 4.5, we can see that only less than 100 iterations are needed for convergence to reach the relative residual 10^{-5}.

4.5.3 International stock markets analysis

We consider a multivariate time series of international stock market indices: the NYSE Composite Index (U.S.), the NASDAQ Composite Index (U.S.), the Frankfurt DAX 30 Composite Stock Index (Germany), the CAC 40 Composite Stock Index (France), the FTSE 100 Share Index (U.K.), the Nikkei 225 Stock index (Japan), the Straits Times Index STI (Singapore), the Hang Seng Stock Composite Index (Hong Kong), the SSE Composite Index (Shanghai, China), and the SZSE Composite Index (Shenzhen, China). All stock index prices are retrieved from yahoo finance from Sep. 15th, 2014 to Dec. 31st, 2014. The variable in the graph is defined to be the return between trading day $t-1$ and t, $r_t = 100 \log(\pi_t/\pi_{t-1})$, where π_t is the closing price on day t. For the 75-day data,

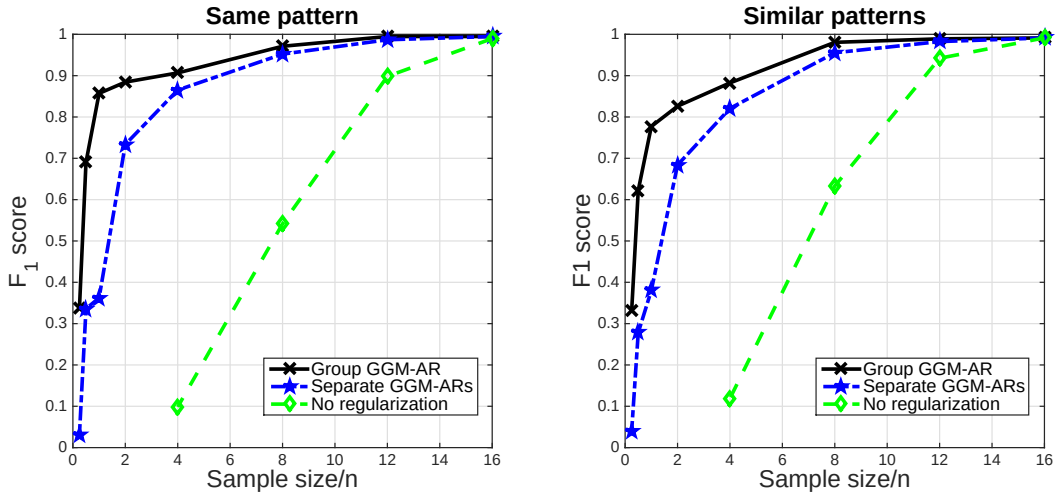


Figure 4.2: F_1 scores for group AR Gaussian graphical models using synthetic data in section 4.5.2. Parameter settings are $K = 3, n = 100, p = 1$. Left: structures with the same pattern but different edge values. Right: structures with similar patterns. The curves show the increment of F_1 score with an increasing sample size.

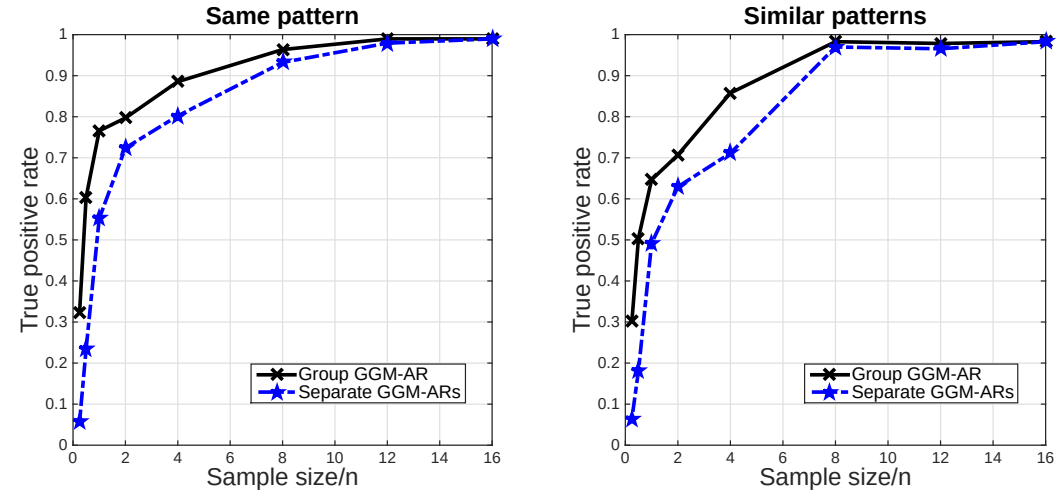


Figure 4.3: True positive rate for group AR Gaussian graphical models using synthetic data in section 4.5.2. Parameter settings are $K = 3, n = 100, p = 1$. Left: structures with the same pattern but different edge values. Right: structures with similar patterns.

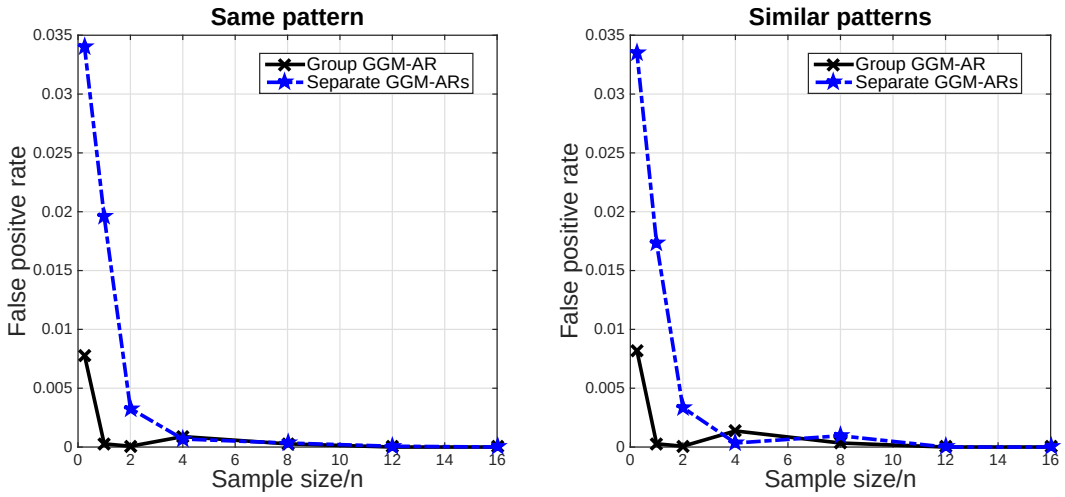


Figure 4.4: False positive rate for group AR Gaussian graphical models using synthetic data in section 4.5.2. Parameter settings are $K = 3, n = 100, p = 1$. Left: structures with the same pattern but different edge values. Right: structures with similar patterns.

we consider day 1-30, day 16-45, day 31-60, day 46-75, as four groups. For each group, half days are overlapping with the next group. We do this intentionally in order to make the groups manifest the change of stock markets over time. The results are shown in Fig. 4.6, Fig. 4.7, and Fig. 4.8.

