Graph-Based Sparse Learning: Models, Algorithms, and Applications

by

Sen Yang

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved November 2014 by the
Graduate Supervisory Committee:

Jieping Ye, Co-Chair
Peter Wonka, Co-Chair
Yalin Wang
Jing Li

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

Sparse learning is a powerful tool to generate models of high-dimensional data with high interpretability, and it has many important applications in areas such as bioinformatics, medical image processing, and computer vision. Recently, the *a priori* structural information has been shown to be powerful for improving the performance of sparse learning models. A graph is a fundamental way to represent structural information of features. This dissertation focuses on graph-based sparse learning. The first part of this dissertation aims to integrate a graph into sparse learning to improve the performance. Specifically, the problem of feature grouping and selection over a given undirected graph is considered. Three models are proposed along with efficient solvers to achieve simultaneous feature grouping and selection, enhancing estimation accuracy. One major challenge is that it is still computationally challenging to solve large scale graph-based sparse learning problems. An efficient, scalable, and parallel algorithm for one widely used graph-based sparse learning approach, called anisotropic total variation regularization is therefore proposed, by explicitly exploring the structure of a graph. The second part of this dissertation focuses on uncovering the graph structure from the data. Two issues in graphical modeling are considered. One is the joint estimation of multiple graphical models using a fused lasso penalty and the other is the estimation of hierarchical graphical models. The key technical contribution is to establish the necessary and sufficient condition for the graphs to be decomposable. Based on this key property, a simple screening rule is presented, which reduces the size of the optimization problem, dramatically reducing the computational cost.

Dedicated to my parents

ACKNOWLEDGEMENTS

Over the past four and a half years I have received encouragement and support from a large number of people. Dr. Jieping Ye, my advisor, is the first person I have to thank. This dissertation would have never been possible without his consistent support and patient guidance. He is an outstanding mentor, and is always there when I need help. I am forever grateful for his mentorship. I also want to thank my co-advisor, Dr. Peter Wonka, for his guidance and support. He is the one who taught me how to start my PhD study. I also want to express my gratitude to my committee members, Dr. Yalin Wang and Dr. Jing Li, for their interactions and feedback.

It is my great fortune to study in the Machine Learning Lab and PRISM at Arizona State University. Every member inspired and helped me in different ways through insightful discussions and useful feedback. Particularly, I would like to give my thanks to Liang Sun, Jianhui Chen, Lei Yuan, Jiayu Zhou, Shuo Xiang, Qian Sun, Tao Yang, Yashu Liu, Qingyang Li, Xinlin Zhao, Zhi Nie, Pinghua Gong, Zheng Wang, Binbin Lin, Jie Wang, Fan Bao, Chi-Han Peng, and Kefei Liu.

I want to thank my parents for their invaluable love, understanding and supporting my graduate studies. Finally, I would like to thank my wife, Ran Wei, for accompanying me during this process.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

The fast advancement of data acquisition technologies in biology, engineering and social science, makes a significant amount of high-dimensional data available. The vast high-dimensional datasets pose a great challenge to statistical inference due to the curse of dimensionality. To make the inference possible, there is a need to develop new statistical and machine learning techniques to represent or approximate the complex high-dimensional dataset using a much smaller number of parameters than the original dimension. Sparse learning has been emerged as a powerful tool to generate models of high dimensional data with high interpretability, and it has many important applications in areas such as bioinformatics, medical image processing, and computer vision. A well-known sparse learning approach is the ℓ_1 regularization approach, known as lasso (Tibshirani, 1996), which can simultaneously perform feature selection and regression/classfication. However, in the presence of highly correlated features lasso tends to only select one of those features resulting in suboptimal performance (Zou and Hastie, 2005). Moreover, lasso lacks the ability to incorporate prior knowledge into the regression/classfication process, which is critical in many applications. As a motivating example, many biological studies have suggested that genes tend to work in groups according to their biological functions, and there are some regulatory relationships between genes (Li and Li, 2008). This biological knowledge can be represented as a graph, where the nodes represent the genes, and the edges imply the regulatory relationships between genes. In previous literature, many variants of group lasso such as standard group lasso, overlapping group lasso, and tree structured group lasso (Bach *et al.*, 2004; Jacob *et al.*, 2009; Yuan and Lin, 2006; Liu and Ye, 2010; Jacob *et al.*, 2009; Liu and Ye, 2010) have demonstrated the benefits of the prior knowledge in improving the performance of regression/classfication. A graph is a fundamental way to represent structural information of features. Many types of structural information can be encoded as a graph. This dissertation addresses the problem how to integrate a graph into the sparse learning process to improve the performance. Specifically, the problem of feature grouping and selection over a given undirected graph is considered. Three models are proposed along with efficient solvers to achieve simultaneous feature grouping and selection, enhancing estimation accuracy.

One major challenge is that it is still computationally challenging to solve large scale graph-based sparse learning models. The well-known total variation regularization uses a special graph structure, where each node is only connected to its neighbors. The wide range of applications of total variation regularization including image restoration, image denoising and deblurring (Barbero and Sra, 2011; Beck and Teboulle, 2009a; Huang *et al.*, 2011; Li *et al.*, 2009; Vogel and Oman, 1998; Yang *et al.*, 2009), underscore its success in signal/image processing. We therefore propose an efficient, scalable, and parallel algorithm for this widely used graph-based sparse learning approach, called the anisotropic total variation regularization model, by explicitly exploring the structure of a graph.

Due to the benefits of graph structural information in sparse learning, there is a need to uncover the graph structure from data sets. The second part of this dissertation focuses on how to estimate graph structures. Traditional approaches usually make simple assumptions such as ignoring dynamic changes among graphs or a single type of edge. These approaches may not be suitable for data nowadays, which can be noisy, high dimensional, and dynamic. A motivating example is the analysis of brain networks of Alzheimer's disease using neuroimaging data. Specifically, we may wish to estimate a brain network for the normal controls (NC), a brain network for the patients with mild cognitive impairment (MCI), and a brain network for Alzheimer's patients (AD). We expect the two brain networks for NC and MCI to share common structures but not to be identical to each other; similarly for the two brain networks for MCI and AD. We consider the problem of estimating multiple graphical models simultaneously. Compared with estimating each graph separately, joint estimation of multiple graphical models can utilize the information of underlying common structure, thus yields a better estimation. In addition to an efficient second-order method, we propose a necessary and sufficient screening rule which decomposes the large graphs into small subgraphs and allows an efficient estimation of multiple independent (small) subgraphs, dramatically reducing the computational cost.

1.1 Notation

In this dissertation, $\Re$ stands for the set of all real numbers, $\Re^n$ denotes the n-dimensional Euclidean space, and the set of all $m \times n$ matrices with real entries is denoted by $\Re^{m\times n}$. All matrices are presented in bold format. The space of symmetric matrices is denoted by $\mathcal{S}^n$. If $\mathbf{X} \in \mathcal{S}^n$ is positive semidefinite (resp. definite), we write $\mathbf{X} \succeq 0$ (resp. $\mathbf{X} \succ 0$). Also, we write $\mathbf{X} \succeq \mathbf{Y}$ to mean $\mathbf{X} - \mathbf{Y} \succeq 0$. The cone of positive semidefinite matrices in $\mathcal{S}^n$ is denoted by $\mathcal{S}^n_+$. Given matrices $\mathbf{X}$ and $\mathbf{Y}$ in $\Re^{m\times n}$, the standard inner product is defined by $\langle \mathbf{X}, \mathbf{Y} \rangle := \mathrm{tr}(\mathbf{X}\mathbf{Y}^T)$, where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. $\mathbf{X} \circ \mathbf{Y}$ and $\mathbf{X} \otimes \mathbf{Y}$ means the Hadamard and Kronecker product of $\mathbf{X}$ and $\mathbf{Y}$, respectively. We denote the identity matrix by $\mathbf{I}$, whose dimension should be clear from the context. The determinant and the minimal eigenvalue of a real symmetric matrix $\mathbf{X}$ are denoted by $\det(\mathbf{X})$ and $\lambda_{\min}(\mathbf{X})$, respectively. Given a matrix $\mathbf{X} \in \Re^{n\times n}$, $\mathrm{diag}(\mathbf{X})$ denotes the vector formed by the diagonal of $\mathbf{X}$, that is, $\mathrm{diag}(\mathbf{X})_i = \mathbf{X}_{ii}$ for $i = 1, \ldots, n$. $\mathrm{Diag}(\mathbf{X})$ is the diagonal matrix which shares the same diagonal as $\mathbf{X}$. vec($\mathbf{X}$) is the vectorization of $\mathbf{X}$. In addition, $\mathbf{X} > 0$ means that all entries of $\mathbf{X}$ are positive. A d-mode tensor (or d-order tensor) is defined as $\mathcal{X} \in \Re^{I_1 \times I_2 \times \cdots \times I_d}$. Its entries are denoted as $x_{j_1,\ldots,j_d}$, where $1 \leq j_k \leq I_k$, $1 \leq k \leq d$. For example, 1-mode tensor is a vector, and 2-mode tensor is a matrix. $\mathbf{x}_{j_1,\ldots,j_{i-1},:,j_{i+1},\ldots,j_d}$ denotes the mode-i fiber at $\{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_d\}$, which is the higher order analogue of matrix rows and columns. The Frobenius norm of a tensor is defined as $\|\mathcal{X}\|_F = (\sum_{j_1,j_2,\ldots,j_d} x^2_{j_1,j_2,\ldots,j_d})^{\frac{1}{2}}$. The inner product in the tensor space is defined as $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{j_1,j_2,\ldots,j_d} x_{j_1,j_2,\ldots,j_d}\, y_{j_1,j_2,\ldots,j_d}$. For simplicity of notation, we use $/\{j_i\}$ to represent the index set excluding j_i, i.e., $\{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_d\}$. For instance, $\sum_{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_d}$ can be simply written as $\sum_{/\{j_i\}}$.

Chapter 2

FEATURE GROUPING AND SELECTION OVER AN UNDIRECTED GRAPH

2.1 Introduction

High-dimensional regression/classification is challenging due to the *curse of dimensionality*. Lasso (Tibshirani, 1996) and its various extensions, which can simultaneously perform feature selection and regression/classification, have received increasing attention in this situation. However, in the presence of highly correlated features lasso tends to only select one of those features resulting in suboptimal performance (Zou and Hastie, 2005). Several methods have been proposed to address this issue in the literature. Shen and Ye (2002) introduce an adaptive model selection procedure that corrects the estimation bias through a data-driven penalty based on generalized degrees of freedom. Elastic Net (Zou and Hastie, 2005) uses an additional l_2 regularizer to encourage highly correlated features to stay together. However, these methods do not incorporate prior knowledge into the regression/classification process, which is critical in many applications. As an example, many biological studies have suggested that genes tend to work in groups according to their biological functions, and there are some regulatory relationships between genes (Li and Li, 2008). This biological knowledge can be represented as a graph, where the nodes represent the genes, and the edges imply the regulatory relationships between genes. Therefore, we want to study how estimation accuracy can be improved using dependency information encoded as a graph.

Given feature grouping information, the group lasso (Bach *et al.*, 2004; Jacob *et al.*, 2009; Yuan and Lin, 2006; Liu and Ye, 2010; Xiang *et al.*, 2013b, 2014, 2013a) yields a solution with grouped sparsity using l_1/l_2 penalty. The orignal group lasso does not consider the overlaps between groups. Zhao *et al.* (2009) extend the group lasso to the case of overlapping groups. Jacob *et al.* (2009) introduce a new penalty function leading to a grouped sparse solution with overlapping groups. Yuan *et al.* (2011) propose an efficient method to solve the overlapping group lasso. Other extensions of group lasso with tree structured regularization include (Liu and Ye, 2010; Jenatton *et al.*, 2010). Prior works have demonstrated the benefit of using feature grouping information for high-dimensional regression/classification. However, these methods need the feature groups to be pre-specified. In other words, they only utilize the grouping information to obtain solutions with grouped sparsity, but lack the capability of identifying groups.

There are also a number of existing methods for feature grouping. Fused lasso (Tibshirani *et al.*, 2005) introduces an l_1 regularization method for estimating subgroups in a certain serial order, but pre-ordering features is required before using fused lasso. A study about parameter estimation of the fused lasso can be found in (Rinaldo, 2009); Shen and Huang (2010) propose a non-convex method to select all possible homogenous subgroups, but it fails to obtain sparse solutions. OSCAR (Bondell and Reich, 2008) employs an l_1 regularizer and a pairwise l_∞ regularizer to perform feature selection and automatic feature grouping. Li and Li (2008) suggest a grouping penalty using a Laplacian matrix to force the coefficients to be similar, which can be considered as a graph version of Elastic Net. When the Laplacian matrix is an identity matrix, Laplacian lasso (Li and Li, 2008; Fei *et al.*, 2010) is identical to Elastic Net. GFlasso employs an l_1 regularization over a graph, which penalizes the difference $|\beta_i - \text{sign}(r_{ij})\beta_j|$, to encourage the coefficients β_i, β_j for features i, j connected by an edge in the graph to be similar when $r_{ij} > 0$, but dissimilar when $r_{ij} < 0$, where r_{ij} is the sample correlation between two features (Kim and Xing, 2009). Although these grouping penalties can improve the performance, they would introduce additional estimation bias due to strict convexity of the penalties or due to possible graph misspecification. For example, additional bias may occur when the signs of coefficients for two features connected by an edge in the graph are different in Laplacian lasso (Li and Li, 2008; Fei *et al.*, 2010), or when the sign of r_{ij} is inaccurate in GFlasso (Kim and Xing, 2009).

In this chapter, we focus on simultaneous estimation of grouping and sparseness structures over a given undirected graph. Features tend to be grouped when they are connected by an edge in a graph. When features are connected by an edge in a graph, the absolute values of the model coefficients for these two features should be similar or identical. We propose one convex and two non-convex penalties to encourage both sparsity and equality of absolute values of coefficients for connected features. The convex penalty includes a pairwise l_∞ regularizer over a graph. The first non-convex penalty improves the convex penalty by penalizing the difference of absolute values of coefficients for connected features. The other one is the extension of the first non-convex penalty using a truncated l_1 regularization to further reduce the estimation bias. These penalties are designed to resolve the aforementioned issues of Laplacian lasso and GFlasso. The non-convex penalties shrink only small differences in absolute values so that estimation bias can be reduced. Through ADMM and DC programming, we develop computational methods to solve the proposed formulations. The proposed methods can combine the benefit of feature selection and that of feature grouping to improve regression/classification performance. Due to the equality of absolute values of coefficients, the model complexity of the learned model can be reduced. We have performed experiments on synthetic data and two real datasets. The results demonstrate the effectiveness of the proposed methods.

The main contributions of this chapter are summarized as follows:

- We propose a convex solution to OSCAR over an arbitrary undirected graph, called graph OSCAR (GOSCAR);
- We propose two non-convex methods for simultaneous feature grouping and selection. The basic method is called ncFGS and the extension using the truncated l_1 regularization is called ncTFGS;
- We show that feature grouping and feature selection are complementary through the proposed non-convex methods.

The rest of the chapter is organized as follows. We introduce the proposed convex method in Section 2.2, and describes the two proposed non-convex methods in Section 2.3. Experimental results are given in Section 2.4. We conclude the chapter in Section 2.5.

2.2 A Convex Formulation

Consider a linear model in which response y_i depends on a vector of p features:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon, \tag{2.1}$$

where $\beta \in \Re^p$ is a vector of coefficients, $\mathbf{X} \in \Re^{n \times p}$ is the data matrix, and ϵ is random noise. Given an undirected graph, we try to build a prediction model (regression or classification) incorporating the graph structure information to estimate the nonzero coefficients of β and identify the feature groups when the number of features p is larger than the sample size n. Let (N, E) be the given undirected graph, where $N = \{1, 2, \dots, p\}$ is a set of nodes, and E is the set of edges. Node i corresponds to feature $\mathbf{x}_i$. If nodes i and j are connected by an edge in E, then features $\mathbf{x}_i$ and $\mathbf{x}_j$ tend to be grouped. The formulation of graph OSCAR (GOSCAR) is given by

$$\min_{\beta} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1\|\beta\|_1 + \lambda_2 \sum_{(i,j)\in E} \max\{|\beta_i|, |\beta_j|\} \tag{2.2}$$

where $\lambda_1,\ \lambda_2$ are regularization parameters. We use a pairwise l_∞ regularizer to encourage the coefficients to be equal (Bondell and Reich, 2008), but we only put grouping constraints over the nodes connected over the given graph. The l_1 regularizer encourages sparseness. The pairwise l_∞ regularizer puts more penalty on the larger coefficients. Note that $\max\{|\beta_i|, |\beta_j|\}$ can be decomposed as

$$\max\{|\beta_i|, |\beta_j|\} = \frac{1}{2}(|\beta_i + \beta_j| + |\beta_i - \beta_j|).$$

$\frac{1}{2}(|\beta_i + \beta_j| + |\beta_i - \beta_j|)$ can be represented by

$$|\mathbf{u}^T\beta| + |\mathbf{v}^T\beta|,$$

where $\mathbf{u}, \mathbf{v}$ are sparse vectors, each with only two non-zero entries $u_i = u_j = \frac{1}{2},\ v_i = -v_j = \frac{1}{2}$. Thus (2.2) can be rewritten in a matrix form as

$$\min_{\beta} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\mathbf{T}\beta\|_1, \tag{2.3}$$

where $\mathbf{T}$ is a sparse matrix constructed from the edge set E.

The proposed formulation is closely related to OSCAR (Bondell and Reich, 2008). The penalty of OSCAR is $\lambda_1\|\beta\|_1 + \lambda_2\sum_{i<j}\max\{|\beta_i|, |\beta_j|\}$. The l_1 regularizer leads to a sparse solution, and the l_∞ regularizer encourages the coefficients to be equal. OSCAR can be efficiently solved by accelerated gradient methods, whose key projection can be solved by a simple iterative group merging algorithm (Zhong and Kwok, 2011). However, OSCAR assumes each node is connected to all the other nodes, which is not sufficient for many applications. Note that OSCAR is a special case of GOSCAR when the graph is complete. GOSCAR, incorporating an arbitrary undirected graph, is much more challenging to solve.

2.2.1 Algorithm

We propose to solve GOSCAR using the alternating direction method of multipliers (ADMM) Boyd *et al.* (2011). ADMM decomposes a large global problem into a series of smaller local subproblems and coordinates the local solutions to identify the globally optimal solution. ADMM attempts to combine the benefits of dual decomposition and augmented Lagrangian methods for constrained optimization (Boyd *et al.*, 2011). The problem solved by ADMM takes the form of

$$\begin{aligned} &\min\nolimits_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \\ &s.t.\ \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}. \end{aligned}$$

ADMM uses a variant of the augmented Lagrangian method and reformulates the problem as follows:

$$L_\rho(\mathbf{x}, \mathbf{z}, \mu) = f(\mathbf{x}) + g(\mathbf{z}) + \mu^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|^2,$$

with μ being the augmented Lagrangian multiplier, and ρ being the non-negative dual update step length. ADMM solves this problem by iteratively minimizing $L_\rho(\mathbf{x}, \mathbf{z}, \mu)$ over $\mathbf{x}$, $\mathbf{z}$, and μ. The update rule for ADMM is given by

$$\begin{aligned} \mathbf{x}^{k+1} &:= \arg\min_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{z}^k, \mu^k), \\ \mathbf{z}^{k+1} &:= \arg\min_{\mathbf{z}} L_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \mu^k), \\ \mu^{k+1} &:= \mu^k + \rho(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}). \end{aligned}$$

Consider the unconstrained optimization problem in (2.3), which is equivalent to the following constrained optimization problem:

$$\begin{aligned} &\min\nolimits_{\beta,\mathbf{q},\mathbf{p}} \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1\|\mathbf{q}\|_1 + \lambda_2\|\mathbf{p}\|_1 \\ &s.t \quad \beta - \mathbf{q} = \mathbf{0},\ \mathbf{T}\beta - \mathbf{p} = \mathbf{0}, \end{aligned} \tag{2.4}$$

where $\mathbf{q}, \mathbf{p}$ are slack variables. (2.4) can then be solved by ADMM. The augmented Lagrangian is

$$\begin{aligned} L_\rho(\beta, \mathbf{q}, \mathbf{p}, \mu, \upsilon) &= \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1\|\mathbf{q}\|_1 + \lambda_2\|\mathbf{p}\|_1 \\ &+ \mu^T(\beta - \mathbf{q}) + \upsilon^T(\mathbf{T}\beta - \mathbf{p}) + \tfrac{\rho}{2}\|\beta - \mathbf{q}\|^2 + \tfrac{\rho}{2}\|\mathbf{T}\beta - \mathbf{p}\|^2, \end{aligned}$$

where μ, υ are augmented Lagrangian multipliers.

Update β: In the $(k+1)$-th iteration, β^{k+1} can be updated by minimizing L_ρ with $\mathbf{q}, \mathbf{p}, \mu, \upsilon$ fixed:

$$\begin{aligned} \beta^{k+1} = \arg\min_\beta \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + (\mu^k + \mathbf{T}^T\upsilon^k)^T\beta \\ + \tfrac{\rho}{2}\|\beta - \mathbf{q}^k\|^2 + \tfrac{\rho}{2}\|\mathbf{T}\beta - \mathbf{p}^k\|^2. \end{aligned} \tag{2.5}$$

The above optimization problem is quadratic. The optimal solution is given by $\beta^{k+1} = \mathbf{F}^{-1}\mathbf{b}^k$, where

$$\begin{aligned} \mathbf{F} &= \mathbf{X}^T\mathbf{X} + \rho(\mathbf{I} + \mathbf{T}^T\mathbf{T}), \\ \mathbf{b}^k &= \mathbf{X}^T\mathbf{y} - \mu^k - \mathbf{T}^T\upsilon^k + \rho\mathbf{T}^T\mathbf{p}^k + \rho\mathbf{q}^k. \end{aligned}$$

The computation of β^{k+1} involves solving a linear system, which is the most time-consuming part in the whole algorithm. To compute β^{k+1} efficiently, we compute the Cholesky factorization of $\mathbf{F}$ at the beginning of the algorithm:

$$\mathbf{F} = \mathbf{R}^T\mathbf{R}.$$

Note that $\mathbf{F}$ is a constant and positive definite matrix. Using the Cholesky factorization we only need to solve the following two linear systems at each iteration:

$$\mathbf{R}^T\hat{\beta} = \mathbf{b}^k, \ \ \mathbf{R}\beta = \hat{\beta}. \tag{2.6}$$

Since $\mathbf{R}$ is an upper triangular matrix, solving these two linear systems is very efficient.

Update q: $\mathbf{q}^{k+1}$ can be obtained by solving

$$\mathbf{q}^{k+1} = \arg\min_{\mathbf{q}} \frac{\rho}{2}\|\mathbf{q} - \beta^{k+1}\|^2 + \lambda_1\|\mathbf{q}\|_1 - (\mu^k)^T\mathbf{q}$$

which is equivalent to the following problem:

$$\mathbf{q}^{k+1} = \arg\min_{\mathbf{q}} \frac{1}{2}\|\mathbf{q} - \beta^{k+1} - \frac{1}{\rho}\mu^k\|^2 + \frac{\lambda_1}{\rho}\|\mathbf{q}\|_1 \tag{2.7}$$

(2.7) has a closed-form solution, known as *soft-thresholding*:

$$\mathbf{q}^{k+1} = S_{\lambda_1/\rho}(\beta^{k+1} + \frac{1}{\rho}\mu^k), \tag{2.8}$$

where the *soft-thresholding operator* is defined as:

$$S_\lambda(x) = \text{sign}(x)\max(|x| - \lambda, 0).$$

Update p: Similar to updating **q**, $\mathbf{p}^{k+1}$ can also be obtained by *soft-thresholding*:

$$\mathbf{p}_i^{k+1} = S_{\lambda_2/\rho}(\mathbf{T}\beta^{k+1} + \frac{1}{\rho}\upsilon^k). \tag{2.9}$$

Update μ, υ:

$$\begin{aligned} \mu^{k+1} &= \mu^k + \rho(\beta^{k+1} - \mathbf{q}^{k+1}), \\ \upsilon^{k+1} &= \upsilon^k + \rho(\mathbf{T}\beta^{k+1} - \mathbf{p}^{k+1}). \end{aligned} \tag{2.10}$$

A summary of GOSCAR is shown in Algorithm 1.

Algorithm 1: The GOSCAR algorithm

```
Input: $\mathbf{X}, \mathbf{y}, E, \lambda_1, \lambda_2, \rho$
Output: $\beta$
Initialization: $\mathbf{p}^0 \leftarrow \mathbf{0}, \mathbf{q}^0 \leftarrow \mathbf{0}, \mu^0 \leftarrow \mathbf{0}, \upsilon^0 \leftarrow \mathbf{0}$;
Compute the Cholesky factorization of $\mathbf{F}$;
do
    Compute $\beta^{k+1}$ according to (2.6).
    Compute $\mathbf{q}^{k+1}$ according to (2.8).
    Compute $\mathbf{p}^{k+1}$ according to (2.9).
    Compute $\mu^{k+1}, \upsilon^{k+1}$ according to (2.10).
Until Convergence;
return $\beta$;
```

In Algorithm 1, the Cholesky factorization only needs to be computed once, and each iteration involves solving one linear system and two soft-thresholding operations. The time complexity of the soft-thresholding operation in (2.8) is $O(p)$. The other one in (2.9) involves a matrix-vector multiplication $\mathbf{T}\beta^{k+1}$. Due to the sparsity of $\mathbf{T}$, its time complexity is $O(n_e)$, where n_e is the number of edges. Solving the linear system involves computing $\mathbf{b}^k$ and solving (2.6), whose total time complexity is $O(p(p+n)+n_e)$. Thus the time complexity of each iteration is $O(p(p+n)+n_e)$.

2.3 Two Non-Convex Formulations

The grouping penalty of GOSCAR overcomes the limitation of Laplacian lasso that the different signs of coefficients can introduce additional penalty. However, under the l_∞ regularizer, even if $|\beta_i|$ and $|\beta_j|$ are close to each other, the penalty on this pair may still be large due to the property of max operator, resulting in the coefficient β_i or β_j being over penalized. The additional penalty would result in biased estimation, especially for large coefficient, as in the lasso case (Tibshirani, 1996). Another related grouping penalty is GFlasso, $|\beta_i - \text{sign}(r_{ij})\beta_j|$, where r_{ij} is the pairwise sample correlation. GFlasso relies on the pairwise sample correlation to decide whether β_i and β_j are enforced to be close or not. When the pairwise sample correlation wrongly estimates the sign between β_i and β_j, additional penalty on β_i and β_j would occur, introducing the estimation bias. This motivates our non-convex grouping penalty, $||\beta_i| - |\beta_j||$, that shrinks only small differences in absolutes values. As a result, estimation bias is reduced as compared to these convex grouping penalties. The proposed non-convex methods perform well even when the graph is wrongly specified, unlike GFlasso. Note that the proposed non-convex grouping penalty does not assume the sign of an edge is given; it only relies on the graph structure.

2.3.1 Non-Convex Formulation I: ncFGS

The proposed non-convex formulation (ncFGS) solves the following optimization problem:

$$\min_{\beta} f(\beta) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1\|\beta\|_1 + \lambda_2 \sum_{(i,j)\in E} ||\beta_i| - |\beta_j||, \tag{2.11}$$

where the grouping penalty $\sum_{(i,j)\in E} ||\beta_i| - |\beta_j||$ controls only magnitudes of differences of coefficients ignoring their signs over the graph. Through the l_1 regularizer and grouping penalty, simultaneous feature grouping and selection are performed, where only large coefficients as well as pairwise differences are shrunk.

A computational method for the non-convex optimization in (2.11) is through DC programming. We will first give a brief review of DC programming.

A particular DC program on $\Re^p$ takes the form of

$$f(\beta) = f_1(\beta) - f_2(\beta)$$

with $f_1(\beta)$ and $f_2(\beta)$ being convex on $\Re^p$. The algorithms to solve DC programming based on the duality and local optimality conditions have been introduced in (Tao and El Bernoussi, 1988). Due to their local characteristic and the non-convexity of DC programming, these algorithms cannot guarantee the computed solution to be globally optimal. In general, these DC algorithms converge to a local solution, but some researchers observed that they converge quite often to a global one (Tao and An, 1997).

To apply DC programming to our problem we need to decompose the objective function into the difference of two convex functions. We propose to use:

$$f_1(\beta) = \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1\|\beta\|_1 + \lambda_2 \textstyle\sum_{(i,j)\in E}(|\beta_i + \beta_j| + |\beta_i - \beta_j|),$$

$$f_2(\beta) = \lambda_2 \textstyle\sum_{(i,j)\in E} (|\beta_i| + |\beta_j|).$$

The above DC decomposition is based on the following identity: $||\beta_i| - |\beta_j|| = |\beta_i + \beta_j| + |\beta_i - \beta_j| - (|\beta_i| + |\beta_j|)$. Note that both $f_1(\beta)$ and $f_2(\beta)$ are convex functions.

Denote $f_2^k(\beta) = f_2(\beta^k) + \langle \beta - \beta^k, \partial f_2(\beta^k) \rangle$ as the affine minorization of $f_2(\beta)$, where $\langle \cdot, \cdot \rangle$ is the inner product. Then DC programming solves (2.11) by iteratively solving a sub-problem as follows:

$$\min_{\beta} f_1(\beta) - f_2^k(\beta). \tag{2.12}$$

Since $\langle \beta^k, \partial f_2(\beta^k) \rangle$ is constant, (2.12) can be rewritten as

$$\min_{\beta} f_1(\beta) - \langle \beta, \partial f_2(\beta^k) \rangle. \tag{2.13}$$

Let $\mathbf{c}^k = \partial f_2(\beta^k)$. Note that

$$c_i^k = \lambda_2 d_i \text{sign}(\beta_i^k) \mathbb{I}(\beta_i^k \neq 0), \tag{2.14}$$

where d_i is the degree of node i, and $\mathbb{I}(\cdot)$ is the indicator function. Hence, the formulation in (2.13) is

$$\min_\beta \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1 \|\beta\|_1 - (\mathbf{c}^k)^T \beta + \lambda_2 \textstyle\sum_{(i,j)\in E} (|\beta_i + \beta_j| + |\beta_i - \beta_j|), \tag{2.15}$$

which is convex. Note that the only differences between the problems in (2.2) and (2.15) are the linear term $(\mathbf{c}^k)^T\beta$ and the second regularization parameter. Similar to GOSCAR, we can solve (2.15) using ADMM, which is equivalent to the following optimization problem:

$$\begin{aligned} &\min_{\beta,\mathbf{q},\mathbf{p}} \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 - (\mathbf{c}^k)^T\beta + \lambda_1\|\mathbf{q}\|_1 + 2\lambda_2\|\mathbf{p}\|_1 \\ &s.t \quad \beta - \mathbf{q} = \mathbf{0},\ \mathbf{T}\beta - \mathbf{p} = \mathbf{0}. \end{aligned} \tag{2.16}$$

There is an additional linear term $(\mathbf{c}^k)^T\beta$ in updating β compared to Algorithm 1. Hence, we can use Algorithm 1 to solve (2.15) with a small change in updating β:

$$\mathbf{F}\beta - \mathbf{b}^s - \mathbf{c}^k = 0.$$

where s represents the iteration number in Algorithm 1.

The key steps of ncFGS are shown in Algorithm 2.

Algorithm 2: The ncFGS algorithm

```
Input: $\mathbf{X}, \mathbf{y}, E, \lambda_1, \lambda_2, \epsilon$
Output: $\beta$
Initialization: $\beta^0 \leftarrow \mathbf{0}$;
while $f(\beta^k) - f(\beta^{k+1}) > \epsilon$ do
    Compute $\mathbf{c}^k$ according to (2.14).
    Compute $\beta^{k+1}$ using Algorithm 1 with $\mathbf{c}^k$ and $\lambda_1, 2\lambda_2$ as regularization
    parameters.
end
return $\beta$;
```

2.3.2 Non-Convex Formulation II: ncTFGS

It is known that the bias of lasso is due to the looseness of convex relaxation of l_0 regularization. The truncated l_1 regularizer, a non-convex regularizer close to the l_0 regularizer, has been proposed to resolve the bias issue (Zhang, 2013). The truncated l_1 regularizer can recover the exact set of nonzero coefficients under a weaker condition, and has a smaller upper error bound than lasso (Zhang, 2013). Therefore, we propose a truncated grouping penalty to further reduce the estimation bias. The proposed formulation based on the truncated grouping penalty is

$$\min_\beta f_T(\beta) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1 p_1(\beta) + \lambda_2 p_2(\beta) \tag{2.17}$$

where

$$p_1(\beta) = \sum_i J_\tau(|\beta_i|),$$
$$p_2(\beta) = \sum_{(i,j)\in E} J_\tau(||\beta_i| - |\beta_j||),$$

and $J_\tau(x) = \min(\frac{x}{\tau}, 1)$ is the truncated l_1 regularizer, a surrogate of the l_0 function; τ is a non-negative tuning parameter. Figure 2.1 shows the difference between l_0 norm, l_1 norm and $J_\tau(|x|)$. When $\tau \to 0$, $J_\tau(|x|)$ is equivalent to the l_0 norm given by the

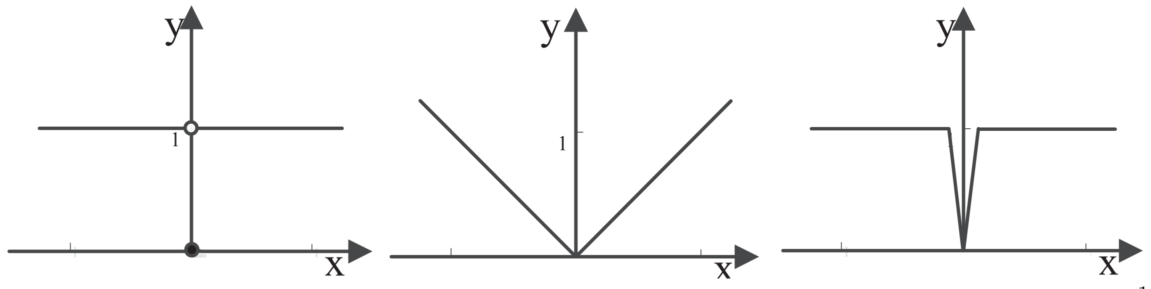


Figure 2.1: Example for l_0 norm (left), l_1 norm (middle), and $J_\tau(|x|)$ with $\tau = \frac{1}{8}$ (right).

number of nonzero entries of a vector. When $\tau \geq |x|$, $\tau J_\tau(|x|)$ is equivalent to the l_1 norm of x.

Note that $J_\tau(||\beta_i| - |\beta_j||)$ can be decomposed as

$$J_\tau(||\beta_i| - |\beta_j||) = \tfrac{1}{\tau}(|\beta_i + \beta_j| + |\beta_i - \beta_j|) - \tfrac{1}{\tau}\max(2|\beta_i| - \tau, 2|\beta_j| - \tau, |\beta_i| + |\beta_j|),$$

and a DC decomposition of $J_\tau(|\beta_i|)$ is

$$J_\tau(|\beta_i|) = \frac{1}{\tau}|\beta_i| - \frac{1}{\tau}\max(|\beta_i| - \tau, 0).$$

Hence, the DC decomposition of $f_T(\beta)$ can be written as

$$f_T(\beta) = f_{T,1}(\beta) - f_{T,2}(\beta),$$

where

$$\begin{aligned} f_{T,1}(\beta) &= \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \tfrac{\lambda_1}{\tau}\|\beta\|_1 + \tfrac{\lambda_2}{\tau}\textstyle\sum_{(i,j)\in E}(|\beta_i + \beta_j| + |\beta_i - \beta_j|), \\ f_{T,2}(\beta) &= \tfrac{\lambda_1}{\tau}\textstyle\sum_i \max(|\beta_i| - \tau, 0) + \tfrac{\lambda_2}{\tau}\textstyle\sum_{(i,j)\in E}\max(2|\beta_i| - \tau, 2|\beta_j| - \tau, |\beta_i| + |\beta_j|). \end{aligned}$$

Let $\mathbf{c}_T^k = \partial f_{T,2}(\beta^k)$ be the subgradient of $f_{T,2}$ in the $(k+1)$-th iteration. We have

$$\begin{aligned} c_{T,i}^k = \operatorname{sign}(\beta_i^k)\big(&\tfrac{\lambda_1}{\tau}\mathbb{I}(|\beta_i^k| > \tau) + \tfrac{\lambda_2}{\tau}\textstyle\sum_{j:(i,j)\in E} \\ &(2\mathbb{I}(|\beta_j^k| < |\beta_i^k| - \tau) + \mathbb{I}(||\beta_i^k| - |\beta_j^k|| < \tau))\big). \end{aligned} \tag{2.18}$$

Then the subproblem of ncTFGS is

$$\min_\beta \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \tfrac{\lambda_1}{\tau}\|\beta\|_1 - (\mathbf{c}_T^k)^T\beta + \tfrac{\lambda_2}{\tau}\textstyle\sum_{(i,j)\in E}(|\beta_i + \beta_j| + |\beta_i - \beta_j|), \tag{2.19}$$

Algorithm 3: The ncTFGS algorithm

```
Input: $\mathbf{X}, \mathbf{y}, E, \lambda_1, \lambda_2, \tau, \epsilon$
Output: $\beta$
Initialization: $\beta^0 \leftarrow \mathbf{0}$;
while $f(\beta^k) - f(\beta^{k+1}) > \epsilon$ do
    Compute $\mathbf{c}_T^k$ according to (2.18).
    Compute $\beta^{k+1}$ using Algorithm 1 with $\mathbf{c}_T^k$ and $\frac{\lambda_1}{\tau}, \frac{2\lambda_2}{\tau}$ as regularization parameters.
end
return $\beta$;
```

which can be solved using Algorithm 1 as in ncFGS.

The key steps of ncTFGS are summarized in Algorithm 3.

ncTFGS is an extension of ncFGS. When $\tau \geq |\beta_i|, \forall i$, ncTFGS with regularization parameters $\tau\lambda_1$ and $\tau\lambda_2$ is identical to ncFGS (see Figure 2.3). ncFGS and ncTFGS have the same time complexity. The subproblems of ncFGS and ncTFGS are solved by Algorithm 1. In our experiments, we observed ncFGS and ncTFGS usually converge in less than 10 iterations.