In Fig. 4.6, Fig. 4.7 and Fig. 4.8, we use the thickness of edges to represent the connectivity between nodes. The larger the partial correlation value is, the thicker the edge is. From the figures, we can see that some pairs like (SSE,SZSE) and (NSYE,NASDAQ) are strongly connected. They have large partial correlations since the markets are within one country, and they highly interact with each other. Other interesting pairs include the pair between CAC40 and DAX. We can see that separate graphical models fail to recover this relation, but the group graphical model is able to demonstrate the relation over different time period. The group graphical model also maintains Nikkei 225's connectivity to NYSE as in the single graphical model, but it is absent in the separate graphical models.

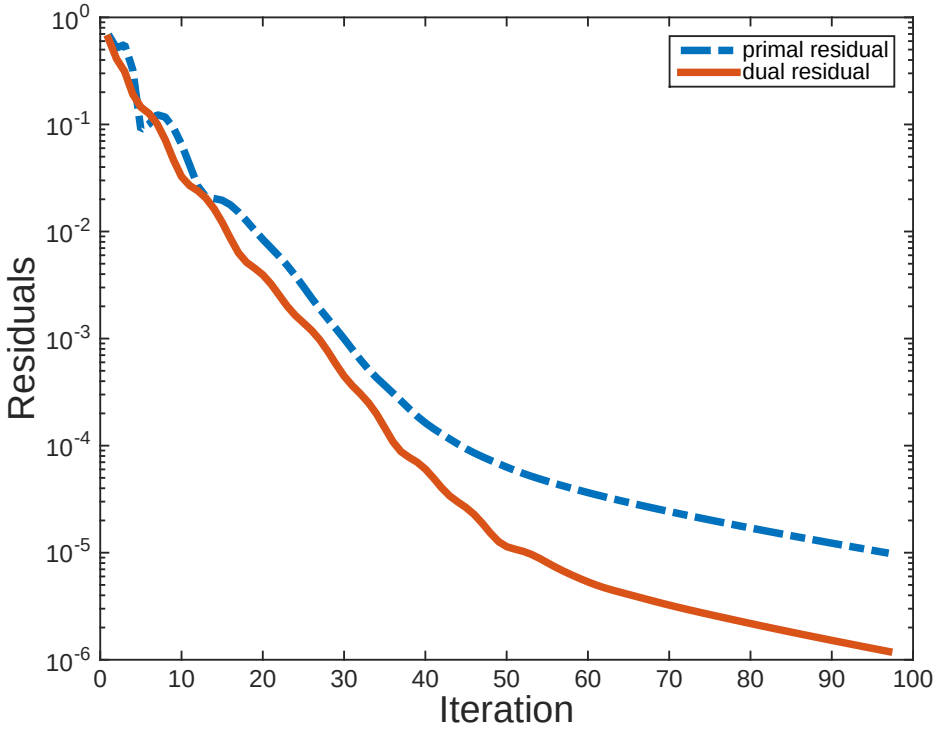


Figure 4.5: Convergence of Spingarn's method for group AR Gaussian graphical models using synthetic data in section 4.5.2. Parameter settings are $K = 3, n = 100, p = 1$

.

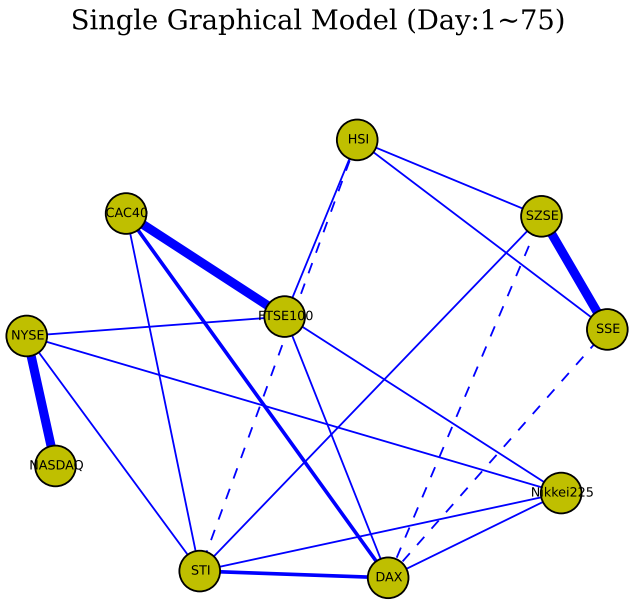


Figure 4.6: International stock market relations via a single Gaussian graphical Model for autoregressive time series. It uses all the data in 75 days.

4.5.4 fMRI brain network

In the brain activity network, all subjects share certain commonality while each individual has their uniqueness. The fMRI dataset we are analyzing is provided by Professor Patrick Dupont at the University of Leuven [VWN+13]. The fMRI experiment was conducted with 33 different subjects, and the reaction in 57 different brain regions were recorded. For each subject, there are 4-6 runs depending on the subject, and for each run, there are 108 scans. For the purpose of demonstration, we only use 5 subjects and 10 regions. Each subject is treated as a group, and the group Gaussian graphical model for autoregressive time series is applied to analyze the data. Fig. 4.9 shows that for all the individuals, some edges like (0,8), (7,9) are always strongly connected, and some edges like (0,1) and (2,3) are

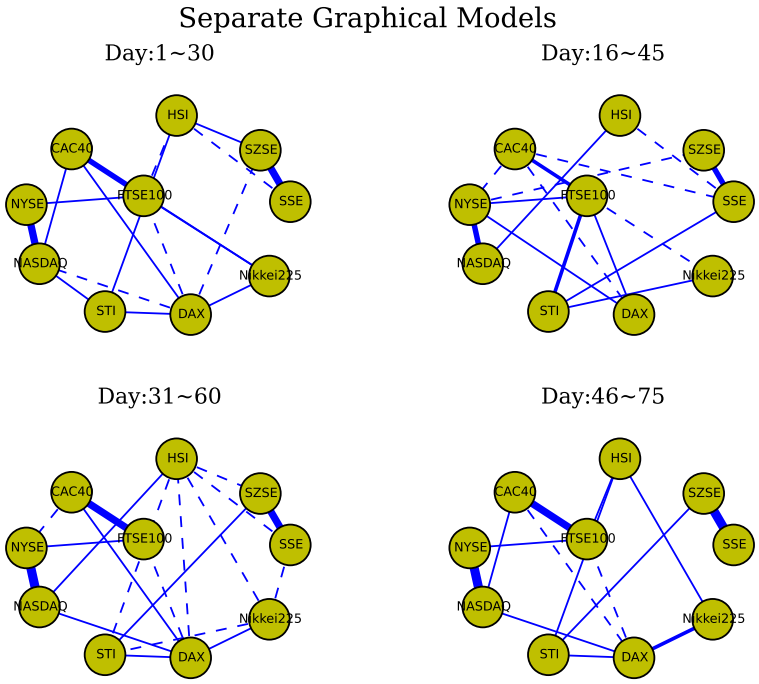


Figure 4.7: International stock market relations via separate Gaussian graphical Models for autoregressive time series. Four graphs represent the graphs for day 1-30, day 16-45, day 31-60, and day 46-75 respectively. The graphs are estimated separately as a single graph.

always disconnected. For different subjects, some edge connectivities are different, such as (1,6) and (0,3). This graph can be used to assist the interpretation of the brain network.

4.6 Conclusion

In this chapter, we have extended joint Gaussian graphical models to autoregressive time series, and applied the Douglas-Rachford method to solve the estimation problems. The model selection experiments have shown that cross validation and BIC work well for group Gaussian graphical models for autoregressive time series.