2.4 Numerical Results

We examine the performance of the proposed methods and compare them against lasso, GFlasso, and OSCAR on synthetic datasets and two real datasets: FDG-PET images [1] and Breast Cancer [2]. The experiments are performed on a PC with dual-core Intel 3.0GHz CPU and 4G memory. The source codes written in MATLAB are

[1]http://adni.loni.ucla.edu/

[2]http://cbio.ensmp.fr/~jvert/publi/

available online [3] . The algorithms and their associated penalties are:

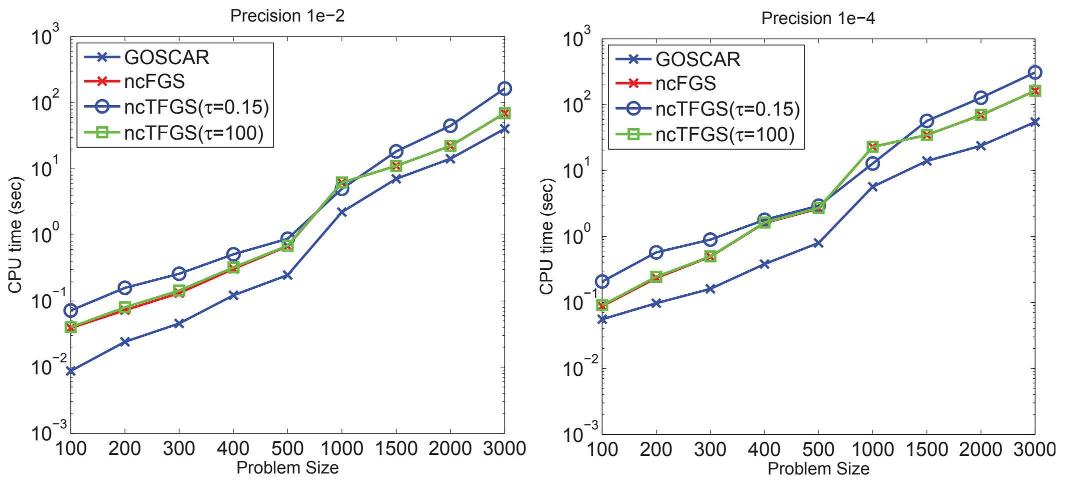
- Lasso: $\lambda_1\|\beta\|_1$;
- OSCAR: $\lambda_1\|\beta\|_1 + \lambda_2 \sum_{i<j} \max\{|\beta_i|, |\beta_j|\}$;
- GFlasso: $\lambda_1\|\beta\|_1 + \lambda_2 \sum_{(i,j)\in E} |\beta_i - \text{sign}(r_{ij})\beta_j|$;
- GOSCAR: $\lambda_1\|\beta\|_1 + \lambda_2 \sum_{(i,j)\in E} \max\{|\beta_i|, |\beta_j|\}$;
- ncFGS: $\lambda_1\|\beta\|_1 + \lambda_2 \sum_{(i,j)\in E} ||\beta_i| - |\beta_j||$;
- ncTFGS: $\lambda_1 \sum_i J_\tau(|\beta_i|) + \lambda_2 \sum_{(i,j)\in E} J_\tau(||\beta_i| - |\beta_j||)$;

2.4.1 Efficiency

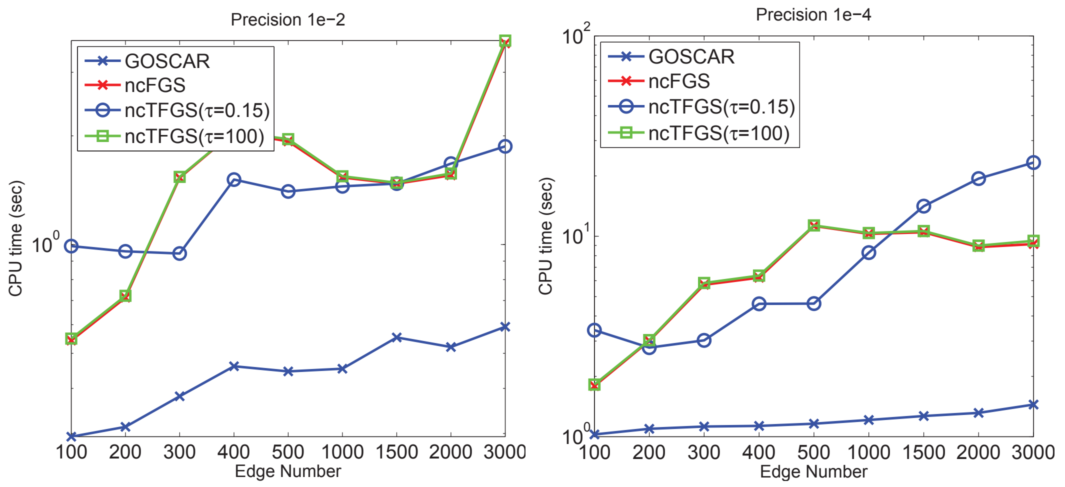
To evaluate the efficiency of the proposed methods, we conduct experiments on a synthetic dataset with a sample size of 100 and dimensions varying from 100 to 3000. The regression model is $\mathbf{y} = \mathbf{X}\beta + \epsilon$, where $\mathbf{X} \sim \mathcal{N}(0, \mathbf{I}_{p\times p})$, $\beta_i \sim \mathcal{N}(0,1)$, and $\epsilon_i \sim \mathcal{N}(0, 0.01^2)$. The graph is randomly generated. The number of edges n_e varies from 100 to 3000. The regularization parameters are set as $\lambda_1 = \lambda_2 = 0.8 \max\{|\beta_i|\}$ with n_e fixed. Since the graph size affects the penalty, λ_1 and λ_2 are scaled by $\frac{1}{n_e}$ to avoid trivial solutions with dimension p fixed. The average computational time based on 30 repetitions is reported in Figure 2.2. As can be seen in Figure 2.2, GOSCAR can achieve 1e-4 precision in less than 10s when the dimension and the number of edges are 1000. The computational time of ncTFGS is about 7 times higher than that of GOSCAR in this experiment. The computational time of ncFGS is the same as that of ncTFGS when $\tau = 100$, and very close to that of ncTFGS when $\tau = 0.15$. We can also observe that the proposed methods scale very well to the number of edges. The computational time of the proposed method increases less than 4 times

[3] http://www.public.asu.edu/~jye02/GraphOSCAR

when the number of edges increases from 100 to 3000. It is not surprising because the time complexity of each iteration in Algorithm 1 is linear with respect to n_e, and the sparsity of $\mathbf{T}$ makes the algorithm much more efficient. The increase of dimension is more costly than that of the number of edges, as the complexity of each iteration is quadratic with respect to p.



(a) The number of edges is fixed to 1000.



(b) The dimension is fixed to 500.

Figure 2.2: Comparison of GOSCAR, ncFGS, ncTFGS ($\tau = 0.15$), and ncTFGS ($\tau = 100$) in terms of computation time with different dimensions, precisions and the numbers of edges (in seconds and in logarithmic scale).

2.4.2 Simulations

We use five synthetic problems that have been commonly used in the sparse learning literature (Bondell and Reich, 2008; Li and Li, 2008) to compare the performance of different methods. The data is generated from the regression model $\mathbf{y} = \mathbf{X}\beta + \epsilon, \;\; \epsilon_i \sim \mathcal{N}(0, \sigma^2)$. The five problems are given by:

1. $n = 100, \; p = 40$, and $\sigma = 2, 5, 10$. The true parameter is given by

$$\beta = (\underbrace{0, \ldots, 0}_{10}, \underbrace{2, \ldots, 2}_{10}, \underbrace{0, \ldots, 0}_{10}, \underbrace{2, \ldots, 2}_{10})^T.$$

$\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{S_{p\times p}})$ with $s_{ii} = 1, \forall i$ and $s_{ij} = 0.5$ for $i \neq j$.

2. $n = 50, \; p = 40, \; \beta = (\underbrace{3, \ldots, 3}_{15}, \underbrace{0, \ldots, 0}_{25})^T$, and $\sigma = 2, 5, 10$. The features are generated as

$$\begin{aligned} \mathbf{x}_i &= Z_1 + \epsilon_i^x, \; Z_1 \sim \mathcal{N}(0, 1), \quad i = 1, \ldots, 5 \\ \mathbf{x}_i &= Z_2 + \epsilon_i^x, \; Z_2 \sim \mathcal{N}(0, 1), \quad i = 6, \ldots, 10 \\ \mathbf{x}_i &= Z_3 + \epsilon_i^x, \; Z_3 \sim \mathcal{N}(0, 1), \quad i = 11, \ldots, 15 \\ \mathbf{x}_i &\sim \mathcal{N}(0, 1) \quad i = 16, \ldots, 40 \end{aligned}$$

with $\epsilon_i^x \sim \mathcal{N}(0, 0.16)$, and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_{40}]$.

3. Consider a regulatory gene network (Li and Li, 2008), where an entire network consists of n_{TF} subnetworks, each with one transcription factor (TF) and its 10 regulatory target genes. The data for each subnetwork can be generated as $\mathbf{X}_i^{TF} \sim \mathcal{N}(0, \mathbf{S}_{11\times 11})$ with $s_{ii} = 1, s_{1i} = s_{i1} = 0.7, \forall i, i \neq 1$ and $s_{ij} = 0$ for $i \neq j, j \neq 1, i \neq 1$. Then $\mathbf{X} = [\mathbf{X}_1^{TF}, \ldots, \mathbf{X}_{n_{TF}}^{TF}]$, $n = 100, \; p = 110$, and $\sigma = 5$. The true parameters are

$$\beta = (\underbrace{\frac{5}{\sqrt{11}}, \ldots, \frac{5}{\sqrt{11}}}_{11}, \underbrace{\frac{-3}{\sqrt{11}}, \ldots, \frac{-3}{\sqrt{11}}}_{11}, \underbrace{0, \ldots, 0}_{p-22})^T.$$

4. Same as 3 except that

$$\beta = (5, \underbrace{\frac{5}{\sqrt{10}}, \dots, \frac{5}{\sqrt{10}}}_{10}, -3, \underbrace{\frac{-3}{\sqrt{10}}, \dots, \frac{-3}{\sqrt{10}}}_{10}, \underbrace{0, \dots, 0}_{p-22})^T$$

5. Same as 3 except that

$$\beta = (5, \underbrace{\frac{5}{\sqrt{10}}, \dots, \frac{5}{\sqrt{10}}}_{10}, -5, \underbrace{\frac{-5}{\sqrt{10}}, \dots, \frac{-5}{\sqrt{10}}}_{10},$$
$$3, \underbrace{\frac{3}{\sqrt{10}}, \dots, \frac{3}{\sqrt{10}}}_{10}, -3, \underbrace{\frac{-3}{\sqrt{10}}, \dots, \frac{-3}{\sqrt{10}}}_{10} \underbrace{0, \dots, 0}_{p-44})^T$$

We assume that the features in the same group are connected in a graph, and those in different groups are not connected. We use MSE to measure the performance of estimation of β, which is defined as

$$MSE(\beta) = (\beta - \beta^*)^T \mathbf{X}^T \mathbf{X} (\beta - \beta^*).$$

For feature grouping and selection, we introduce two separate metrics to measure the accuracy of feature grouping and selection. Denote $I_i, i = 0, 1, 2, ..., K$ as the index of different groups, where I_0 is the index of zero coefficients. Then the metric for feature selection is defined as

$$s_0 = \frac{\sum_{i \in I_0} \mathbb{I}(\beta_i = 0) + \sum_{i \notin I_0} \mathbb{I}(\beta_i \neq 0)}{p},$$

and the metric for feature grouping is defined as

$$s = \frac{\sum_{i=1}^{K} s_i + s_0}{K+1},$$

where

$$s_i = \frac{\sum_{i \neq j, i, j \in I_i} \mathbb{I}(|\beta_i| = |\beta_j|) + \sum_{i \neq j, i \in I_i, j \notin I_i} \mathbb{I}(|\beta_i| \neq |\beta_j|)}{|I_i|(p-1)}.$$

s_i measures the grouping accuracy of group i under the assumption that the absolute values of entries in the same group should be the same, but different from those in different groups. s_0 measures the accuracy of feature selection. It is clear that $0 \leq s_0, s_i, s \leq 1$.

For each dataset, we generate n samples for training, as well as n samples for testing. To make the synthetic datasets more challenging, we first randomly select $\lfloor n/2 \rfloor$ coefficients, and change their signs, as well as those of the corresponding features. Denote $\tilde{\beta}$ and $\tilde{\mathbf{X}}$ as the coefficients and features after changing signs. Then $\tilde{\beta}_i = -\beta_i$, $\tilde{\mathbf{x}}_i = -\mathbf{x}_i$, if the i-th coefficient is selected; otherwise, $\tilde{\beta}_i = \beta_i$, $\tilde{\mathbf{x}}_i = \mathbf{x}_i$. So that $\tilde{\mathbf{X}}\tilde{\beta} = \mathbf{X}\beta$. We apply different approaches on $\tilde{\mathbf{X}}$. The covariance matrix of $\mathbf{X}$ is used in GFlasso to simulate the graph misspecification. The results of β converted from $\tilde{\beta}$ are reported.

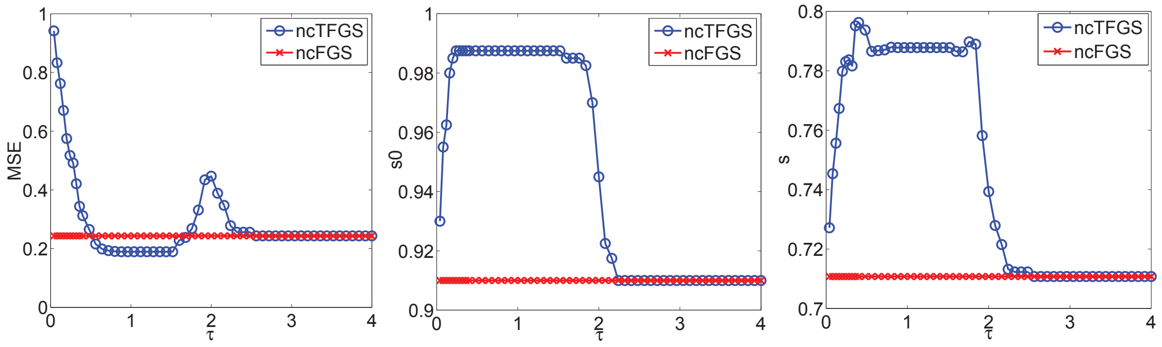


Figure 2.3: MSEs (left), s_0 (middle), and s (right) of ncFGS and ncTFGS on dataset 1 for fixed λ_1 and λ_2. The regularization parameters for ncTFGS are $\tau\lambda_1$ and $\tau\lambda_2$. τ ranges from 0.04 to 4.

Figure 2.3 shows that ncFGS obtains the same results as ncTFGS on dataset 1 with $\sigma = 2$ when τ is larger than $|\beta_i|$. The regularization parameters are $\tau\lambda_1$ and $\tau\lambda_2$ for ncTFGS, and λ_1 and λ_2 for ncFGS. Figure 2.4 shows the average nonzero coefficients obtained on dataset 1 with $\sigma = 2$. As can be seen in Figure 2.4, GOSCAR, ncFGS, and ncTFGS are able to utilize the graph information, and achieve good parameter estimation. Although GFlasso can use the graph information, it performs worse than GOSCAR, ncFGS, and ncTFGS due to the graph misspecification.

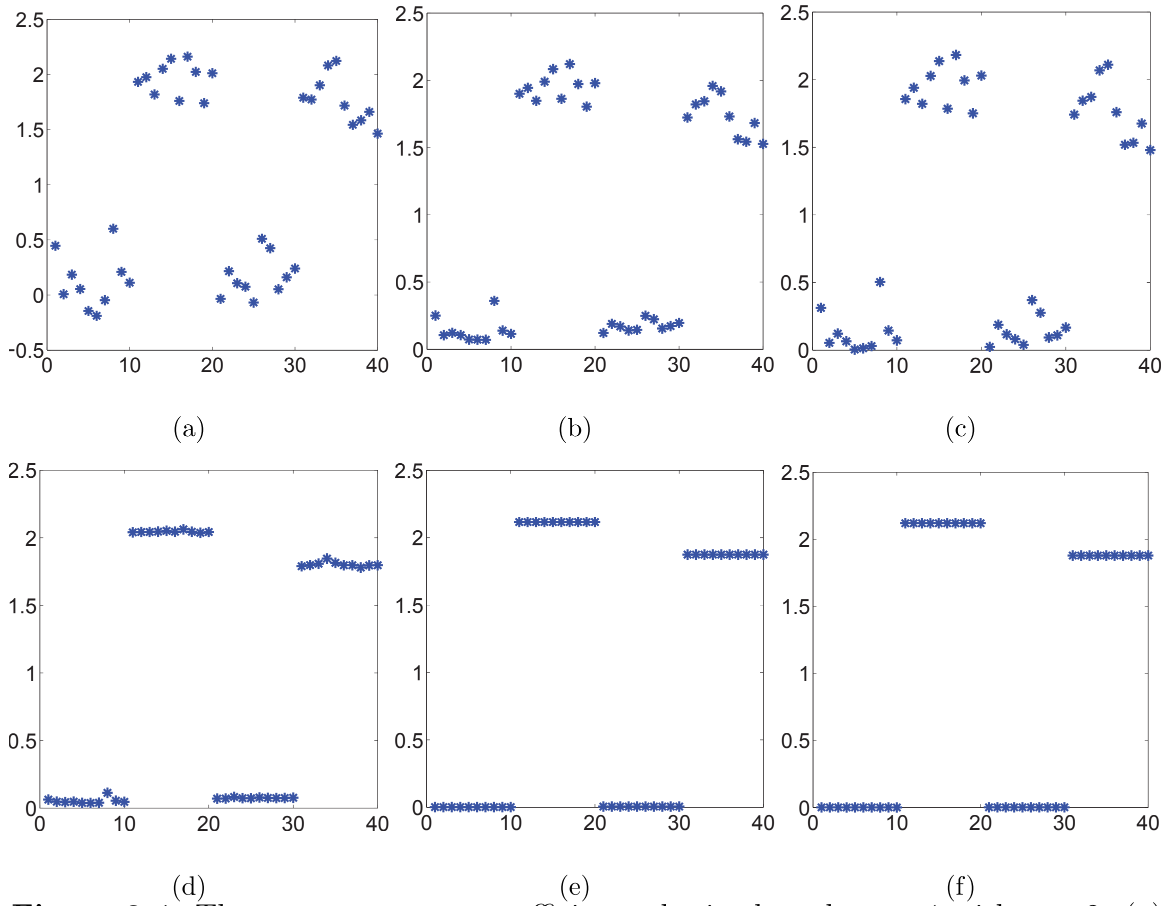


Figure 2.4: The average nonzero coefficients obtained on dataset 1 with $\sigma = 2$: (a) Lasso; (b) GFlasso; (c) OSCAR; (d) GOSCAR; (e); ncFGS; (f) ncTGS

The performance in terms of MSEs averaged over 30 simulations are shown in Table 3.1. As indicated in Table 3.1, among existing methods (Lasso, GFlasso, OSCAR), GFlasso is the best, except in the two cases where OSCAR is better. GOSCAR is better than the best existing method in all cases except for two, and ncFGS and ncTFGS outperform all the other methods.

Table 2.3 shows the results in terms of accuracy of feature grouping and selection. Since Lasso does not perform feature grouping, we only report the results of the other five methods: OSCAR, GFlasso, GOSCAR, ncFGS, and ncTFGS. Table 2.3 shows that ncFGS and ncTFGS achieve higher accuracy than other methods.

Table 2.1 shows the comparison of feature selection alone ($\lambda_2 = 0$), feature grouping alone ($\lambda_1 = 0$), and simultaneous feature grouping and selection using ncTFGS. From Table 2.1, we can observe that simultaneous feature grouping and selection outperforms either feature grouping or feature selection, demonstrating the benefit of joint feature grouping and selection in the proposed non-convex method.

Table 2.1: Comparison of feature selection alone (FS), feature grouping alone (FG), and simultaneous feature grouping and feature selection (Both). The average results based on 30 replications of three datasets with $\sigma = 5$: Data3 (top), Data4 (middle), and Data5 (bottom) are reported. The numbers in parentheses are the standard deviations.

Meth.	MSE	s_0	s
FG	2.774(0.967)	0.252(0.156)	0.696(0.006)
FS	6.005(1.410)	0.945(0.012)	0.773(0.037)
Both	**0.348(0.283)**	**0.996(0.014)**	**0.978(0.028)**
FG	9.4930(1.810)	0.613(0.115)	0.770(0.038)
FS	6.437(1.803)	0.947(0.016)	0.782(0.046)
Both	**4.944(0.764)**	**0.951(0.166)**	**0.890(0.074)**
FG	10.830(2.161)	0.434(0.043)	0.847(0.014)
FS	10.276(1.438)	0.891(0.018)	0.768(0.026)
Both	**7.601(1.038)**	**0.894(0.132)**	**0.919(0.057)**

Table 2.2: Comparison of performance in terms of MSEs and estimated standard deviations (in parentheses) for different methods based on 30 simulations on different synthetic datasets.

Datasets	Lasso	OSCAR	GFlasso	GOSCAR	ncFGS	ncTFGS
Data1 ($\sigma = 2$)	1.807(0.331)	1.441(0.318)	1.080(0.276)	0.315(0.157)	0.123(0.075)	**0.116(0.075)**
Data1 ($\sigma = 5$)	5.294(0.983)	5.328(1.080)	3.480(1.072)	1.262(0.764)	0.356(0.395)	**0.356(0.395)**
Data1 ($\sigma = 10$)	12.628(3.931)	13.880(4.031)	13.411(4.540)	6.061(4.022)	1.963(1.600)	**1.958(1.593)**
Data2 ($\sigma = 2$)	1.308(0.435)	1.084(0.439)	0.623(0.250)	0.291(0.208)	0.226(0.175)	**0.223(0.135)**
Data2 ($\sigma = 5$)	4.907(1.496)	4.868(1.625)	2.538(0.656)	0.781(0.598)	0.721(0.532)	**0.705(0.535)**
Data2 ($\sigma = 10$)	18.175(6.611)	18.353(6.611)	6.930(2.858)	4.601(2.623)	4.232(2.561)	**4.196(2.577)**
Data3 ($\sigma = 5$)	5.163(1.708)	4.503(1.677)	4.236(1.476)	3.336(1.725)	0.349(0.282)	**0.348(0.283)**
Data4 ($\sigma = 5$)	7.664(2.502)	7.167(2.492)	7.516(2.441)	7.527(2.434)	5.097(0.780)	**4.943(0.764)**
Data5 ($\sigma = 5$)	9.893(1.965)	7.907(2.194)	9.622(2.025)	9.810(2.068)	7.684(1.1191)	**7.601(1.038)**

Table 2.3: Accuracy of feature grouping and selection based on 30 simulations for five feature grouping methods: the first row for each dataset corresponds to the accuracy of feature selection; the second row corresponds to the accuracy of feature grouping. The numbers in parentheses are the standard deviations.

Datasets	OSCAR	GFlasso	GOSCAR	ncFGS	ncTFGS
Data1 ($\sigma = 2$)	0.675(0.098)	0.553(0.064)	0.513(0.036)	0.983(0.063)	**1.000(0.000)**
	0.708(0.021)	0.709(0.017)	0.702(0.009)	0.994(0.022)	**1.000(0.000)**
Data1 ($\sigma = 5$)	0.565(0.084)	0.502(0.009)	0.585(0.085)	**1.000(0.000)**	**1.000(0.000)**
	0.691(0.011)	0.709(0.016)	0.708(0.017)	**1.000(0.000)**	**1.000(0.000)**
Data1 ($\sigma = 10$)	0.532(0.069)	0.568(0.088)	0.577(0.061)	**0.983(0.063)**	**1.000(0.000)**
	0.675(0.031)	0.725(0.022)	0.708(0.020)	**0.994(0.022)**	**0.999(0.001)**
Data2 ($\sigma = 2$)	0.739(0.108)	0.544(0.272)	**1.000(0.000)**	0.958(0.159)	0.958(0.159)
	0.625(0.052)	0.823(0.029)	0.837(0.014)	0.831(0.052)	**0.846(0.041)**
Data2 ($\sigma = 5$)	0.763(0.114)	0.717(0.275)	**0.999(0.005)**	0.979(0.114)	0.975(0.115)
	0.650(0.066)	0.741(0.062)	0.833(0.011)	**0.845(0.030)**	0.842(0.037)
Data2 ($\sigma = 10$)	0.726(0.101)	0.818(0.149)	0.993(0.024)	**1.000(0.000)**	**1.000(0.000)**
	0.597(0.058)	0.680(0.049)	0.829(0.025)	0.851(0.015)	**0.856(0.014)**
Data3 ($\sigma = 5$)	0.886(0.135)	0.736(0.103)	0.382(0.084)	0.992(0.026)	**0.996(0.014)**
	0.841(0.056)	0.739(0.041)	0.689(0.013)	**0.995(0.017)**	0.978(0.028)
Data4 ($\sigma = 5$)	0.875(0.033)	0.881(0.026)	0.882(0.037)	0.796(0.245)	**0.950(0.012)**
	0.834(0.030)	0.805(0.035)	0.805(0.036)	**0.895(0.114)**	0.890(0.074)
Data5 ($\sigma = 5$)	0.760(0.203)	0.802(0.153)	0.861(0.051)	0.881(0.174)	**0.894(0.132)**
	0.858(0.031)	0.821(0.037)	0.805(0.037)	**0.920(0.056)**	0.919(0.057)

Table 2.4: Comparison of classification accuracy, sensitivity, specificity, degrees of freedom, and the number of nonzero coefficients averaged over 20 replications for different methods on FDG-PET dataset.

Metrics	Lasso	OSCAR	GFlasso	GOSCAR	ncFGS	ncTFGS
acc.	0.886(0.026)	0.891(0.026)	0.901(0.029)	0.909(0.026)	0.909(0.031)	**0.920(0.024)**
sen.	0.870(0.041)	0.876(0.038)	0.904(0.038)	0.909(0.041)	0.915(0.043)	**0.933(0.047)**
pec.	0.913(0.0446)	**0.917(0.050)**	0.902(0.046)	0.915(0.046)	0.908(0.047)	0.915(0.052)
dof.	22.150	29.150	24.350	21.300	31.250	**18.250**
nonzero coeff.	22.150	38.000	41.900	24.250	37.350	**19.350**

Table 2.5: Comparison of classification accuracy, sensitivity, specificity, degrees of freedom, and the number of nonzero coefficients averaged over 30 replications for various methods on Breast Cancer dataset.

Metrics	Lasso	OSCAR	GFlasso	GOSCAR	ncFGS	ncTFGS
acc.	0.739(0.054)	0.755(0.055)	0.771(0.050)	0.783(0.042)	0.779(0.041)	**0.790(0.036)**
sen.	0.707(0.056)	0.720(0.060)	0.749(0.060)	0.755(0.050)	0.755(0.055)	**0.762(0.044)**
pec.	0.794(0.071)	0.810(0.068)	0.805(0.056)	0.827(0.061)	0.819(0.058)	**0.834(0.060)**
dof.	239.267	165.633	108.633	70.267	57.233	**45.600**
nonzero coeff.	239.267	243.867	144.867	140.667	**79.833**	116.567

2.4.3 Real Data

We conduct experiments on two real datasets: FDG-PET and Breast Cancer. The metrics to measure the performance of different algorithms include accuracy (acc.), sensitivity (sen.), specificity (spe.), degrees of freedom (dof.), and the number of nonzero coefficients (nonzero coeff.). The dof. of lasso is the number of nonzero coefficients (Tibshirani, 1996). For the algorithms capable of feature grouping, we use the same definition of dof. in (Bondell and Reich, 2008), which is the number of estimated groups.

FDG-PET

In this experiment, we use FDG-PET 3D images from 74 Alzheimer's disease (AD), 172 mild cognitive impairment (MCI), and 81 normal control (NC) subjects downloaded from the Alzheimer's disease neuroimaging initiative (ADNI) database. The different regions of whole brain volume can be represented by 116 anatomical volumes of interest (AVOI), defined by Automated Anatomical Labeling (AAL) (Tzourio-Mazoyer *et al.*, 2002). Then we extracted data from each of the 116 AVOIs, and derived average of each AVOI for each subject.

In our study, we compare different methods in distinguishing AD and NC subjects, which is a two-class classification problem over a dataset with 155 samples and 116 features. The dataset is randomly split into two subset, one training set consisting of 104 samples, and one testing set consisting of the remaining 51 samples. The tuning of the parameter is achieved by 5-fold cross validation. Sparse inverse covariance estimation (SICE) has been recognized as an effective tool for identifying the structure of the inverse covariance matrix. We use SICE developed in (Huang *et al.*, 2009) to model the connectivity of brain regions. Figure 2.5 shows sample subgraphs built by

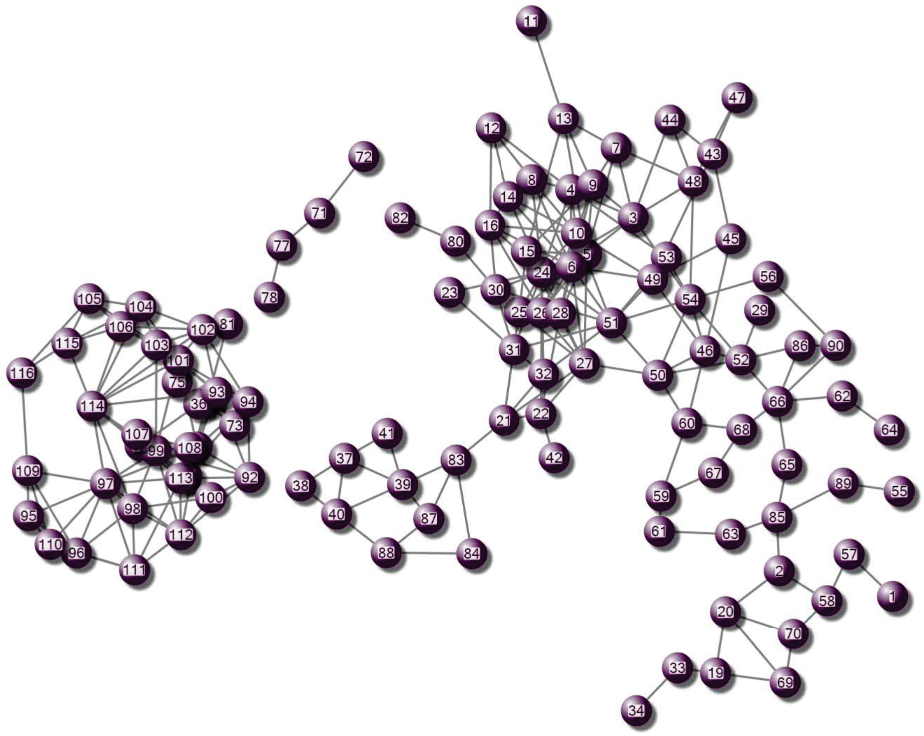
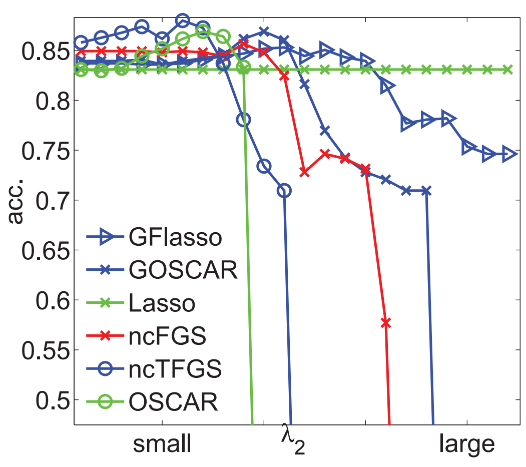


Figure 2.5: Subgraphs of the graph built by SICE on FDG-PET dataset, which consists of 265 edges.

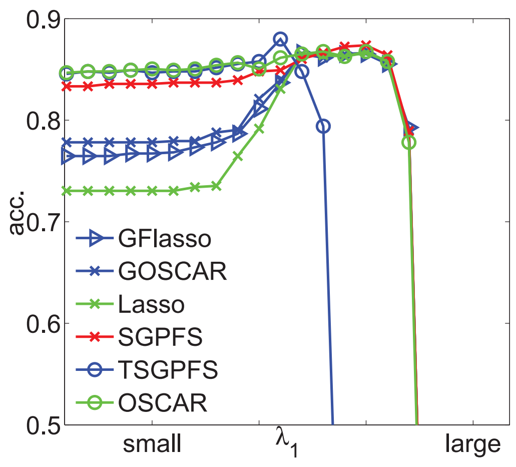
SICE consisting of 115 nodes and 265 edges.

The results based on 20 replications are shown in Table 2.4. From Table 2.4, we can see that ncTFGS achieves more accurate classification while obtaining smaller degrees of freedom. ncFGS and GOSCAR achieve similar classification, while ncFGS selects more features than GOSCAR.

Figure 2.6 shows the comparison of accuracy with either λ_1 or λ_2 fixed. The λ_1 and λ_2 values range from 1e-4 to 100. As we can see, the performance of ncTFGS is slightly better than that of the other competitors. Since the regularization parameters of subproblems in ncTFGS are $\frac{\lambda_1}{\tau}$ and $\frac{2\lambda_2}{\tau}$, the solution of ncTFGS is more sparse than those of other competitors when λ_1 and λ_2 are large and τ is small ($\tau = 0.15$ in this case).



(a) λ_1 fixed



(b) λ_2 fixed

Figure 2.6: Comparison of accuracies for various methods with λ_1 fixed (left) and λ_2 fixed (right) on FDT-PET dataset.

Breast Cancer

We conduct experiments on the breast cancer datasets, which consists of gene expression data for 8141 genes in 295 breast cancer tumors (78 metastatic and 217 non-metastatic). The network described in Chuang *et al.* (2007) is used as the input graph in this experiment. Figure 2.7 shows a subgraph consisting of 80 nodes of the used graph. We restrict our analysis to the 566 genes most correlated to the output, but also connected in the graph. 2/3 data is randomly chosen as training data, and the remaining 1/3 data is used as testing data. The tuning parameter is estimated by 5-fold cross validation. Table 2.5 shows the results averaged over 30 replications. As indicated in Table 2.5, GOSCAR, ncFGS and ncTFGS outperform the other three methods, and ncTFGS achieves the best performance.

2.5 Conclusion

In this chapter, we consider simultaneous feature grouping and selection over a given undirected graph. We propose one convex and two non-convex penalties to

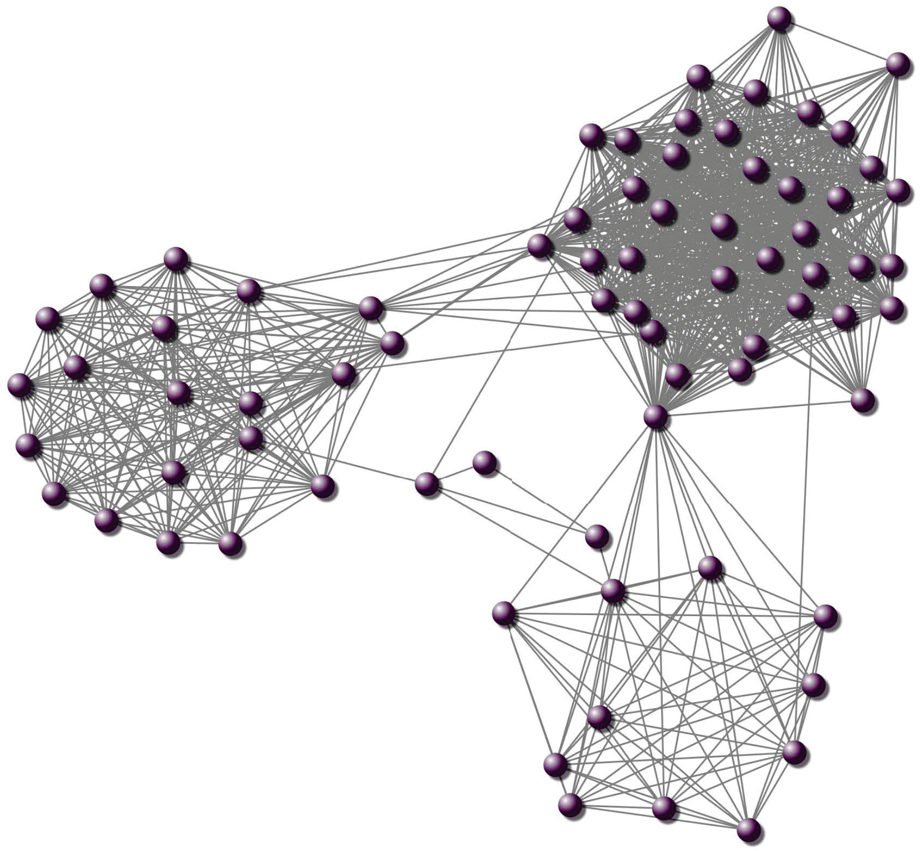


Figure 2.7: A subgraph of the network in Breast Cancer dataset (Chuang *et al.*, 2007). The subgraph consists of 80 nodes.

encourage both sparsity and equality of absolute values of coefficients for features connected in the graph. We employ ADMM and DC programming to solve the proposed formulations. Numerical experiments on synthetic and real data demonstrate the effectiveness of the proposed methods. Our results also demonstrate the benefit of simultaneous feature grouping and feature selection through the proposed non-convex methods. In this chapter, we focus on undirected graphs. A possible future direction is to extend the formulations to directed graphs. In addition, we plan to study the generalization performance of the proposed formulations.

Chapter 3

AN EFFICIENT ADMM ALGORITHM FOR MULTIDIMENSIONAL ANISOTROPIC TOTAL VARIATION REGULARIZATION PROBLEMS

3.1 Introduction

The presence of noise in signals is unavoidable. To recover original signals, many noise reduction techniques have been developed to reduce or remove the noise. Noisy signals usually have high total variation (TV). Several total variation regularization approaches have been developed to exploit the special properties of noisy signals and they have been widely used in noise reduction in signal processing. The total variation model was first introduced by Rudin *et al.* (1992) as a regularization approach to remove noise and handle proper edges in images. More recently, the total variation models have been applied successfully for image reconstruction, e.g. Magnetic Resonance (MR) image reconstruction (Huang *et al.*, 2011; Ma *et al.*, 2008). The wide range of applications including image restoration, image denoising and deblurring (Barbero and Sra, 2011; Beck and Teboulle, 2009a; Huang *et al.*, 2011; Li *et al.*, 2009; Vogel and Oman, 1998; Yang *et al.*, 2009; Wang *et al.*, 2014; Yang *et al.*, 2013b), underscore its success in signal/image processing. The discrete penalized version of the TV-based image denoising model solves an unconstrained convex minimization problem of the following form:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2 + \lambda\|\mathbf{X}\|_{TV}, \tag{3.1}$$

where $\|\cdot\|_F$ is the Frobenius norm defined as $\|\mathbf{X}\|_F = \sqrt{\sum_{i,j} x_{i,j}^2}$, $\mathbf{Y}$ is the observed image, $\mathbf{X}$ is the desired unknown image to be recovered, and $\|\cdot\|_{TV}$ is the discrete TV norm defined below. The nonnegative regularization parameter λ provides a tradeoff between the noise sensitivity and closeness to the observed image. There are two popular choices for the discrete TV norm: ℓ_2-based *isotropic* TV defined by

$$\|\mathbf{X}\|_{TV} = \sum_{i=1}^{m}\sum_{j=1}^{n} \|\nabla x_{i,j}\|_2, \ \mathbf{X} \in \Re^{m\times n},$$

and the ℓ_1-based *anisotropic* TV defined by

$$\|\mathbf{X}\|_{TV} = \sum_{i=1}^{m}\sum_{j=1}^{n} \|\nabla x_{i,j}\|_1, \ \mathbf{X} \in \Re^{m\times n},$$

where ∇ denotes the forward finite difference operators on the vertical and horizonal directions, i.e., $\nabla x_{i,j} = (\nabla_1 x_{i,j},\, \nabla_2 x_{i,j})^T$:

$$\nabla_1 x_{i,j} = \begin{cases} x_{i,j} - x_{i+1,j} & \text{if } 1 \le i < m \\ 0 & \text{if } j = n \end{cases}, \nabla_2 x_{i,j} = \begin{cases} x_{i,j} - x_{i,j+1} & \text{if } 1 \le j < n \\ 0 & \text{if } i = m. \end{cases}$$

Despite the simple form of the TV norm, it is a challenge to solve TV-based regularization problems efficiently. One of the key difficulties in the TV-based image denoising problem is the nonsmoothness of the TV norm. Continued research efforts have been made to build fast and scalable numerical methods in the last few years. Existing methods aim to balance the tradeoff between the convergence rate and the simplicity of each iterative step. For example, computing the exact optimal solution at each iteration leads to a better convergence rate (Schmidt *et al.*, 2011). However, this usually requires heavy computations, for instance, a large linear system of equations. Simple methods with less computation efforts at each iteration are more suitable for large-scale problems, but usually they have a slow convergence rate. To this end, we propose a fast but simple ADMM algorithm to solve TV-based problems. The key idea of the proposed method is to decompose the large problem into a set of smaller and independent problems, which can be solved efficiently and exactly. Moreover, these small problems are decoupled, thus they can be solved in parallel. Therefore, the proposed method scales to large-size problems.