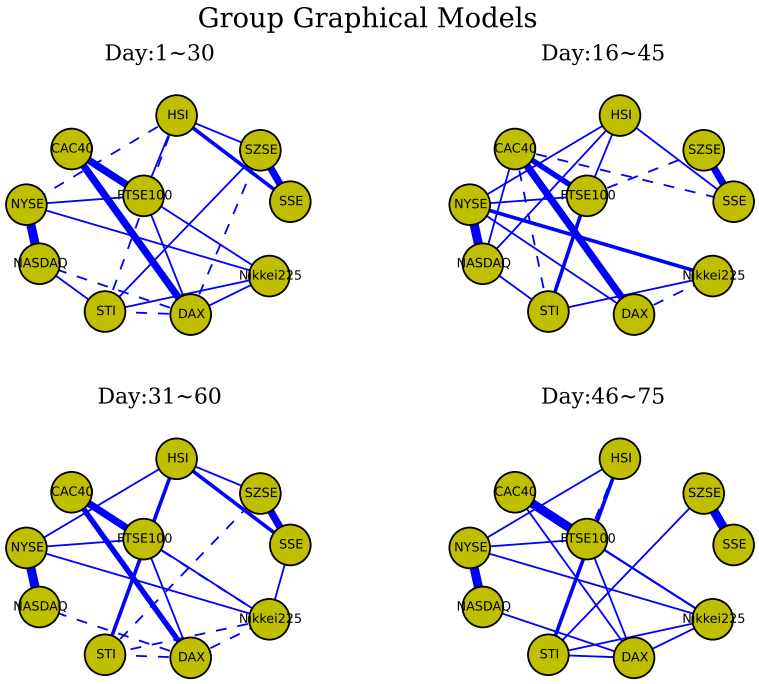


Figure 4.8: International stock market relations via group Gaussian graphical Models for autoregressive time series. Four graphs represent the graphs for day 1-30, day 16-45, day 31-60, and day 46-75 respectively. The graphs are estimated together using group Gaussian graphical Models for autoregressive time series.

The synthetic experiments have shown that the performance of the group Gaussian graphical model for autoregressive time series is better than the performance of estimating multiple graphical models separately. Lastly, the group model has been applied to real applications including international stock markets analysis and fMRI brain network analysis to increase the interpretability.

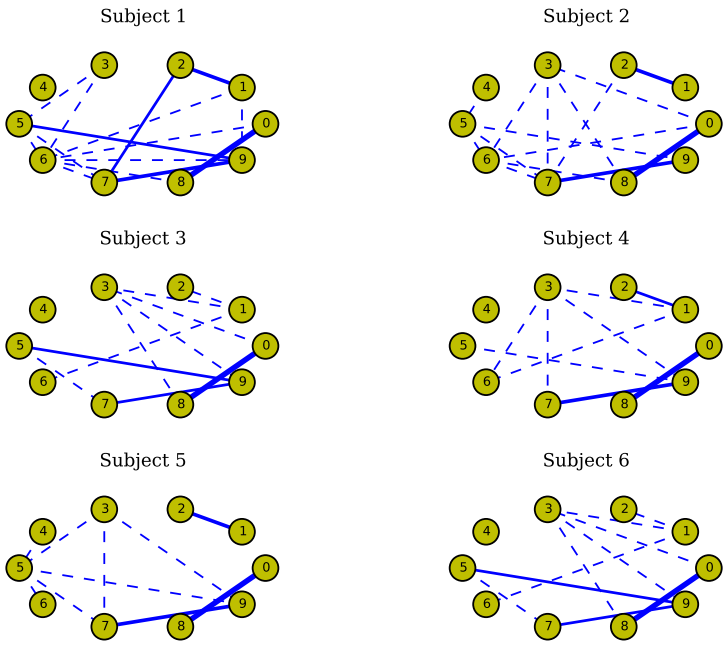


Figure 4.9: fMRI brain network using group Gaussian graphical models for autoregressive time series

CHAPTER 5

Time series with latent variables

The structure of a Gaussian graphical model is characterized by conditional independence, and requires the information of all nodes when we evaluate the connectivity between two nodes. This requirement is stringent, since some nodes may be hidden or latent, and once they are considered in the model, the topology can change significantly. In this chapter, we focus on graphical models with latent variables. We first explain the effect of latent variables on Gaussian graphical model (for time series), then we apply the Douglas-Rachford method to solve Gaussian time series graphical models with latent variables. Last, we provide some experiments to illustrate the effect of latent models and demonstrate the performance of the algorithm.

5.1 Latent variables in Gaussian graphical models

In this section, we start with discussing the effect of latent variables on conditional independence graphs. Then we review the existing work on Gaussian graphical models with latent variables [CPW10], and provide some explanations about the effect of latent variables on Gaussian graphical models.

5.1.1 Effect of latent variables

Based on the definition of conditional independence, whether two variables x_i and x_j are conditionally independent depends on the entire vector. Adding or removing

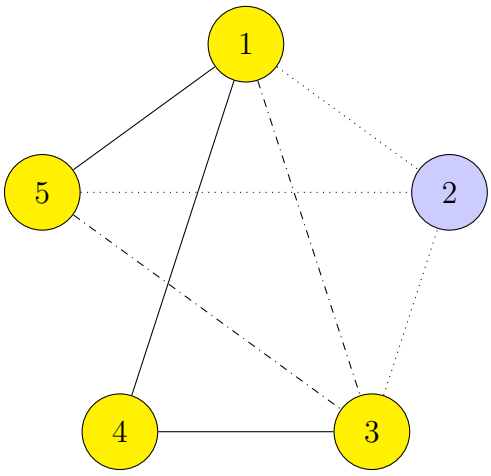


Figure 5.1: Effect of latent variables. Node 1,3,4,5 are observable variables, and node 2 is a latent variable. Solid lines: true conditional dependence between observable variables. Dotted lines: true conditional dependence between an observable variable and a latent variable. Dashed-dot lines: conditional dependence result produced by lack of the information of node 2.

a third variable from the vector can change the topology. In some applications this is no problem because one may have a good list of all the variables that are of importance. But in applications where there may be important variables that are hidden or latent, and observable variables do not include all important factors, one may get misleading results. For one example, Fig. 5.1 shows a graphical model with 4 observable variables and 1 latent variable. We can observe the set of nodes $\{1, 3, 4, 5\}$, while node 2 is hidden. The solid lines and dotted lines represent the true conditional dependence. However, if node 2 is hidden, the lack of information from node 2 produces two extra edges ($\{1, 3\}$ and $\{3, 5\}$) as shown in the dashed-dot line segments), since these two sets of nodes are conditionally independent only based on extra information from node 2.