Although the TV problems have been extensively studied for matrices (e.g. two-dimensional images), there is not much work on tensors, a higher-dimensional extension of matrices. Tensor data is common in real world applications, for instance, functional magnetic resonance imaging (fMRI) is a 3-mode tensor and a color video is a 4-mode tensor. Another contribution of this chapter is that the proposed ADMM algorithm is designed to solve TV problems for tensors, e.g., multidimensional TV problems. The 2D TV problem can be solved efficiently by a special case of the proposed algorithm (for matrices). Our experiments show that the proposed method is more efficient than state-of-the-art approaches for solving 2D TV problems. We further demonstrate the efficiency of the proposed method for multidimensional TV problems in image reconstruction, video denoising and image deblurring.

3.1.1 Related Work

Due to the nonsmoothness of the TV norm, solving large-scale TV problems efficiently continues to be a challenging issue despite its simple form. In the past, considerable efforts have been devoted to develop an efficient and scalable algorithm for TV problems. The 1D total variation, also known as the fused signal approximator, has been widely used in signal noise reduction. Liu *et al.* (2010) propose an efficient method to solve the fused signal approximator using a warm start technique. It has been shown to be very efficient in practice, though the convergence rate has not been established. Barbero and Sra (2011) introduce a fast Newton-type method for 1D total variation regularization, and solve the 2D total variation problem using the Dyktra's method (Combettes and Pesquet, 2011). Wahlberg *et al.* (2012) propose an ADMM method to solve the 1D total variation problem. A linear system of equations has to be solved at each iteration. Recently, a very fast direct, noniterative, algorithm for 1D total variation problem has been proposed in (Condat, 2013). A dual-based approach to solve the 2D total variation problems is introduced in (Chambolle, 2004). Beck and Teboulle (2009a) propose a fast gradient-based method by combining the dual-based approach with the acceleration technique in Fast Iterative Shrinkage Thresholding Algorithm (FISTA) (Beck and Teboulle, 2009b). One potential drawback of the dual-based approaches is that it may not scale well. Goldstein and Osher introduce the split Bregman method to solve the 2D total variation problem, which is an application of split Bregman method solving ℓ_1 based problems. The total variation has also been widely used in Magnetic Resonance (MR) image reconstruction (Huang *et al.*, 2011; Ma *et al.*, 2008). Ma *et al.* (2008) introduce an operator-splitting algorithm (TVCMRI) to solve the MR image reconstruction problem. By combining the composite splitting algorithm (Combettes and Pesquet, 2011) and the acceleration technique in FISTA, Huang *et al.* (2011) propose an efficient MR image reconstruction algorithm called FCSA. We show that our proposed method is much more efficient than these methods for solving 2D TV problems.

The rest of this chapter is organized as follows. We present the multidimensional total variation regularization problems and the proposed ADMM method in Section 3.2. One of the key steps in the proposed algorithm involves the solution of a 1D TV problem; we show how to estimate the active regularization parameter range for 1D TV problem in Section 3.3. We report empirical results in Section 3.4, and conclude this chapter in Section 3.5.

3.2 The Proposed Algorithm for Multidimensional TV Problems

We first introduce the multidimensional total variation regularization problems in Section 3.2.1. In Section 3.2.2, we present the details of the proposed algorithm.

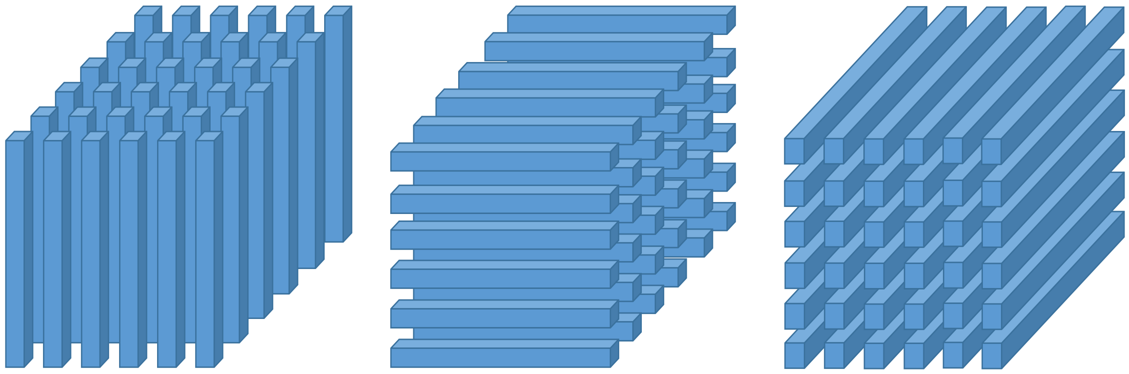


Figure 3.1: Fibers of a 3-mode tensor: mode-1 fibers $\mathbf{x}_{:,j_2,j_3}$, mode-2 fibers $\mathbf{x}_{j_1,:,j_3}$, and mode-3 fibers $\mathbf{x}_{j_1,j_2,:}$(left to right).

The global convergence is established in Section 3.2.3. Section 3.2.4 presents the time complexity of the proposed algorithm.

3.2.1 The Multidimensional TV Problem

Denote $\mathfrak{F}_i(\mathcal{X})$ as the fused operator along the i-th mode of $\mathcal{X}$ taking the form of

$$\mathfrak{F}_i(\mathcal{X}) = \sum_{/\{j_i\}} \sum_{j_i=1}^{I_i-1} |x_{j_1,\ldots,j_i,\ldots,j_d} - x_{j_1,\ldots,(j_i+1),\ldots,j_d}|.$$

It is not hard to see that $\mathfrak{F}_i(\mathcal{X})$ is decomposable with respect to mode-i fibers, which are the higher order analogue of matrix rows and columns (see Figure 3.1 for an illustration). In the case of matrix, $\mathfrak{F}_i(\mathbf{X})$ only involves the rows or columns of X. For example, $\mathfrak{F}_1(\mathbf{X}) = \sum_{j=1}^{n} \sum_{i=1}^{m-1} |x_{i,j} - x_{i+1,j}|, \mathbf{X} \in \Re^{m\times n}$. It is clear that the ℓ_1-based *anisotropic* TV norm for matrices can be rewritten as $\sum_{i=1}^{2} \mathfrak{F}_i(\mathbf{X})$. The tensor is the generalization of the matrix concept. We generalize the TV norm for the matrix case to higher-order tensors by the following tensor TV norm:

$$\|\mathcal{X}\|_{TV} = \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{X}).$$

Based on the definition above, the TV-based denoising problem for the matrix case can be generalized to tensors by solving the following optimization problem:

$$\min_{\mathcal{X}} \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\|_F^2 + \lambda \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{X}), \tag{3.2}$$

where $\mathcal{Y} \in \Re^{I_1 \times I_2 \times \ldots, \times I_d}$ is the observed data represented as a tensor, $\mathcal{X} \in \Re^{I_1 \times I_2 \times \ldots, \times I_d}$ is the unknown tensor to be estimated, $\sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{X})$ is the tensor TV norm, and λ is a nonnegative regularization parameter. The tensor TV regularization encourages $\mathcal{X}$ to be smooth along all dimensions.

3.2.2 The Proposed Algorithm

We propose to solve the multidimensional TV problem (MTV) using ADMM (Boyd *et al.*, 2011). ADMM decomposes a large global problem into a series of smaller local subproblems, and coordinates the local solutions to compute the globally optimal solution. ADMM attempts to combine the benefits of augmented Lagrangian methods and dual decomposition for constrained optimization problems (Boyd *et al.*, 2011). The problem solved by ADMM takes the following form:

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \\ &s.t. \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, \end{aligned} \tag{3.3}$$

where x, z are unknown variables to be estimated.

ADMM reformulates the problem using a variant of the augmented Lagrangian method as follows:

$$\mathfrak{L}_\rho(\mathbf{x}, \mathbf{z}, \mu) = f(\mathbf{x}) + g(\mathbf{z}) + \mu^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|^2$$

with μ being the augmented Lagrangian multiplier, and ρ being the nonnegative penalty parameter (or dual update length). ADMM solves the original constrained problem by iteratively minimizing $\mathfrak{L}_\rho(\mathbf{x}, \mathbf{z}, \mu)$ over $\mathbf{x}$, $\mathbf{z}$, and updating μ according to the following update rule:

$$\begin{aligned}
\mathbf{x}^{k+1} =& \text{argmin}_{\mathbf{x}} \mathfrak{L}_\rho(\mathbf{x}, \mathbf{z}^k, \mu^k) \\
\mathbf{z}^{k+1} =& \text{argmin}_{\mathbf{z}} \mathfrak{L}_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \mu^k) \\
\mu^{k+1} =& \mu^k + \rho(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}).
\end{aligned}$$

Consider the unconstrained optimization problem in (3.2), which can be reformulated as the following constrained optimization problem:

$$\begin{aligned}
&\min_{\mathcal{X}, \mathcal{Z}_i} \frac{1}{2} \|\mathcal{Y} - \mathcal{X}\|_F^2 + \lambda \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{Z}_i) \\
&s.t. \ \ \mathcal{X} = \mathcal{Z}_i, \text{for } 1 \leq i \leq d,
\end{aligned} \tag{3.4}$$

where $\mathcal{Z}_i, 1 \leq i \leq d$ are slack variables. The optimization problem in (3.4) can be solved by ADMM. The augmented Lagrangian of (3.4) is given by

$$\begin{aligned}
\mathcal{L}(\mathcal{X}, \mathcal{Z}_i, \mathcal{U}_i) =& \frac{1}{2} \|\mathcal{Y} - \mathcal{X}\|_F^2 + \lambda \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{Z}_i) + \\
& \sum_{i=1}^{d} \langle \mathcal{U}_i, \mathcal{Z}_i - \mathcal{X} \rangle + \frac{\rho}{2} \sum_{i=1}^{d} \|\mathcal{Z}_i - \mathcal{X}\|_F^2.
\end{aligned} \tag{3.5}$$

Applying ADMM, we carry out the following steps at each iteration:

Step 1 Update $\mathcal{X}^{k+1}$ with $\mathcal{Z}_i^k$ and $\mathcal{U}_i^k$ fixed:

$$\mathcal{X}^{k+1} = \text{argmin}_{\mathcal{X}} \frac{1}{2} \|\mathcal{Y} - \mathcal{X}\|_F^2 - \sum_{i=1}^{d} \langle \mathcal{U}_i^k, \mathcal{X} \rangle + \frac{\rho}{2} \sum_{i=1}^{d} \|\mathcal{Z}_i^k - \mathcal{X}\|_F^2. \tag{3.6}$$

The optimal solution is given by

$$\mathcal{X}^{k+1} = \frac{\mathcal{Y} + \sum_{i=1}^{d} (\mathcal{U}_i^k + \rho \mathcal{Z}_i^k)}{1 + d\rho}. \tag{3.7}$$

Step 2 Compute $\mathcal{Z}_i^{k+1}, i = 1, \cdots, d$ with $\mathcal{X}^{k+1}$, and $\mathcal{U}_i^k, i = 1, \cdots, d$ fixed:

$$\{\mathcal{Z}_i^{k+1}\} = \text{argmin}_{\{\mathcal{Z}_i\}} \frac{\rho}{2} \sum_{i=1}^{d} \|\mathcal{Z}_i - \mathcal{X}^{k+1}\|_F^2 + \sum_{i=1}^{d} \langle \mathcal{U}_i^k, \mathcal{Z}_i \rangle + \lambda \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{Z}_i), \tag{3.8}$$

where $\{\mathcal{Z}_i\}$ denotes the set $\{\mathcal{Z}_1, \ldots, \mathcal{Z}_d\}$. This problem is decomposable, i.e., we can solve $\mathcal{Z}_i^{k+1}, 1 \leq i \leq d$ separately,

$$\mathcal{Z}_i^{k+1} = \mathrm{argmin}_{\mathcal{Z}_i} \frac{\rho}{2} \|\mathcal{Z}_i - \mathcal{X}^{k+1}\|_F^2 + \langle \mathcal{U}_i^k, \mathcal{Z}_i \rangle + \lambda \mathfrak{F}_i(\mathcal{Z}_i),$$

which can be equivalently written as

$$\mathcal{Z}_i^{k+1} = \mathrm{argmin}_{\mathcal{Z}_i} \frac{1}{2} \|\mathcal{Z}_i - \mathcal{T}_i\|_F^2 + \frac{\lambda}{\rho} \mathfrak{F}_i(\mathcal{Z}_i) \tag{3.9}$$

with $\mathcal{T}_i = -\frac{1}{\rho}\mathcal{U}_i^k + \mathcal{X}^{k+1}$. The problem in (3.9) is decomposable for different mode-i fibers. Denote $\mathbf{z}_{j_1,\ldots,j_{i-1},:,j_{i+1},\ldots,j_d}$ as a mode-i fiber to be estimated, which is a vector of I_i length. For simplicity, we use $\mathbf{v}$ to represent the vector $\mathbf{z}_{j_1,\ldots,j_{i-1},:,j_{i+1},\ldots,j_d}$. Then, (3.9) can be decomposed into a set of independent and much smaller problems:

$$\begin{gathered} \mathbf{v}^{k+1} = \mathrm{argmin}_{\mathbf{v}} \frac{1}{2} \|\mathbf{v} - \mathbf{t}\|^2 + \frac{\lambda}{\rho} \sum_{i=1}^{I_i - 1} |v_i - v_{i+1}|, \\ \forall j_1, \ldots, j_{i-1}, j_{i+1}, \ldots, j_d, \end{gathered} \tag{3.10}$$

where $\mathbf{t}$ is the corresponding mode-i fiber of $\mathcal{T}_i$. (3.10) is the formulation of 1D total variation regularization problem, which can be solved exactly and very efficiently (Condat, 2013; Liu *et al.*, 2010).

The problem of computing $\mathcal{Z}_i^{k+1}, 1 \leq i \leq d$ in (3.8) is therefore decomposed into a set of much smaller problems of computing fibers. Each fiber problem is independent, enabling that the whole set of problems can be computed in parallel.

Step 3 Update $\mathcal{U}_i^{k+1}, i = 1, \ldots, d$:

$$\mathcal{U}_i^{k+1} = \mathcal{U}_i^k + \rho(\mathcal{Z}_i^{k+1} - \mathcal{X}^{k+1}). \tag{3.11}$$

A summary of the proposed method is shown in Algorithm 4 below.

The algorithm stops when the primal and dual residuals (Boyd *et al.*, 2011) satisfy a certain stopping criterion. The stopping criterion can be specified by two thresholds:

Algorithm 4: The proposed ADMM algorithm for multi-dimensional total variation

```
Input: $\mathcal{Y}, \lambda, \rho$
Output: $\mathcal{X}$
Initialization: $\mathcal{Z}_i^0 = \mathcal{X}^0 \leftarrow \mathcal{Y}, \mathcal{U}_i^0 \leftarrow 0$;
do
    Compute $\mathcal{X}^{k+1}$ according to (3.7).
    Compute $\mathcal{Z}_i^{k+1}, i = 1, \ldots, d$ according to (3.9).
    Compute $\mathcal{U}_i^{k+1}, i = 1, \ldots, d$ according to (3.11).
Until Convergence;
return $\mathcal{X}$;
```

absolute tolerance ϵ_{abs} and relative tolerance ϵ_{rel} (see Boyd *et al.* (2011) for more details). The penalty parameter ρ affects the primal and dual residuals, hence affects the termination of the algorithm. A large ρ tends to produce small primal residuals, but increases the dual residuals (Boyd *et al.*, 2011). A fixed ρ (say 10) is commonly used. But there are some schemes of varying the penalty parameter to achieve better convergence. We refer interested readers to Boyd *et al.* (2011) for more details.

Remark 1. *We can add the ℓ_1 regularization in the formulation of multidimensional TV problems for a sparse solution. The subproblem with ℓ_1 regularization is called the fused signal approximator. The optimal solution can be obtained by first solving 1D total variation problem, then applying soft-thresholding (Friedman* et al.*, 2007; Liu* et al.*, 2010).*

3.2.3 Convergence Analysis

The convergence of ADMM to solve the standard form (3.3) has been extensively studied (Boyd *et al.*, 2011; Eckstein and Bertsekas, 1992; He and Yuan, 2012). We establish the convergence of Algorithm 4 by transforming the MTV problem in (3.4) into a standard form (3.3), and show that the transformed optimization problem satisfies the condition needed to establish the convergence.

Denote $\mathbf{x}$ as the vectorization of $\mathcal{X}$, i.e., $\mathbf{x} = \text{vec}(\mathcal{X}) \in \Re^{\prod_i I_i \times 1}$, $\mathbf{y} = \text{vec}(\mathcal{Y}) \in \Re^{\prod_i I_i \times 1}$, $\mathbf{z} = [\text{vec}(\mathcal{Z}_1)^T, \ldots, \text{vec}(\mathcal{Z}_d)^T\ ]^T \in \Re^{d \prod_i I_i \times 1}$, $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2$, and $g(\mathbf{z}) = \lambda \sum_{i=1}^{d} \mathfrak{F}_i(\mathcal{Z}_i)$. Then the MTV problem in (3.4) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}), \\ &s.t. \quad \mathbf{A}\mathbf{x} - \mathbf{z} = \mathbf{0}, \end{aligned} \tag{3.12}$$

where $\mathbf{A} = [\mathbf{I}, \ldots, \mathbf{I}]^T \in \Re^{d \prod_i I_i \times \prod_i I_i}$, and $\mathbf{I}$ is the identity matrix of size $\prod_i I_i \times \prod_i I_i$. The first and second steps of Algorithm 4 are exactly the steps of updating $\mathbf{x}$ and $\mathbf{z}$ in the standard form. Since f, g are proper, closed, and convex, and $\mathbf{A}$ is of column full rank, the convergence of Algorithm 4 directly follows from the results in (Boyd *et al.*, 2011; Eckstein and Bertsekas, 1992; He and Yuan, 2012). Moreover, an $O(1/k)$ convergence rate of Algorithm 4 can be established following the conclusion in (He and Yuan, 2012).

3.2.4 Time Complexity Analysis

The first step of Algorithm 4 involves computations of $\mathcal{X}_i^{k+1}, i = 1, \ldots, d$. Computing $\mathcal{X}_i^{k+1}$ needs to compute $\prod_{j \neq i} I_j$ mode-i fibers of I_i length by the 1D total variation algorithm. The complexity of solving the 1D total variation is $O(I_i)$, but $O(I_i^2)$ in the worst case (Condat, 2013). However, we observe that the empirical complexity is $O(I_i)$ in our experiments (see Figure 3.2). Thus, the complexity of the first step is $O(d \prod_j I_j)$. The time complexity of the second and third steps are $O(\prod_j I_j)$. Hence, the complexity of each iteration is $O(d \prod_j I_j)$. The number of iterations in Algorithm 4 to obtain an ϵ-optimal solution is $O(1/\epsilon)$ (He and Yuan, 2012). Thus, the total complexity of Algorithm 4 is $O(d\prod_j I_j/\epsilon)$ for achieving an ϵ-optimal solution. Since each step of Algorithm 4 can be solved in parallel, the complexity of the parallel version of Algorithm 4 is $O(d\prod_j I_j/n_p\epsilon)$ with n_p processors.

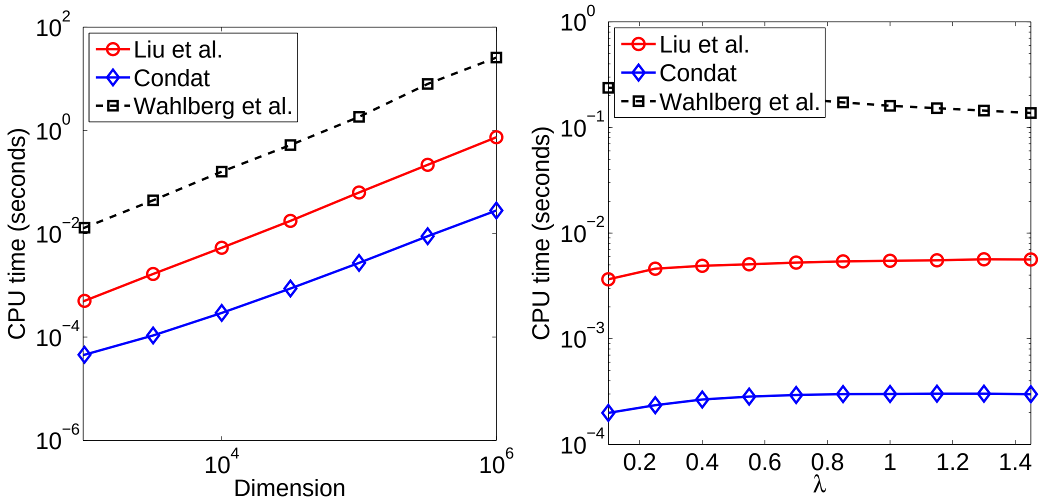


Figure 3.2: Computational time (seconds) of three efficient 1D total variation algorithms: Liu *et al.* (2010), Condat (2013), and Wahlberg *et al.* (2012). Left: dimension varies from 10^3 to 10^6 with $\lambda = 1$. Right: λ varies from 0.15 to 1.45 with dimension 10^4. The data is sampled from standard normal distribution.

3.3 Active Regularization Range for 1D Total Variation

The most time-consuming part of the proposed ADMM algorithm is the first step, which involves the computation of $\mathcal{X}_i^{k+1}, 1 \leq i \leq d$. We decompose the problem of computing $\mathcal{X}_i^{k+1}, 1 \leq i \leq d$ into a set of small 1D total variation problems. Thus, the computation of the proposed method highly depends on that of 1D total variation. In this section, we show how to estimate the active regularization range for 1D total variation, which only relies on the regularization parameter and the observed vector, to directly compute the optimal solution. More specifically, we compute λ_{min} and λ_{max} based on the observed vector; if $\lambda \notin (\lambda_{min}, \lambda_{max})$, the optimal solution can be computed in a closed form, thus significantly improving the efficiency.

Consider the formulation of 1D total variation, i.e.,

$$\inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda \sum_{i=1}^{n-1} |x_i - x_{i+1}|,$$

which can be rewritten as

$$\inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda \|\mathbf{G}\mathbf{x}\|_1 \tag{3.13}$$

in which $\mathbf{y}, \mathbf{x} \in \Re^n$. $\mathbf{G} \in \Re^{(n-1)\times n}$ encodes the structure of the 1D TV norm. We have

$$g_{i,j} = \begin{cases} 1 & \text{if } j = i+1 \\ -1 & \text{if } j = i \\ 0 & \text{otherwise.} \end{cases} \tag{3.14}$$

3.3.1 The Dual Problem

Before we derive the dual formualtion of problem in (3.13) (Boyd and Vandenberghe, 2004; Dhara and Dutta, 2012), we first introduce some useful definitions and lemmas.

Definition 1. (Coercivity). *(Dhara and Dutta, 2012) A function $\phi : \Re^n \to \bar{\Re}$ is said to be coercive over a set $\mathcal{S} \subset \Re^n$ if for every sequence $\{x_k\} \subset \mathcal{S}$*

$$\lim_{k\to\infty} \phi(x_k) = +\infty \textit{ whenever } \|x_k\| \to +\infty.$$

For $\mathcal{S} = \Re^n$, ϕ is simply called coercive.

Denote the objective function in problem (3.13) as:

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda \|\mathbf{G}\mathbf{x}\|_1. \tag{3.15}$$

It is easy to see that $f(\mathbf{x})$ is coercive. For each $\alpha \in \Re$, we define the α sublevel set of $f(\mathbf{x})$ as $\mathcal{S}_\alpha = \{\mathbf{x} : f(\mathbf{x}) \le \alpha\}$. Then we have the following lemma.

Lemma 1. *For any $\alpha \in \Re$, the sublevel set $\mathcal{S}_\alpha = \{\mathbf{x} : f(\mathbf{x}) \leq \alpha\}$ is bounded.*

Proof. We prove the lemma by contradiction. Suppose there exists an α such that $\mathcal{S}_\alpha$ is unbounded. Then we can find a sequence $\{\mathbf{x}_k\} \subset \mathcal{S}_\alpha$ such that $\lim_{k\to\infty} \|\mathbf{x}_k\| = \infty$. Because $f(\mathbf{x})$ is coercive, we can conclude that $\lim_{k\to\infty} f(\mathbf{x}_k) = +\infty$. However, since $\{\mathbf{x}_k\} \subset \mathcal{S}_\alpha$, we know $f(\mathbf{x}_k) \leq \alpha$ for all k, which leads to a contradiction. Therefore, the proof is complete. $\square$

We derive the dual formulation of problem (3.13) via the Sion's Minimax Theorem (Dhara and Dutta, 2012; Sion, 1958). Let $\mathcal{B} = \{\mathbf{x} : \mathbf{y} - \lambda \mathbf{G}^T\mathbf{s}, \|\mathbf{s}\|_\infty \leq 1\}$ and $\mathbf{x}' = \arg\max_{\mathbf{x}\in\mathcal{B}} f(\mathbf{x})$. Because $\mathcal{B}$ is compact, $\mathbf{x}'$ must exist. Denote $\alpha' = f(\mathbf{x}')$ and $\mathcal{S}' = \mathcal{S}_{\alpha'}$.

$$\begin{aligned}\inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\|\mathbf{G}\mathbf{x}\|_1 &= \inf_{\mathbf{x}\in\mathcal{S}'} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\|\mathbf{G}\mathbf{x}\|_1 \\ &= \inf_{\mathbf{x}\in\mathcal{S}'} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda \sup_{\|\mathbf{s}\|_\infty\leq 1} \langle \mathbf{s}, \mathbf{G}\mathbf{x}\rangle \\ &= \inf_{\mathbf{x}\in\mathcal{S}'} \sup_{\|\mathbf{s}\|_\infty\leq 1} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, \mathbf{G}\mathbf{x}\rangle.\end{aligned} \tag{3.16}$$

By Lemma 1, we know that $\mathcal{S}'$ is compact. Moreover, the function

$$\frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, \mathbf{G}\mathbf{x}\rangle$$

is convex and concave with respect to $\mathbf{x}$ and $\mathbf{s}$ respectively. Thus, by the Sion's Minimax Theorem (Sion, 1958), we have

$$\begin{aligned}&\inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\|\mathbf{G}\mathbf{x}\|_1 \qquad (3.17)\\ =& \inf_{\mathbf{x}\in\mathcal{S}'} \sup_{\|\mathbf{s}\|_\infty\leq 1} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, \mathbf{G}\mathbf{x}\rangle \\ =& \sup_{\|\mathbf{s}\|_\infty\leq 1} \inf_{\mathbf{x}\in\mathcal{S}'} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, \mathbf{G}\mathbf{x}\rangle.\end{aligned}$$

We can see that

$$\mathbf{x}^*(\mathbf{s}) = \mathbf{y} - \lambda\mathbf{G}^T\mathbf{s} = \text{argmin}_{\mathbf{x}\in\mathcal{S}'} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, \mathbf{G}\mathbf{x}\rangle \tag{3.18}$$

and thus

$$\begin{aligned}\inf_{\mathbf{x}\in\mathcal{S}'} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\langle \mathbf{s}, \mathbf{G}\mathbf{x}\rangle &= -\frac{\lambda^2}{2}\|\mathbf{G}^T\mathbf{s}\|^2 + \lambda\langle \mathbf{s}, \mathbf{G}\mathbf{y}\rangle \\ &= \frac{1}{2}\|\mathbf{y}\|^2 - \frac{\lambda^2}{2}\|\frac{\mathbf{y}}{\lambda} - \mathbf{G}^T\mathbf{s}\|^2.\end{aligned}$$

Therefore the primal problem (3.16) is transformed to its dual problem:

$$\sup_{\|\mathbf{s}\|_\infty\le 1} \frac{1}{2}\|\mathbf{y}\|^2 - \frac{\lambda^2}{2}\|\frac{\mathbf{y}}{\lambda} - \mathbf{G}^T\mathbf{s}\|^2, \tag{3.19}$$

which is equivalent to

$$\begin{aligned}&\min_{\mathbf{s}} \|\frac{\mathbf{y}}{\lambda} - \mathbf{G}^T\mathbf{s}\|^2 \\ &\text{s.t. } \|\mathbf{s}\|_\infty \le 1.\end{aligned} \tag{3.20}$$

3.3.2 Computing the Maximal Value for λ

Consider 1D total variation problem, the matrix $\mathbf{G} \in \Re^{(n-1)\times n}$ can be written as:

$$\mathbf{G} = \begin{pmatrix} -1 & 1 & \cdots & \cdots & 0 \\ \vdots & -1 & 1 & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & -1 & 1 \end{pmatrix}.$$

Then, it follows that $\mathbf{G}^T$ has full column rank. Denote $e = (1, \cdots, 1)^T \in \Re^n$, and the subspace spanned by the rows of $\mathbf{G}$ and e as $V_{\mathbf{G}^T}$ and V_e. Clearly $\Re^n = V_{\mathbf{G}^T} \oplus V_e$ and $V_{\mathbf{G}^T} \perp V_e$.

Let $\mathbf{P} = I - \frac{ee^T}{\langle e,e\rangle}$ and $\mathbf{P}^\perp$ denote the projection operator into $V_{\mathbf{G}^T}$ and V_e respectively. Therefore the equation

$$\mathbf{P}\mathbf{y} = \mathbf{G}^T\mathbf{s} \tag{3.21}$$

must have a unique solution for each $\mathbf{y}$.

Let $\mathbf{Py} = \widetilde{\mathbf{y}}$, (3.21) can be written as:

$$\begin{pmatrix} -1 & \cdots & \cdots & \cdots \\ 1 & -1 & \cdots & \vdots \\ \vdots & 1 & -1 & \vdots \\ \vdots & & \ddots & -1 \\ 0 & \cdots & \cdots & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n-1} \end{pmatrix} = \begin{pmatrix} \widetilde{y}_1 \\ \widetilde{y}_2 \\ \vdots \\ \widetilde{y}_n \end{pmatrix},$$

then it follows that

$$s_i = -\sum_{j=1}^{i} \widetilde{y}_j, \quad \forall i = 1, \cdots, n-1$$

and clearly $s_{n-1} = -\sum_{j=1}^{n-1} \widetilde{y}_j = \widetilde{y}_n$ since $\langle e, \widetilde{\mathbf{y}} \rangle = 0$. Denote

$$\lambda_{max} = \|\mathbf{s}\|_\infty = \max\{|\sum_{j=1}^{i} \widetilde{y}_j| : i = 1, \cdots, n-1\}. \tag{3.22}$$

From the above analysis, it is easy to see that when $\lambda \geq \lambda_{max}$, there is an $\mathbf{s}^*$ such that

$$\frac{\mathbf{Py}}{\lambda} = \mathbf{G}^T \mathbf{s}^* \quad \text{and} \quad \|\mathbf{s}^*\|_\infty \leq 1.$$

According to (3.18), we have

$$\mathbf{x}^* = (\mathbf{P} + \mathbf{P}^\perp)\mathbf{y} - \lambda \mathbf{G}^T \mathbf{s}^* = \mathbf{P}^\perp \mathbf{y} = \frac{\langle e, \mathbf{y} \rangle}{\langle e, e \rangle} e = \frac{\langle e, \mathbf{y} \rangle}{n} e. \tag{3.23}$$

The maximal value for λ has been studied in Liu *et al.* (2010). However, a linear system has to be solved. From (3.22), it is easy to see that the maximal value can be obtained by a close form solution. Thus, our approach is more efficient.

3.3.3 Computing the Minimum Value for λ

We rewrite the dual problem (3.19) as:

$$\min_{\mathbf{s}} \frac{1}{2} \|\mathbf{y} - \lambda \mathbf{G}^T \mathbf{s}\|^2 \tag{3.24}$$

$$\text{s.t. } \|\mathbf{s}\|_\infty \leq 1.$$

Denote $g(\mathbf{s}) = \frac{1}{2}\|\mathbf{y} - \lambda\mathbf{G}^T\mathbf{s}\|^2$. The gradient of $g(\mathbf{s})$ can be found as: $g'(\mathbf{s}) = -\lambda\mathbf{G}(\mathbf{y} - \lambda\mathbf{G}^T\mathbf{s})$.

Let $\mathcal{B}_\infty$ denote the unit ∞ norm ball. We know that $\mathbf{s}^*$ is the unique optimal solution to the problem (3.24) if and only if

$$-g'(\mathbf{s}^*) \in N_{\mathcal{B}_\infty}(\mathbf{s}^*),$$

where $N_{\mathcal{B}_\infty}(\mathbf{s}^*)$ is the normal cone at $\mathbf{s}^*$ with respect to $\mathcal{B}_\infty$.

Let $\mathcal{I}^+(\mathbf{s}) = \{i : s_i = 1\}$, $\mathcal{I}^-(\mathbf{s}) = \{i : s_i = -1\}$, and $\mathcal{I}^\circ(\mathbf{s}) = \{i : s_i \in (-1, 1)\}$. Assume $\mathbf{d} \in N_{\mathcal{B}_\infty}(\mathbf{s})$, then $\mathbf{d}$ can be found as:

$$d_i \in \begin{cases} [0, +\infty), & \text{if } i \in \mathcal{I}^+(\mathbf{s}) \\ (-\infty, 0], & \text{if } i \in \mathcal{I}^-(\mathbf{s}) \\ 0, & \text{if } i \in \mathcal{I}^\circ(\mathbf{s}) \end{cases}$$

Therefore the optimality condition can be expressed as:

$$\mathbf{s}^* = \text{argmin}_{\mathbf{s}} \frac{1}{2}\|\mathbf{y} - \lambda\mathbf{G}^T\mathbf{s}\|^2$$

if and only if

$$\lambda\mathbf{G}(\mathbf{y} - \lambda\mathbf{G}^T\mathbf{s}^*) \in N_{\mathcal{B}_\infty}(\mathbf{s}^*).$$

Because $\lambda > 0$, $\lambda\mathbf{G}(\mathbf{y} - \lambda\mathbf{G}^T\mathbf{s}) \in N_{\mathcal{B}_\infty}(\mathbf{s}^*)$ is equivalent to

$$\mathbf{G}(\mathbf{y} - \lambda\mathbf{G}^T\mathbf{s}^*) \in N_{\mathcal{B}_\infty}(\mathbf{s}^*). \tag{3.25}$$

According to (3.18), we have

$$\begin{aligned} x_1^* &= y_1 + \lambda s_1^* \\ x_i^* &= y_i - \lambda(s_{i-1}^* - s_i^*), \text{for } 1 < i < n \\ x_n^* &= y_n - \lambda s_n^*, \end{aligned} \tag{3.26}$$

and

$$\mathbf{G}(\mathbf{y} - \lambda \mathbf{G}^T \mathbf{s}^*) = \begin{pmatrix} x_2^* - x_1^* \\ \vdots \\ x_n^* - x_{n-1}^* \end{pmatrix}. \tag{3.27}$$

By (3.25) and (3.27), we have the following observations:

B1. If $x_{i+1}^* > x_i^*$, $s_i^* = 1$;

B2. If $x_{i+1}^* = x_i^*$, $s_i^* \in [-1, 1]$;

B3. If $x_{i+1}^* < x_i^*$, $s_i^* = -1$.

Notice that, from (3.26), we can estimate a range for every x_i^*, which is not necessarily the tightest one. In fact, we have

$$x_i^* \in \begin{cases} [y_i - \lambda, y_i + \lambda], & \text{if } i \in \{1, n\} \\ [y_i - 2\lambda, y_i + 2\lambda], & \text{otherwise.} \end{cases} \tag{3.28}$$

Define

$$\lambda_{min} = \min \left\{ \frac{|y_{i+1} - y_i|}{3}, i \in \{1, n-1\}; \frac{|y_{i+1} - y_i|}{4}, i \in \{2, \ldots, n-2\} \right\}. \tag{3.29}$$

It follows that when $\lambda < \lambda_{min}$, the solution to (3.26) is fixed and can be found as:

$$s_i^* = \text{sign}(y_{i+1} - y_i), \quad i = 1, \ldots, n-1. \tag{3.30}$$

Then x_i^* can be computed accordingly by (3.26).

3.4 Experimental Results

In this section, we evaluate the efficiency of the proposed algorithm on synthetic and real-word data, and show several applications of the proposed algorithm.

3.4.1 Efficiency Comparison

We examine the efficiency of the proposed algorithm using synthetic datasets on 2D and 3D cases. For the 2D case, the competitors include

- SplitBregman written in C [1] (Goldstein and Osher, 2009);
- ADAL written in C faithfully based on the paper (Qin *et al.*, 2011);
- The dual method in Matlab [2] (Beck and Teboulle, 2009a);
- Dykstra written in C (Combettes and Pesquet, 2011);

For the 3D case, only the Dykstra's method and the proposed method (MTV) are compared, since the other algorithms are designed specifically for the 2D case.

The experiments are performed on a PC with quad-core Intel 2.67GHz CPU and 9GB memory. The code of MTV is written in C. Since the proposed method and the Dykstra's method can be implemented in parallel, we also compare their parallel versions implemented with OpenMP.

2D case

We generate synthetic images $\mathbf{Y} \in \Re^{N\times N}$ of different N. The value of each pixel is 1 or 0. A Gaussian noise $\epsilon = \mathcal{N}(0, 0.2^2)$ is added to each image as $\tilde{\mathbf{Y}} = \mathbf{Y} + \epsilon$. A synthetic example image is shown in Figure 3.3. The comparisons are based on the computation time. For a given λ, we first run MTV until a certain precision level specified by ϵ_{abs} and ϵ_{rel} is reached, and then run the others until they achieve an objective function value smaller than or equal to that of MTV. Different precision levels of the solutions are evaluated such that a fair comparison can be made. In

[1] `http://tag7.web.rice.edu/Split_Bregman.html`

[2] `http://iew3.technion.ac.il/~becka/papers/tv_fista.zip`

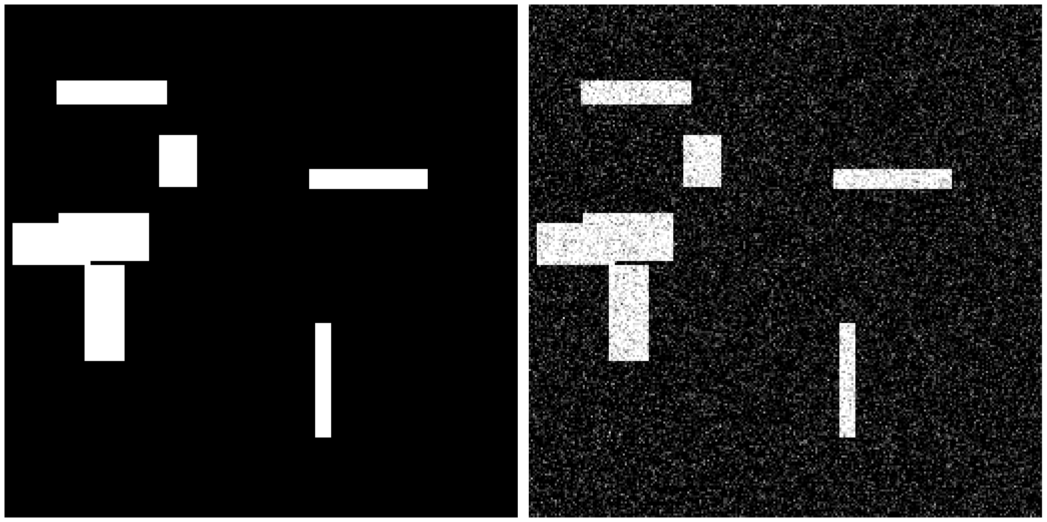


Figure 3.3: Synthetic images. Left: clean image; right: noisy image;

addition, we set the maximal iteration number of all methods to be 2000 in order to avoid slow convergence. The penalty parameters ρ for MTV and ADAL are fixed to 10. We vary the size of image ($N \times N$) from 50×50 to 2000×2000 with $\lambda = 0.35$, and vary the regularization parameter λ from 0.15 to 1 with a step size of 0.05 with a fixed $N = 500$. For each setting, we perform 20 trials and report the average computational time (seconds). The results are shown in Figure 3.4.

From Figure 3.4, we observe that the proposed method is much more efficient than its competitors. The non-parallel version of MTV is about 70 times faster than the dual method, and 8 times fasters than ADAL when N is 2000 and $\epsilon_{abs} = \epsilon_{rel} = 1e-3$. Although the subproblems of MTV and Dykstra are the same, Dykstra is about 12 times slower than MTV, demonstrating that MTV has faster convergence than Dykstra. Utilizing parallel computing, the parallel version of MTV and Dykstra are about 3.5 times more efficient than their non-parallel version in a quad-core PC. We also observe that the Split Bregman method, dual method, and ADAL need more iterations to achieve a similar precision to that of MTV when the regularization parameter λ increases, i.e., the portion of the nonsmooth part increases. However, MTV and

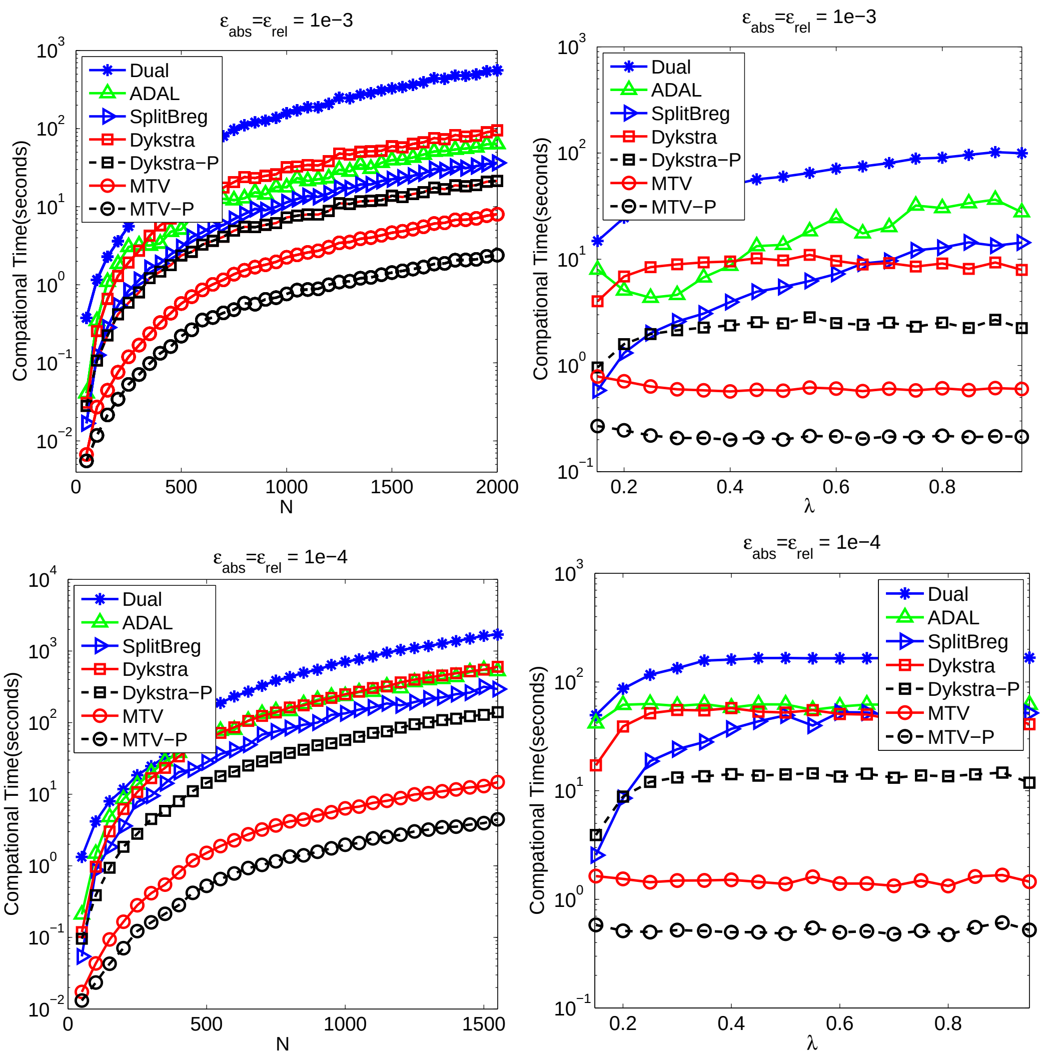


Figure 3.4: Comparison of SplitBregman (Goldstein and Osher, 2009), ADAL (Qin *et al.*, 2011), Dual Method (Beck and Teboulle, 2009a), Dykstra (Combettes and Pesquet, 2011), and our proposed MTV algorithm in terms of computational time (in seconds and in the logarithmic scale). Dykstra-P and MTV-P are the parallel version of Dykstra and MTV. Different precision levels are used for comparison. The size of image is $N \times N$. Left column: $\lambda = 0.35$ with N varying from 50 to 2000; right column: $N = 500$ with λ varying from 0.15 to 0.95.

Dykstra are more stable when λ varies. The reason is that we directly compute the exact optimal solution of the proximal operator of the fused regularization in the subproblems of MTV and Drystra, unlike ADAL and the Split Bregman method which perform soft-thresholding.

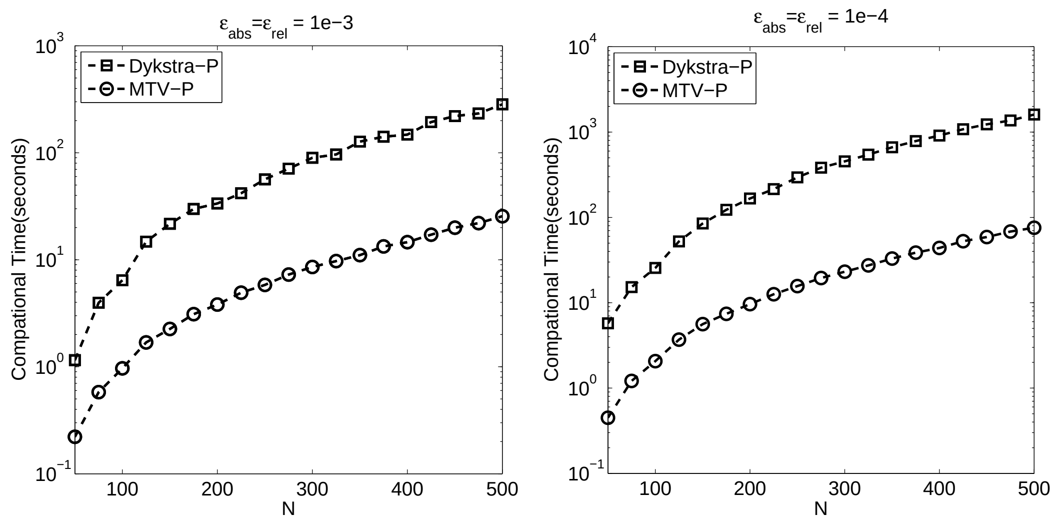


Figure 3.5: Comparison of Dykstra and the proposed MTV in terms of computational time (in seconds and in the logarithmic scale) in the 3D case. Different precision levels are used for comparison. The size of 3D images is $N \times N \times 50$, and N varies from 50 to 500 with $\lambda = 0.35$.

3D case

The synthetic 3D images are generated in a similar manner to the 2D case. Gaussian noise $\epsilon = \mathcal{N}(0, 0.2^2)$ is added to each pixel. We set the size of 3D images to $N \times N \times 50$, and vary N from 50 to 500 with a step size of 25. The regularization parameter λ is set to 0.35. We apply the Dykstra's method and MTV on the noisy 3D images. In this experiment, we compare the computational time of Dykstra and MTV in a similar setting to the 2D case. Figure 3.5 shows the comparison between the Dykstra's method and MTV. From Figure 3.5, we can see that MTV is much more efficient than Dykstra, demonstrating the efficiency of MTV. MTV is about 20 times faster than Dykstra when $N = 500$ and $\epsilon_{abs} = \epsilon_{rel} = 1e - 4$.

Scalability

We conduct experiments to evaluate the scalability of the proposed method. The experiments are performed on a Linux server with 4 quad-core Intel Xeon 2.93GHz

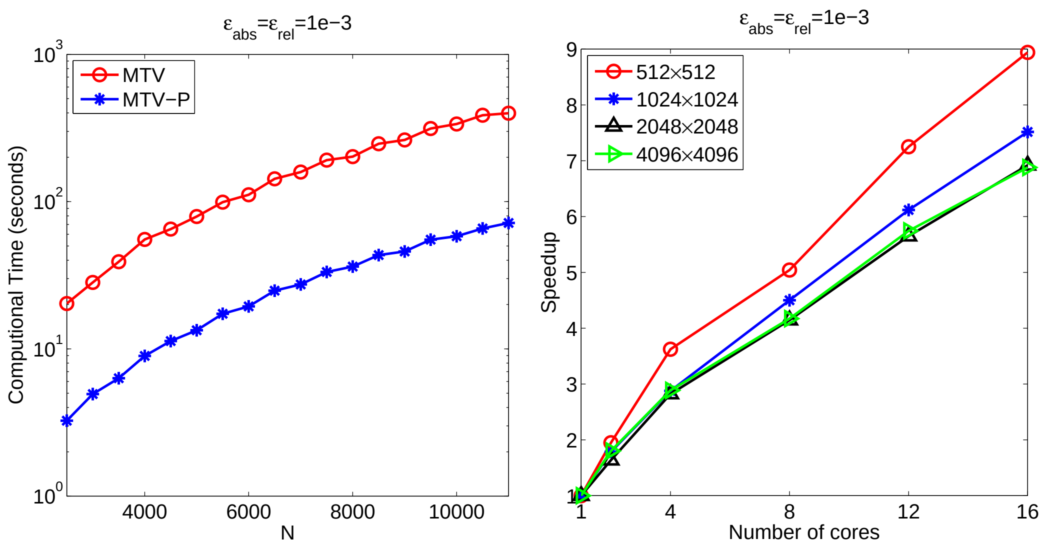


Figure 3.6: Scalability of the proposed method. The size of image is $N \times N$, and $\lambda = 0.35$. Left: the computational time of MTV and MTV-P with 12 processors and N varying from 2500 to 11000; right: the speedup of MTV-P with respect to the number of processors varying from 1 to 16.

CPUs and 65GB memory. We vary the size of images ($N \times N$) from 2500×2500 to 11000×11000 with 12 processors, and the number of processors from 1 to 16 with a fixed image size. The regularization parameter λ is set to be 0.35. For each setting, the average computational time of 10 trials is reported to demonstrate the efficiency/speedup of MTV-P (Figure 3.6). As shown in Figure 3.6, the computational time of MTV-P is less than 100 seconds when $N = 11000$, demonstrating the superiority of the proposed method. We also observe that the speedup increases almost linearly with the number of processors used. The speedup is less than the number of processors used because of the parallel overhead.

3.4.2 Applications

Image reconstruction

Due to the excellent depiction of soft tissue changes, Magnetic Resonance Imaging (MRI) has been widely used in medical diagnosis. Based on the compressive sensing theory, it is possible to reconstruct perfect signals from a limited number of samples by taking advantage of the sparse nature of the signals in a transform domain. In the case of MRI, an accurate reconstruction of MR images from undersampled K-space data is possible, reducing the cost of scanning. The formulation of image reconstruction is given by

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathfrak{R}(\mathbf{X}) - \mathbf{b}\|^2 + \lambda_1\|\mathfrak{W}(\mathbf{X})\|_1 + \lambda_2\|\mathbf{X}\|_{TV} \tag{3.31}$$

where $\mathbf{b}$ is the undersampled measurements of K-space data, $\mathfrak{R}$ is partial Fourier transformation and $\mathfrak{W}$ is wavelet transform. We try to reconstruct the image $\mathbf{X} \in \Re^{m\times n}$ from the undersampled measurements $\mathbf{b}$. A fast algorithm, FCSA, is introduced by Huang *et al.* (2011). One of the key steps in FCSA is the proximal operator of the 2D TV norm, which is a special case of MTV. In Huang *et al.* (2011), the dual method proposed in Beck and Teboulle (2009a) is used to solve the proximal operator. We follow the same framework as FCSA, but apply the proposed MTV to solve the proximal operator to achieve a speedup gain.

We compare two approaches: FCSA with the dual method (FSCA-Dual)(Huang *et al.*, 2011) and FCSA with MTV (FSCA-MTV). We apply these two methods on four 2D MR images [3] : cardiac, brain, chest, and artery. We follow the same sampling strategy as in (Huang *et al.*, 2011). The sample ratio is set to about 25%. A Gaussian noise $\epsilon = \mathcal{N}(0, 0.01^2)$ is added to the observed measurements $\mathbf{b}$. For a fair comparison, we first run FCSA-MTV and keep track of the objective function values of MTV in each iteration, then run FCSA-Dual. In each outer iteration, the dual method stops when its objective function value is equal to or smaller than the corresponding tracked objective function value of MTV. Both FCSA-Dual and FCSA-MTV run 50 iterations. Only the computational time of the proximal operator by dual method and MTV, is recorded. The precision parameters of MTV are set to $\epsilon_{abs} = \epsilon_{rel} = 1e-3$, and the

[3] `http://ranger.uta.edu/~huang/R_CSMRI.htm`

Table 3.1: Comparison of the dual method and MTV in FCSA in terms of average computational time of 50 iterations (seconds).

Methods	Cardiac	Brain	Chest	Artery
Dual	0.6762	0.5855	0.5813	0.7588
MTV	0.0066	0.0061	0.0056	0.0078
Speedup	102.45	95.98	103.80	97.28

dual update step length ρ is set to 10. Since the objective function of both methods are identical, and the precision of each iteration are about the same, the solutions of both methods are expected to be the same.

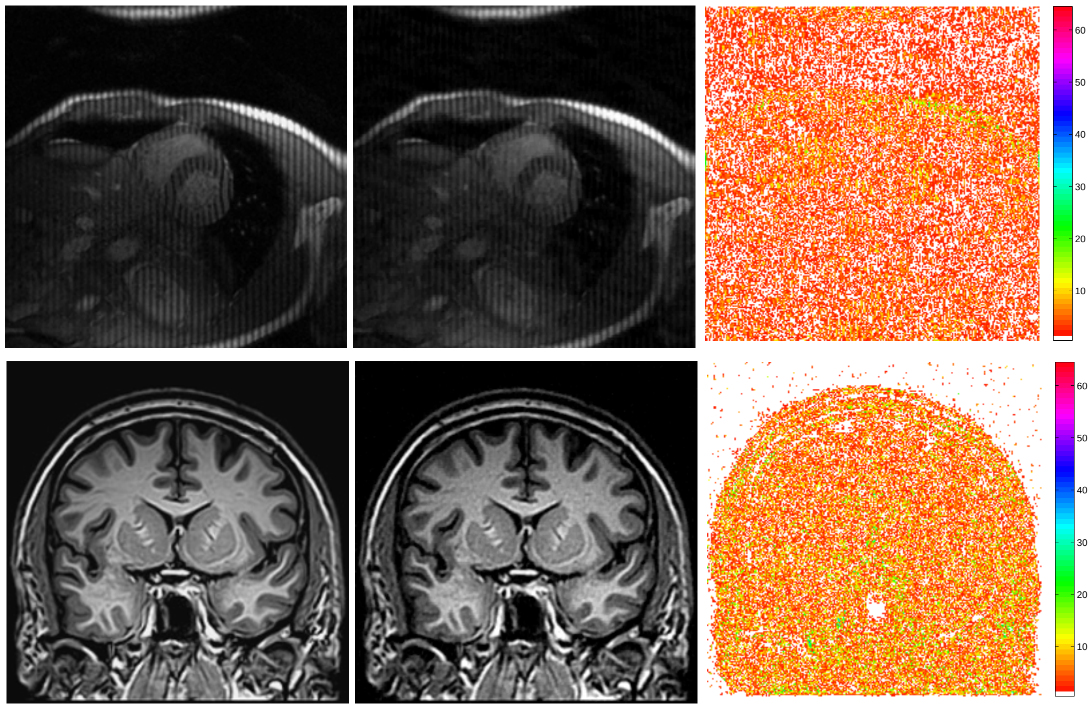


Figure 3.7: MRI reconstruction. Columns: original (left), FCSA-Dual and FCSA-MTV(middle), and the difference image between original image and reconstructed image (right); (top) Cardiac: SNR of two methods are 17.615; (bottom) Brain: SNR are 20.376;

The reconstruction results of the MR images are shown in Figure 3.7 and Figure 3.8. Table 3.1 shows the average time of dual method and MTV for 50 iterations.

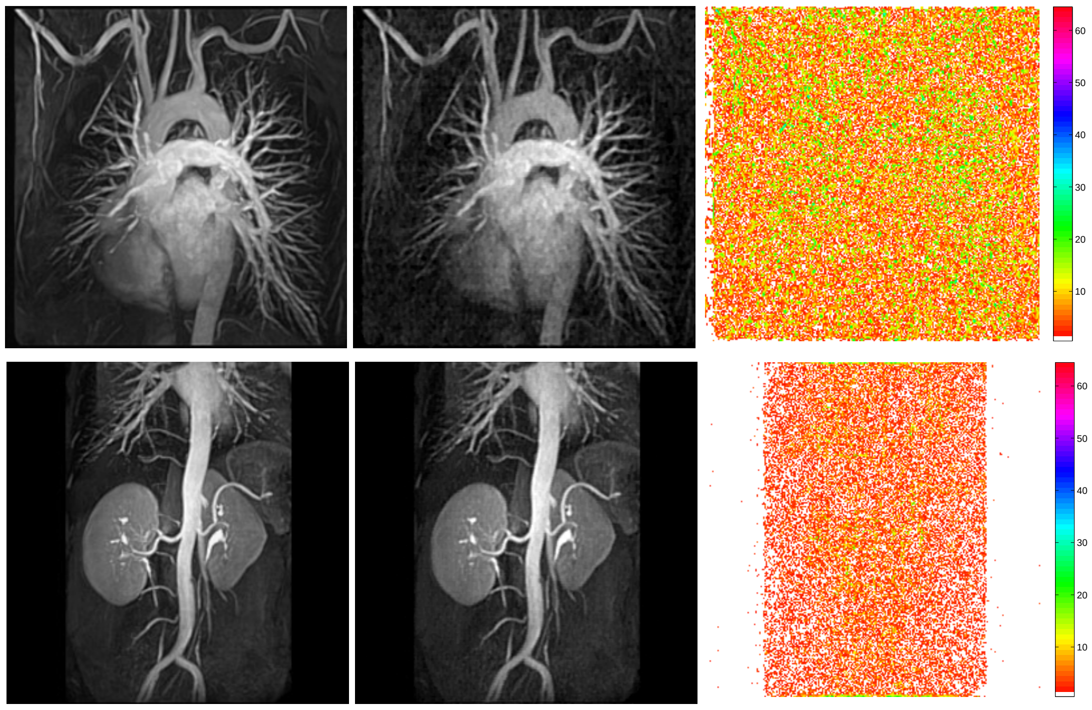


Figure 3.8: MRI reconstruction. Columns: original (left), FCSA-Dual and FCSA-MTV(middle), and the difference image between original image and reconstructed image (right); (top) Chest: both SNR are 16.082; (bottom) Artery: both SNR are 23.769;

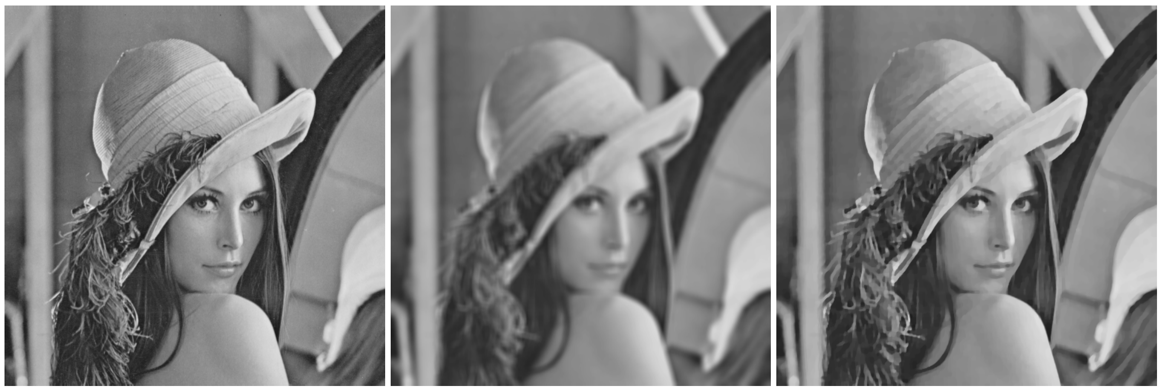


Figure 3.9: Image deblurring: original image(left), blurred and noisy image (middle), and deblurred image (right). The SNR of the blurred image is 11.01, and the SNR of the deblurred image is 17.23.

Since each iteration of FCSA-MTV and FCSA-Dual are the same, FCSA-MTV and FCSA-Dual have the same SNR. But we can observe from Table 3.1 that MTV is more efficient than dual method(about 100 times speedup), thus FCSA-MTV is more efficient than FCSA-Dual.

Image deblurring

The proposed method can be used to deblur images. The formulation of TV-based image deblurring model is given by

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathfrak{B}(\mathbf{X}) - \mathbf{Y}\|^2 + \lambda\|\mathbf{X}\|_{TV}, \tag{3.32}$$

where $\mathbf{Y} \in \Re^{m\times n}$ is the observed blurred and noisy image, $\mathfrak{B} : \Re^{m\times n} \to \Re^{m\times n}$ is a linear transformation encoding the blurring operator, and $\mathbf{X} \in \Re^{m\times n}$ is the image to be restored. A popular approach to solve the convex optimization problem in (3.32) is FISTA (Beck and Teboulle, 2009a,b). One of the key steps is the proximal operator of TV regularization. Similar to the previous experiment, we use MTV instead of the dual method (Beck and Teboulle, 2009a) to solve the proximal operator of TV regularization to achieve a speedup gain. The "lena" image of size 512×512 is used in this experiment. The image is rescaled to [0,1], and then blurred by an average filter of size 9×9. Furthermore, a Gaussian noise, $\mathcal{N}(0, 0.001^2)$, is added to the blurred image. The parameter setting of MTV is the same as the previous experiment. The regularization parameter λ is set to 0.001. The results are shown in Figure 3.9. The average computation time of the dual method for 100 iterations is 1.066 seconds, while that of MTV is 0.037 seconds. The proposed MTV method achieves about 29 times speedup.

Video denoising

A video is a 3-mode tensor. The proposed method in the 3D case can be used to denoise video. We expect that pixel values should be smooth along all 3 modes. In this experiment, we use a time series of 2D MR images of heart beats downloaded

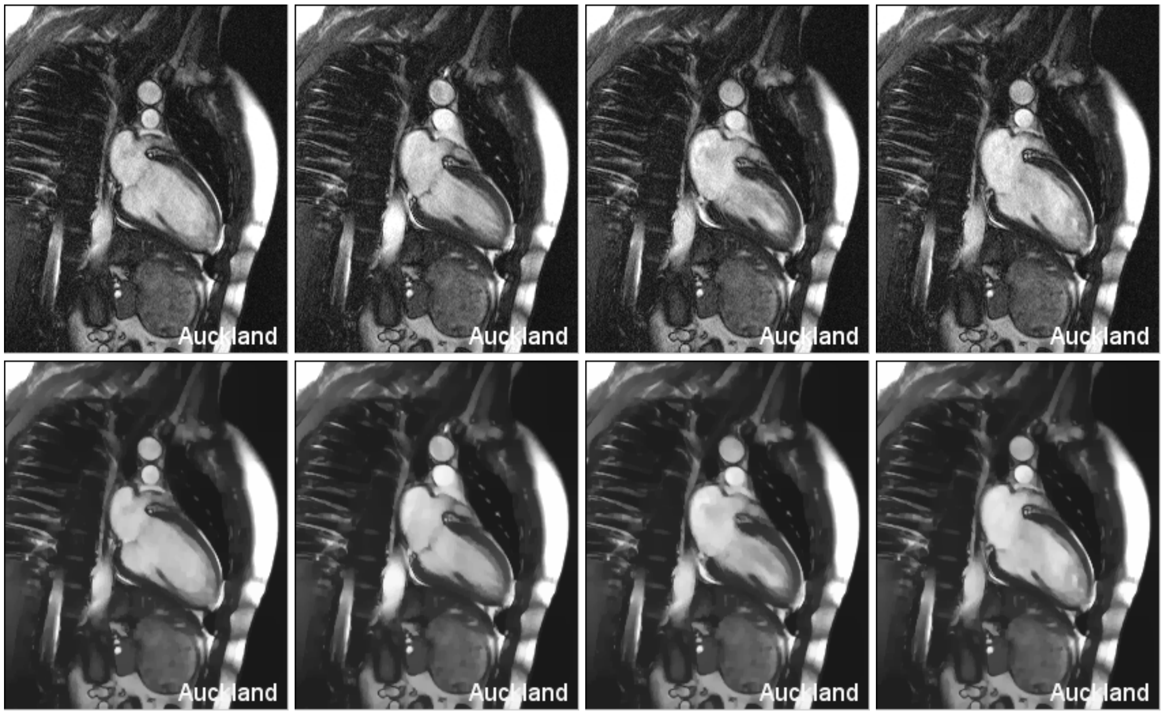


Figure 3.10: Sample frames of video denoising: original frames (top), and denoised frames (bottom) (best viewed on a screen).

from the website of the Cardiac Atlas [4] . The 2D MR images are in the format of avi, which includes 32 frames. We applied the proposed method and the Dystra's method to denoise all the MR images as a 3-mode tensor of size $257 \times 209 \times 32$. The computational time of MTV is 4.482 seconds, and the computational time of the Dykstra's method is 43.751 seconds. The speedup is about 10 times. Some sample result frames are shown in Figure 3.10. This experiment demonstrates the effectiveness of total variation regularization in video denoising.

3.5 Conclusion

In this chapter, we propose an efficient optimization of the multidimensional total variation regularization problems. We employ an efficient ADMM algorithm to solve the formulation. The key idea of our algorithm is to decompose the original problem

[4] `http://atlas.scmr.org/download.html`

into a set of independent and small problems, which can be solved exactly and efficiently. Furthermore, the set of independent problems can be solved in parallel. Thus, the proposed method can handle large-scale problems efficiently. We also establish the global convergence of the proposed algorithm. The experimental results demonstrate the efficiency of the proposed algorithm. The proposed algorithm opens the possibility of utilizing the power of GPU computing to further improve the efficiency of the proposed algorithm. We will explore the GPU computing in the future work. Moreover, we plan to apply the proposed algorithm to other real-world applications, such as MBB (mobile broad band) data and 3G network data, both are big data problems.

Chapter 4

FUSED MULTIPLE GRAPHICAL LASSO

4.1 Introduction

Undirected graphical models explore the relationships among a set of random variables through their joint distribution. The estimation of undirected graphical models has applications in many domains, such as computer vision, biology, and medicine (Guo *et al.*, 2011; Huang *et al.*, 2009; Yang *et al.*, 2012). One instance is the analysis of gene expression data. As shown in many biological studies, genes tend to work in groups based on their biological functions, and there exist some regulatory relationships between genes (Chuang *et al.*, 2007). Such biological knowledge can be represented as a graph, where nodes are the genes, and edges describe the regulatory relationships. Graphical models provide a useful tool for modeling these relationships, and can be used to explore gene activities. One of the most widely used graphical models is the Gaussian graphical model (GGM), which assumes the variables to be Gaussian distributed (Banerjee *et al.*, 2008; Yuan and Lin, 2007). In the framework of GGM, the problem of learning a graph is equivalent to estimating the inverse of the covariance matrix (precision matrix), since the nonzero off-diagonal elements of the precision matrix represent edges in the graph (Banerjee *et al.*, 2008; Yuan and Lin, 2007).

In recent years many research efforts have focused on estimating the precision matrix and the corresponding graphical model (see, for example (Banerjee *et al.*, 2008; Friedman *et al.*, 2008; Hsieh *et al.*, 2011; Huang *et al.*, 2009; Li and Toh, 2010; Liu *et al.*, 2011; Lu, 2009, 2010; Mazumder and Hastie, 2012b; Meinshausen and Bühlmann, 2006; Olsen *et al.*, 2012; Yuan and Lin, 2007). Meinshausen and Bühlmann (2006) estimated edges for each node in the graph by fitting a lasso problem (Tibshirani, 1996) using the remaining variables as predictors. Yuan and Lin (2007) and Banerjee *et al.* (2008) propose a penalized maximum likelihood model using ℓ_1 regularization to estimate the sparse precision matrix. Numerous methods have been developed for solving this model. For example, d'Aspremont *et al.* (2008) and Lu (2009, 2010) studied Nesterov's smooth gradient methods (Nesterov, 2005) for solving this problem or its dual. Banerjee *et al.* (2008) and Friedman *et al.* (2008) propose block coordinate ascent methods for solving the dual problem. The latter method (Friedman *et al.*, 2008) is widely referred to as Graphical lasso (GLasso). Mazumder and Hastie (2012b) propose a new algorithm called DP-GLasso, each step of which is a box-constrained QP problem. Scheinberg and Rish (2009) propose a coordinate descent method for solving this model in a greedy approach. Yuan (2012) and Scheinberg *et al.* (2010) apply alternating direction method of multipliers (ADMM) (Boyd *et al.*, 2011) to solve this problem. Li and Toh (2010) and Yuan and Lin (2007) propose to solve this problem using interior point methods. Wang *et al.* (2010), Hsieh *et al.* (2011), Olsen *et al.* (2012), and Dinh *et al.* (2013) studied Newton method for solving this model. The main challenge of estimating a sparse precision matrix for the problems with a large number of nodes (variables) is its intensive computation. Witten *et al.* (2011) and Mazumder and Hastie (2012a) independently derive a necessary and sufficient condition for the solution of a single graphical lasso to be block diagonal (subject to some rearrangement of variables). This can be used as a simple screening test to identify the associated blocks, and the original problem can thus be decomposed into a group of smaller sized but independent problems corresponding to these blocks. When the number of blocks is large, it can achieve massive computational gain. However, these formulations assume that observations are independently drawn from a single Gaussian distribution. In many applications the observations may be drawn from multiple Gaussian distributions; in this case, multiple graphical models need to be estimated.

There are some recent works on the estimation of multiple precision matrices (Danaher *et al.*, 2013; Guo *et al.*, 2011; Hara and Washio, 2011; Honorio and Samaras, 2010; Kolar *et al.*, 2010; Kolar and Xing, 2011; Mohan *et al.*, 2012; Zhou *et al.*, 2010). Guo *et al.* (2011) propose a method to jointly estimate multiple graphical models using a hierarchical penalty. However, their model is not convex. Honorio and Samaras (2010) propose a convex formulation to estimate multiple graphical models using the $\ell_{1,\infty}$ regularizer. Hara and Washio (2011) introduce a method to learn common substructures among multiple graphical models. Danaher *et al.* (2013) estimate multiple precision matrices simultaneously using a pairwise fused penalty and grouping penalty. ADMM is used to solve the problem, but it requires computing multiple eigen decompositions at each iteration. Mohan *et al.* (2012) propose to estimate multiple precision matrices based on the assumption that the network differences are generated from node perturbations. Compared with single graphical model learning, learning multiple precision matrices jointly is even more challenging to solve. Recently, a necessary and sufficient condition for multiple graphs to be decomposable is proposed in (Danaher *et al.*, 2013). However, such necessary and sufficient condition is restricted to two graphs only when the fused penalty is used. It is not clear whether this screening rule can be extended to the more general case with more than two graphs, which is the case in brain network modeling.

There are several types of fused penalties that can be used for estimating multiple (more than two) graphs such as pairwise fused penalty and sequential fused penalty (Tibshirani *et al.*, 2005). In this chapter we set out to address the sequential fused case first, because we work on practical applications that can be more appropriately formulated using the sequential formulation. Specifically, we consider the problem of estimating multiple graphical models by maximizing a penalized log likelihood with ℓ_1 and sequential fused regularization. The ℓ_1 regularization yields a sparse solution, and the fused regularization encourages adjacent graphs to be similar. The graphs considered in this chapter have a natural order, which is common in many applications. A motivating example is the modeling of brain networks for Alzheimer's disease using neuroimaging data such as Positron emission tomography (PET). In this case, we want to estimate graphical models for three groups: normal controls (NC), patients of mild cognitive impairment (MCI), and Alzheimer's patients (AD). These networks are expected to share some common connections, but they are not identical. Furthermore, the networks are expected to evolve over time, in the order of disease progression from NC to MCI to AD. Estimating the graphical models separately fails to exploit the common structures among them. It is thus desirable to jointly estimate the three networks (graphs). Our key technical contribution is to establish the necessary and sufficient condition for the solution of the fused multiple graphical lasso (FMGL) to be block diagonal. The duality theory and several other tools in linear programming are used to derive the necessary and sufficient condition. Based on this crucial property of FMGL, we develop a screening rule which enables the efficient estimation of large multiple precision matrices for FMGL. The proposed screening rule can be combined with any algorithms to reduce computational cost. We employ a second-order method (Hsieh *et al.*, 2011; Lee *et al.*, 2012; Tseng and Yun, 2009) to solve the fused multiple graphical lasso, where each step is solved by the spectral projected gradient method (Lu and Zhang, 2011; Wright *et al.*, 2009). In addition, we propose an active set identification scheme to identify the variables to be updated in each step of the second-order method, which reduces the computation cost of each step. We conduct experiments on both synthetic and real data; our results demonstrate the effectiveness and efficiency of the proposed approach.

The rest of this chapter is organized as follows. We introduce the fused multiple graphical lasso formulation in Section 4.2. The screening rule is presented in Section 4.3. The proposed second-order method is presented in Section 4.4. The experimental results are shown in Section 4.5. We conclude the chapter in Section 4.6.

4.2 Fused Multiple Graphical Lasso

Assume we are given K data sets, $x^{(k)} \in \Re^{n_k \times p},\ k = 1, \ldots, K$ with $K \geq 2$, where n_k is the number of samples, and p is the number of features. The p features are common for all K data sets, and all $\sum_{k=1}^{K} n_k$ samples are independent. Furthermore, the samples within each data set $x^{(k)}$ are identically distributed with a p-variate Gaussian distribution with zero mean and positive definite covariance matrix $\mathbf{\Sigma}^{(k)}$, and there are many conditionally independent pairs of features, i.e., the precision matrix $\mathbf{\Theta}^{(k)} = (\mathbf{\Sigma}^{(k)})^{-1}$ should be sparse. For notational simplicity, we assume that $n_1 = \cdots = n_K = n$. Denote the sample covariance matrix for each data set $x^{(k)}$ as $\mathbf{S}^{(k)}$ with $\mathbf{S}^{(k)} = \frac{1}{n}(x^{(k)})^T x^{(k)}$, and $\mathbf{\Theta} = (\mathbf{\Theta}^{(1)}, \ldots, \mathbf{\Theta}^{(K)})$. Then the negative log likelihood for the data takes the form of

$$\sum_{k=1}^{K} \left(-\log\det(\mathbf{\Theta}^{(k)}) + \operatorname{tr}(\mathbf{S}^{(k)}\mathbf{\Theta}^{(k)}) \right). \tag{4.1}$$

Clearly, minimizing (4.1) leads to the maximum likelihood estimate (MLE) $\widehat{\mathbf{\Theta}}^{(k)} = (\mathbf{S}^{(k)})^{-1}$. However, the MLE fails when $\mathbf{S}^{(k)}$ is singular. Furthermore, the MLE is usually dense. The ℓ_1 regularization has been employed to induce sparsity, resulting in the sparse inverse covariance estimation Banerjee *et al.* (2008); Friedman *et al.* (2008); Yuan and Lin (2006). In this chapter, we employ both the ℓ_1 regularization and the fused regularization for simultaneously estimating multiple graphs. The ℓ_1 regularization leads to a sparse solution, and the fused penalty encourages $\mathbf{\Theta}^{(k)}$ to be similar to its neighbors. Mathematically, we solve the following formulation:

$$\min_{\mathbf{\Theta}^{(k)} \succ 0, k=1\ldots K} \sum_{k=1}^{K} \left(-\log\det(\mathbf{\Theta}^{(k)}) + \operatorname{tr}(\mathbf{S}^{(k)}\mathbf{\Theta}^{(k)})\right) + P(\mathbf{\Theta}), \tag{4.2}$$

where

$$P(\mathbf{\Theta}) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\mathbf{\Theta}_{ij}^{(k)}| + \lambda_2 \sum_{k=1}^{K-1} \sum_{i \neq j} |\mathbf{\Theta}_{ij}^{(k)} - \mathbf{\Theta}_{ij}^{(k+1)}|,$$

$\lambda_1 > 0$ and $\lambda_2 > 0$ are positive regularization parameters. This model is referred to as the fused multiple graphical lasso (FMGL).

To ensure the existence of a solution for problem (4.2), we assume throughout this chapter that $\operatorname{diag}(\mathbf{S}^{(k)}) > 0, k = 1, \ldots, K$. Recall that $\mathbf{S}^{(k)}$ is a sample covariance matrix, and hence $\operatorname{diag}(\mathbf{S}^{(k)}) \geq 0$. The diagonal entries may be not, however, strictly positive. But we can always add a small perturbation (say 10^{-8}) to ensure the above assumption holds. The following theorem shows that under this assumption the FMGL (4.2) has a unique solution.

Theorem 2. *Under the assumption that* $\operatorname{diag}(\mathbf{S}^{(k)}) > 0, k = 1, \ldots, K$, *problem (4.2) has a unique optimal solution.*

To prove Theorem 2, we first establish a technical lemma which regards the existence of a solution for a standard graphical lasso problem.