5.1.2 Estimation of Gaussian graphical models with latent variables

Given the effect of latent variables on the estimation of conditional independence graph, it is important to extend Gaussian graphical models to latent variables. Chandrasekharan et al. [CPW10] have developed a technique to deal with this problem, based on the assumption that the number of latent variables is small. They view a Gaussian graph model as a sparse matrix plus a low rank matrix. To be specific, for a vector of variables $x = (x_{\mathrm{o}}, x_{\mathrm{h}})$, we use x_{o} to represent a vector of observable variables, and x_{h} to represent a vector of latent variables. The covariance matrix of joint Gaussian random variables can be written as

$$\Sigma = \begin{bmatrix} \Sigma_{\mathrm{oo}} & \Sigma_{\mathrm{oh}} \\ \Sigma_{\mathrm{ho}} & \Sigma_{\mathrm{hh}} \end{bmatrix},$$

and the corresponding inverse covariance matrix (precision matrix) can be denoted as

$$K = \Sigma^{-1} = \begin{bmatrix} K_{\mathrm{oo}} & K_{\mathrm{oh}} \\ K_{\mathrm{ho}} & K_{\mathrm{hh}} \end{bmatrix}.$$

By Schur complement,

$$\Sigma_{\mathrm{oo}}^{-1} = K_{\mathrm{oo}} - K_{\mathrm{oh}} K_{\mathrm{hh}}^{-1} K_{\mathrm{ho}}. \tag{5.1}$$

In (5.1), K_{oo} characterizes the conditional independence of observable variables, and is typically sparse. $K_{\mathrm{oh}} K_{\mathrm{hh}}^{-1} K_{\mathrm{ho}}$ summarizes the effect of marginalization over the latent variables x_{h}. This term has a small rank assuming that the number of latent variables n_{h} is small relative to the number of observed variables n_{o}. Thus the decomposition (5.1) reveals that the estimated structure using observable variables is a mixture of the true structure of observable variables plus the marginalization effect of latent variables. The former is characterized by a sparse matrix, while the latter is characterized by a low rank matrix. This means that some otherwise conditionally independent elements becomes dependent with extra edges added due to the effect of latent variables.

If more latent variables are added to the graph, from (5.1) we can see that the augmented edges are determined by K_{oh}. If all the elements in row i of K_{oh} are zero, there will be no extra edges added to node i, and thus the marginalization of the latent variables has no effect on node i. For rows with some non-zero entries in K_{oh}, K_{hh}^{-1} and the entry values in K_{oh} determine the coefficients of added edges. To be more specific, let us first assume that all latent variables are independent of each other, *i.e.*, K_{hh} is a diagonal matrix, then the marginalization effect can be written as

$$K_{\text{oh}}K_{\text{hh}}^{-1}K_{\text{ho}} = \sum_{i=1}^{n_{\text{h}}}(K_{\text{hh}})_{ii}(K_{\text{oh}})_i(K_{\text{oh}})_i^T, \tag{5.2}$$

where $(K_{\text{oh}})_i$ denotes the i-th column of K_{oh}. In this special case, each latent variable serves as an independent factor and introduces a separate set of edges (structure) to the graph. With n_{h} latent variables, n_{h} different sets of edges are added, each of which defines a feature shared among all nodes involved in the graph. For example, if the graph is used to model correlated purchase behavior of the consumers in the social network, then the sparsity of the graph illustrates that there is little unique common interest shared among individuals, and the low rank matrix shows the presence of some common properties shared between individuals.

Then let us consider the case when K_{hh} is not a diagonal matrix, *i.e.*, latent variables are not conditionally independent. With eigenvalue decomposition on K_{hh}, we can obtain $K_{\text{hh}} = UDU^T$, where U is an orthogonal matrix, and D is a diagonal matrix. U can be interpreted as a transformation matrix that is used to extract orthogonal features from latent variables, and reorganize the information of latent variables so that they become conditionally independent. By defining $\tilde{K}_{\text{oh}} = K_{\text{oh}}U$, this problem boils down to the special case of diagonal matrix K_{hh}. If some latent variables are similar with each other, then after applying the above shown procedure, some eigenvalues approach zero, and thus the corresponding effects on the graph become very weak and ignorable.

For Gaussian graphical models, Chandrasekaran et al. [CPW10] have shown that if we apply conditional maximum likelihood estimation with an ℓ_1-norm penalization to the sparse matrix K_{oo} and a nuclear norm penalization on the low rank matrix $K_{\mathrm{oh}}K_{\mathrm{hh}}^{-1}K_{\mathrm{ho}}$, the penalized estimation can be expressed as in (5.3):

$$\begin{array}{ll} \text{minimize} & -\log\det(X-L)+\mathbf{tr}(C(X-L))+\gamma\|X\|_1+\lambda\,\mathbf{tr}(L) \\ \text{subject to} & X-L\succ 0 \\ & L\succeq 0, \end{array} \tag{5.3}$$

where $K_{\mathrm{oh}}K_{\mathrm{hh}}^{-1}K_{\mathrm{ho}}$ is substituted with L, K_{oo} is substituted with X, and γ, λ are parameters used to control the sparsity of X and the rank of L respectively.

5.2 Latent variables for time series

In this section, we focus on the effect of latent variables on Gaussian time series $x(t)$. This has been analyzed by Zorzi and Sepulchre in [MS14], and we will review the approach in this section.

Let us consider a zero-mean stationary Gaussian process $x(t)$ with n_{o} observable variables and n_{h} latent variables with $n_{\mathrm{o}} \gg n_{\mathrm{h}}$. $x_i(t)$ and $x_j(t)$ are conditionally independent if $(S(\omega))^{-1}_{ij} = 0$. However, $(S(\omega))^{-1}_{ij}$ cannot be obtained due to the lack of data for latent variables. We assume $x(t) = (x_{\mathrm{o}}(t), x_{\mathrm{h}}(t))$ and that we can only access observable variables $x_{\mathrm{o}}(t)$, and the dimension of $x_{\mathrm{h}}(t)$ is unknown. The whole spectral density matrix can be written as

$$S(\omega) = \begin{bmatrix} S_{\mathrm{oo}}(\omega) & S_{\mathrm{oh}}(\omega) \\ S_{\mathrm{ho}}(\omega) & S_{\mathrm{hh}}(\omega) \end{bmatrix},$$

and only the component $S_{\mathrm{oo}}(\omega)$ can be inferred from data. We also denote the inverse spectrum $S(\omega)^{-1}$ as

$$S(\omega)^{-1} = K(\omega) = \begin{bmatrix} K_{\mathrm{oo}}(\omega) & K_{\mathrm{oh}}(\omega) \\ K_{\mathrm{ho}}(\omega) & K_{\mathrm{hh}}(\omega) \end{bmatrix}.$$

With this notation, $(x_{\mathrm{o}}(t))_i$ and $(x_{\mathrm{o}}(t))_j$ are conditionally independent if and only if $(K_{\mathrm{oo}}(\omega))_{ij} = 0$ for all ω. As analogous to static Gaussian graphical models, by applying Schur complement, we can obtain

$$S_{\mathrm{oo}}(\omega)^{-1} = K_{\mathrm{oo}}(\omega) - K_{\mathrm{oh}}(\omega)K_{\mathrm{hh}}(\omega)^{-1}K_{\mathrm{ho}}(\omega). \tag{5.4}$$

Suppose the observable data sequence follows p-lag AR model in (4.2), $S_{\mathrm{oo}}(\omega)^{-1}$ can be represented as

$$S_{\mathrm{oo}}(\omega)^{-1} = Y_0 + \sum_{k=1}^{p} \left(e^{-jk\omega} Y_k + e^{jk\omega} Y_k^T \right), \tag{5.5}$$

where $Y_k = \sum_{l=0}^{p-k} B_l^T B_{l+k}$. We assume the low rank term $K_{\mathrm{oh}}(\omega)K_{\mathrm{hh}}(\omega)^{-1}K_{\mathrm{ho}}(\omega)$ can be expanded as

$$K_{\mathrm{oh}}(\omega)K_{\mathrm{hh}}(\omega)^{-1}K_{\mathrm{ho}}(\omega) = L_0 + \sum_{k=1}^{\infty} \left(e^{-jk\omega} L_k + e^{jk\omega} L_k^T \right). \tag{5.6}$$