Lemma 3. *Let* $\mathbf{S} \in \mathcal{S}_+^p$ *and* $\mathbf{\Lambda} \in \mathcal{S}^p$ *be such that* $\operatorname{Diag}(\mathbf{S}) + \mathbf{\Lambda} > 0$ *and* $\operatorname{diag}(\mathbf{\Lambda}) \geq 0$. *Consider the problem*

$$\min_{\mathbf{X} \succ 0} \underbrace{-\log\det(\mathbf{X}) + \operatorname{tr}(\mathbf{S}\mathbf{X}) + \sum_{ij} \mathbf{\Lambda}_{ij} |\mathbf{X}_{ij}|}_{f(\mathbf{X})} . \tag{4.3}$$

Then the following statements hold:

(a) Problem (4.3) has a unique optimal solution;

(b) The sub-level set $\mathcal{L} = \{\mathbf{X} \succ 0 : f(\mathbf{X}) \le \alpha\}$ is compact for any $\alpha \ge f^*$, where f^* is the optimal value of (4.3).

Proof. (a) Let $\mathcal{U} = \{\mathbf{U} \in \mathcal{S}^p : \mathbf{U}_{ij} \in [-1, 1],\ \forall i, j\}$. Consider the problem

$$\max_{\mathbf{U}\in\mathcal{U}} \{\log\det(\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}) : \mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U} \succ 0\}. \tag{4.4}$$

We first claim that the feasible region of problem (4.4) is nonempty, or equivalently, there exists $\bar{\mathbf{U}} \in \mathcal{U}$ such that $\lambda_{\min}(\mathbf{S} + \mathbf{\Lambda} \circ \bar{\mathbf{U}}) > 0$. Indeed, one can observe that

$$\begin{aligned} \max_{\mathbf{U}\in\mathcal{U}} \lambda_{\min}(\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}) &= \max_{t,\mathbf{U}\in\mathcal{U}} \{t : \mathbf{\Lambda} \circ \mathbf{U} + \mathbf{S} - t\mathbf{I} \succeq 0\}, \\ &= \min_{\mathbf{X}\succeq 0} \max_{t,\mathbf{U}\in\mathcal{U}} \{t + \operatorname{tr}(\mathbf{X}(\mathbf{\Lambda} \circ \mathbf{U} + \mathbf{S} - t\mathbf{I}))\}, \\ &= \min_{\mathbf{X}\succeq 0} \left\{ \operatorname{tr}(\mathbf{S}\mathbf{X}) + \sum_{ij} \mathbf{\Lambda}_{ij}|\mathbf{X}_{ij}| :\ \operatorname{tr}(\mathbf{X}) = 1 \right\}, \end{aligned} \tag{4.5}$$

where the second equality follows from the Lagrangian duality since its associated Slater condition is satisfied. Let $\Omega := \{\mathbf{X} \in \mathcal{S}^p : \operatorname{tr}(\mathbf{X}) = 1,\ \mathbf{X} \succeq 0\}$. By the assumption $\operatorname{Diag}(\mathbf{S}) + \mathbf{\Lambda} > 0$, we see that $\mathbf{\Lambda}_{ij} > 0$ for all $i \ne j$ and $\mathbf{S}_{ii} + \mathbf{\Lambda}_{ii} > 0$ for every i. Since $\Omega \subset \mathcal{S}^p_+$, we have $\operatorname{tr}(\mathbf{S}\mathbf{X}) \ge 0$ for all $\mathbf{X} \in \Omega$. If there exists some $k \ne l$ such that $\mathbf{X}_{kl} > 0$, then $\sum\limits_{i\ne j} \mathbf{\Lambda}_{ij}|\mathbf{X}_{ij}| > 0$ and hence,

$$\operatorname{tr}(\mathbf{S}\mathbf{X}) + \sum_{ij} \mathbf{\Lambda}_{ij}|\mathbf{X}_{ij}| \ > \ 0,\ \forall \mathbf{X} \in \Omega. \tag{4.6}$$

Otherwise, one has $\mathbf{X}_{ij} = 0$ for all $i \ne j$, which, together with the facts that $\mathbf{S}_{ii} + \mathbf{\Lambda}_{ii} > 0$ for all i and $\operatorname{tr}(\mathbf{X}) = 1$, implies that for all $\mathbf{X} \in \Omega$,

$$\operatorname{tr}(\mathbf{S}\mathbf{X}) + \sum_{ij} \mathbf{\Lambda}_{ij}|\mathbf{X}_{ij}| \ = \ \sum_i (\mathbf{S}_{ii} + \mathbf{\Lambda}_{ii})\mathbf{X}_{ii} \ \ge \ \operatorname{tr}(\mathbf{X}) \min_i (\mathbf{S}_{ii} + \mathbf{\Lambda}_{ii}) \ > \ 0.$$

Hence, (4.6) again holds. Combining (4.5) with (4.6), one can see that $\max\limits_{\mathbf{U}\in\mathcal{U}} \lambda_{\min}(\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}) \ > \ 0$. Therefore, problem (4.4) has at least a feasible solution.

We next show that problem (4.4) has an optimal solution. Let $\bar{\mathbf{U}}$ be a feasible point of (4.4), and

$$\bar{\Omega} := \{\mathbf{U} \in \mathcal{U} : \log\det(\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}) \;\geq\; \log\det(\mathbf{S} + \mathbf{\Lambda} \circ \bar{\mathbf{U}}),\; \mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U} \succ 0\}.$$

One can observe that $\{\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U} : \mathbf{U} \in \mathcal{U}\}$ is compact. Using this fact, it is not hard to see that $\log\det(\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}) \to -\infty$ as $\mathbf{U} \in \mathcal{U}$ and $\lambda_{\min}(\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}) \downarrow 0$. Thus there exists some $\delta > 0$ such that

$$\bar{\Omega} \subseteq \{\mathbf{U} \in \mathcal{U} : \mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U} \succeq \delta I\},$$

which implies that

$$\bar{\Omega} = \{\mathbf{U} \in \mathcal{U} : \log\det(\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}) \;\geq\; \log\det(\mathbf{S} + \mathbf{\Lambda} \circ \bar{\mathbf{U}}),\; \mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U} \succeq \delta I\}.$$

Hence, $\bar{\Omega}$ is a compact set. In addition, one can observe that problem (4.4) is equivalent to

$$\max_{\mathbf{U} \in \bar{\Omega}} \log\det(\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}).$$

The latter problem clearly has an optimal solution and so is problem (4.4).

Finally we show that $\mathbf{X}^* = (\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}^*)^{-1}$ is the unique optimal solution of (4.3), where $\mathbf{U}^*$ is an optimal solution of (4.4). Since $\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}^* \succ 0$, we have $\mathbf{X}^* \succ 0$. By the definitions of $\mathcal{U}$ and $\mathbf{X}^*$, and the first-order optimality conditions of (4.4) at $\mathbf{U}^*$, one can have

$$\mathbf{U}^*_{ij} = \begin{cases} 1 & \text{if } \mathbf{X}^*_{ij} > 0; \\ \beta \in [-1, 1] & \text{if } \mathbf{X}^*_{ij} = 0; \\ -1 & \text{otherwise.} \end{cases}$$

It follows that $\mathbf{\Lambda} \circ \mathbf{U}^* \in \partial(\sum_{ij} \mathbf{\Lambda}_{ij}|\mathbf{X}_{ij}|)$ at $\mathbf{X} = \mathbf{X}^*$, where $\partial(\cdot)$ stands for the subdifferential of the associated convex function. For convenience, let $f(\mathbf{X})$ denote the objective function of (4.3). Then we have

$$-(\mathbf{X}^*)^{-1} + \mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}^* \in \partial f(\mathbf{X}^*),$$

which, together with $\mathbf{X}^* = (\mathbf{S} + \mathbf{\Lambda} \circ \mathbf{U}^*)^{-1}$, implies that $0 \in \partial f(\mathbf{X}^*)$. Hence, $\mathbf{X}^*$ is an optimal solution of (4.3) and moreover it is unique due to the strict convexity of $-\log\det(\cdot)$.

(b) By statement (a), problem (4.3) has a finite optimal value f^*. Hence, the above sub-level set $\mathcal{L}$ is nonempty. We can observe that for any $\mathbf{X} \in \mathcal{L}$,

$$\begin{aligned} \frac{1}{2}\sum_{ij} \mathbf{\Lambda}_{ij}|\mathbf{X}_{ij}| &= f(\mathbf{X}) - \underbrace{[-\log\det(\mathbf{X}) + \operatorname{tr}(\mathbf{SX}) + \frac{1}{2}\sum_{ij}\mathbf{\Lambda}_{ij}|\mathbf{X}_{ij}|]}_{\underline{f}(\mathbf{X})}, \\ &\leq \alpha - \underline{f}^*, \end{aligned} \tag{4.7}$$

where $\underline{f}^* := \inf\{\underline{f}(\mathbf{X}) : \mathbf{X} \succ 0\}$. By the assumption $\operatorname{Diag}(\mathbf{S}) + \mathbf{\Lambda} > 0$, one has $\operatorname{Diag}(\mathbf{S}) + \mathbf{\Lambda}/2 > 0$. This together with statement (a) yields $\underline{f}^* \in \Re$. Notice that $\mathbf{\Lambda}_{ij} > 0$ for all $i \neq j$. This relation and (4.7) imply that $\mathbf{X}_{ij}$ is bounded for all $\mathbf{X} \in \mathcal{L}$ and $i \neq j$. In addition, it is well-known that $\det(\mathbf{X}) \leq \mathbf{X}_{11}\mathbf{X}_{22}\cdots\mathbf{X}_{pp}$ for all $\mathbf{X} \succeq 0$. Using this relation, the definition of $f(\cdot)$, and the boundedness of $\mathbf{X}_{ij}$ for all $\mathbf{X} \in \mathcal{L}$ and $i \neq j$, we have that for every $\mathbf{X} \in \mathcal{L}$,

$$\begin{aligned} \sum_i -\log(\mathbf{X}_{ii}) + (\mathbf{S}_{ii} + \Lambda_{ii})\mathbf{X}_{ii} &\leq f(\mathbf{X}) - \sum_{i \neq j}(\mathbf{S}_{ij}\mathbf{X}_{ij} + \Lambda_{ij}|\mathbf{X}_{ij}|), \\ &\leq \alpha - \sum_{i\neq j}(\mathbf{S}_{ij}\mathbf{X}_{ij} + \Lambda_{ij}|\mathbf{X}_{ij}|) \leq \delta \end{aligned} \tag{4.8}$$

for some $\delta > 0$. In addition, notice from the assumption that $\mathbf{S}_{ii} + \Lambda_{ii} > 0$ for all i, and hence

$$-\log(\mathbf{X}_{ii}) + (\mathbf{S}_{ii} + \Lambda_{ii})\mathbf{X}_{ii} \geq 1 + \min_k \log(\mathbf{S}_{kk} + \Lambda_{kk}) =: \sigma$$

for all i. This relation together with (4.8) implies that for every $\mathbf{X} \in \mathcal{L}$ and all i,

$$-\log(\mathbf{X}_{ii}) + (\mathbf{S}_{ii} + \Lambda_{ii})\mathbf{X}_{ii} \leq \delta - (p-1)\sigma,$$

and hence $\mathbf{X}_{ii}$ is bounded for all i and $\mathbf{X} \in \mathcal{L}$. We thus conclude that $\mathcal{L}$ is bounded. In view of this result and the definition of f, it is not hard to see that there exists some $\nu > 0$ such that $\lambda_{\min}(\mathbf{X}) \geq \nu$ for all $\mathbf{X} \in \mathcal{L}$. Hence, one has

$$\mathcal{L} = \{\mathbf{X} \succeq \nu I : f(\mathbf{X}) \leq \alpha\}.$$

By the continuity of f on $\{\mathbf{X} : \mathbf{X} \succeq \nu I\}$, it follows that $\mathcal{L}$ is closed. Hence, $\mathcal{L}$ is compact. □

We are now ready to prove Theorem 2.

Proof. Since $\lambda_1 > 0$ and $\operatorname{diag}(\mathbf{S}^{(k)}) > 0, k = 1, \ldots, K$, it follows from Lemma 3 that there exists some δ such that for each $k = 1, \ldots, K$,

$$-\log\det(\boldsymbol{\Theta}^{(k)}) + \operatorname{tr}(\mathbf{S}^{(k)}\boldsymbol{\Theta}^{(k)}) + \lambda_1 \sum_{i \neq j} |\boldsymbol{\Theta}_{ij}^{(k)}| \;\geq\; \delta, \quad \forall \boldsymbol{\Theta}^{(k)} \succ 0.$$

For convenience, let $h(\boldsymbol{\Theta})$ denote the objective function of (4.2) and $\bar{\boldsymbol{\Theta}} = (\bar{\boldsymbol{\Theta}}^{(1)}, \ldots, \bar{\boldsymbol{\Theta}}^{(K)})$ an arbitrary feasible point of (4.2). Let

$$\begin{aligned} \Omega &= \left\{\boldsymbol{\Theta} = (\boldsymbol{\Theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(K)}) : h(\boldsymbol{\Theta}) \leq h(\bar{\boldsymbol{\Theta}}),\; \boldsymbol{\Theta}^{(k)} \succ 0, k = 1, \ldots, K\right\}, \\ \Omega_k &= \left\{\boldsymbol{\Theta}^{(k)} \succ 0 : -\log\det(\boldsymbol{\Theta}^{(k)}) + \operatorname{tr}(\mathbf{S}^{(k)}\boldsymbol{\Theta}^{(k)}) + \lambda_1 \textstyle\sum_{i \neq j} |\boldsymbol{\Theta}_{ij}^{(k)}| \;\leq\; \bar{\delta}\right\} \end{aligned}$$

for $k = 1, \ldots, K$, where $\bar{\delta} = h(\bar{\boldsymbol{\Theta}}) - (K-1)\delta$. Then it is not hard to observe that $\Omega \subseteq \bar{\Omega} := \Omega_1 \times \cdots \times \Omega_K$. Moreover, problem (4.2) is equivalent to

$$\min_{\boldsymbol{\Theta} \in \bar{\Omega}} h(\boldsymbol{\Theta}). \tag{4.9}$$

In view of Lemma 3, we know that Ω_k is compact for all k, which implies that $\bar{\Omega}$ is also compact. Notice that h is continuous and strictly convex on $\bar{\Omega}$. Hence, problem (4.9) has a unique optimal solution and so is problem (4.2). □

4.3 The Screening Rule for Fused Multiple Graphical Lasso

Due to the presence of the log determinant, it is challenging to solve the formulations involving the penalized log-likelihood efficiently. The existing methods for

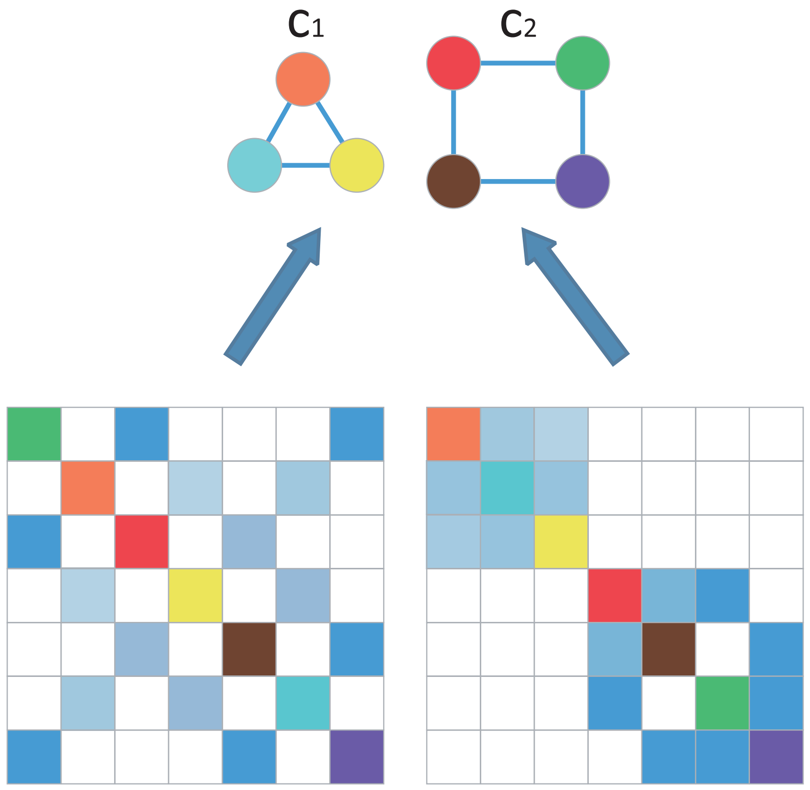


Figure 4.1: Two precision matrices (bottom) whose nodes are in the different order corresponds to the same graph with two connected components (top). The white color in precision matrices represents 0.

single graphical lasso are not scalable to the problems with a large amount of features because of the high computational complexity. Recent studies have shown that the graphical model may contain many connected components, which are disjoint with each other, due to the sparsity of the graphical model, i.e., the corresponding precision matrix has a block diagonal structure (subject to some rearrangement of features, see Figure 4.1 for illustration). To reduce the computational complexity, it is advantageous to first identify the block structure and then compute the diagonal blocks of the precision matrix instead of the whole matrix. Danaher *et al.* (2013) develop a similar necessary and sufficient condition for fused graphical lasso with two graphs, thus the block structure can be identified. However, it remains a challenge to derive the necessary and sufficient condition for the solution of fused multiple graphical lasso to be block diagonal for $K > 2$ graphs.

In this section, we first present a theorem demonstrating that FMGL can be decomposable once its solution has a block diagonal structure. Then we derive a necessary and sufficient condition for the solution of FMGL to be block diagonal for arbitrary number of graphs.

Let $C_1, \ldots, C_L$ be a partition of the p features into L non-overlapping sets, with $C_l \cap C_{l'} = \emptyset,\ \forall l \neq l'$ and $\bigcup_{l=1}^{L} C_l = \{1, \ldots, p\}$. We say that the solution $\widehat{\mathbf{\Theta}}$ of FMGL (4.2) is block diagonal with L known blocks consisting of features in the sets $C_l,\ l = 1, \ldots, L$ if there exists a permutation matrix $\mathbf{U} \in \Re^{p \times p}$ such that each estimation precision matrix takes the form of

$$\widehat{\mathbf{\Theta}}^{(k)} = \mathbf{U} \begin{pmatrix} \widehat{\mathbf{\Theta}}_1^{(k)} & & \\ & \ddots & \\ & & \widehat{\mathbf{\Theta}}_L^{(k)} \end{pmatrix} \mathbf{U}^T,\ k = 1, \ldots, K. \tag{4.10}$$

For simplicity of presentation, we assume throughout this chapter that $\mathbf{U} = \mathbf{I}$.

The following decomposition result for problem (4.2) is straightforward. Its proof is thus omitted.

Theorem 4. *Suppose that the solution $\widehat{\mathbf{\Theta}}$ of FMGL (4.2) is block diagonal with L known $C_l,\ l = 1, \ldots, L$, i.e., each estimated precision matrix has the form (4.10) with $\mathbf{U} = \mathbf{I}$. Let $\widehat{\mathbf{\Theta}}_l = (\widehat{\mathbf{\Theta}}_l^{(1)}, \ldots, \widehat{\mathbf{\Theta}}_l^{(K)})$ for $l = 1, \ldots, L$. Then there holds:*

$$\widehat{\mathbf{\Theta}}_l = \arg\min_{\mathbf{\Theta}_l \succ 0} \sum_{k=1}^{K} \left(-\log\det(\mathbf{\Theta}_l^{(k)}) + \mathrm{tr}(\mathbf{S}_l^{(k)} \mathbf{\Theta}_l^{(k)}) \right) + P(\mathbf{\Theta}_l),\ l = 1, \ldots, L, \tag{4.11}$$

where $\mathbf{\Theta}_l^{(k)}$ and $\mathbf{S}_l^{(k)}$ are the $|C_l| \times |C_l|$ symmetric submatrices of $\mathbf{\Theta}^{(k)}$ and $\mathbf{S}^{(k)}$ corresponding to the l-th diagonal block, respectively, for $k = 1, \ldots, K$, and $\mathbf{\Theta}_l = (\mathbf{\Theta}_l^{(1)}, \ldots, \mathbf{\Theta}_l^{(K)})$ for $l = 1, \ldots, L$.

The above theorem demonstrates that if a large-scale FMGL problem has a block diagonal solution, it can then be decomposed into a group of smaller sized FMGL problems. The computational cost for the latter problems can be much cheaper. Now one natural question is how to efficiently identify the block diagonal structure of the FMGL solution before solving the problem. We address this question in the remaining part of this section.

The following theorem provides a necessary and sufficient condition for the solution of FMGL to be block diagonal with L blocks $C_l, l = 1, \ldots, L$, which is a key for developing efficient decomposition scheme for solving FMGL. Since its proof requires some substantial development of other technical results, we shall postpone the proof until the end of this section.

Theorem 5. *The FMGL (4.2) has a block diagonal solution $\widehat{\mathbf{\Theta}}^{(k)}, k = 1, \ldots, K$ with L known blocks $C_l, l = 1, \ldots, L$ if and only if $\mathbf{S}^{(k)}, k = 1, \ldots, K$ satisfy the following inequalities:*

$$\begin{cases} |\sum_{k=1}^{t} \mathbf{S}_{ij}^{(k)}| \leq t\lambda_1 + \lambda_2, \\ |\sum_{k=0}^{t-1} \mathbf{S}_{ij}^{(r+k)}| \leq t\lambda_1 + 2\lambda_2,\ 2 \leq r \leq K - t, \\ |\sum_{k=1}^{t} \mathbf{S}_{ij}^{(K-t+k)}| \leq t\lambda_1 + \lambda_2, \\ |\sum_{k=1}^{K} \mathbf{S}_{ij}^{(k)}| \leq K\lambda_1 \end{cases} \tag{4.12}$$

for $t = 1, \ldots, K-1$, $i \in C_l, j \in C_{l'}, l \neq l'$.

One immediate consequence of Theorem 5 is that the conditions (4.12) can be used as a screening rule to identify the block diagonal structure of the FMGL solution. The steps about this rule are described as follows.

1. Construct an adjacency matrix $\mathbf{E} = \mathbf{I}_{p \times p}$. Set $\mathbf{E}_{ij} = \mathbf{E}_{ji} = 0$ if $\mathbf{S}_{ij}^{(k)}, k = 1, \ldots, K$ satisfy the conditions (4.12). Otherwise, set $\mathbf{E}_{ij} = \mathbf{E}_{ji} = 1$.

2. Identify the connected components of the adjacency matrix $\mathbf{E}$ (for example, it can be done by calling the Matlab function "graphconncomp").

In view of Theorem 5, it is not hard to observe that the resulting connected components are the partition of the p features into nonoverlapping sets. It then follows from Theorem 4 that a large-scale FMGL problem can be decomposed into a group of smaller sized FMGL problems restricted to the features in each connected component. The computational cost for the latter problems can be much cheaper. Therefore, this approach may enable us to solve large-scale FMGL problems very efficiently.

In the remainder of this section we provide a proof for Theorem 5. Before proceeding, we establish several technical lemmas as follows.

Lemma 6. *Given any two arbitrary index sets $I \subseteq \{1, \cdots, n\}$ and $J \subseteq \{1, \cdots, n-1\}$, let $\bar{I}$ and $\bar{J}$ be the complement of I and J with respect to $\{1, \cdots, n\}$ and $\{1, \cdots, n-1\}$, respectively. Define*

$$P_{I,J} = \{y \in \Re^n : y_I \geq 0,\ y_{\bar{I}} \leq 0,\ y_J - y_{J+1} \geq 0,\ y_{\bar{J}} - y_{\bar{J}+1} \leq 0\}, \tag{4.13}$$

where $J+1 = \{j+1 : j \in J\}$ and $\bar{J}+1 = \{j+1 : j \in \bar{J}\}$. Then, the following statements hold:

(i) Either $P_{I,J} = \{0\}$ or $P_{I,J}$ is unbounded;

(ii) 0 is the unique extreme point of $P_{I,J}$;

(iii) Suppose that $P_{I,J}$ is unbounded. Then, $\emptyset \neq \mathrm{ext}(P_{I,J}) \subseteq Q$, where $\mathrm{ext}(P_{I,J})$ denotes the set of all extreme rays of $P_{I,J}$, and

$$Q := \{\alpha(\underbrace{0, \cdots, 0}_{m}, \underbrace{1, \cdots, 1}_{l}, 0, \cdots, 0)^T \in \Re^n : \alpha \neq 0, m \geq 0, 1 \leq l \leq n\}. \tag{4.14}$$

Proof. (i) We observe that $0 \in P_{I,J}$. If $P_{I,J} \neq \{0\}$, then there exists $0 \neq y \in P_{I,J}$. Hence, $\{\alpha y : \alpha \geq 0\} \subseteq P_{I,J}$, which implies that $P_{I,J}$ is unbounded.

(ii) It is easy to see that $0 \in P_{I,J}$ and moreover there exist n linearly independent active inequalities at 0. Hence, 0 is an extreme point of $P_{I,J}$. On the other hand, suppose y is an arbitrary extreme point of $P_{I,J}$. Then there exist n linearly independent active inequalities at y, which together with the definition of $P_{I,J}$ immediately implies $y = 0$. Therefore, 0 is the unique extreme point of $P_{I,J}$.

(iii) Suppose that $P_{I,J}$ is unbounded. By statement (ii), we know that $P_{I,J}$ has a unique extreme point. Using Minkowski's resolution theorem (e.g., see Bertsekas and Tsitsiklis (1997)), we conclude that $\text{ext}(P_{I,J}) \neq \emptyset$. Let $d \in \text{ext}(P_{I,J})$ be arbitrarily chosen. Then $d \neq 0$. It follows from (4.13) that d satisfies the inequalities

$$d_I \geq 0, \; d_{\bar{I}} \leq 0, \; d_J - d_{J+1} \geq 0, \; d_{\bar{J}} - d_{\bar{J}+1} \leq 0, \tag{4.15}$$

and moreover, the number of independent active inequalities at d is $n-1$. If all entries of d are nonzero, then d must satisfy $d_J - d_{J+1} = 0$ and $d_{\bar{J}} - d_{\bar{J}+1} = 0$ (with a total number $n-1$), which implies $d_1 = d_2 = \cdots = d_n$ and thus $d \in Q$. We now assume that d has at least one zero entry. Then, there exist positive integers k, $\{m_i\}_{i=1}^k$ and $\{n_i\}_{i=1}^k$ satisfying $m_i \leq n_i < m_{i+1} \leq n_{i+1}$ for $i = 1, \ldots, k-1$ such that

$$\{i : d_i = 0\} = \{m_1, \cdots, n_1\} \cup \{m_2, \cdots, n_2\} \cup \cdots \cup \{m_k, \cdots, n_k\}. \tag{4.16}$$

One can immediately observe that

$$d_{m_i} = \cdots = d_{n_i} = 0, \quad d_j - d_{j+1} = 0, \quad m_i \leq j \leq n_i - 1, \; 1 \leq i \leq k. \tag{4.17}$$

We next divide the rest of proof into four cases.

Case (a): $m_1 = 1$ and $n_k = n$. In view of (4.16), one can observe that $d_{m_i - 1} - d_{m_i} \neq 0$ and $d_{n_{i-1}} - d_{n_{i-1}+1} \neq 0$ for $i = 2, \ldots, k$. We then see from (4.15) that except the active inequalities given in (4.17), all other possible active inequalities at d are

$$d_j - d_{j+1} = 0, \quad n_{i-1} < j < m_i - 1,\ 2 \le i \le k \tag{4.18}$$

(with a total number $\sum_{i=2}^k (m_i - n_{i-1} - 2)$). Notice that the total number of independent active inequalities given in (4.17) is $\sum_{i=1}^k (n_i - m_i + 1)$. Hence, the number of independent active inequalities at d is at most

$$\sum_{i=1}^k (n_i - m_i + 1) + \sum_{i=2}^k (m_i - n_{i-1} - 2) = n_k - m_1 - k + 2 = n - k + 1.$$

Recall that the number of independent active inequalities at d is $n-1$. Hence, we have $n - k + 1 \ge n - 1$, which implies $k \le 2$. Due to $d \ne 0$, we observe that $k \ne 1$ holds for this case. Also, we know that $k > 0$. Hence, $k = 2$. We then see that all possible active inequalities described in (4.18) must be active at d, which together with $k = 2$ immediately implies that $d \in Q$.

Case (b): $m_1 = 1$ and $n_k < n$. Using (4.16), we observe that $d_{m_i - 1} - d_{m_i} \ne 0$ for $i = 2, \ldots, k$ and $d_{n_i} - d_{n_i+1} \ne 0$ for $i = 1, \ldots, k$. In view of these relations and a similar argument as in case (a), one can see that the number of independent active inequalities at d is at most

$$\sum_{i=1}^k (n_i - m_i + 1) + \sum_{i=2}^k (m_i - n_{i-1} - 2) + n - n_k - 1 = n - m_1 - k + 1 = n - k.$$

Similarly as in case (a), we can conclude from the above relation that $k = 1$ and $d \in Q$.

Case (c): $m_1 > 1$ and $n_k = n$. By (4.16), one can observe that $d_{m_i-1} - d_{m_i} \ne 0$ for $i = 1, \ldots, k$ and $d_{n_i} - d_{n_i+1} \ne 0$ for $i = 1, \ldots, k-1$. Using these relations and a similar argument as in case (a), we see that the number of independent active inequalities at d is at most

$$m_1 - 2 + \sum_{i=1}^k (n_i - m_i + 1) + \sum_{i=2}^k (m_i - n_{i-1} - 2) = n_k - k = n - k.$$

Similarly as in case (a), we can conclude from the above relation that $k = 1$ and $d \in Q$.

Case (d): $m_1 > 1$ and $n_k < n$. From (4.16), one can observe that $d_{m_i-1} - d_{m_i} \neq 0$ for $i = 1, \dots, k$ and $d_{n_i} - d_{n_i+1} \neq 0$ for $i = 1, \dots, k$. By virtue of these relations and a similar argument as in case (a), one can see that the number of independent active inequalities at d is at most

$$m_1 - 2 + \sum_{i=1}^{k}(n_i - m_i + 1) + \sum_{i=2}^{k}(m_i - n_{i-1} - 2) + n - n_k - 1 = n - k - 1.$$

Recall that $k \geq 1$ and the number of independent active inequalities at d is $n - 1$. Hence, this case cannot occur.

Combining the above four cases, we conclude that $\text{ext}(P_{I,J}) \subseteq Q$. □

Lemma 7. *Let P_{IJ} and Q be defined in (4.13) and (4.14), respectively. Then,*

$$\cup\{\text{ext}(P_{I,J}) : I \subseteq \{1, \cdots, n\},\ J \subseteq \{1, \cdots, n-1\}\} = Q.$$

Proof. It follows from Lemma 6 (iii) that

$$\cup\{\text{ext}(P_{I,J}) : I \subseteq \{1, \cdots, n\},\ J \subseteq \{1, \cdots, n-1\}\} \ \subseteq \ Q.$$

We next show that

$$\cup\{\text{ext}(P_{I,J}) : I \subseteq \{1, \cdots, n\},\ J \subseteq \{1, \cdots, n-1\}\} \ \supseteq \ Q.$$

Indeed, let $d \in Q$ be arbitrarily chosen. Then, there exist $\alpha \neq 0$ and positive integers m_1 and n_1 satisfying $1 \leq m_1 \leq n_1$ such that $d_i = \alpha$ for $m_1 \leq i \leq n_1$ and the rest of d_i's are 0. If $\alpha > 0$, it is not hard to see that $d \in \text{ext}(P_{I,J})$ with $I = \{1, \cdots, n\}$ and $J = \{m_1, \cdots, n-1\}$. Similarly, if $\alpha < 0$, $d \in \text{ext}(P_{I,J})$ with $I = \emptyset$ and J being the complement of $\bar{J} = \{m_1, \cdots, n-1\}$. Hence, $d \in \cup\{\text{ext}(P_{I,J}) : I \subseteq \{1, \cdots, n\},\ J \subseteq \{1, \cdots, n-1\}\}$. □

Lemma 8. *Let $x \in \Re^n$, λ_1, $\lambda_2 \geq 0$ be given, and let*

$$f(y) := x^T y - \lambda_1 \sum_{i=1}^{n} |y_i| - \lambda_2 \sum_{i=1}^{n-1} |y_i - y_{i+1}|.$$

Then, $f(y) \leq 0$ for all $y \in \Re^n$ if and only if x satisfies the following inequalities:

$$\begin{cases} |\sum_{j=1}^{k} x_j| \leq k\lambda_1 + \lambda_2, \\ |\sum_{j=0}^{k-1} x_{i+j}| \leq k\lambda_1 + 2\lambda_2, \ 2 \leq i \leq n-k, \\ |\sum_{j=1}^{k} x_{n-k+j}| \leq k\lambda_1 + \lambda_2, \\ |\sum_{j=1}^{n} x_j| \leq n\lambda_1 \end{cases}$$

for $k = 1, \ldots, n-1$.

Proof. Let $P_{I,J}$ be defined in (4.13) for any $I \subseteq \{1, \ldots, n\}$ and $J \subseteq \{1, \ldots, n-1\}$. We observe that

(a) $\Re^n = \cup \{P_{I,J} : \ I \subseteq \{1, \ldots, n\}, \ J \subseteq \{1, \ldots, n-1\}\}$;

(b) $f(y) \leq 0$ for all $y \in \Re^n$ if and only if $f(y) \leq 0$ for all $y \in P_{I,J}$, and every $I \subseteq \{1, \ldots, n\}$ and $J \subseteq \{1, \ldots, n-1\}$;

(c) $f(y)$ is a linear function of y when restricted to the set $P_{I,J}$ for every $I \subseteq \{1, \ldots, n\}$ and $J \subseteq \{1, \ldots, n-1\}$.

If $P_{I,J}$ is bounded, we have $P_{I,J} = \{0\}$ and $f(y) = 0$ for $y \in P_{I,J}$. Suppose that $P_{I,J}$ is unbounded. By Lemma 6 and Minkowski's resolution theorem, $P_{I,J}$ equals the finitely generated cone by $\text{ext}(P_{I,J})$. It then follows that $f(y) \leq 0$ for all $y \in P_{I,J}$ if and only if $f(d) \leq 0$ for all $d \in \text{ext}(P_{I,J})$. Using these facts and Lemma 7, we see that $f(y) \leq 0$ for all $y \in \Re^n$ if and only if $f(d) \leq 0$ for all $d \in Q$, where Q is defined in (4.14). By the definitions of Q and f, we further observe that $f(y) \leq 0$ for all $y \in \Re^n$ if and only if $f(d) \leq 0$ for all

$$d \in \left\{ \pm(\underbrace{0, \cdots, 0}_{m}, \underbrace{1, \cdots, 1}_{l}, 0, \cdots, 0)^T \in \Re^n : m \geq 0, 1 \leq l \leq n \right\},$$

which together with the definition of f immediately implies that the conclusion of this lemma holds. □

Lemma 9. *Let $x \in \Re^n$, λ_1, $\lambda_2 \geq 0$ be given. The linear system*

$$\begin{cases} x_1 + \lambda_1\gamma_1 + \lambda_2 v_1 = 0, \\ x_i + \lambda_1\gamma_i + \lambda_2(v_i - v_{i-1}) = 0,\ 2 \leq i \leq n-1, \\ x_n + \lambda_1\gamma_n - \lambda_2 v_{n-1} = 0, \\ -1 \leq \gamma_i \leq 1,\ i = 1,\ldots,n, \\ -1 \leq v_i \leq 1,\ i = 1,\ldots,n-1 \end{cases} \tag{4.19}$$

has a solution (γ, v) if and only if $(x, \lambda_1, \lambda_2)$ satisfies the following inequalities:

$$\begin{cases} |\sum_{j=1}^{k} x_j| \leq k\lambda_1 + \lambda_2, \\ |\sum_{j=0}^{k-1} x_{i+j}| \leq k\lambda_1 + 2\lambda_2,\ 2 \leq i \leq n-k, \\ |\sum_{j=1}^{k} x_{n-k+j}| \leq k\lambda_1 + \lambda_2, \\ |\sum_{j=1}^{n} x_j| \leq n\lambda_1 \end{cases}$$

for $k = 1,\ldots,n-1$.

Proof. The linear system (4.19) has a solution if and only if the linear programming

$$\min_{\gamma,v}\{0^T\gamma + 0^T v : (\gamma, v) \text{ satisfies (4.19)}\} \tag{4.20}$$

has an optimal solution. The Lagrangian dual of (4.20) is

$$\max_y \min_{\gamma,v} \left\{ x^T y + \lambda_1 \sum_{i=1}^{n} y_i\gamma_i + \lambda_2 \sum_{i=1}^{n-1}(y_i - y_{i+1})v_i :\ -1 \leq \gamma, v \leq 1 \right\},$$

which is equivalent to

$$\max_y f(y) := x^T y - \lambda_1 \sum_{i=1}^{n} |y_i| - \lambda_2 \sum_{i=1}^{n-1} |y_i - y_{i+1}|. \tag{4.21}$$

By the Lagrangian duality theory, problem (4.20) has an optimal solution if and only if its dual problem (4.21) has optimal value 0, which is equivalent to $f(y) \leq 0$ for all $y \in \Re^n$. The conclusion of this lemma then immediately follows from Lemma 8. $\square$

We are now ready to prove Theorem 5.

Proof. For the sake of convenience, we denote the inverse of $\widehat{\mathbf{\Theta}}^{(k)}$ as $\widehat{\mathbf{W}}^{(k)}$ for $k = 1, \ldots, K$. By the first-order optimality conditions, we observe that $\widehat{\mathbf{\Theta}}^{(k)} \succ 0, k = 1, \ldots, K$ is the optimal solution of problem (4.2) if and only if it satisfies

$$-\widehat{\mathbf{W}}_{ii}^{(k)} + \mathbf{S}_{ii}^{(k)} = 0,\ 1 \leq k \leq K, \tag{4.22}$$

$$-\widehat{\mathbf{W}}_{ij}^{(1)} + \mathbf{S}_{ij}^{(1)} + \lambda_1 \gamma_{ij}^{(1)} + \lambda_2 \upsilon_{ij}^{(1,2)} = 0, \tag{4.23}$$

$$-\widehat{\mathbf{W}}_{ij}^{(k)} + \mathbf{S}_{ij}^{(k)} + \lambda_1 \gamma_{ij}^{(k)} + \lambda_2 (-\upsilon_{ij}^{(k-1,k)} + \upsilon_{ij}^{(k,k+1)}) = 0,\ 2 \leq k \leq K-1, \tag{4.24}$$

$$-\widehat{\mathbf{W}}_{ij}^{(K)} + \mathbf{S}_{ij}^{(K)} + \lambda_1 \gamma_{ij}^{(K)} - \lambda_2 \upsilon_{ij}^{(K-1,K)} = 0 \tag{4.25}$$

for all $i, j = 1, \ldots, p, i \neq j$, where $\gamma_{ij}^{(k)}$ is a subgradient of $|\mathbf{\Theta}_{ij}^{(k)}|$ at $\mathbf{\Theta}_{ij}^{(k)} = \widehat{\mathbf{\Theta}}_{ij}^{(k)}$; and $\upsilon_{ij}^{(k,k+1)}$ is a subgradient of $|\mathbf{\Theta}_{ij}^{(k)} - \mathbf{\Theta}_{ij}^{(k+1)}|$ with respect to $\mathbf{\Theta}_{ij}^{(k)}$ at $(\mathbf{\Theta}_{ij}^{(k)}, \mathbf{\Theta}_{ij}^{(k+1)}) = (\widehat{\mathbf{\Theta}}_{ij}^{(k)}, \widehat{\mathbf{\Theta}}_{ij}^{(k+1)})$, that is, $\upsilon_{ij}^{(k,k+1)} = 1$ if $\widehat{\mathbf{\Theta}}_{ij}^{(k)} > \widehat{\mathbf{\Theta}}_{ij}^{(k+1)}$, $\upsilon_{ij}^{(k,k+1)} = -1$ if $\widehat{\mathbf{\Theta}}_{ij}^{(k)} < \widehat{\mathbf{\Theta}}_{ij}^{(k+1)}$, and $\upsilon_{ij}^{(k,k+1)} \in [-1, 1]$ if $\widehat{\mathbf{\Theta}}_{ij}^{(k)} = \widehat{\mathbf{\Theta}}_{ij}^{(k+1)}$.

Necessity: Suppose that $\widehat{\mathbf{\Theta}}^{(k)}, k = 1, \ldots, K$ is a block diagonal optimal solution of problem (4.2) with L known blocks $C_l, l = 1, \ldots, L$. Note that $\widehat{\mathbf{W}}^{(k)}$ has the same block diagonal structure as $\widehat{\mathbf{\Theta}}^{(k)}$. Hence, $\widehat{\mathbf{W}}_{ij}^{(k)} = \widehat{\mathbf{\Theta}}_{ij}^{(k)} = 0$ for $i \in C_l, j \in C_{l'}, l \neq l'$. This together with (4.23)-(4.25) implies that for each $i \in C_l, j \in C_{l'}, l \neq l'$, there exist

$(\gamma_{ij}^{(k)}, \upsilon_{ij}^{(k,k+1)}), k = 1, \ldots, K-1$ and $\gamma_{ij}^{(K)}$ such that

$$\begin{aligned}
&\mathbf{S}_{ij}^{(1)} + \lambda_1 \gamma_{ij}^{(1)} + \lambda_2 \upsilon_{ij}^{(1,2)} = 0, \\
&\mathbf{S}_{ij}^{(k)} + \lambda_1 \gamma_{ij}^{(k)} + \lambda_2(-\upsilon_{ij}^{(k-1,k)} + \upsilon_{ij}^{(k,k+1)}) = 0, \ 2 \le k \le K-1, \\
&\mathbf{S}_{ij}^{(K)} + \lambda_1 \gamma_{ij}^{(K)} - \lambda_2 \upsilon_{ij}^{(K-1,K)} = 0, \qquad (4.26) \\
&-1 \le \gamma_{ij}^{(k)} \le 1, \ 1 \le k \le K, \\
&-1 \le \upsilon_{ij}^{(k,k+1)} \le 1, \ \ 1 \le k \le K-1.
\end{aligned}$$

Using (4.26) and Lemma 9, we see that (4.12) holds for $t = 1, \ldots, K-1$, $i \in C_l, j \in C_{l'}, l \neq l'$.

Sufficiency: Suppose that (4.12) holds for $t = 1, \ldots, K-1$, $i \in C_l, j \in C_{l'}, l \neq l'$. It then follows from Lemma 9 that for each $i \in C_l, j \in C_{l'}, l \neq l'$, there exist $(\gamma_{ij}^{(k)}, \upsilon_{ij}^{(k,k+1)}), k = 1, \ldots, K-1$ and $\gamma_{ij}^{(K)}$ such that (4.26) holds. Now let $\widehat{\mathbf{\Theta}}^{(k)}, k = 1, \ldots, K$ be a block diagonal matrix as defined in (4.10) with $\mathbf{U} = \mathbf{I}$, where $\widehat{\mathbf{\Theta}}_l = (\widehat{\mathbf{\Theta}}_l^{(1)}, \ldots, \widehat{\mathbf{\Theta}}_l^{(K)})$ is given by (4.11) for $l = 1, \ldots, L$. Also, let $\widehat{\mathbf{W}}^{(k)}$ be the inverse of $\widehat{\mathbf{\Theta}}^{(k)}$ for $k = 1, \ldots, K$. Since $\widehat{\mathbf{\Theta}}_l$ is the optimal solution of problem (4.11), the first-order optimality conditions imply that (4.22)-(4.25) hold for all $i, j \in C_l, i \neq j, l = 1, \ldots, L$. Notice that $\widehat{\mathbf{\Theta}}_{ij}^{(k)} = \widehat{\mathbf{W}}_{ij}^{(k)} = 0$ for every $i \in C_l, j \in C_{l'}, l \neq l'$. Using this fact and (4.26), we observe that (4.22)-(4.25) also hold for all $i \in C_l, j \in C_{l'}, l \neq l'$. It then follows that $\widehat{\mathbf{\Theta}}^{(k)}, k = 1, \ldots, K$ is an optimal solution of problem (4.2). In addition, $\widehat{\mathbf{\Theta}}^{(k)}, k = 1, \ldots, K$ is block diagonal with L known blocks $C_l, l = 1, \ldots, L$. The conclusion thus holds. □

4.3.1 Extension to Other Regularizations

We show how to establish a similar necessary and sufficient condition for general fused regularization (i.e., graph fused regularization). Denote $G = (V, E)$ as an undirected graph, where the nodes are $V = \{1, \ldots, K\}$ and E is a set of edges.

Assume there is no redundancy in E (i.e., if $(u,v) \in E$, $(v,u) \notin E$). Then we define the graph fused regularization by

$$P(\mathbf{\Theta}) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\mathbf{\Theta}_{ij}^{(k)}| + \lambda_2 \sum_{i \neq j} \sum_{(u,v) \in E} |\mathbf{\Theta}_{ij}^{(u)} - \mathbf{\Theta}_{ij}^{(v)}|. \tag{4.27}$$

Clearly, the sequential fused and pairwise fused regularization are special cases of the graph fused regularization. The graph fused regularization is decomposable based on the connected components of the given graph G. Without loss of generality, we assume that G has only *one connected component*, which means that there exists an edge across any two set partition of V. The technique used in the sequential fused case can be extended to the case of graph fused regularization. The key is to prove the results similar to those in Lemma 6 and Lemma 7 for graph fused regularization.

Denote $\mathcal{G} = \{G_1, G_2, \ldots, G_M\}$ as the set of subgraphs in graph G such that each subgraph G_m has only one connected component. For example, a fully connected graph with 3 nodes has 7 such subgraphs. According to the assumption that G has only one connected component, we have $G \in \mathcal{G}$. Let $\mathcal{V} = \{V_1, V_2, \ldots, V_M\}$ where V_m represents the nodes of subgraph G_m. Then we have the following results:

Lemma 10. *Given an undirected graph $G = (V, E)$, where the nodes are $V = \{1, \ldots, n\}$ and E is a set of edges of size $|E|$. Given any two arbitrary index sets $I \subseteq \{1, \cdots, n\}$, $J \subseteq \{1, \cdots, |E|\}$, let $\bar{I}$ and $\bar{J}$ be the complement of I and J with respect to $\{1, \cdots, n\}$ and $\{1, \cdots, |E|\}$, respectively. Define*

$$\begin{aligned} P_{I,J} = \{y \in \Re^n : y_I \geq 0,\ y_{\bar{I}} \leq 0,\ y_u - y_v \geq 0, \forall (u,v) \in E_J, \\ y_u - y_v \leq 0, \forall (u,v) \in E_{\bar{J}}\}, \end{aligned} \tag{4.28}$$

where E_J and $E_{\bar{J}}$ denote the sets of edges whose indexes are in J and $\bar{J}$, respectively. Then, the following statements hold:

(i) Either $P_{I,J} = \{0\}$ or $P_{I,J}$ is unbounded;

(ii) 0 *is the unique extreme point of* $P_{I,J}$*;*

(iii) Suppose that $P_{I,J}$ *is unbounded. Then,* $\emptyset \neq \operatorname{ext}(P_{I,J}) \subseteq Q$*, where*

$$Q := \left\{ \alpha d \in \Re^n : \alpha \neq 0, d_i = \left\{ \begin{array}{ll} 1, & i \in V_m \\ 0, & i \notin V_m \end{array} \right. , \forall V_m \in \mathcal{V} \right\}. \tag{4.29}$$

(iv) $\cup \{\operatorname{ext}(P_{I,J}) : I \subseteq \{1, \cdots, n\}, J \subseteq \{1, \cdots, |E|\}\} = Q.$

Proof. (i) and (ii) can be proved in a similar way to Lemma 6.

(iii) Similar to Lemma 6, we can show that $\operatorname{ext}(P_{I,J}) \neq \emptyset$. Next we show that $\cup \{\operatorname{ext}(P_{I,J}) :\ I \subseteq \{1, \cdots, n\}, J \subseteq \{1, \cdots, |E|\}\} \subseteq Q$. Denote G_J and $G_{\bar{J}}$ as the subgraphs with edges only in E_J and $E_{\bar{J}}$ respectively. Accordingly, $\mathcal{G}_J$ represents the set of all possible subgraphs with only one connected component in G_J, and $\mathcal{V}_J$ denotes the corresponding node sets of $\mathcal{G}_J$. Then we have $\mathcal{V}_J \cup \mathcal{V}_{\bar{J}} \subseteq \mathcal{V}$. Moreover, $\cup \{\mathcal{V}_J \cup \mathcal{V}_{\bar{J}}, J \subseteq \{1, \ldots, |E|\}\} = \mathcal{V}$.

Let $d \in \cup \{\operatorname{ext}(P_{I,J}) : I \subseteq \{1, \cdots, n\}, J \subseteq \{1, \ldots, |E|\}\}$. Then $d \neq 0$ and the number of independent active inequalities at d is $n-1$. It is clear that the maximum number of independent active inequalities restricted to the nodes in $V_m \in \mathcal{V}$ is $|V_m|$ which is achieved when $d_i = 0, \forall i \in V_m$. If $d_i \neq 0, \forall i \in V_m, V_m \neq \emptyset$, it is not hard to show that the maximum number of independent active inequalities restricted to V_m is $|V_m| - 1$ which is achieved when $d_i = d_j, \forall i, j \in V_m$. Suppose there exist two nonempty and nonoverlapping sets V_l and V_m such that $d_i = d_j \neq 0, \forall i, j \in V_l$ and $d_i = d_j \neq 0, \forall i, j \in V_m$. We consider the following two cases: (a) there is no edge across V_l and V_m. In this case, the maximum number of independent active inequalities is $|V_m| - 1 + |V_l| - 1 + n - |V_m| - |V_l| = n-2$; (b) $d_i \neq d_j, i \in V_l, j \in V_m$, thus inequalities from the edges across V_l and V_m are inactive. In this case, the maximum number of independent active inequalities is $|V_m| - 1 + |V_l| - 1 + n - |V_m| - |V_l| = n-2$. This is a contradiction to the definition of extreme ray d. Combining the arguments above, we show that all nodes in V with a nonzero value in d form a set in $\mathcal{V}$. Therefore, $\cup\{\text{ext}(P_{I,J}) : I \subseteq \{1, \cdots, n\}, J \subseteq \{1, \ldots, |E|\}\} \subseteq Q$.

(iv) Let $d \in Q$ be arbitrarily chosen. Then, there exist $\alpha \neq 0$ and a $V_m \in \mathcal{V}$ such that $d_i = \alpha, i \in V_m$ and the rest of $d_i's$ are 0. If $\alpha > 0$, it is not hard to see that $d \in \text{ext}(P_{I,J})$ with $I = \{1, \ldots, n\}$ and J such that $E_J = \{(u,v) : u, v \in V_m,\ (u,v) \in E\} \cup \{(u,v) : u \in V_m, v \in \bar{V}_m, (u,v) \in E\}$, where $\bar{V}_m$ is the complement of V_m. If $\alpha < 0$, $d \in \text{ext}(P_{I,J})$ with $I = \emptyset$ and J such that $E_J = \{(u,v) : u, v \in V_m,\ (u,v) \in E\} \cup \{(u,v) : u \in \bar{V}_m, v \in V_m, (u,v) \in E\}$. Hence, $d \in \cup\{\text{ext}(P_{I,J}) : I \subseteq \{1, \cdots, n\}, J \subseteq \{1, \ldots, |E|\}\}$. Combined with (iii), we have $\cup\{\text{ext}(P_{I,J}) : I \subseteq \{1, \cdots, n\}, J \subseteq \{1, \ldots, |E|\}\} = Q$. □

After we obtain the set of all extreme rays, the remaining steps can be proved in the same manner as in the fused case. Let $|E_{\setminus V_m}|$ be the number of edges across V_m and its complement, and let $|V_m|$ be the number of nodes in V_m. Then the necessary and sufficient condition for graph fused regularization is

$$\left| \sum_{k=1}^{|V_m|} \mathbf{S}_{ij}^{(u_k)} \right| \leq |V_m|\lambda_1 + |E_{\setminus V_m}|\lambda_2, u_k \in V_m, \forall V_m \in \mathcal{V}. \tag{4.30}$$

The complexity of verifying the necessary and sufficient condition for an arbitrary graph is exponential due to all possible subgraphs with only one connected component. Exploring the structure of the given graph may reduce redundancy of the conditions (4.30). We defer to future work.

4.3.2 Screening Rule for General Structured Multiple Graphical Lasso

We consider the following general structured multiple graphical lasso (SMGL):

$$\min_{\mathbf{\Theta}^{(k)} \succ 0, k=1\ldots K} \sum_{k=1}^{K} \left(-\log\det(\mathbf{\Theta}^{(k)}) + \text{tr}(\mathbf{S}^{(k)}\mathbf{\Theta}^{(k)}) \right) + \sum_{i \neq j} \phi(\mathbf{\Theta}_{ij}), \tag{4.31}$$

where $\mathbf{\Theta}_{ij} = (\mathbf{\Theta}_{ij}^{(1)}, \ldots, \mathbf{\Theta}_{ij}^{(K)})^T \in \Re^K$, and $\phi(x)$ is a convex regularization that encourages estimated graph models to have a certain structure. Besides fused and graph regularizations, there are other examples including but not limited to

- Overlapping group regularization:

$$\phi(x) = \lambda_1 \|x\|_1 + \lambda_2 \sum_{i=1}^{g} \|x_{G_i}\|_2,$$

  where $G_i, i = 1, \ldots, g$ are g groups such that $\bigcup_{i=1}^{g} G_i = \{1, \ldots, K\}$. Different groups may overlap.

- Tree structured group regularization:

$$\phi(x) = \sum_{i=1}^{d} \sum_{j=1}^{n_i} w_j^i \|x_{G_j^i}\|_2,$$

  where w_j^i is a positive weight, and the groups $G_j^i, j = 1, \ldots, n_i,\ i = 1, \ldots, d$ exhibit a tree structure (Liu and Ye, 2010).

Theorem 11. *The SMGL (4.31) has a block diagonal solution $\widehat{\mathbf{\Theta}}^{(k)}, k = 1, \ldots, K$ with L blocks $C_l, l = 1, \ldots, L$ if and only if 0 is the optimal solution of the following problem:*

$$\min_x \frac{1}{2} \|x + \mathbf{S}_{ij}\|_2^2 + \phi(x) \tag{4.32}$$

for $i \in C_l, j \in C_{l'}, l \neq l'$.

Proof. By the first-order optimality conditions, $\widehat{\mathbf{\Theta}}^{(k)} \succ 0, k = 1, \ldots, K$ is the optimal solution of problem (4.2) if and only if it satisfies

$$-\widehat{\mathbf{W}}_{ii}^{(k)} + \mathbf{S}_{ii}^{(k)} = 0,\ 1 \leq k \leq K, \tag{4.33}$$

$$-\widehat{\mathbf{W}}_{ij} + \mathbf{S}_{ij} + \partial \phi_{ij} = 0, \tag{4.34}$$

for all $i, j = 1, \ldots, p, i \neq j$, where $\widehat{\mathbf{W}}_{ij} = (\widehat{\mathbf{W}}_{ij}^{(1)}, \ldots, \widehat{\mathbf{W}}_{ij}^{(K)})^T$, $\mathbf{S}_{ij} = (\mathbf{S}_{ij}^{(1)}, \ldots, \mathbf{S}_{ij}^{(K)})^T$, and $\partial \phi_{ij}$ is a subgradient of $\phi(\mathbf{\Theta}_{ij})$ at $\mathbf{\Theta}_{ij} = \widehat{\mathbf{\Theta}}_{ij}$.

Suppose that $\widehat{\mathbf{\Theta}}^{(k)}, k = 1, \ldots, K$ is a block diagonal optimal solution of problem (4.2) with L known blocks $C_l, l = 1, \ldots, L$. $\widehat{\mathbf{W}}_{ij}^{(k)} = \widehat{\mathbf{\Theta}}_{ij}^{(k)} = 0$ for $i \in C_l, j \in C_{l'}, l \neq l'$. This together with (4.34) implies that for each $i \in C_l, j \in C_{l'}, l \neq l'$, there exists a $\partial\phi_{ij}$ such that

$$\mathbf{S}_{ij} + \partial\phi_{ij} = 0,$$

which directly shows that 0 is the optimal solution of (4.32). The sufficiency can be proved in a similar way to Theorem 5. □

Theorem 11 can be used as a screening rule for SMGL. If (4.32) has a closed form solution as in the case of tree structured group regularization (Liu and Ye, 2010), the screening rule results in an exact block diagonal structure. However, if (4.32) does not have a closed form solution, the screening rule may not identify an exact block diagonal structure due to numerical error. Although the identified structure may be inexact, it can still be used to find a good initial solution as shown in Hsieh *et al.* (2012). An interesting future direction is to study the error bound between the identified and exact block diagonal structures.

4.4 Second-order Method

The screening rule proposed in Section 4.3 is capable of partitioning all features into a group of smaller sized blocks. Accordingly, a large-scale FMGL (4.2) can be decomposed into a number of smaller sized FMGL problems. For each block l, we need to compute its individual precision matrix $\mathbf{\Theta}_l^{(k)}$ by solving the FMGL (4.2) with $\mathbf{S}^{(k)}$ replaced by $\mathbf{S}_l^{(k)}$. In this section, we show how to solve those single block FMGL problems efficiently. For simplicity of presentation, we assume throughout this section that the FMGL (4.2) has only one block, that is, $L = 1$.

We now propose a second-order method to solve the FMGL (4.2). For simplicity of notation, we let $\mathbf{\Theta} := (\mathbf{\Theta}^{(1)}, \ldots, \mathbf{\Theta}^{(K)})$ and use t to denote the Newton iteration index. Let $\mathbf{\Theta}_t = (\mathbf{\Theta}_t^{(1)}, \ldots, \mathbf{\Theta}_t^{(K)})$ be the approximate solution obtained at the t-th Newton iteration.

The optimization problem (4.2) can be rewritten as

$$\min_{\mathbf{\Theta} \succ 0} F(\mathbf{\Theta}) := \sum_{k=1}^{K} f_k(\mathbf{\Theta}^{(k)}) + P(\mathbf{\Theta}), \tag{4.35}$$

where

$$f_k(\mathbf{\Theta}^{(k)}) = -\log\det(\mathbf{\Theta}^{(k)}) + \mathrm{tr}(\mathbf{S}^{(k)}\mathbf{\Theta}^{(k)}).$$

In the second-order method, we approximate the objective function $F(\mathbf{\Theta})$ at the current iterate $\mathbf{\Theta}_t$ by a "quadratic" model $Q_t(\mathbf{\Theta})$:

$$\min_{\mathbf{\Theta}} Q_t(\mathbf{\Theta}) := \sum_{k=1}^{K} q_k(\mathbf{\Theta}^{(k)}) + P(\mathbf{\Theta}), \tag{4.36}$$

where q_k is the quadratic approximation of f_k at $\mathbf{\Theta}_t^{(k)}$, that is,

$$q_k(\mathbf{\Theta}^{(k)}) = \frac{1}{2}\mathrm{tr}(\mathbf{W}_t^{(k)}\mathbf{D}^{(k)}\mathbf{W}_t^{(k)}\mathbf{D}^{(k)}) + \mathrm{tr}((\mathbf{S}^{(k)} - \mathbf{W}_t^{(k)})\mathbf{D}^{(k)}) + f_k(\mathbf{\Theta}_t^{(k)})$$

with $\mathbf{W}_t^{(k)} = (\mathbf{\Theta}_t^{(k)})^{-1}$ and $\mathbf{D}^{(k)} = \mathbf{\Theta}^{(k)} - \mathbf{\Theta}_t^{(k)}$. Suppose that $\bar{\mathbf{\Theta}}_{t+1}$ is the optimal solution of (4.36). Then we obtain the Newton search direction

$$\mathbf{D} = \bar{\mathbf{\Theta}}_{t+1} - \mathbf{\Theta}_t. \tag{4.37}$$

We shall mention that the subproblem (4.36) can be suitably solved by the non-monotone spectral projected gradient (NSPG) method (see, for example, Lu and Zhang (2011); Wright *et al.* (2009)). It was shown by Lu and Zhang (2011) that the NSPG method is locally linearly convergent. Numerous computational studies have demonstrated that the NSPG method is very efficient though its global convergence rate is so far unknown. When applied to (4.36), the NSPG method requires solving the proximal subproblems in the form of

$$prox_{\alpha P}(\mathbf{Z}_r) := \arg\min_{\mathbf{\Theta}} \frac{1}{2}\|\mathbf{\Theta} - \mathbf{Z}_r\|_F^2 + \alpha P(\mathbf{\Theta}), \tag{4.38}$$

where r represents the r-th iteration in NSPG, $\|\mathbf{\Theta} - \mathbf{Z}_r\|_F^2 = \sum_{k=1}^K \|\mathbf{\Theta}^{(k)} - \mathbf{Z}_r^{(k)}\|_F^2$, $\mathbf{Z}_r = \mathbf{\Theta}_r - \alpha\mathbf{G}_r$, and $\mathbf{G}_r^{(k)} = \mathbf{S}^{(k)} - 2\mathbf{W}_t^{(k)} + \mathbf{W}_t^{(k)}\mathbf{\Theta}_r^{(k)}\mathbf{W}_t^{(k)}$. Denote $\mathbf{R} = \mathbf{\Theta}_r - \mathbf{\Theta}_{r-1}$ and $\bar{\alpha} = \sum_{k=1}^K \operatorname{tr}(\mathbf{R}^{(k)}\mathbf{W}_t^{(k)}\mathbf{R}^{(k)}\mathbf{W}_t^{(k)}) / \sum_{k=1}^K \|\mathbf{R}^{(k)}\|_F^2$. Then α is given by $\alpha = \max(\alpha_{min}, \min(1/\bar{\alpha}, \alpha_{max}))$, where $[\alpha_{min}, \alpha_{max}]$ is a given safeguard (Lu and Zhang, 2011; Wright *et al.*, 2009).

By the definition of $P(\mathbf{\Theta})$, it is not hard to see that problem (4.38) can be decomposed into a set of independent and smaller sized problems

$$\min_{\mathbf{\Theta}_{ij}^{(k)}, k=1,\ldots,K} \frac{1}{2}\sum_{k=1}^K (\mathbf{\Theta}_{ij}^{(k)} - \mathbf{Z}_{r,ij}^{(k)})^2 + \alpha_1 \sum_{k=1}^K |\mathbf{\Theta}_{ij}^{(k)}| + \alpha_2 \sum_{k=1}^{K-1} |\mathbf{\Theta}_{ij}^{(k)} - \mathbf{\Theta}_{ij}^{(k+1)}| \tag{4.39}$$

for all $i > j$, $(\alpha_1, \alpha_2) = \alpha(\lambda_1, \lambda_2)$, and for $i = j$, $\alpha_1, \alpha_2 = 0$, $j = 1, \ldots, p$. The problem (4.39) is known as the fused lasso signal approximator, which can be solved very efficiently and exactly (Condat, 2013; Liu *et al.*, 2010). In addition, they are independent from each other and thus can be solved in parallel.

Given the current search direction $\mathbf{D} = (\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(K)})$ that is computed above, we need to find the suitable step length $\beta \in (0, 1]$ to ensure a sufficient reduction in the objective function of (4.2) and positive definiteness of the next iterate $\mathbf{\Theta}_{t+1}^{(k)} = \mathbf{\Theta}_t^{(k)} + \beta\mathbf{D}^{(k)}, k = 1, \ldots, K$. In the context of the standard (single) graphical lasso, Hsieh *et al.* (2011) have shown that a step length satisfying the above requirements always exists. We can similarly prove that the desired step length also exists for the FMGL (4.2).

Lemma 12. *Let $\mathbf{\Theta}_t = (\mathbf{\Theta}_t^{(1)}, \ldots, \mathbf{\Theta}_t^{(K)})$ be such that $\mathbf{\Theta}_t^{(k)} \succ 0$ for $k = 1, \ldots, K$, and let $\mathbf{D} = (\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(K)})$ be the associated Newton search direction computed according to (4.36). Suppose $\mathbf{D} \neq 0$.* [1] *Then there exists a $\bar{\beta} > 0$ such that $\mathbf{\Theta}_t^{(k)} + \beta\mathbf{D}^{(k)} \succ 0$ and the sufficient reduction condition*

$$F(\mathbf{\Theta}_t + \beta\mathbf{D}) \leq F(\mathbf{\Theta}_t) + \sigma\beta\delta_t \tag{4.40}$$

[1] It is well known that if $\mathbf{D} = 0$, $\mathbf{\Theta}_t$ is the optimal solution of problem (4.2).

holds for all $0 < \beta < \bar{\beta}$, *where* $\sigma \in (0, 1/2)$ *is a given constant and*

$$\delta_t = \sum_{k=1}^{K} \mathrm{tr}((\mathbf{S}^{(k)} - \mathbf{W}_t^{(k)})\mathbf{D}^{(k)}) + P(\mathbf{\Theta}_t + \mathbf{D}) - P(\mathbf{\Theta}_t). \tag{4.41}$$

Proof. Let $\tilde{\beta} = 1/\max\{\|(\mathbf{\Theta}_t^{(k)})^{-1}\mathbf{D}^{(k)}\|_2 : k = 1, \ldots, K\}$, where $\|\cdot\|_2$ denotes the spectral norm of a matrix. Since $\mathbf{D} \neq 0$ and $\mathbf{\Theta}_t^{(k)} \succ 0, k = 1, \ldots, K$, we see that $\tilde{\beta} > 0$. Moreover, we have for all $0 < \beta < \tilde{\beta}$ and $k = 1, \ldots, K$,

$$\begin{aligned}(\mathbf{\Theta}_t^{(k)})^{-\frac{1}{2}} \left(\mathbf{\Theta}_t^{(k)} + \beta\mathbf{D}^{(k)}\right) (\mathbf{\Theta}_t^{(k)})^{-\frac{1}{2}} &= \mathbf{I} + \beta(\mathbf{\Theta}_t^{(k)})^{-\frac{1}{2}}\mathbf{D}^{(k)}(\mathbf{\Theta}_t^{(k)})^{-\frac{1}{2}} \\ &\succeq (1 - \beta\|(\mathbf{\Theta}_t^{(k)})^{-1}\mathbf{D}^{(k)}\|_2)\mathbf{I} \succ 0.\end{aligned}$$

By the definition of $\mathbf{D}$ and (4.36), one can easily show that

$$\delta \le -\sum_{k=1}^{K} \mathrm{tr}(\mathbf{W}_t^{(k)}\mathbf{D}^{(k)}\mathbf{W}_t^{(k)}\mathbf{D}^{(k)}),$$

which together with the fact that $\mathbf{W}_t^{(k)} \succ 0, k = 1, \ldots, K$ and $\mathbf{D} \neq 0$ implies that $\delta < 0$. Using differentiability of f_k, convexity of P, and the definition of δ, we obtain that for all sufficiently small $\beta > 0$,

$$\begin{aligned}F(\mathbf{\Theta}_t + \beta\mathbf{D}) - F(\mathbf{\Theta}_t) &= \textstyle\sum_{k=1}^{K}(f_k(\mathbf{\Theta}_t^{(k)} + \beta\mathbf{D}^{(k)}) - f_k(\mathbf{\Theta}_t^{(k)})) + P(\mathbf{\Theta}_t + \beta\mathbf{D}) - P(\mathbf{\Theta}_t), \\ &= \textstyle\sum_{k=1}^{K} \mathrm{tr}((\mathbf{S}^{(k)} - \mathbf{W}_t^{(k)})\mathbf{D}^{(k)})\beta + o(\beta) + P(\beta(\mathbf{\Theta}_t + \mathbf{D}) + (1-\beta)\mathbf{\Theta}_t) - P(\mathbf{\Theta}_t), \\ &\le \textstyle\sum_{k=1}^{K} \mathrm{tr}((\mathbf{S}^{(k)} - \mathbf{W}_t^{(k)})\mathbf{D}^{(k)})\beta + o(\beta) + \beta P(\mathbf{\Theta}_t + \mathbf{D}) + (1-\beta)P(\mathbf{\Theta}_t) - P(\mathbf{\Theta}_t), \\ &\le \beta\delta + o(\beta).\end{aligned}$$

This inequality together with $\delta < 0$ and $\sigma \in (0, 1)$ implies that there exists $\hat{\beta} > 0$ such that for all $\beta \in (0, \hat{\beta})$, $F(\mathbf{\Theta}_t + \beta\mathbf{D}) - F(\mathbf{\Theta}_t) \le \sigma\beta\delta$. It then follows that the conclusion of this lemma holds for $\bar{\beta} = \min\{\tilde{\beta}, \hat{\beta}\}$. $\square$

By virtue of Lemma 12, we can adopt the well-known Armijo's backtracking line search rule (Tseng and Yun, 2009) to select a step length $\beta \in (0, 1]$ so that $\mathbf{\Theta}_t^{(k)} + \beta\mathbf{D}^{(k)} \succ 0$ and (4.40) holds. In particular, we choose β to be the largest number of the sequence $\{1, 1/2, \dots, 1/2^i, \dots\}$ that satisfies these requirements. We can use the Cholesky factorization to check the positive definiteness of $\mathbf{\Theta}_t^{(k)} + \beta \mathbf{D}^{(k)}, k = 1, \dots, K$. In addition, the associated terms $\log\det(\mathbf{\Theta}_t^{(k)} + \beta \mathbf{D}^{(k)})$ and $(\mathbf{\Theta}_t^{(k)} + \beta \mathbf{D}^{(k)})^{-1}$ can be efficiently computed as a byproduct of the Cholesky decomposition of $\mathbf{\Theta}_t^{(k)} + \beta \mathbf{D}^{(k)}$.

4.4.1 Active Set Identification

Given the large number of unknown variables in (4.36), it is advantageous to minimize (4.36) in a reduced space. The issue now is how to identify the reduced space. In the case of a single graph ($K = 1$), problem (4.36) degenerates to a lasso problem of size p^2. Hsieh *et al.* (2011) propose a strategy to determine a subset of variables that are allowed to be updated in each Newton iteration for single graphical lasso. Specifically, the p^2 variables in single graphical lasso are partitioned into two sets, including a free set $\mathcal{F}$ and an active set $\mathcal{A}$, based on the gradient at the start of each Newton iteration, and then the minimization is only performed on the variables in $\mathcal{F}$. We call this technique "active set identification" in this chapter. Due to the sparsity of the precision matrix, the size of $\mathcal{F}$ is usually much smaller than p^2. Moreover, it has been shown in the single graph case that the size of $\mathcal{F}$ will decrease quickly (Hsieh *et al.*, 2011). The active set identification can thus improve the computational efficiency. This technique was also successfully used in (Joachims, 1999; Oberlin and Wright, 2006; Olsen *et al.*, 2012; Yuan *et al.*, 2012). We show that active set identification can be extended to the fused multiple graphical lasso based on the results established in Section 4.3.

Denote the gradient of f_k at t-th iteration by $\widetilde{\mathbf{G}}_t^{(k)} = \mathbf{S}^{(k)} - \mathbf{W}_t^{(k)}$, and its (i, j)-th element by $\widetilde{\mathbf{G}}_{t,ij}^{(k)}$. Then we have the following result.

Lemma 13. *For $\mathbf{\Theta}_t$ in the t-th iteration, define the active set $\mathcal{A}$ as*

$$\mathcal{A} = \{(i,j)|\mathbf{\Theta}_{t,ij}^{(1)} = \cdots = \mathbf{\Theta}_{t,ij}^{(K)} = 0 \textit{ and } \widetilde{\mathbf{G}}_{t,ij}^{(1)}, \ldots, \widetilde{\mathbf{G}}_{t,ij}^{(K)} \textit{ satisfy the inequalities below}\}.$$

$$\begin{cases} |\sum_{k=1}^{u} \widetilde{\mathbf{G}}_{t,ij}^{(k)}| < u\lambda_1 + \lambda_2, \\ |\sum_{k=0}^{u-1} \widetilde{\mathbf{G}}_{t,ij}^{(r+k)}| < u\lambda_1 + 2\lambda_2,\ 2 \le r \le K-u, \\ |\sum_{k=1}^{u} \widetilde{\mathbf{G}}_{t,ij}^{(K-u+k)}| < u\lambda_1 + \lambda_2, \\ |\sum_{k=1}^{K} \widetilde{\mathbf{G}}_{t,ij}^{(k)}| < K\lambda_1 \end{cases} \tag{4.42}$$

for $u = 1, \ldots, K-1$.

Then, the solution of the following optimization problem is $\mathbf{D}^{(1)} = \cdots = \mathbf{D}^{(K)} = 0$:

$$\min_{\mathbf{D}} Q_t(\mathbf{\Theta}_t + \mathbf{D}) \textit{ such that } \mathbf{D}_{ij}^{(1)} = \cdots = \mathbf{D}_{ij}^{(K)} = 0, (i,j) \notin \mathcal{A}. \tag{4.43}$$

Proof. Consider problem (4.43), which can be reformulated to

$$\begin{aligned} \min_{\mathbf{D}} \quad & \sum_{k=1}^{K} \left(\tfrac{1}{2}\text{vec}(\mathbf{D}^{(k)})^T \mathbf{H}_t^{(k)} \text{vec}(\mathbf{D}^{(k)}) + \text{vec}(\widetilde{\mathbf{G}}_t^{(k)})^T \text{vec}(\mathbf{D}^{(k)})\right) \\ & + P(\mathbf{\Theta}_t + \mathbf{D}), \\ & s.t. \quad \mathbf{D}_{ij}^{(1)} = \cdots = \mathbf{D}_{ij}^{(K)} = 0, (i,j) \notin \mathcal{A}, \end{aligned} \tag{4.44}$$

where $\mathbf{H}_t^{(k)} = \mathbf{W}_t^{(k)} \otimes \mathbf{W}_t^{(k)}$. Because of the constraint $\mathbf{D}_{ij}^{(1)} = \cdots = \mathbf{D}_{ij}^{(K)} = 0, (i,j) \notin \mathcal{A}$, we only consider the variables in the set $\mathcal{A}$. According to Lemma 9, it is easy to see that $\mathbf{D}_{\mathcal{A}} = 0$ satisfies the optimality condition of the following problem

$$\min_{\mathbf{D}_{\mathcal{A}}} \sum_{k=1}^{K} \text{vec}(\widetilde{\mathbf{G}}_{t,\mathcal{A}}^{(k)})^T \text{vec}(\mathbf{D}_{\mathcal{A}}^{(k)}) + P(\mathbf{D}_{\mathcal{A}}).$$

Since $\sum_{k=1}^{K} \text{vec}(\mathbf{D}^{(k)})^T \mathbf{H}_t^{(k)} \text{vec}(\mathbf{D}^{(k)}) \ge 0$, the optimal solution of (4.43) is given by $\mathbf{D}^{(1)} = \cdots = \mathbf{D}^{(K)} = 0$. □

Lemma 13 provides an active set identification scheme to partition the variables into the free set $\mathcal{F}$ and the active set $\mathcal{A}$. Lemma 13 shows that when the variables in the free set $\mathcal{F}$ are fixed, no update is needed for the variables in the active set $\mathcal{A}$.

The resulting second-order method with active set identification for solving the fused multiple graphical lasso is summarized in Algorithm 5.

Algorithm 5: Proposed Second-Order Method for Fused Multiple Graphical Lasso (FMGL)

```
Input: $\mathbf{S}^{(k)}, k = 1, \dots, K, \lambda_1, \lambda_2$
Output: $\mathbf{\Theta}^{(k)}, k = 1, \dots, K$
Initialization: $\mathbf{\Theta}_0^{(k)} = (\mathrm{Diag}(\mathbf{S}^{(k)}))^{-1}$;
while Not Converged do
    Determine the sets of free and fixed indices $\mathcal{F}$ and $\mathcal{A}$ using Lemma 13.
    Compute the Newton direction $\mathbf{D}^{(k)}, k = 1, \dots, K$ by solving (4.36) and
    (4.37) over the free variables $\mathcal{F}$.
    Choose $\mathbf{\Theta}_{t+1}^{(k)}$ by performing the Armijo backtracking line search along
    $\mathbf{\Theta}_t^{(k)} + \beta \mathbf{D}^{(k)}$ for $k = 1, \dots, K$.
end
return $\mathbf{\Theta}^{(k)}, k = 1, \dots, K$;
```

4.4.2 Convergence

Convergence of proximal Newton-type methods has been studied in previous literature (Byrd *et al.*, 2013; Hsieh *et al.*, 2011; Lee *et al.*, 2012; Scheinberg and Tang, 2014; Tseng and Yun, 2009). Under the assumption that the subproblems are solved exactly, a local quadratic convergence rate can be achieved when the exact Hessian is used (i.e., proximal Newton method) (Hsieh *et al.*, 2011; Lee *et al.*, 2012; Tseng and Yun, 2009). When an approximate Hessian is used (i.e., proximal quasi-Newton method), the local convergence rate is linear or superlinear (Lee *et al.*, 2012; Tseng and Yun, 2009). We show that the FMGL algorithm (with active set identification) falls into the proximal quasi-Newton framework. Denote the approximate Hessian by

$$\widetilde{\mathbf{H}}_t^{(k)} = \begin{pmatrix} \mathbf{H}_{t,\mathcal{F}}^{(k)} & \\ & \mathbf{H}_{t,\mathcal{A}}^{(k)} \end{pmatrix} \tag{4.45}$$

where $\mathbf{H}_{t,\mathcal{J}}^{(k)}$ is the submatrix of the exact Hessian $\mathbf{H}_t^{(k)}$ with variables in $\mathcal{J}$. Using $\widetilde{\mathbf{H}}_t^{(k)}$ instead, the subproblem (4.36) can be decomposed into the following two problems:

$$\begin{aligned} &\min_{\mathbf{D}_\mathcal{J}} \textstyle\sum_{k=1}^K \left(\frac{1}{2}\mathrm{vec}(\mathbf{D}_\mathcal{J}^{(k)})^T \mathbf{H}_{t,\mathcal{J}}^{(k)} \mathrm{vec}(\mathbf{D}_\mathcal{J}^{(k)}) + \mathrm{vec}(\widetilde{\mathbf{G}}_{t,\mathcal{J}}^{(k)})^T \mathrm{vec}(\mathbf{D}_\mathcal{J}^{(k)})\right) \\ &\qquad + P(\mathbf{\Theta}_{t,\mathcal{J}} + \mathbf{D}_\mathcal{J}), \qquad \mathcal{J} = \mathcal{F},\ \mathcal{A}. \end{aligned} \tag{4.46}$$

Consider the problem with respect to the variables in $\mathcal{A}$:

$$\min_{\mathbf{D}_\mathcal{A}} \textstyle\sum_{k=1}^K \left(\frac{1}{2}\mathrm{vec}(\mathbf{D}_\mathcal{A}^{(k)})^T \mathbf{H}_{t,\mathcal{A}}^{(k)} \mathrm{vec}(\mathbf{D}_\mathcal{A}^{(k)}) + \mathrm{vec}(\widetilde{\mathbf{G}}_{t,\mathcal{A}}^{(k)})^T \mathrm{vec}(\mathbf{D}_\mathcal{A}^{(k)})\right) + P(\mathbf{\Theta}_{t,\mathcal{A}} + \mathbf{D}_\mathcal{A}),$$

which is equivalent to problem (4.44). According to the definition of the active set $\mathcal{A}$, it follows Lemma 13 that the optimal solution is $\mathbf{D}_\mathcal{A}^{(k)} = 0,\ k = 1, \ldots, K$. Thus, FMGL in Algorithm 5 is a proximal quasi-Newton method. The global convergence to the unique optimal solution is therefore guaranteed (Lee *et al.*, 2012).