We also define

$$S(\omega)^{-1} = K(\omega) = Z_0 + \sum_{k=1}^{\infty} \left(e^{-jk\omega} Z_k + e^{jk\omega} Z_k^T \right),$$

then

$$K_{\mathrm{oo}}(\omega) = (Z_{\mathrm{oo}})_0 + \sum_{k=1}^{\infty} \left(e^{-jk\omega} (Z_{\mathrm{oo}})_k + e^{jk\omega} (Z_{\mathrm{oo}})_k^T \right), \tag{5.7}$$

where $(Z_{\mathrm{oo}})_k$ corresponds to the observable part of $(Z)_k$. With these notations, $(x_{\mathrm{o}}(t))_i$ and $(x_{\mathrm{o}}(t))_j$ are conditionally independent if

$$((Z_{\mathrm{oo}})_k)_{ij} = 0, \text{ or equivalently } \begin{cases} (L_k + Y_k)_{ij} = 0, & k = 0, \ldots, p. \\ (L_k)_{ij} = 0, & k \geq p+1. \end{cases} \tag{5.8}$$

As a heuristic approximation, we only consider the constraints for $k = 0, \ldots, p$. Then the conditional independence condition (5.8) is equivalent to

$$\left(\mathcal{D}_k(B^T B + L) \right)_{ij} = 0, \quad k = 0, \ldots, p, \quad (i,j) \notin E,$$

where $B = \begin{bmatrix} B_0 & B_1 & \cdots & B_p \end{bmatrix}$, and $L = \mathcal{T}\left(\begin{bmatrix} L_0 & L_1 & \cdots & L_p \end{bmatrix}\right)$. Before deriving the low rank constraints, we define $\Delta(e^{j\omega})$ as

$$\Delta(e^{j\omega}) = \begin{bmatrix} I_m & e^{j\omega} I_m & \cdots & e^{jp\omega} I_m \end{bmatrix}. \tag{5.9}$$

Then $K_{\mathrm{oh}}(\omega) K_{\mathrm{hh}}(\omega)^{-1} K_{\mathrm{ho}}(\omega)$ can be expressed as $\Delta(e^{j\omega}) L \Delta(e^{j\omega})^*$. Then the corresponding nuclear norm can be expressed as

$$\mathbf{tr}\left(\Delta(e^{j\omega}) L \Delta(e^{j\omega})^*\right) = \mathbf{tr}\left(L \Delta(e^{j\omega})^* \Delta(e^{j\omega})\right) = \mathbf{tr}(L), \quad \forall \omega.$$

The constrained latent Gaussian graphical model for autoregressive time series can be formulated as

$$\begin{array}{ll} \text{minimize} & -2 \log\det B_0 + \mathbf{tr}(C B^T B) + \lambda\, \mathbf{tr}(L) \\ \text{subject to} & \mathcal{D}_k(B^T B + L)_{ij} = 0, \quad k = 0, \ldots, p, \quad (i,j) \notin E \\ & L \succeq 0. \end{array} \tag{5.10}$$

If we define $\tilde{h}$ as the indicator function:

$$\tilde{h}(Y)_{k,ij} = \begin{cases} 0, & (i,j) \notin E, \quad k = 0, \ldots, p \\ +\infty, & \text{otherwise}, \end{cases}$$

problem (5.10) can be written as

$$\begin{array}{ll} \text{minimize} & -2 \log\det B_0 + \mathbf{tr}(C B^T B) + \lambda\, \mathbf{tr}(L) + \tilde{h}\left(\mathcal{D}(B^T B + L)\right) \\ \text{subject to} & L \succeq 0. \end{array} \tag{5.11}$$

The function $\tilde{h}$ can also be extended to other convex functions. For penalized estimation, one choice of $\tilde{h}$ is

$$\tilde{h}(Y) = \gamma \sum_{i>j} \max_{k=0,\ldots,p} \{|Y_{k,ij}|, |Y_{k,ji}|\}. \tag{5.12}$$

Problem (5.11) is non-convex due to the quadratic term $B^T B$. A convex relaxation can be made by variable substitution $X = B^T B$, *i.e.*, to solve

$$\begin{array}{ll} \text{minimize} & -\log\det X_{00} + \mathbf{tr}(C X) + \lambda\, \mathbf{tr}(L) + \tilde{h}\left(\mathcal{D}(X + L)\right) \\ \text{subject to} & L \succeq 0 \\ & X \succeq 0. \end{array} \tag{5.13}$$

5.3 Algorithms

Zorzi and Sepulchre in [MS14] did not discuss practical methods for solving (5.13). As one contribution in this thesis, we apply the Douglas-Rachford method to solve (5.13). First, we reformulate problem (5.13) as

$$\begin{array}{ll} \text{minimize} & f(X,L)+g(U,Y) \\ \text{subject to} & (U,Y)=\mathcal{A}(X,L). \end{array} \tag{5.14}$$

- $f(X,L)=\delta(X)+\delta(L)+\mathbf{tr}(CX)+\gamma\,\mathbf{tr}(L)$, where δ is an indicator function of positive semidefinite cone $\mathbf{S}_+^{(p+1)n_o}$.
- $g(U,Y)=-\log\det U+\tilde{h}(Y)$.
- $\mathcal{A}(X,L)=(X_{00},\mathcal{D}(X+L))$.

If we apply Spingarn's method or the primal-dual Douglas-Rachford method (see chapter 2) to problem 5.14, we need to evaluate the proximal operator of f, $(I+t^2\mathcal{A}^{\text{adj}}\mathcal{A})^{-1}$, and the proximal operator of g. The major operations are summarized as follows.

Proximal operator of f This is involved with projections on the positive semidefinite cone for X and L respectively.

Evaluation of $(I+t^2\mathcal{A}^{\text{adj}}\mathcal{A})^{-1}$ The adjoint operator of $\mathcal{A}$ is

$$\mathcal{A}^{\text{adj}}(U,Y)=\left(\begin{bmatrix} U & 0 \\ 0 & 0 \end{bmatrix}+\mathcal{T}(Y),\mathcal{T}(Y)\right).$$

Then the evaluation of $(I+t^2\mathcal{A}^{\text{adj}}\mathcal{A})^{-1}$ is equivalent to solving the linear equation

$$(X,L)+t^2\mathcal{A}^{\text{adj}}(\mathcal{A}(X,L)))=\left(B^{(1)},B^{(2)}\right).$$

This is equivalent to solving

$$\begin{aligned} X + t^2 \begin{bmatrix} X_{00} & 0 \\ 0 & 0 \end{bmatrix} + t^2 \mathcal{T}(\mathcal{D}(X+L)) &= B^{(1)}, \\ L + t^2 \mathcal{T}(\mathcal{D}(X+L)) &= B^{(2)}, \end{aligned} \tag{5.15}$$

$(I + t^2 \mathcal{A}^{\text{adj}} \mathcal{A})^{-1}$ can be evaluated by solving the linear equations in (5.15). More importantly an analytical solution can be obtained from (5.15), and the related coefficients only need to be calculated for once.

Proximal operator of g The proximal operator of function $g(U, Y)$ can be computed by evaluating two independent proximal operators:

- proximal operator of $-\log\det U$. It requires eigenvalue decomposition.
- proximal operator of $\tilde{h}(Y)$. If it is for constrained estimation, we need to compute the projection onto the sparsity pattern; if it is for penalized estimation, we need to compute the projection onto an ℓ_1-norm ball and Moreau decomposition.

The details of evaluating proximal operators can be found in chapter 2 section 2.2.2.

5.4 Numerical examples

For the static model, it has been shown that with a proper choice of parameters (λ and γ) that make the minimum nonzero singular value of the low-rank matrix L and minimum nonzero entry of the matrix $\mathcal{D}(X + L)$ bounded away from zero, the model provides estimates with the correct sparsity pattern and rank [CPW10, Theorem 4.1]. However, this theoretical value is unknown beforehand. As an alternative method, in this section, we demonstrate the performance of latent Graphical models for autoregressive time series by iterating through different

values of λ and γ using synthetic examples, and compare the ROC curve with the corresponding non-latent model.

In the simulation, we consider a latent variable autoregressive graphical model with $n_{\mathrm{o}} = 100$ observational variables, and $n_{\mathrm{h}} = 8$ latent variables. The order of the autoregressive model is set as $p = 3$. For the coefficients A_k in the full autoregressive model, A_0 is the identity matrix, and $A_k, k = 1, \ldots, p$ share the same sparsity pattern, where 99% entries are selected to be zeros, and the values for nonzero entries are set as 0.5 or -0.5 with equal probability. Then we generate data samples with $N = 10n_{\mathrm{o}}$. Spingarn's method is applied to problem (5.13), and the simulation results are shown in Fig. 5.2 and Fig. 5.3. The primal residual and dual residual in Fig. 5.2 are defined in chapter 2 section 2.3.1. Fig. 5.2 shows that Spingarn's method converges within 200 iterations to reach the relative residual 10^{-5} where X, L are 400×400 positive semidefinite matrices in the experiment. Fig. 5.3 presents the ROC curves for both the latent and non-latent Gaussian graphical model for autoregressive time series. TPR and FPR are defined by comparing the recovered partial correlation graph with the true one. Edges with partial correlation greater than 0.1 are considered as positive, and the remaining edges are considered negative. Fig. 5.3 shows that the latent model has a better accuracy than the non-latent model. Therefore, the latent model recovers the graph structure better than the non-latent model if latent variables exist in the graph.

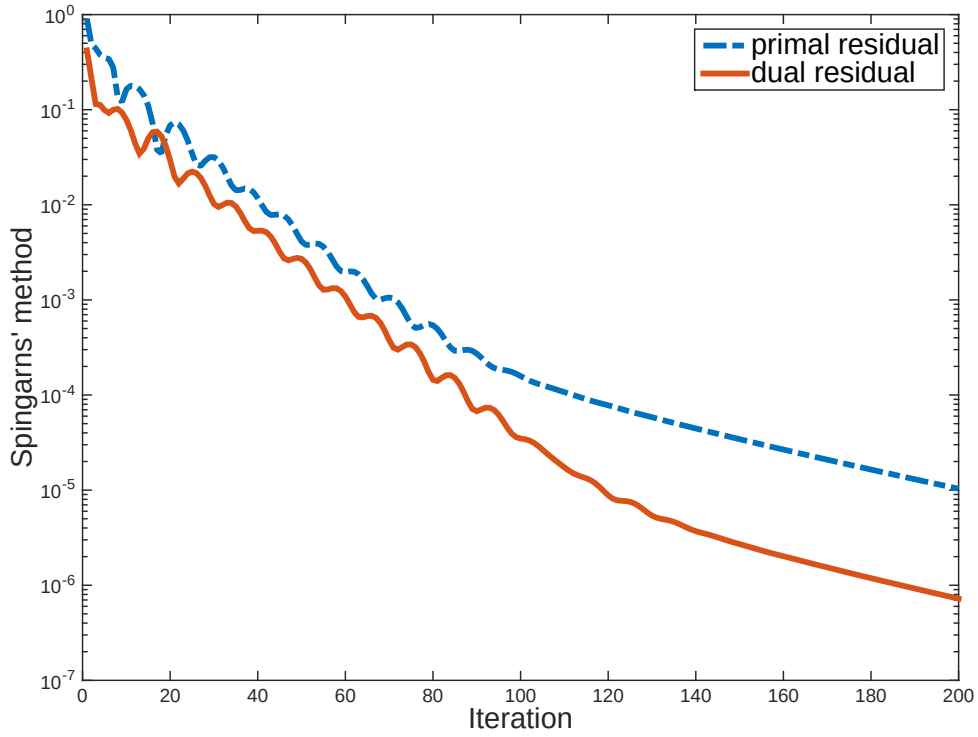


Figure 5.2: Spingarn's Method applied to the latent Gaussian graphical model for autoregressive time series with size $n_{\mathrm{o}} = 100, p = 3$.

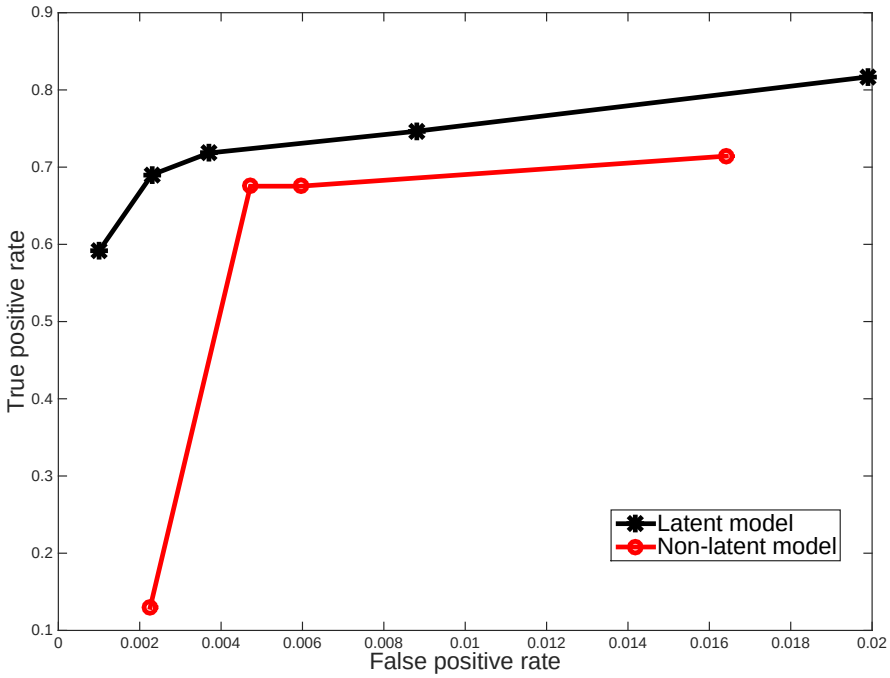


Figure 5.3: Black line: ROC curve using the latent graphical model. Red line: ROC curve using the original Gaussian graphical model for autoregressive time series.