In the case when the subproblems are solved inexactly (i.e., *inexact* FMGL), we can adopt the following adaptive stopping criterion proposed in (Byrd *et al.*, 2013; Lee *et al.*, 2012) to achieve the global convergence:

$$\|M_{\tau\bar{q}}(\bar{\mathbf{\Theta}})\| \le \eta_t \|M_{\tau\bar{f}}(\mathbf{\Theta}_t)\|, \quad Q_t^H(\bar{\mathbf{\Theta}}) - Q_t^H(\mathbf{\Theta}_t) \le \zeta(L_t(\bar{\mathbf{\Theta}}) - L_t(\mathbf{\Theta}_t)), \tag{4.47}$$

for some $\tau > 0$, where $\bar{\mathbf{\Theta}}$ is an *inexact* solution of the subproblem, $\eta_t \in (0,1)$ is a forcing term, $\zeta \in (\sigma, 1/2)$, $L_t(\mathbf{\Theta})$ is defined by

$$L_t(\mathbf{\Theta}) = \bar{f}(\mathbf{\Theta}_t) + \mathrm{vec}(\nabla \bar{f}(\mathbf{\Theta}))^T \mathrm{vec}(\mathbf{\Theta} - \mathbf{\Theta}_t) + P(\mathbf{\Theta}),$$

and the composite gradient step $M_{\tau\bar{f}}(\mathbf{\Theta})$ is defined by

$$M_{\tau\bar{f}}(\mathbf{\Theta}) = \frac{1}{\tau}\left(\mathbf{\Theta} - prox_{\tau P}(\mathbf{\Theta} - \tau\nabla\bar{f}(\mathbf{\Theta}))\right).$$

The function $\bar{q}(\boldsymbol{\Theta})$ and $\bar{f}(\boldsymbol{\Theta})$ are defined by

$$\bar{q}(\boldsymbol{\Theta}) = \sum_{k=1}^{K} q_k^H(\boldsymbol{\Theta}^{(k)}), \ \bar{f}(\boldsymbol{\Theta}) = \sum_{k=1}^{K} f_k(\boldsymbol{\Theta}^{(k)}).$$

The superscript in Q_t^H and q_k^H represents the "quadratic" approximate functions Q_t and q_k using the approximate Hessian in (4.45) rather than the exact Hessian. According to the definition of $\mathcal{A}$ (i.e., $\mathbf{D}_{\mathcal{A}} = 0$ and $\boldsymbol{\Theta}_{t,\mathcal{A}} = 0$), the adaptive stopping criterion in (4.47) can only be verified over the variables in the free set $\mathcal{F}$. Following (Byrd *et al.*, 2013), the sufficient reduction condition in the line search of inexact FMGL uses $L_t(\boldsymbol{\Theta}_t + \beta\mathbf{D}) - L_t(\boldsymbol{\Theta}_t)$ instead of $\beta\delta_t$ in (4.40).

Although the global convergence of inexact proximal Newton-type (including Newton and quasi-Newton) methods is guaranteed, it is still challenging to prove a convergence rate for inexact proximal quasi-Newton methods such as inexact FMGL where an approximate Hessian is used. The local convergence rate of inexact proximal Newton method has been studied in Byrd *et al.* (2013); Lee *et al.* (2012). However, their proofs require the Hessian to be exact, which is not the case in inexact FMGL. It is worthy of noting that Scheinberg and Tang (2014) have recently shown a sublinear global convergence rate for inexact proximal quasi-Newton methods. In order to have such global convergence rate, their method uses a prox-parameter updating mechanism instead of line search for acceptance of iterates (Scheinberg and Tang, 2014). It is difficult to apply their technique to our formulation, since the conditions in Scheinberg and Tang (2014) for the global convergence rates may not hold for inexact FMGL. The property of the selected active set $\mathcal{A}$ and the special structure of the approximate Hessian may be the key to establish a faster local convergence rate for inexact FMGL. We defer these analysis to future work.

4.5 Experimental Results

In this section, we evaluate the proposed algorithm and screening rule on synthetic datasets and two real datasets: ADHD-200 [2] and FDG-PET images [3] . The experiments are performed on a PC with quad-core Intel 2.67GHz CPU and 9GB memory.

4.5.1 Simulation

We conduct experiments to demonstrate the effectiveness of the proposed screening rule and the efficiency of our method FMGL. The following algorithms are included in our comparisons:

- FMGL: the proposed second-order method in Algorithm 5.
- ADMM: ADMM method.
- FMGL-S: FMGL with screening.
- ADMM-S: ADMM with screening.

Both FMGL and ADMM are written in Matlab, and they are available online [4] . Since both methods involve solving (4.38) which involves a double loop, we implement the sub-routine for solving (4.38) in C for a fair comparison.

The synthetic covariance matrices are generated as follows. We first generate K block diagonal ground truth precision matrices $\mathbf{\Theta}^{(k)}$ with L blocks, and each block $\mathbf{\Theta}_l^{(k)}$ is of size $(p/L) \times (p/L)$. Each $\mathbf{\Theta}_l^{(k)}, l = 1, \ldots, L, k = 1, \ldots, K$ has random sparsity structures. We control the number of nonzeros in each $\mathbf{\Theta}_l^{(k)}$ to be about

[2]`http://fcon_1000.projects.nitrc.org/indi/adhd200/`

[3]`http://adni.loni.ucla.edu/`

[4]`http://www.public.asu.edu/~jye02/Software/MGL/`

$10p/L$ so that the total number of nonzeros in the K precision matrices is $10Kp$. Given the precision matrices, we draw $5p$ samples from each Gaussian distribution to compute the sample covariance matrices. The fused penalty parameter λ_2 is fixed to 0.1, and the ℓ_1 regularization parameter λ_1 is selected so that the total number of nonzeros in the solution is about $10Kp$.

Convergence

We first explore the convergence behavior of FMGL with different stopping criteria in NSPG. Three stopping criteria are considered:

- 1E-6: stop when the relative error $\frac{\max\{\|\mathbf{\Theta}_r^{(k)} - \mathbf{\Theta}_{r-1}^{(k)}\|_\infty\}}{\max\{\|\mathbf{\Theta}_{r-1}^{(k)}\|_\infty\}} \leq 1e\text{-}6$.
- Exact: the subproblems are solved accurately as in (Lee *et al.*, 2012) (More precisely, NSPG stops when $\frac{\max\{\|\mathbf{\Theta}_r^{(k)} - \mathbf{\Theta}_{r-1}^{(k)}\|_\infty\}}{\max\{\|\mathbf{\Theta}_{r-1}^{(k)}\|_\infty\}} \leq 1e\text{-}12$).
- Adaptive: stop when adaptive stopping criterion (4.47) is satisfied. The forcing term η_k is chosen as in (Lee *et al.*, 2012).

We plot the relative error of objective value versus Newton iterations and time on a synthetic dataset ($K = 5,\ L = 1,\ p = 500$) in Figure 4.2. We observe from Figure 4.2 that the exact stopping criterion has the fastest convergence with respect to Newton iterations. Considering computational time, the adaptive criterion has the best convergence behavior. Although the criterion 1E-6 has almost the same convergence behavior as the exact criterion in the first few steps, FMGL with this constant stopping criterion converges slower when the approximated solution is close enough to the optimal solution. We also include the convergence of ADMM in Figure 4.2. We can see that ADMM converges much slower than FMGL.

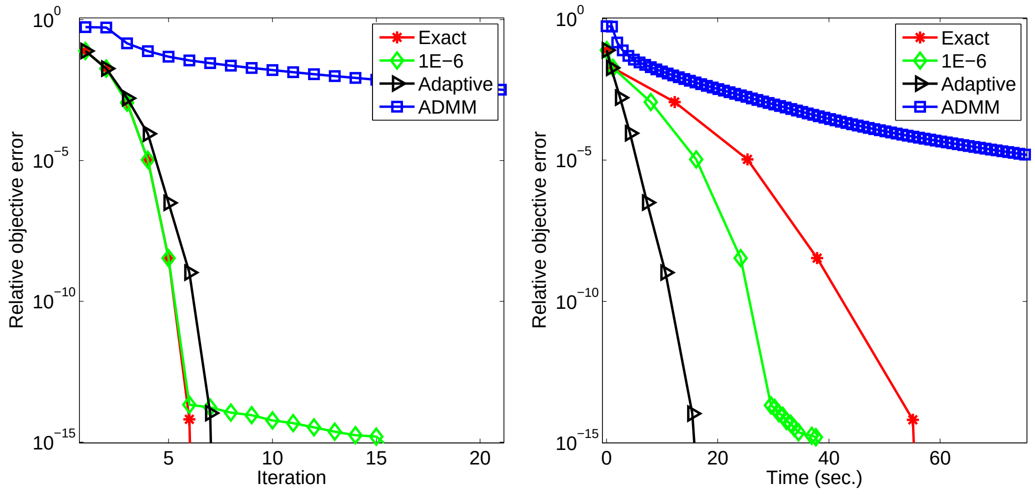


Figure 4.2: Convergence behavior of FMGL with 3 stopping criteria (exact, adaptive and 1E-6) and ADMM.

Screening

We conduct experiments to show the effectiveness of the proposed screening rule. NSPG is terminated using the adaptive stop criterion. FMGL is terminated when the relative error of the objective value is smaller than $1e$-5, and ADMM stops when it achieves an objective value equal to or smaller than that of FMGL. The results presented in Table 4.1 show that FMGL is consistently faster than ADMM. Moreover, the screening rule can achieve great computational gain. The speedup with the screening rule is about 10 and 20 times for $L = 5$ and 10 respectively.

Stability

We conduct experiments to demonstrate the effectiveness of FMGL. The synthetic sparse precision matrices are generated in the following way: we set the first precision matrix $\mathbf{\Theta}^{(1)}$ as $0.25I_{p\times p}$, where $p = 100$. When adding an edge (i, j) in the graph, we add σ to $\theta_{ii}^{(1)}$ and $\theta_{jj}^{(1)}$, and subtract σ from $\theta_{ij}^{(1)}$ and $\theta_{ji}^{(1)}$ to keep the positive definiteness of $\mathbf{\Theta}^{(1)}$, where σ is uniformly drawn from $[0.1, 0.3]$. When deleting an edge (i, j) from the graph, we reverse the above steps with $\sigma = \theta_{ij}^{(1)}$. We randomly

Table 4.1: Comparison of the proposed FMGL and ADMM with and without screening in terms of average computational time (seconds). FMGL-S and ADMM-S are FMGL and ADMM with screening respectively. p stands for the dimension, K is the number of graphs, L is the number of blocks, and λ_1 is the ℓ_1 regularization parameter. The fused penalty parameter λ_2 is fixed to 0.1. $\|\mathbf{\Theta}\|_0$ represents the total number of nonzero entries in ground truth precision matrices $\mathbf{\Theta}^{(k)}, k = 1, \ldots, K$, and $\|\mathbf{\Theta}^*\|_0$ is the number of nonzeros in the solution.

Data and parameter setting						Computational time			
p	K	L	$\|\mathbf{\Theta}\|_0$	λ_1	$\|\mathbf{\Theta}^*\|_0$	FMGL-S	FMGL	ADMM-S	ADMM
500	2	5	9848	0.08	9810	**0.44**	4.13	13.30	100.79
1000			20388	0.088	19090	**2.25**	17.88	57.44	617.88
500	5		24866	0.055	23304	**0.97**	12.23	32.40	286.98
1000			50598	0.054	44030	**5.16**	50.95	174.91	1595.91
500	10		49092	0.051	45474	**2.33**	24.35	63.75	458.51
1000			100804	0.046	84310	**10.27**	111.78	302.86	2966.72
500	2	10	9348	0.07	9386	**0.32**	4.87	6.82	105.01
1000			19750	0.08	20198	**0.76**	17.93	25.62	674.28
500	5		23538	0.055	22900	**0.77**	14.96	15.09	256.33
1000			49184	0.054	45766	**1.92**	53.96	64.31	1314.18
500	10		47184	0.051	47814	**1.66**	52.32	29.86	455.43
1000			98564	0.046	94566	**4.44**	126.26	128.52	2654.24

assign 200 edges for $\mathbf{\Theta}^{(1)}$. $\mathbf{\Theta}^{(2)}$ is obtained by adding 25 edges and deleting 25 different edges from $\mathbf{\Theta}^{(1)}$. $\mathbf{\Theta}^{(3)}$ is obtained from $\mathbf{\Theta}^{(2)}$ in the same way. For each precision matrix, we randomly draw n samples from the Gaussian distribution with the corresponding precision matrix, where n varies from 40 to 200 with a step of 20. We perform 500 replications for each n. For each n, λ_2 is fixed to 0.08, and λ_1 is adjusted to make sure that the edge number is about 200. The accuracy n_d/n_g is used to measure the performance of FMGL and GLasso, where n_d is the number of true edges detected by FGML and GLasso, and n_g is the number of true edges. The results are shown in Figure 4.3. We can see from the figure that FMGL achieves higher accuracies, demonstrating the effectiveness of FMGL for learning multiple graphical models simultaneously.

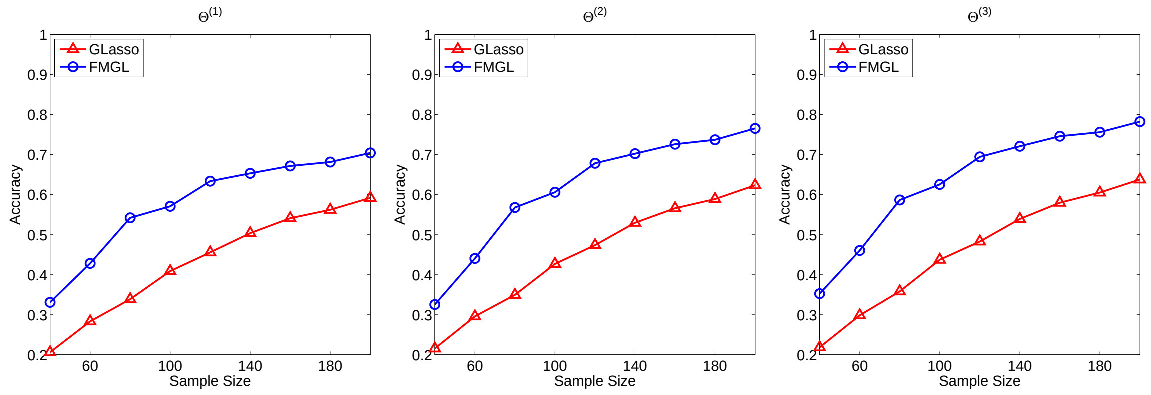


Figure 4.3: Comparison of FMGL and GLasso in detecting true edges. Sample size varies from 40 to 200 with a step of 20.

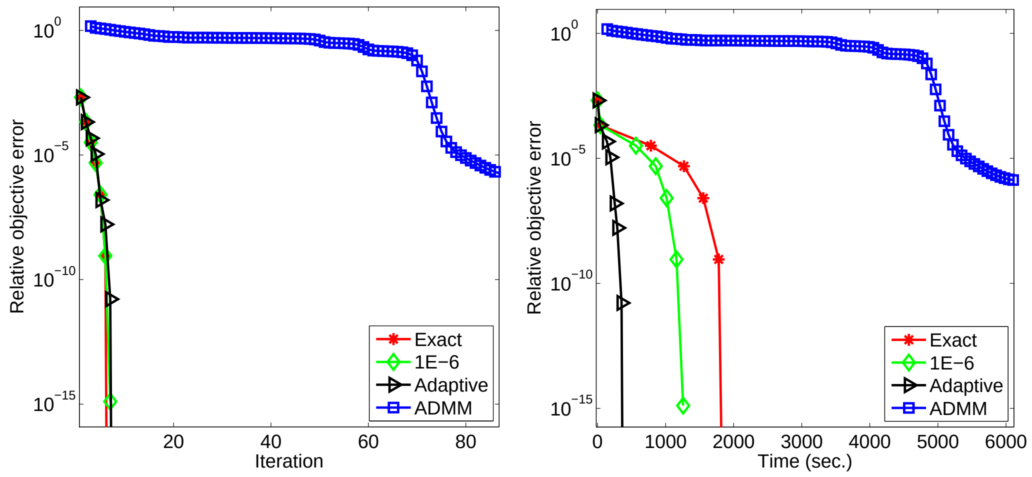


Figure 4.4: Comparison of FMGL with 3 stopping criteria and ADMM in terms of objective value curve on the ADHD-200 dataset. The dimension p is 2834, and the number of graphs K is 3.

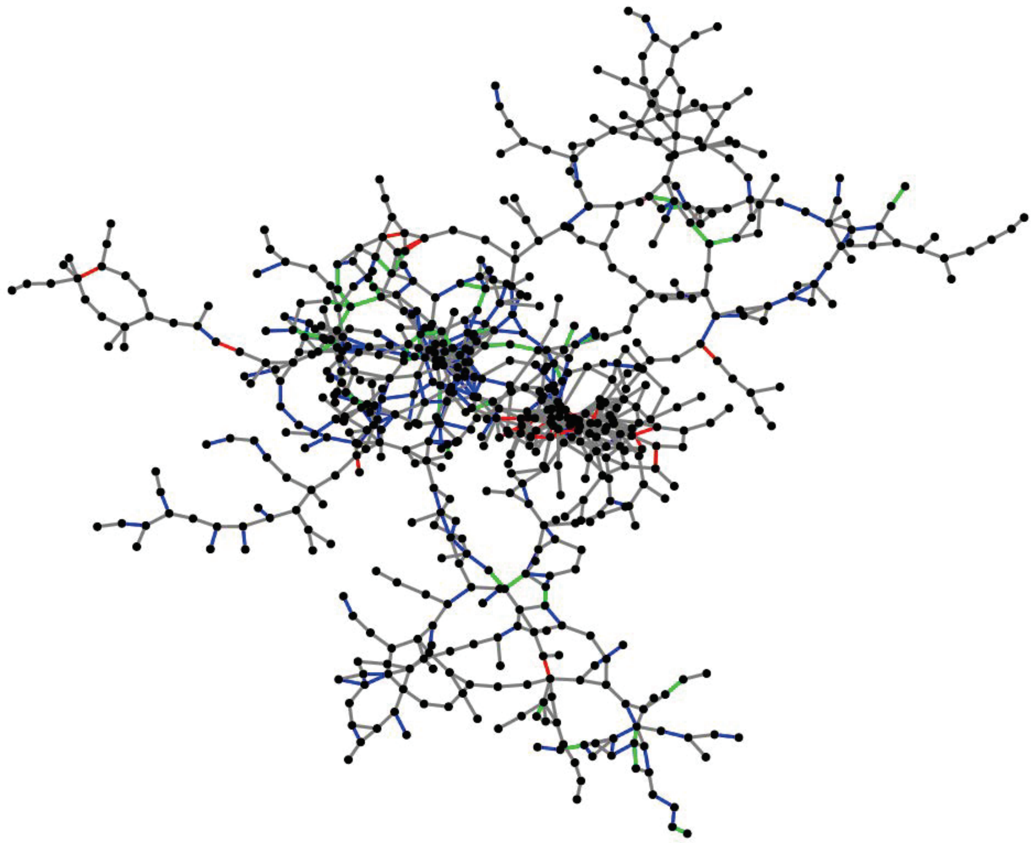


Figure 4.5: A subgraph of ADHD-200 identified by FMGL with the proposed screening rule. The grey edges are common edges among the three graphs; the red, green, and blue edges are the specific edges for TDC, ADHD-I, and ADHD-C respectively.

4.5.2 Real Data

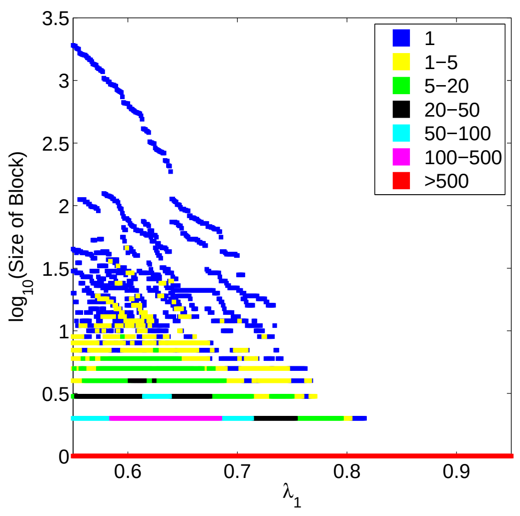
ADHD-200

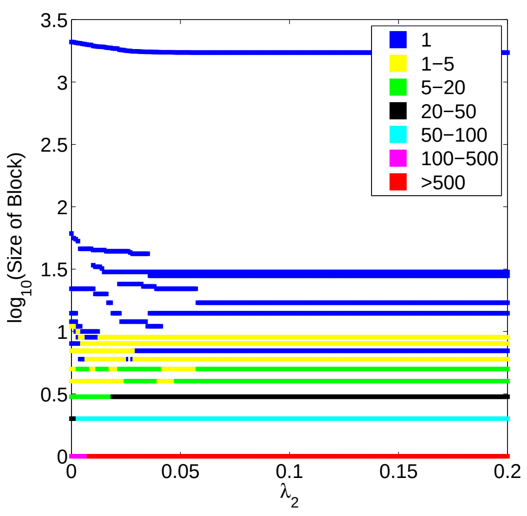
Attention Deficit Hyperactivity Disorder (ADHD) affects at least 5-10% of school-age children with annual costs exceeding 36 billion/year in the United States. The ADHD-200 project has released resting-state functional magnetic resonance images (fMRI) of 491 typically developing children and 285 ADHD children, aiming to encourage the research on ADHD. The data used in this experiment is preprocessed using the NIAK pipeline, and downloaded from neurobureau [5] . More details about the prepro-

[5]`http://www.nitrc.org/plugins/mwiki/index.php?title=neurobureau:NIAKPipeline/`

cessing strategy can be found in the same website. The dataset we choose includes 116 typically developing children (TDC), 29 ADHD-Combined (ADHD-C), and 49 ADHD-Inattentive (ADHD-I). There are 231 time series and 2834 brain regions for each subject. We want to estimate the graphs of the three groups simultaneously. The sample covariance matrix is computed using all data from the same group. Since the number of brain regions p is 2834, obtaining the precision matrices is computationally intensive. We use this data to test the effectiveness of the proposed screening rule. λ_1 and λ_2 are set to 0.6 and 0.015. The comparison of FMGL with 3 stopping criteria and ADMM in terms of the objective value curve is shown in Figure 4.4. The result shows that FMGL converges much faster than ADMM. To obtain a solution of precision $1e$-5, the computational times of FMGL (Adaptive), FMGL (1E-6), FMGL (Exact), and ADMM are 252.78, 855.86, 1269.75 and 5410.48 seconds respectively. However, with the screening, the computational times of FMGL-S (Adaptive), FMGL-S (1E-6), FMGL-S (Exact), and ADMM-S are reduced to 4.02, 12.51, 19.55, and 80.52 seconds respectively, demonstrating the superiority of the proposed screening rule. The obtained solution has 1443 blocks. The largest one including 634 nodes is shown in Figure 4.5.

The block structures of the FMGL solution are the same as those identified by the screening rule. The screening rule can be used to analyze the rough structures of the graphs. The cost of identifying blocks using the screening rule is negligible compared to that of estimating the graphs. For high-dimensional data such as ADHD-200, it is practical to use the screening rule to identify the block structure before estimating the large graphs. We use the screening rule to identify block structures on ADHD-200 data with varying λ_1 and λ_2. The size distribution is shown in Figure 4.6. We can observe that the number of blocks increases, and the size of blocks deceases when the regularization parameter value increases.





(a) (b)

Figure 4.6: The size distribution of blocks (in the logarithmic scale) identified by the proposed screening rule. The color represents the number of blocks of a specified size. (a): λ_1 varies from 0.5 to 0.95 with λ_2 fixed to 0.015. (b): λ_2 varies from 0 to 0.2 with λ_1 fixed to 0.55.

FDG-PET

In this experiment, we use FDG-PET images from 74 Alzhei-mer's disease (AD), 172 mild cognitive impairment (MCI), and 81 normal control (NC) subjects downloaded from the Alzheimer's disease neuroimaging initiative (ADNI) database. The different regions of the whole brain volume can be represented by 116 anatomical volumes of interest (AVOI), defined by Automated Anatomical Labeling (AAL) (Tzourio-Mazoyer *et al.*, 2002). Then we extracted data from each of the 116 AVOIs, and derived the average of each AVOI for each subject. The 116 AVOIs can be categorized into 10 groups: prefrontal lobe, other parts of the frontal lobe, parietal lobe, occipital lobe, thalamus, insula, temporal lobe, corpus striatum, cerebellum, and vermis. More details about the categories can be found in (Tzourio-Mazoyer *et al.*, 2002; Wang *et al.*, 2007). We remove two small groups (thalamus and insula) containing only 4 AVOIs in our experiments.

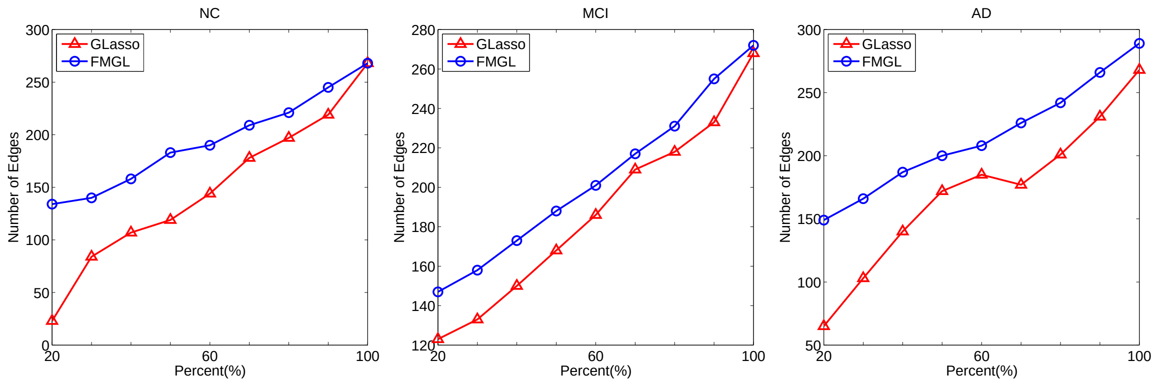


Figure 4.7: The average number of stable edges detected by FMGL and GLasso in NC, MCI, and AD of 500 replications. Sample size varies from 20% to 100% with a step of 10%.

To examine whether FMGL can effectively utilize the information of common structures, we randomly select g percent samples from each group, where g varies from 20 to 100 with a step size of 10. For each g, λ_2 is fixed to 0.1, and λ_1 is adjusted to make sure the number of edges in each group is about the same. We perform 500 replications for each g. The edges with probability larger than 0.85 are considered as stable edges. The results showing the numbers of stable edges are summarized in Figure 4.7. We can observe that FMGL is more stable than GLasso. When the sample size is too small (say 20%), there are only 20 stable edges in the graph of NC obtained by GLasso. But the graph of NC obtained by FMGL still has about 140 stable edges, illustrating the superiority of FMGL in stability.

The brain connectivity models obtained by FMGL are shown in Figure 4.8. We can see that the number of connections within the prefrontal lobe significantly increases, and the number of connections within the temporal lobe significantly decreases from NC to AD, which are supported by previous literatures (Azari *et al.*, 1992; Horwitz *et al.*, 1987). The connections between the prefrontal and occipital lobes increase from NC to AD, and connections within cerebellum decrease. We can also find that the adjacent graphs are similar, indicating that FMGL can identify the common

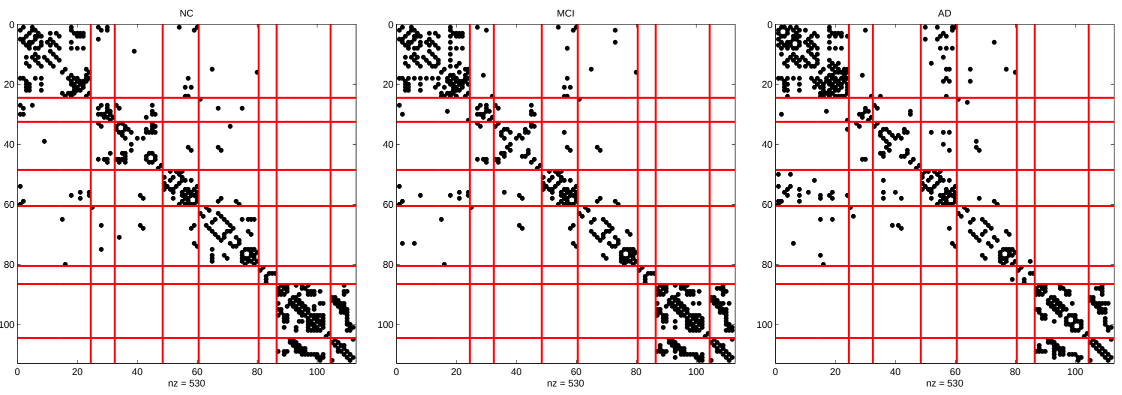


Figure 4.8: Brain connection models with 265 edges: NC, MCI, and AD. In each figure, the diagonal blocks are prefrontal lobe, other parts of frontal lobe, parietal lobe, occipital lobe, temporal lobe, corpus striatum, cerebellum, and vermis respectively.

structures, but also keep the meaningful differences.

4.6 Conclusion

In this chapter, we consider simultaneously estimating multiple graphical models by maximizing a fused penalized log likelihood. We have derived a set of necessary and sufficient conditions for the FMGL solution to be block diagonal for an arbitrary number of graphs. A screening rule has been developed to enable the efficient estimation of large multiple graphs. The second-order method is employed to solve the fused multiple graphical lasso, which is shown to be equivalent to a proximal quasi-Newton method. The global convergence of the proposed method with an adaptive stopping criterion is guaranteed. An active set identification scheme is proposed to identify the variables to be updated during the Newton iterations, thus reduces the computation. Numerical experiments on synthetic and real data demonstrate the efficiency and effectiveness of the proposed method and the screening rule. We plan to further explore the convergence properties of the second-order methods when the subproblems are solved inexactly. Due to the active set identification scheme, the proposed second-order method is suitable for warm-start techniques. A good initial solution can further speedup the computation. As part of the future work, we plan to explore how to efficiently find a good initial solution to further improve the efficiency of the proposed method. One possibility is to use divide-and-conquer techniques Hsieh *et al.* (2012).

Chapter 5

TREE-GUIDED GRAPHICAL LASSO

In this chapter, we describe a hierarchical graphical model framework where the features have a hierarchical structure. A motivating example is the estimation of brain network. The brain is a multi-level system, and the brain network has a native hierarchical structure as shown in Figure 5.1: hundreds of thousands of voxels form regions, and regions form systems. We present a second-order method to efficiently solve the proposed formulation. In addition, we derive a necessary and sufficient condition for the graph to be decomposable based on its connected components. Based on this property, we propose a simple screening rule which significantly reduces the size of the optimization problem, thus improving the computational efficiency. The proposed screening only relies on the data and the used parameters, thus it can be combined with any algorithms to reduce the computational cost. The experiments on both synthetic and real data demonstrate the effectiveness of our approaches.

5.1 Formulation

Suppose we are given a data set $\mathbf{X} \in \Re^{n \times p}$ with n samples, and p features (or variables). The n samples are independently and identically distributed with a p-variate Gaussian distribution with zero mean and positive definite covariance matrix Σ. Even all features are correlated, there usually are many conditional independences among these features. In other words, a sparse precision matrix $\mathbf{\Theta} = \Sigma^{-1}$ is of interest in most cases. This Gaussian graphical model (GMM) is also referred to *Gaussian Markov Random Field (GMRF).* The negative log likelihood for the data $\mathbf{X}$ takes the

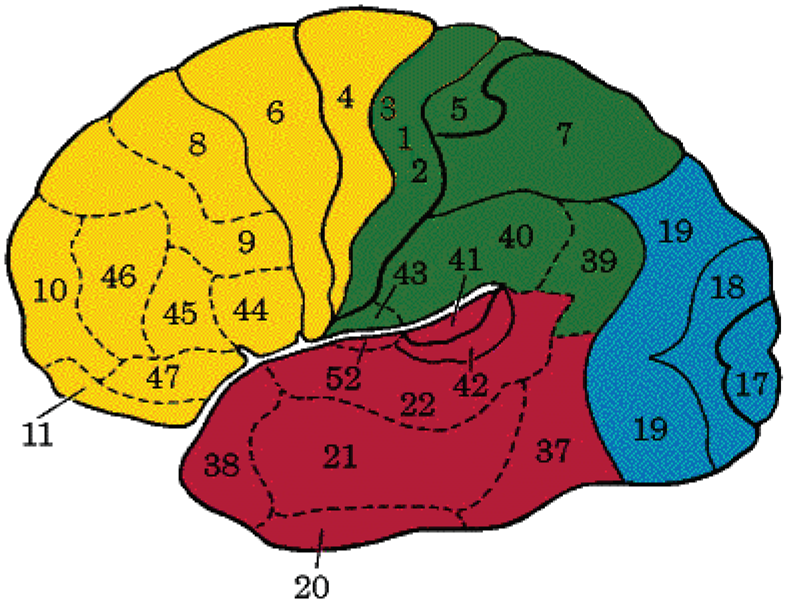


Figure 5.1: Brain image (Umich, 2014). Yellow: frontal lobe; green: parietal lobe; red: temporal lobe; blue: occipital lobe. Number represents brain regions within lobes.

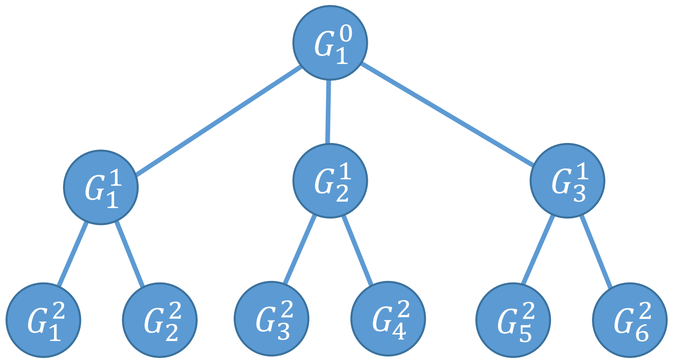


Figure 5.2: A sample index tree. Root: $G_1^0 = \{1,2,3,4,5,6,7,8\}$. Depth 1: $G_1^1 = \{1,2\}$, $G_2^1 = \{3,4,5,6\}$, $G_3^1 = \{7,8\}$. Depth 2: $G_1^2 = \{1\}$, $G_2^2 = \{2\}$, $G_3^2 = \{3,4,5\}$, $G_4^2 = \{6\}$, $G_5^2 = \{7\}$, $G_6^2 = \{8\}$.

form of

$$\mathcal{L}(\mathbf{\Theta}) := -\log\det(\mathbf{\Theta}) + \operatorname{tr}(\mathbf{S}\mathbf{\Theta}), \tag{5.1}$$

where $\mathbf{S}$ is the sample covariance matrix given by $\mathbf{S} = \frac{1}{n}\mathbf{X}^T\mathbf{X}$. Minimizing (5.1) leads to the maximum likelihood estimation (MLE) $\mathbf{\Theta}^* = \mathbf{S}^{-1}$. However, there are some

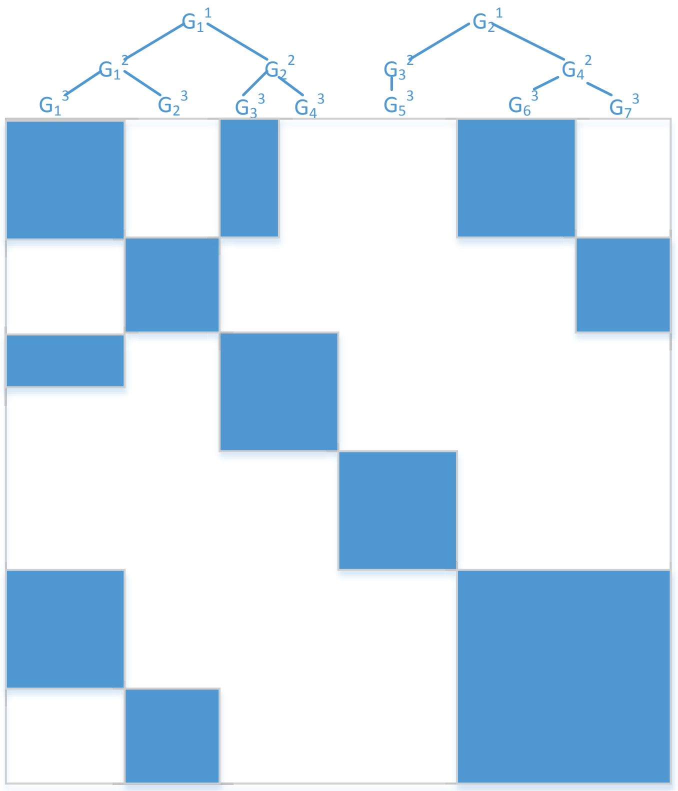


Figure 5.3: Illustration of a hierarchical graphical model. The features have a hierarchical structure specified by tree groups $\{G_i^j\}$. The blue blocks represent the nonzero blocks in the precision matrix.

issues with MLE. MLE fails in high-dimensional setting ($n < p$). In this setting, $\mathbf{S}$ is singular, thus $\mathbf{\Theta}^*$ does not exists. On the other hand, $\mathbf{\Theta}^*$ is unlikely to be sparse even $\mathbf{S}$ is invertible. The ℓ_1 regularization has been employed to induce the sparsity, resulting in sparse precision matrix estimation. In this chapter, we employ the tree structured group regularization to encourage the estimated graph to have a hierarchical structure. Mathematically, we solve the following formulation:

$$\min_{\mathbf{\Theta} \succ 0} - \log \det(\mathbf{\Theta}) + \operatorname{tr}(\mathbf{S\Theta}) + \phi(\mathbf{\Theta}) \tag{5.2}$$

where

$$\phi(\mathbf{\Theta}) = \sum_j \left(\sum_{i \neq i'} w_{ii'}^j \|\mathbf{\Theta}_{G_i^j, G_{i'}^j}\|_F + w_{ii}^j \|\mathbf{\Theta}_{G_i^j, G_i^j, off}\|_F \right),$$

the groups G_i^j is defined in Definition 2 (see Figure 5.2 for illustration), $\mathbf{\Theta}_{G_i^j, G_{i'}^j}$ denotes the submatrix of $\mathbf{\Theta}$ consisting of features in $G_i^j, G_{i'}^j$, and $w_{ii'}^j = w_{i'i}^j$ is a positive weight for $\mathbf{\Theta}_{G_i^j, G_{i'}^j}$. $\mathbf{\Theta}_{\cdot,\cdot,off}$ represents the matrix $\mathbf{\Theta}_{\cdot,\cdot}$ excluding the diagonal elements. We do not penalize the diagonal elements of $\mathbf{\Theta}$ since $\mathbf{\Theta}$ is required to be positive definite. For simplicity of notation, we use $\mathbf{\Theta}_{ii'}^j$ to represent $\mathbf{\Theta}_{G_i^j, G_{i'}^j}$, and $\mathbf{\Theta}_{ii/}^j$ to represent $\mathbf{\Theta}_{G_i^j, G_i^j, off}$. It is clear that $\mathbf{\Theta}_{ii'}^j = (\mathbf{\Theta}_{i'i}^j)^T$, thus we require $w_{ii'}^j = w_{i'i}^j$. The regularization $\phi(\mathbf{\Theta})$ encourages the estimated precision matrix to be tree structured (see Figure 5.3 for example).