CHAPTER 6

Conclusions

We have looked at three extensions of Gaussian graphical models based on prior structural information. The first model extends the ℓ_1-norm penalized inverse covariance selection problem. We assume that the zero pattern is partially known and penalize the remaining entries using an ℓ_1-norm penalty. We refer to this problem as restricted sparse inverse covariance selection. By taking advantage of the knowledge of the partial zero pattern, the number of variables is largely decreased, and thus the estimation variance is decreased. The nonzero pattern constraints can be interpreted as a combination of an extended chordal pattern via chording embedding plus a penalty on the extended entries using an indicator function. With this reformulation, the estimation can be restricted to chordal structure. A proximal Newton method with inexact proximal Newton steps has been proposed to solve this problem. This method is attractive since the key computations involved are gradient and Hessian evaluations of the log-determinant term, which can be be computed efficiently with fast algorithms associated with choral sparse matrices. A theoretical analysis of the convergence property of proximal Newton methods with inexact proximal Newton steps for self-concordant functions has been provided, and it has been shown that if the proximal Newton step is exact, the algorithm is quadratic convergent; if the proximal Newton step approaches to the exact value as the algorithm converges, it is super-linear convergent; if the relative inexactness of the proximal Newton step is fixed, it is linearly convergent. Experiments with synthetic data have been provided to show the performance

of proximal Newton method with inexact proximal Newton steps calculated by FISTA.

The second model extends the Gaussian graphical model for autoregressive time series. It is extended to deal with applications where multiple graphical models are of interest, with similar but not identical topologies or coefficients. Estimating these models together as a joint graphical model is useful because the joint model considers the shared feature among different models and each model's uniqueness simultaneously. This also increases the estimation accuracy for applications where the number of samples is limited. The Douglas-Rachford algorithm has been applied to solve the joint graphical model. Numerical examples have been provided to demonstrate the performance of joint graphical models. The experiment with synthetic data has shown that BIC and cross validation are good model selection methods, and that the joint graphical model outperforms the method of estimating the models separately. The real data examples of international stock markets and brain networks via fMRI data analysis have shown that the joint graphical model can capture features that separate models cannot detect, and make the estimation result easier to interpret.

The third extension is the latent Gaussian graphical model for autoregressive time series. We have reviewed existing models, and applied Spingarn's method to solve the latent time series model. Synthetic experiments have been conducted to compare the performance between the latent time series graphical model and the original time series graphical model.

Several suggestions for future work are listed as follows:

- In chapter 3, we have used FISTA to calculate inexact proximal Newton steps with a fixed θ. As we have seen, the evaluation of the inexact proximal Newton step is a lasso problem, and can be solved by many different iterative algorithms [Tib96, EHJT04, Zou06, FHHT07, WL08]. The choice

of algorithm for the inexact proximal Newton steps is one of the most important questions for further research. Also, the formulation of good strategies for adaptive control of the accuracy θ is of great importance.

- In chapter 4 and chapter 5, we have discussed two extensions of Gaussian graphical models for autoregressive time series. One future research can focus on the statistical analysis of the asymptotic properties of these models [WJ08, CPW10], and provide theoretical methods for choosing tuning parameters.

Bibliography

[ADV13] M. S. Andersen, J. Dahl, and L. Vandenberghe. Logarithmic barriers for sparse matrix cones. *Optimization Methods and Software*, 28(3):396–423, 2013.

[ADV15] M. Andersen, J. Dahl, and L. Vandenberghe. *CVXOPT: A Python Package for Convex Optimization*. `www.cvxopt.org`, 2015.

[Ahl79] L. Ahlfors. *Convex Analysis,Third Edition*. McGraw-Hill, 1979.

[ALW13] E. Avventi, A. G. Lindquist, and B. Wahlberg. ARMA identification of graphical models. *IEEE Transactions on Automatic Control*, 58(5):1167–1178, 2013.

[AV15] M. S. Andersen and L. Vandenberghe. *CHOMPACK: A Python Package for Chordal Matrix Computations, Version 2.2.1*, 2015. `cvxopt.github.io/chompack`.

[BA02] K. P Burnham and D. R Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer, 2002.

[BB01] P. Baldi and S. Brunak. *Bioinformatics: the machine learning approach*. MIT press, 2001.

[BCG11] S. R. Becker, E. J. Candès, and M. C. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathemetical Programming Computation*, 3(3):165–218, 2011.

[BEd08] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

[Ber09] D. P. Bertsekas. *Convex Optimization Theory*. Athena Scientific, 2009.

[BGd08] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9(485-516), 2008.

[Bis06] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[BJ04] F. R. Bach and M. I. Jordan. Learning graphical models for stationary time series. *IEEE Transactions on Signal Processing*, 32:2189–2199, 2004.

[BJR11] G. Box, G. Jenkins, and G. Reinsel. *Time series analysis: forecasting and control*, volume 734. John Wiley & Sons, 2011.

[BNO15] R. H. Byrd, J. Nocedal, and F. Oztoprak. An inexact successive quadratic approximation method for L-1 regularized optimization. *Mathematical Programming*, pages 1–22, 2015.

[BP93] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In A. George, J. R. Gilbert, and J. W. H. Liu, editors, *Graph Theory and Sparse Matrix Computation*. Springer-Verlag, 1993.

[BPC+11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, pages 1–122, 2011.

[Bri01] D. R. Brillinger. *Time Series. Data Analysis and Theory.* Society for Industrial and Applied Mathematics, 2001. First published by Holden Day in 1981.

[BT09] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[Bur75] J. P. Burg. *Maximum entropy spectral analysis.* PhD thesis, Stanford University, 1975.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, 2004.

[CP07] P. L. Combettes and J.-C. Pesquet. A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):564–574, 2007.

[CPW10] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 1610–1613. IEEE, 2010.

[Dah00] R. Dahlhaus. Graphical interaction models for multivariate time series. *Metrika*, 51(2):157–172, 2000.

[Dar09] A. Darwiche. *Modeling and Reasoning with Bayesian Networks.* Cambridge University Press, 2009.

[dBG08] A. d'Aspremont, O. Banerjee, and L. El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66, 2008.

[Dem72] A. P. Dempster. Covariance selection. *Biometrics*, 28:157–175, 1972.

[DGK08] J. Duchi, S. Gould, and D. Koller. Projected subgradient methods for learning sparse Gaussians. In *Proceedings of the Conference on Uncertainty in AI*, 2008.

[DH11] T. A. Davis and Y. Hu. The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software*, 38:1–25, 2011.

[DR83] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Transactions on Mathematical Software*, 9(3):302–325, 1983.

[DVR08] J. Dahl, L. Vandenberghe, and V. Roychowdhury. Covariance selection for non-chordal graphs via chordal embedding. *Optimization Methods and Software*, 23(4):501–520, 2008.

[DWW14] P. Danaher, P. Wang, and D. M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

[EB92] J. Eckstein and D. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318, 1992.

[EHJT04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[FHHT07] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 2007.

[FHT01] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[FHT07] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 2007.

[FHT08] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[FHT10] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 2010.

[Fri11] K. Friston. Functional and effective connectivity: a review. *Brain connectivity*, 1(1):13–36, 2011.

[GLMZ11] J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint estimation of multiple graphical models. *Biometrika*, page asq060, 2011.

[GMS13] D. Goldfarb, S. Ma, and K. Scheinberg. Fast alternating linearization methods for minimizing the sum of two convex functions. *Mathematical Programming*, 141(1-2):349–382, 2013.

[HDRB12a] C. Hsieh, I. S Dhillon, P. Ravikumar, and A. Banerjee. A divide-and-conquer procedure for sparse inverse covariance estimation. NIPS, 2012.

[HDRB12b] C.-J. Hsieh, I. S. Dhillon, P. Ravikumar, and A. Banerjee. A divide-and-conquer procedure for sparse inverse covariance estimation. In *Advances in Neural Information Processing (NIPS)*, volume 25, pages 2339–2347, 2012.

[HDRS11] C. Hsieh, I. S Dhillon, P. Ravikumar, and M. Sustik. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems*, pages 2330–2338, 2011.

[Hec95] D. Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, March 1995.

[HSDR11] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing (NIPS)*, volume 24, pages 2330–2338, 2011.

[HSDR14] C. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. Quic: Quadratic approximation for sparse inverse covariance estimation. *Journal of Machine Learning Research*, 15:2911–2947, 2014.

[HUL93] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*, volume 305 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, New York, 1993.

[Jor99] M. I. Jordan, editor. *Learning in Graphical Models*. MIT Press, 1999.

[Jor04] M. I. Jordan. Graphical models. *Statistical Science*, 19:140–155, 2004.

[KF09] D. Koller and N. Friedman. *Probabilistic Graphical Models. Principles and Techniques*. MIT Press, 2009.

[KSAX10] M. Kolar, L. Song, A. Ahmed, and E. Xing. Estimating time-varying networks. *The Annals of Applied Statistics*, pages 94–123, 2010.

[Lau96] S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.

[LSS12] J. D. Lee, Y. Sun, and M. A. Saunders. Proximal Newton-type methods for convex optimization. In *Advances in Neural Information Processing (NIPS)*, volume 25, pages 836–844, 2012.

[LSS14] J. D. Lee, Y. Sun, and M. A. Saunders. Proximal Newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443, 2014.

[LT10] L. Li and K.-C. Toh. An inexact interior point method for L1-regularized sparse covariance selection. *Mathematical Programming Computation*, 2:291–315, 2010.

[Lu09] Z. Lu. Smooth optimization approach for sparse covariance selection. *SIAM Journal on Optimization*, 19(4):1807–1827, 2009.

[Lu10] Z. Lu. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2000–2016, 2010.

[MB06] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

[MCH+12] K. Mohan, M. Chung, S. Han, D. Witten, S. Lee, and M. Fazel. Structured learning of gaussian graphical models. In *Advances in neural information processing systems*, pages 620–628, 2012.

[MH12] R. Mazumder and T. Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, 13:781–794, 2012.

[MLF+14] K. Mohan, P. London, M. Fazel, D. Witten, and S. Lee. Node-based learning of multiple gaussian graphical models. *The Journal of Machine Learning Research*, 15(1):445–488, 2014.

[Mor65] J. J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Math. Soc. France*, 93:273–299, 1965.

[MS14] Z. Mattia and R. Sepulchre. Ar identification of latent-variable graphical models. 2014.

[Mur12] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[MvdGB08] L. Meier, S. van de Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

[Nes04] Yu. Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[Nes05] Yu. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming Series A*, 103:127–152, 2005.

[Nes12] Y. Nesterov. Towards non-symmetric conic optimization. *Optimization Methods and Software*, 27(4-5):893–917, 2012.

[NN94] Yu. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Methods in Convex Programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[OONR12] P. A. Olsen, F. Oztoprak, J. Nocedal, and S. J. Rennie. Newton-like methods for sparse inverse covariance estimation. In *Advances in Neural Information Processing (NIPS)*, volume 25, pages 764–772, 2012.

[PB13] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

[QHLC14] H. Qiu, F. Han, H. Liu, and B. Caffo. Joint estimation of multiple graphical models from high dimensional time series. *arXiv preprint arXiv:1311.0219*, 2014.

[Ren01] J. Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*. SIAM, 2001.

[RH05] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

[Roc70] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[SDV10] J. Songsiri, J. Dahl, and L. Vandenberghe. Graphical models of autoregressive processes. In Y. Eldar and D. Palomar, editors, *Convex Optimization in Signal Processing and Communications*, pages 89–116. Cambridge University Press, Cambridge, 2010.

[SGB14] K. Scheinberg, D. Goldfarb, and X. Bai. Fast first-order methods for composite convex optimization with backtracking. *Foundations of Computational Mathematics*, 14:389–417, 2014.

[SM97] P. Stoica and R. L. Moses. *Introduction to Spectral Analysis*. Prentice Hall, London, 1997.

[SMG10] K. Scheinberg, S. Ma, and D. Goldfarb. Sparse inverse covariance selection via alternating linearization methods. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2101–2109. 2010.

[SR09] K. Scheinberg and I. Rish. SINCO - a greedy coordinate ascent method for sparse inverse covariance selection problem. Technical report, 2009. IBM Resesarch Report.

[ST13] K. Scheinberg and X. Tang. Complexity of inexact proximal Newton methods. Technical Report 13T-02-R1, COR@L, Lehigh University, 2013.

[ST15] K. Scheinberg and X. Tang. Practical inexact proximal quasi-Newton method with global complexity analysis, 2015. arXiv:1311.6547.

[SV10] J. Songsiri and L. Vandenberghe. Topology selection in graphical models of autoregressive processes. *Journal of Machine Learning Research*, 11:2671–2705, 2010.

[TDKC15] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. Composite self-concordant minimization. *Journal of Machine Learning Research*, 16:371–416, 2015.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[TLM+14] K. M. Tan, P. London, K. Mohan, S. Lee, M. Fazel, and D. Witten. Learning graphical models with hubs. *The Journal of Machine Learning Research*, 15(1):3297–3331, 2014.

[TSR+05] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[TY84] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13(3):566–579, 1984.

[VA14] L. Vandenberghe and M. S. Andersen. Chordal graphs and semidefinite optimization. *Foundations and Trends in Optimization*, 1(4):241–433, 2014.

[VWN+13] R. Vandenberghe, Y. Wang, N. Nelissen, M. Vandenbulcke, T. Dhollander, S. Sunaert, and P. Dupont. The associative-semantic network for words and pictures: Effective connectivity and graph analysis. *Brain and language*, 127(2):264–272, 2013.

[WJ08] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inferencing. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[WL08] T. Wu and K. Lange. Coordinate descent procedures for lasso penalized regression. *The Annals of Applied Statistics*, 2, 2008.

[WST10] C. Wang, D. Sun, and K. Toh. Solving log-determinant optimization problems by a newton-cg primal proximal point algorithm. *SIAM Journal on Optimization*, 20(6):2994–3013, 2010.

[YL06] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B. Statistical Methodology*, 68(1):49–67, 2006.

[YL07] M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19, 2007.

[Yua09] X. Yuan. Alternating direction methods for sparse covariance selection. 2009. Preprint available at `Optimization Online` (2009.09.2390).

[ZH05] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[Zou06] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101, 2006.

[ZS14] M. Zorzi and R. Sepulchre. AR identification of latent-variable graphical models, 2014. `arxiv.org/1405.0027`.

[ZW12] B. Zhang and Y. Wang. Learning structural changes of gaussian graphical models in controlled experiments. *arXiv preprint arXiv:1203.3532*, 2012.