Definition 2. *(Liu and Ye, 2010) For an index tree T of depth U, we let $T_u = \{G_1, \dots, G_{n_i}\}$ contain all the nodes corresponding to depth u, where $n_0 = 1$, $G_1^0 = \{1, \dots, K\}$ and $n_i \geq 1, i = 1, \dots, U$. The nodes satisfy the following conditions: 1) the nodes from the same depth level have non-overlapping indices, i.e., $G_j^u \cap G_k^u = \emptyset, \forall u = 1, \dots, U, j \neq k, 1 \leq j, k \neq n_i$; 2) let G_{j0}^{u-1} be the parent node of a non-root node G_j^u, then $G_j^u \subseteq G_{j0}^{u-1}$.*

5.2 Algorithm

We employ the second-order method to solve tree-guided graphical lasso (5.2). Let $f(\mathbf{\Theta})$ be the smooth function in (5.2) such that

$$f(\mathbf{\Theta}) = -\log\det(\mathbf{\Theta}) + \mathrm{tr}(\mathbf{S}\mathbf{\Theta}).$$

(5.2) can be rewritten as

$$\min_{\mathbf{\Theta} \succ 0} f(\mathbf{\Theta}) + \phi(\mathbf{\Theta}). \tag{5.3}$$

In the second-order method, we solve a "quadratic" model of (5.2) at each iteration defined by

$$\min_{\mathbf{\Theta}} \frac{1}{2}\mathrm{tr}(\mathbf{W}_t\mathbf{D}\mathbf{W}_t\mathbf{D}) + \mathrm{tr}((\mathbf{S} - \mathbf{W}_t)\mathbf{D}) + \phi(\mathbf{\Theta}), \tag{5.4}$$

where $\mathbf{W}_t = \mathbf{\Theta}_t^{-1}$ and $\mathbf{D} = \mathbf{\Theta} - \mathbf{\Theta}_t$, and t represents the t-th Newton iteration.

The subproblem (5.4) can be solved by non-monotone spectral projected gradient (NSPG) method (Wright *et al.*, 2009). When applied to (5.4), NSPG needs to solve the proximal subproblem taking form of

$$\min_{\mathbf{\Theta}} \frac{1}{2}\|\mathbf{\Theta} - \mathbf{G}_r\|_F^2 + \alpha\phi(\mathbf{\Theta}), \tag{5.5}$$

where

$$\mathbf{G}_r = \mathbf{\Theta}_r - \alpha(\mathbf{S} - 2\mathbf{W}_t + \mathbf{W}_t\mathbf{\Theta}_r\mathbf{W}_t)$$

and r denotes the r-th inner iteration in NSPG. Denote $\mathbf{R} = \mathbf{\Theta}_r - \mathbf{\Theta}_{r-1}$ and $\bar{\alpha} = \operatorname{tr}(\mathbf{R}\mathbf{W}_t\mathbf{R}\mathbf{W}_t)/\|\mathbf{R}\|_F^2$, then α is given by $\alpha = \max(\alpha_{min}, \min(1/\bar{\alpha}, \alpha_{max}))$, where $[\alpha_{min}, \alpha_{max}]$ is a given safeguard.

After obtaining the optimal solution of (5.4) $\mathbf{\Theta}^*$, the Newton direction $\mathbf{D}$ can be computed as

$$\mathbf{D} = \mathbf{\Theta}^* - \mathbf{\Theta}_t. \tag{5.6}$$

Once the Newton direction is obtained, we need to find an appropriate step size $\beta \in (0, 1]$ to ensure a sufficient reduction in the objective function in (5.3). Because of the positive definite constraint in (5.3), we need to ensure the next iterate $\mathbf{\Theta}_{t+1} = \mathbf{\Theta}_t + \beta\mathbf{D}$ to be positive definite. In Chapter 4, we prove that such step size satisfying the above requirements always exits. Thus, we can adopt the Amrmijo's backtracking line search rule to select a step length $\beta \in (0, 1]$. We use the Cholesky decomposition to check the positive definiteness of $\mathbf{\Theta}_{t+1} = \mathbf{\Theta}_t + \beta\mathbf{D}$. In addition, the $\log\det(\mathbf{\Theta}_{t+1})$ and $\mathbf{\Theta}_{t+1}^{-1}$ can be efficiently computed as a byproduct of the Cholesky decomposition of $\mathbf{\Theta}_{t+1}$. The algorithm can be summarized in Algorithm 6.

Under the assumption that the subproblem (5.4) is solved exactly, the convergence rate of the second-order method is locally quadratic when the exact Hessian is used (Hsieh *et al.*, 2011; Lee *et al.*, 2012; Tseng and Yun, 2009). If the subproblem (5.4)

Algorithm 6: Tree-Guided Graphical Lasso (TGL)

```
Input: $\mathbf{S}, \{G_i^j\}, \{w_{ii'}^j\}$
Output: $\mathbf{\Theta}$
Initialization: $\mathbf{\Theta}_0 = (\mathrm{Diag}(\mathbf{S}))^{-1}$;
while Not Converged do
    Compute the Newton direction $\mathbf{D}$ by solving (5.4) and (5.6).
    Choose $\mathbf{\Theta}_{t+1}$ by performing the Armijo backtracking line search along
    $\mathbf{\Theta}_t + \beta\mathbf{D}$.
end
return $\mathbf{\Theta}_{t+1}$;
```

is solved inexactly, the convergence rate of the second method is locally superlinear by adopting an adaptive stopping criteria in NSPG (Lee *et al.*, 2012). Due to the use of Cholesky decomposition and the need of computing $\mathrm{tr}(\mathbf{W}_t\mathbf{D}\mathbf{W}_t\mathbf{D})$ in (5.4), the complexity of Algorithm 6 is $O(p^3)$.

5.3 Screening

Due to the existence of log determination, it is computationally challenging to solve the penalized log likelihood approach. Screening has commonly been employed to reduce the size of optimization problem so that a missive computational gain can be achieved. In this section, we derive a necessary and sufficient condition for the solution of TGL to be block diagonal (subject to some rearrangement of features). Since the elements in off diagonal blocks are zero, the original optimization problem can be thus reduced to a small problem restricted to the elements in diagonal blocks, resulting in a great computational gain.

Let $C_1, \ldots, C_L$ be a partition of the p features into L non-overlapping sets such that $C_l \cap C_{l'} = \emptyset$, $\forall l \neq l'$. We say that the solution $\widehat{\mathbf{\Theta}}$ of TGL (5.2) is block diagonal (subject to some rearrangement of features) with L known blocks $C_l,\ l=1,\dots,L$ if $\widehat{\mathbf{\Theta}}_{ij}=\widehat{\mathbf{\Theta}}_{ji}=0$ for $i\in C_l,\ j\in C_{l'},\ l\neq l'$. Without loss of generality, we assume that a block diagonal solution $\widehat{\mathbf{\Theta}}$ with L blocks $C_l,\ l=1,\dots,L$ takes the form of

$$\widehat{\mathbf{\Theta}}=\begin{pmatrix}\widehat{\mathbf{\Theta}}_1 & & \\ & \ddots & \\ & & \widehat{\mathbf{\Theta}}_L\end{pmatrix}, \tag{5.7}$$

where $\widehat{\mathbf{\Theta}}_l$ is the $|C_l|\times|C_l|$ symmetric submatrix of $\widehat{\mathbf{\Theta}}$ consisting of features in C_l.

Theorem 14. *The TGL (5.2) has a block diagonal solution $\widehat{\mathbf{\Theta}}$ with L blocks $C_l,\ l=1,\dots,L$ if and only if the solution to the below problem is block diagonal with blocks $C_l,\ l=1\dots,L$:*

$$\min_{\mathbf{X}}\frac{1}{2}\|\mathbf{X}+\mathbf{S}\|_F^2+\phi(\mathbf{X}). \tag{5.8}$$

Proof. By the first-order optimality condition, $\widehat{\mathbf{\Theta}}$ is the optimal solution of problem (5.2) if and only if it satisfies

$$-(\widehat{\mathbf{\Theta}})^{-1}+\mathbf{S}+\partial\phi(\widehat{\mathbf{\Theta}})=0. \tag{5.9}$$

For simplicity of notation, we use $\mathbf{W}$ and $\partial\phi$ to represent $\mathbf{\Theta}^{-1}$ and $\partial\phi(\mathbf{\Theta})$, respectively.

If: Let $\widehat{\mathbf{X}}$ be the optimal solution of (5.8). Suppose that $\widehat{\mathbf{X}}$ has a block structure $C_l,\ l=1\dots,L$, then we have $\widehat{\mathbf{X}}_{ij}=0, i\in C_l, j\in C_{l'}, l\neq l'$. According to the first optimality condition, we have

$$\mathbf{S}_{ij}+\partial\phi_{ij}=0$$

for $i\in C_l, j\in C_{l'}, l\neq l'$.

Now Let $\widehat{\mathbf{\Theta}}$ be a block diagonal matrix with blocks $C_l,\ l=1\dots,L$. It is clear to see that the optimality condition of (5.2) for off diagonal elements can be satisfied. We can let the elements in diagonal block of $\widehat{\mathbf{\Theta}}$ be the solution of the following problem:

$$\min_{\mathbf{\Theta}_l,l=1,\dots,L}\sum_{l=1}^{L}(-\log\det(\mathbf{\Theta}_l)+\mathrm{tr}(\mathbf{S}_l\mathbf{\Theta}_l))+\phi(\mathbf{\Theta}).$$

Note that zero elements in the off diagonal blocks of $\mathbf{\Theta}$ do not affect $\phi(\mathbf{\Theta})$. The first optimality condition (5.9) holds for $\widehat{\mathbf{\Theta}}$, thus $\widehat{\mathbf{\Theta}}$ is the optimal solution of (5.2).

Only if: Suppose that the optimal solution of (5.2) $\widehat{\mathbf{\Theta}}$ has a block diagonal structure $C_l, l = 1, \ldots, L$. Note that $\widehat{\mathbf{W}}$ has the same block diagonal structure as $\widehat{\mathbf{\Theta}}$, thus we have

$$\mathbf{S}_{ij} + \partial\phi_{ij} = 0$$

for $i \in C_l, j \in C_{l'}, l \neq l'$. It is not hard to see that the optimal solution of (5.8) has the same block structure $C_l, l = 1 \ldots, L$. $\square$

Theorem 14 can be used as screening rule to determine the elements in the identified off-diagonal blocks to be zero in advance. Assume that there are L blocks of the same size identified by the screening rule, $p^2(1 - \frac{1}{L})$ elements do not need to be computed as the optimal value for these elements are determined as 0 by the screening. Recall that the complexity of the proposed second-order method is $O(p^3)$ due to Cholesky decomposition and computation of $\mathrm{tr}(\mathbf{W}_t\mathbf{D}\mathbf{W}_t\mathbf{D})$. The complexity of solving the proximal operator (5.8) is $O(p^2)$ (Liu and Ye, 2010). By applying the screening rule, the complexity of Cholesky decomposition and computation of $\mathrm{tr}(\mathbf{W}_t\mathbf{D}\mathbf{W}_t\mathbf{D})$ are reduced to $O(p^3/L^2)$, and the complexity of solving (5.8) is reduced to $O(p^2/L)$. Therefore, the complexity of the second-order method with screening is $O(p^3/L^2)$ since $L \leq p$. When L is large, applying the screening rule can achieve a great computational gain.

5.4 Experimental Results

In this section, we conduct experiments to demonstrate the effectiveness of the proposed tree-guided graphical lasso (TGL). The experiments are performed on a PC with quad-core Intel i7 3.4GHz CPU and 16GB memory. TGL is written in Matlab, while its sub-routine for solving the subproblem (5.5) is written in C. We compare TGL with standard graphical lasso (GLasso) in the following experiments.

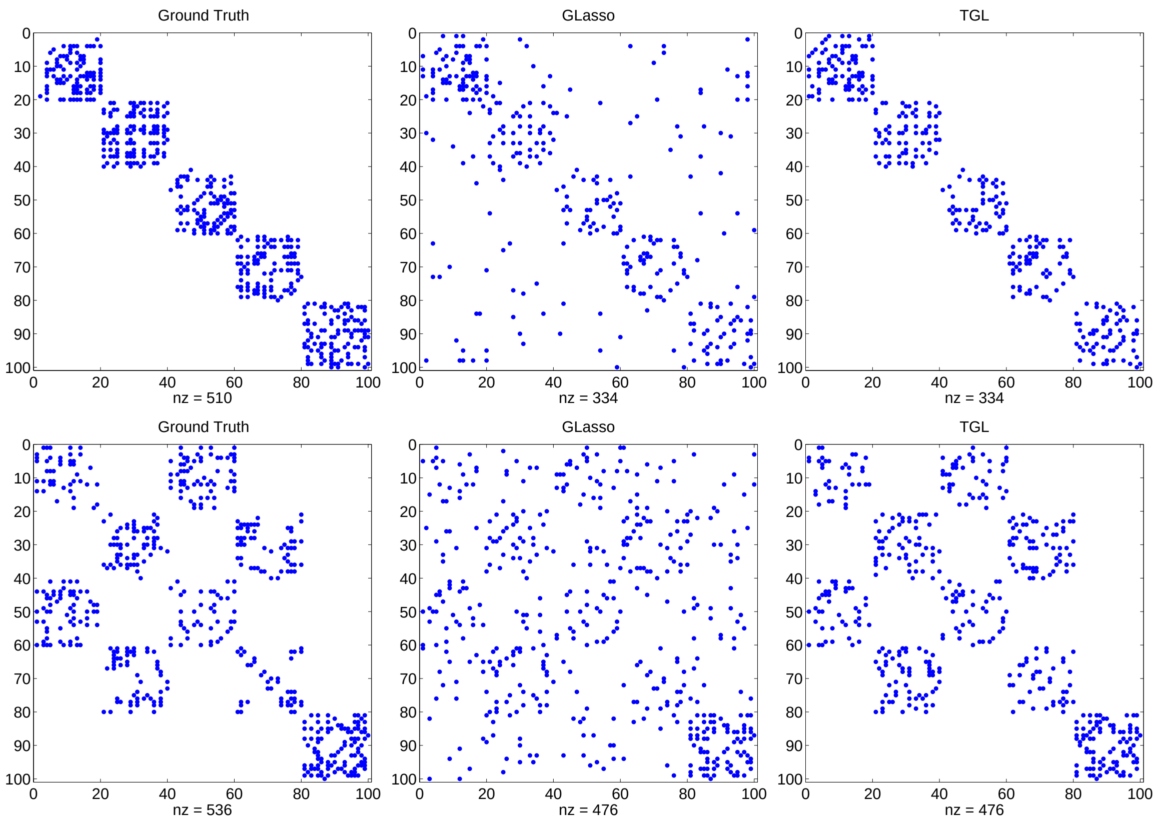


Figure 5.4: Comparison between TGL and GLasso in terms of edge detection. Left: the ground truth precision matrix; middle: the precision matrix estimated by GLasso; right: the precision matrix estimated by TGL.

5.4.1 Synthetic Data

The synthetic covariance matrix is generated in a similar way to Yang *et al.* (2013a): we first generate the ground truth precision matrix $\mathbf{\Theta}$ with random block nonzero patterns. Each nonzero block has a random sparse structure. Given the precision matrix $\mathbf{\Theta}$, we sample from Gaussian distribution to compute the sample covariance matrix. The weights for tree structured group regularization take the form of $w_{ii'}^j = \frac{\rho}{\sqrt{|\mathbf{\Theta}_{ii'}^j|}}$, where ρ is a given positive parameter and $|\mathbf{\Theta}_{ii'}^j|$ is the number of elements in $\mathbf{\Theta}^j_{ii'}$. We control the regularization parameters of TGL and GLasso to ensure the edge number of both estimations to be the same, so that a fair comparison can be made.

Figure 5.4 shows the comparison between TGL and GLasso in terms of edge detection. The first column of Figure 5.4 shows the nonzero patterns (i.e. edges) of two ground truth precision matrices. In both cases, the same index tree is used, which is $\{G^3_i = \{i\}, i = 1, \ldots, 100; G^2_i = \{20i + 1 : 20(i + 1)\}, i = 0, \ldots, 4; G^1_1 = \{1 : 60\}, G^1_2 = \{61 : 100\}\}$. We can observe from Figure 5.4 that the nonzero patterns of the precision matrices estimated by TGL are more similar to the ground truth than GLasso, demonstrating the superiority of TGL over GLasso.

Screening

We conduct experiments to show the effectiveness of the proposed screening rule. NSPG is terminated when $\frac{\|\mathbf{\Theta}^{(k)}_r - \mathbf{\Theta}^{(k)}_{r-1}\|_\infty}{\|\mathbf{\Theta}^{(k)}_{r-1}\|_\infty} \leq 1e\text{-}6$. TGL is terminated when the relative error of the objective value is smaller than $1e$-5. The used index tree is given by $\{G^3_i = \{i\}, i = 1, \ldots, p; G^2_i = \{\frac{ip}{2L} + 1 : \frac{(i+1)p}{2L}\}, i = 0, \ldots, 2L - 1; G^1_i = \{\frac{ip}{L} + 1 : \frac{(i+1)p}{L}\}, i = 0, \ldots, L - 1;\}$, where L is the number of blocks. We can observe from Table 5.1 that the computational time of screening is negligible compared with solving multiple TGL (i.e., TGLs). Since the complexity of identifying the connected components is $\mathrm{O}(\|\mathbf{\Theta}^*\|_0)$, the computational time of screening is almost linear with respect to $\|\mathbf{\Theta}^*\|_0$. Table 5.1 shows that the screening rule can achieve great computational gain. The larger the L is, the higher the speedup is.

5.4.2 Real Data

We apply the proposed TGL method to the voxel-level gene expression and brain connectivity data from Allen Developing Mouse Brain Atlas (2013) to demonstrate

Table 5.1: Comparison of the proposed TGL with and without screening in terms of average computational time (seconds). TGL-S is TGL with screening. The computational time of TGL-S is the summation of screening and TGLs. p stands for the dimension, and L is the number of blocks. $\|\mathbf{\Theta}\|_0$ represents the total number of nonzero entries in ground truth precision matrix $\mathbf{\Theta}$, and $\|\mathbf{\Theta}^*\|_0$ is the number of nonzeros in the solution.

Data setting				Computational time (seconds)		
p	L	$\|\mathbf{\Theta}\|_0$	$\|\mathbf{\Theta}^*\|_0$	TGL-S		TGL
				screening	TGLs	
1000	5	11442	11914	**0.0109**	**0.1715**	2.8219
2000		23694	23854	**0.0395**	**1.0839**	12.2679
1000	10	11142	9782	**0.0105**	**0.2286**	6.481
2000		23308	23862	**0.0366**	**0.4257**	19.1117

the effectiveness of TGL and the proposed screening rule. The data consists of 1724 genes and 7796 structural voxels. The structural voxels have a hierarchical structure which can be obtained from Allen Developing Mouse Brain Atlas (2013). We use such hierarchical structure as the input prior knowledge for our algorithm TGL. We compare TGL with standard GLasso on this data. Figure 5.5 shows the comparison between the precision matrices estimated by TGL and GLasso. From Figure 5.5, we can see that the precision matrix estimated by TGL has a clear pattern which fits the input hierarchical structure. To obtain a solution with precision $1e$-6, the computational time of TGL is 57189.6 seconds. Applying the screening, the computational time of TGL-S (with screening) is reduced to 2781.5 seconds, demonstrating the superiority of the proposed screening rule.

5.5 Conclusion

In this chapter, we propose a hierarchical graphical model framework called tree-guided graphical lasso. The second-order method is employed to solve the proposed

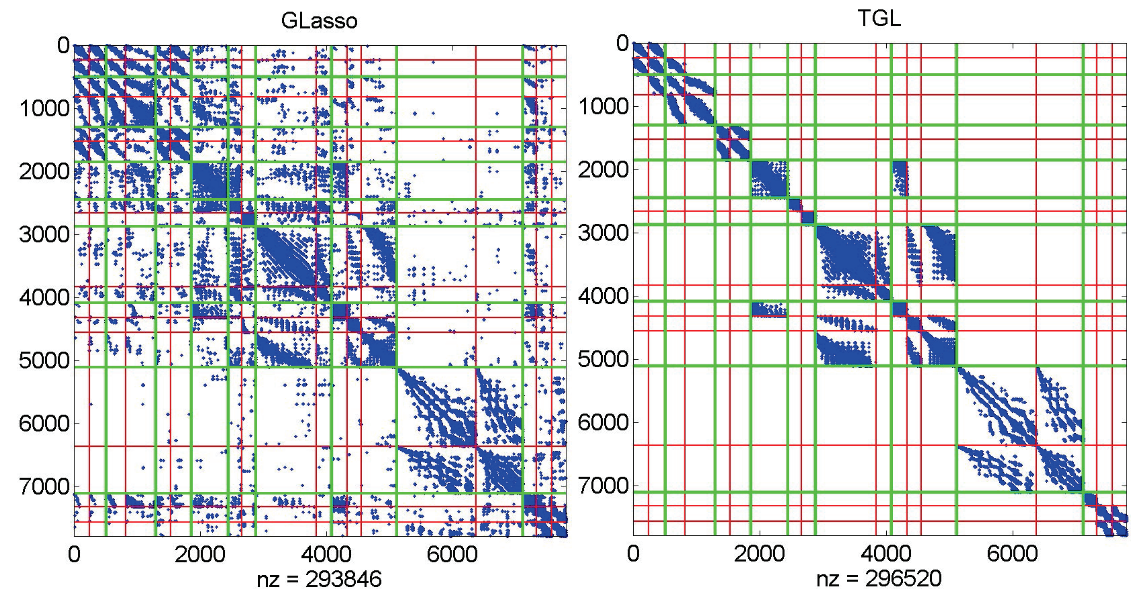


Figure 5.5: Comparison between TGL and GLasso in terms of edge detection on Allen developing mouse brain atlas data. Left: the precision matrix estimated by GLasso; right: the precision matrix estimated by TGL. The red and green grids visualize the tree structured groups in two layers.

formulation. In addition, we derive a necessary and sufficient condition for the TGL solution to be block diagonal. Based on this condition, a simple screening rule has been developed to allow our algorithm scaling up to the large-scale problems. Numerical experiments on synthetic and real data demonstrate the efficiency and effectiveness of the proposed method and the screening rule.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The main goal of this dissertation is to uncover the structural information from the high-dimensional data, and is to develop flexible and advanced learning algorithms with integration of the structural information (e.g., graph) to improve the learning performance. In this dissertation, we focus on addressing the following two specific questions:

- How can we select and group relevant features from high-dimensional and noisy data while taking advantage of structural information among features?
- How can we estimate graphs with certain structure from data in a fast and reliable way?

For the first question, we propose three new feature grouping and selection methods incorporating graph structural information to improve the performance of feature selection and grouping. The first method employs a convex function to penalize the pairwise l_∞ norm of connected regression/classification coefficients, achieving simultaneous feature grouping and selection. The second method improves the first one by utilizing a non-convex function to reduce the estimation bias. The third one is the extension of the second method using a truncated l_1 regularization to further reduce the estimation bias. The proposed methods combine feature grouping and feature selection to enhance estimation accuracy. We employ the alternating direction method of multipliers (ADMM) and difference of convex functions (DC) programming to solve the proposed formulations. Our experimental results on synthetic data and two real datasets demonstrate the effectiveness of the proposed methods.

In addition, we also consider a special case of graph-based sparse learning algorithm, anisotropic total variation regularization problem, which has important applications in signal processing including image denoising, image blurring, and image reconstruction. We propose an efficient optimization of multidimensional total variation regularization problems. The key contribution is to decompose the original problem into a set of independent and small problems which can be solved in parallel. Thus, the proposed algorithm can handle large-scale problems efficiently. Our experimental results show that our algorithm is more efficient than state-of-the-art methods.

For the second question, we consider the problem of estimating multiple graphical models simultaneously using the fused lasso penalty, which encourages adjacent graphs to share similar structures. We propose a second-order method to solve the proposed formulation. The developed approach is applied to the analysis of brain networks of Alzheimer's disease. Our preliminary results show that joint estimation of multiple graphical models leads to a better result than current state-of-the-art methods. To allow our method scaling up to large-scale problems, we establish a necessary and sufficient screening rule, which decomposes the large graphs into small subgraphs and allows an efficient estimation of multiple independent (small) subgraphs. Thus, a huge computational gain can be achieved. In addition to fused penalty, we extend our approaches and screening rule to other general structural penalties, such as overlapping group penalty and tree group structural penalty.

6.2 Future Work

The proposed work can be improved from the following aspects. It will be very interesting to develop a distributed and efficient solution for general graph-based sparse learning problems similar to that for the anisotropic total variation regularization problem. One possible solution is to cluster the nodes within a graph into several connected components such that few connections between different connected components exists. One computer is assigned to solve a small graph-based sparse learning restricted to one connected component. Stochastic ADMM can be employed to solve this problem. The partition of the graph into connected components is important, since it directly affects the communication among computers.

We mainly consider the undirected graph models (i.e., Gaussian graphical model) in this dissertation. One limitation of undirected graph models is that it does not reflect the causal information among the variables. Directed graph models such as directed acyclic graphical models are widely used to make causal inferences for the random variables in multivariate systems. It will be interesting to integrate directed acyclic graph structure information into learning processes in order to improve the learning performance. Estimating directed acyclic graphs from data is also an interesting and challenging problem for future research.

REFERENCES

Allen Developing Mouse Brain Atlas, "http://developingmouse.brain-map.org", (2013).

Azari, N., S. Rapoport, C. Grady, M. Schapiro, J. Salerno, A. Gonzales-Aviles and B. Horwitz, "Patterns of interregional correlations of cerebral glucose metabolic rates in patients with dementia of the alzheimer type", Neurodegeneration **1**, 101–111 (1992).

Bach, F., G. Lanckriet and M. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm", in "Proceedings of The 21th International Conference on Machine Learning (ICML)", (2004).

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

Barbero, A. and S. Sra, "Fast newton-type methods for total variation regularization", in "Proceedings of The 28th International Conference on Machine Learning (ICML)", (2011).

Beck, A. and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems", Image Processing, IEEE Transactions on **18**, 11, 2419–2434 (2009a).

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

Bertsekas, D. and J. N. Tsitsiklis, "Introduction to linear optimization", (1997).

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

Boyd, S., 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 **3**, 1, 1–122 (2011).

Boyd, S. and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, 2004).

Byrd, R. H., J. Nocedal and F. Oztoprak, "An inexact successive quadratic approximation method for convex l-1 regularized optimization", arXiv:1309.3529 [math] (2013).

Chambolle, A., "An algorithm for total variation minimization and applications", Journal of Mathematical imaging and vision **20**, 1, 89–97 (2004).

Chuang, H., E. Lee, Y. Liu, D. Lee and T. Ideker, "Network-based classification of breast cancer metastasis", Molecular systems biology **3**, 1 (2007).

Combettes, P. and J. Pesquet, "Proximal splitting methods in signal processing", Fixed-Point Algorithms for Inverse Problems in Science and Engineering pp. 185–212 (2011).

Condat, L., "A direct algorithm for 1-d total variation denoising", IEEE Signal Processing Letters **20**, 1054–1057 (2013).

Danaher, P., 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) (2013).

d'Aspremont, A., 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).

Dhara, A. and J. Dutta, *Optimality Conditions in Convex Optimization, A finite-dimensional view* (CRC Press, 2012).

Dinh, Q., A. Kyrillidis and V. Cevher, "A proximal newton framework for composite minimization: Graph learning without cholesky decompositions and matrix inversions", arXiv preprint arXiv:1301.1459 (2013).

Eckstein, J. and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators", Mathematical Programming **55**, 1, 293–318 (1992).

Fei, H., B. Quanz and J. Huan, "Regularization and feature selection for networked features", in "Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM)", pp. 1893–1896 (2010).

Friedman, J., T. Hastie, H. Höfling and R. Tibshirani, "Pathwise coordinate optimization", The Annals of Applied Statistics **1**, 2, 302–332 (2007).

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

Goldstein, T. and S. Osher, "The split Bregman method for l1-regularized problems", SIAM Journal on Imaging Sciences **2**, 2, 323–343 (2009).

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

Hara, S. and T. Washio, "Common substructure learning of multiple graphical gaussian models", MLKDD pp. 1–16 (2011).

He, B. and X. Yuan, "On the o(1/n) convergence rate of the Douglas-Rachford alternating direction method", SIAM Journal on Numerical Analysis **50**, 2, 700–709 (2012).

Honorio, J. and D. Samaras, "Multi-task learning of gaussian graphical models", in "Proceedings of The 27th International Conference on Machine Learning (ICML)", (2010).

Horwitz, B., C. Grady, N. Schlageter, R. Duara and S. Rapoport, "Intercorrelations of regional cerebral glucose metabolic rates in alzheimer's disease", Brain research **407**, 2, 294–306 (1987).

Hsieh, C., I. Dhillon, P. Ravikumar and A. Banerjee, "A divide-and-conquer method for sparse inverse covariance estimation", in "Advances in Neural Information Processing Systems (NIPS)", pp. 2339–2347 (2012).

Hsieh, C., M. A. Sustik, I. S. Dhillon and P. Ravikumar, "Sparse inverse covariance matrix estimation using quadratic approximation", Advances in Neural Information Processing Systems (NIPS) **24** (2011).

Huang, J., S. Zhang and D. Metaxas, "Efficient MR image reconstruction for compressed mr imaging", Medical Image Analysis **15**, 5, 670–679 (2011).

Huang, S., J. Li, L. Sun, J. Liu, T. Wu, K. Chen, A. Fleisher, E. Reiman and J. Ye, "Learning brain connectivity of Alzheimer's disease from neuroimaging data", Advances in Neural Information Processing Systems (NIPS) **22**, 808–816 (2009).

Jacob, L., G. Obozinski and J. Vert, "Group lasso with overlap and graph lasso", in "Proceedings of The 26th International Conference on Machine Learning (ICML)", pp. 433–440 (2009).

Jenatton, R., J. Mairal, G. Obozinski and F. Bach, "Proximal methods for sparse hierarchical dictionary learning", in "Proceedings of The 27th International Conference on Machine Learning (ICML)", (2010).

Joachims, T., "Making large-scale support vector machine learning practical", in "Advances in kernel methods", pp. 169–184 (MIT Press, 1999).

Kim, S. and E. Xing, "Statistical estimation of correlated genome associations to a quantitative trait network", PLoS genetics **5**, 8, e1000587 (2009).

Kolar, M., L. Song, A. Ahmed and E. Xing, "Estimating time-varying networks", The Annals of Applied Statistics **4**, 1, 94–123 (2010).

Kolar, M. and E. Xing, "On time varying undirected graphs", in "International Conference on Artificial Intelligence and Statistics (AISTAT)", (2011).

Lee, J. D., Y. Sun and M. A. Saunders, "Proximal newton-type methods for minimizing convex objective functions in composite form", arXiv preprint arXiv:1206.1623 (2012).

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

Li, C., W. Yin and Y. Zhang, "Tval3: Tv minimization by augmented lagrangian and alternating direction algorithms", (2009).

Li, L. and K. Toh, "An inexact interior point method for l 1-regularized sparse covariance selection", Mathematical Programming Computation **2**, 3, 291–315 (2010).

Liu, H., K. Roeder and L. Wasserman, "Stability approach to regularization selection (StARS) for high dimensional graphical models", in "Advances in Neural Information Processing Systems (NIPS)", (2011).

Liu, J. and J. Ye, "Moreau-Yosida regularization for grouped tree structure learning", Advances in Neural Information Processing Systems (NIPS) (2010).

Liu, J., L. Yuan and J. Ye, "An efficient algorithm for a class of fused lasso problems", in "Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)", pp. 323–332 (2010).

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

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

Lu, Z. and Y. Zhang, "An augmented lagrangian approach for sparse principal component analysis", Mathematical Programming pp. 1–45 (2011).

Ma, S., W. Yin, Y. Zhang and A. Chakraborty, "An efficient algorithm for compressed mr imaging using total variation and wavelets", in "IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pp. 1–8 (2008).

Mazumder, R. and T. Hastie, "Exact covariance thresholding into connected components for large-scale graphical lasso", The Journal of Machine Learning Research **13**, 781–794 (2012a).

Mazumder, R. and T. Hastie, "The graphical lasso: New insights and alternatives", Electronic Journal of Statistics **6**, 2125–2149 (2012b).

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

Mohan, K., M. Chung, S. Han, D. Witten, S. Lee and M. Fazel, "Structured learning of gaussian graphical models", in "Advances in Neural Information Processing Systems (NIPS)", pp. 629–637 (2012).

Nesterov, Y., "Smooth minimization of non-smooth functions", Mathematical Programming **103**, 1, 127–152 (2005).

Oberlin, C. and S. J. Wright, "Active set identification in nonlinear programming", SIAM Journal on Optimization **17**, 2, 577–605 (2006).

Olsen, P. A., F. Oztoprak, J. Nocedal and S. J. Rennie, "Newton-like methods for sparse inverse covariance estimation", in "Advances in Neural Information Processing Systems (NIPS)", (2012).

Qin, Z., D. Goldfarb and S. Ma, "An alternating direction method for total variation denoising", arXiv preprint arXiv:1108.1587 (2011).

Rinaldo, A., "Properties and refinements of the fused lasso", The Annals of Statistics **37**, 5B, 2922–2952 (2009).

Rudin, L., S. Osher and E. Fatemi, "Nonlinear total variation based noise removal algorithms", Physica D: Nonlinear Phenomena **60**, 1, 259–268 (1992).

Scheinberg, K., S. Ma and D. Goldfarb, "Sparse inverse covariance selection via alternating linearization methods", in "Advances in Neural Information Processing Systems (NIPS)", (2010).

Scheinberg, K. and I. Rish, "Sinco-a greedy coordinate ascent method for sparse inverse covariance selection problem", preprint (2009).

Scheinberg, K. and X. Tang, "Practical inexact proximal quasi-newton method with global complexity analysis", arXiv preprint arXiv:1311.6547v3 (2014).

Schmidt, M., N. Roux and F. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization", Advances in Neural Information Processing Systems (NIPS) (2011).

Shen, X. and H. Huang, "Grouping pursuit through a regularization solution surface", Journal of the American Statistical Association **105**, 490, 727–739 (2010).

Shen, X. and J. Ye, "Adaptive model selection", Journal of the American Statistical Association **97**, 457, 210–221 (2002).

Sion, M., "On general minimax theorems", Pacific J. Math **8**, 1, 171–176 (1958).

Tao, P. and L. An, "Convex analysis approach to DC programming: Theory, algorithms and applications", Acta Math. Vietnam **22**, 1, 289–355 (1997).

Tao, P. and S. El Bernoussi, "Duality in DC (difference of convex functions) optimization. subgradient methods", Trends in Mathematical Optimization **84**, 277–293 (1988).

Tibshirani, R., "Regression shrinkage and selection via the lasso", Journal of the Royal Statistical Society. Series B pp. 267–288 (1996).

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

Tseng, P. and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization", Mathematical Programming **117**, 1, 387–423 (2009).

Tzourio-Mazoyer, N., B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer and M. Joliot, "Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain", Neuroimage **15**, 1, 273–289 (2002).

Umich, "http://www.umich.edu/∼cogneuro/jpg/brodmann.html", (2014).

Vogel, C. and M. Oman, “Fast, robust total variation-based reconstruction of noisy, blurred images”, Image Processing, IEEE Transactions on **7**, 6, 813–824 (1998).

Wahlberg, B., S. Boyd, M. Annergren and Y. Wang, “An admm algorithm for a class of total variation regularized estimation problems”, IFAC (2012).

Wang, C., 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).

Wang, J., Q. Li, S. Yang, W. Fan, P. Wonka and J. Ye, “A highly scalable parallel algorithm for isotropic total variation models”, in “Proceedings of The 31th International Conference on Machine Learning (ICML)”, (2014).

Wang, K., M. Liang, L. Wang, L. Tian, X. Zhang, K. Li and T. Jiang, “Altered functional connectivity in early alzheimer’s disease: A resting-state fMRI study”, Human brain mapping **28**, 10, 967–978 (2007).

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

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

Xiang, S., X. Shen and J. Ye, “Efficient sparse group feature selection via nonconvex optimization”, in “Proceedings of The 30th International Conference on Machine Learning (ICML)”, pp. 284–292 (2013a).

Xiang, S., T. Yang and J. Ye, “Simultaneous feature and feature group selection through hard thresholding”, in “Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 532–541 (ACM, 2014).

Xiang, S., L. Yuan, W. Fan, Y. Wang, P. M. Thompson and J. Ye, “Multi-source learning with block-wise missing data for alzheimer’s disease prediction”, in “Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 185–193 (ACM, 2013b).

Yang, J., Y. Zhang and W. Yin, “An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise”, SIAM Journal on Scientific Computing **31**, 4, 2842–2865 (2009).

Yang, S., Z. Lu, X. Shen, P. Wonka and J. Ye, “Fused multiple graphical lasso”, arXiv preprint arXiv:1209.2139v2 (2013a).

Yang, S., J. Wang, W. Fan, Z. Xiatian, P. Wonka and J. Ye, “An efficient ADMM algorithm for multi dimensional anisotropic total variation regularization problems”, in “Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)”, (2013b).

Yang, S., L. Yuan, Y. Lai, X. Shen, P. Wonka and J. Ye, “Feature grouping and selection over an undirected graph”, in “Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)”, pp. 922–930 (ACM, 2012).

Yuan, G., C. Ho and C. Lin, “An improved glmnet for l1-regularized logistic regression”, The Journal of Machine Learning Research **13**, 1999–2030 (2012).

Yuan, L., J. Liu and J. Ye, “Efficient methods for overlapping group lasso”, Advances in Neural Information Processing Systems (NIPS) (2011).

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

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

Yuan, X., “Alternating direction method for covariance selection models”, Journal of Scientific Computing **51**, 2, 261–273 (2012).

Zhang, T., “Multi-stage convex relaxation for feature selection”, Bernoulli **19**, 5B, 2277–2293 (2013).

Zhao, P., G. Rocha and B. Yu, “The composite absolute penalties family for grouped and hierarchical variable selection”, The Annals of Statistics **37**, 6A, 3468–3497 (2009).

Zhong, L. and J. Kwok, “Efficient sparse modeling with automatic feature grouping”, Proceedings of The 28th International Conference on Machine Learning (ICML) (2011).

Zhou, S., J. Lafferty and L. Wasserman, “Time varying undirected graphs”, Machine Learning **80**, 2-3, 295–319 (2010).

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