entropy

MDPI

Article

Efficient Proximal Gradient Algorithms for Joint Graphical Lasso

Jie Chen *, Ryosuke Shimmura and Joe Suzuki

Graduate School of Engineering Science, Osaka University, Osaka 560-0043, Japan; shimmura@sigmath.es.osaka-u.ac.jp (R.S.); j-suzuki@sigmath.es.osaka-u.ac.jp (J.S.)
* Correspondence: chen@sigmath.es.osaka-u.ac.jp

Abstract: We consider learning as an undirected graphical model from sparse data. While several efficient algorithms have been proposed for graphical lasso (GL), the alternating direction method of multipliers (ADMM) is the main approach taken concerning joint graphical lasso (JGL). We propose proximal gradient procedures with and without a backtracking option for the JGL. These procedures are first-order methods and relatively simple, and the subproblems are solved efficiently in closed form. We further show the boundedness for the solution of the JGL problem and the iterates in the algorithms. The numerical results indicate that the proposed algorithms can achieve high accuracy and precision, and their efficiency is competitive with state-of-the-art algorithms.

Keywords: Gaussian graphical model; joint graphical lasso; proximal gradient descent method

Citation: Chen, J.; Shimmura, R.; Suzuki, J. Efficient Proximal Gradient Algorithms for Joint Graphical Lasso. *Entropy* **2021**, *23*, 1623. https://doi.org/10.3390/e23121623

Academic Editor: Mohamed Medhat Gaber

Received: 4 November 2021
Accepted: 27 November 2021
Published: 2 December 2021

1. Introduction

Graphical models are widely used to describe the relationships among interacting objects [1]. Such models have been extensively used in various domains, such as bioinformatics, text mining, and social networks. The graph provides a visual way to understand the joint distribution of an entire set of variables.

In this paper, we consider learning Gaussian graphical models that are expressed by undirected graphs, which represent the relationship among continuous variables that follow a joint Gaussian distribution. In an undirected graph, $\mathcal{G} = (V, E)$, and edge set E represents the conditional dependencies among the variables in vertex set V.

Let $X_1, \ldots, X_p$ $(p \geq 1)$ be Gaussian variables with covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{p \times p}$, and $\mathbf{\Theta} := \mathbf{\Sigma}^{-1}$ be the precision matrix, if it exists. We remove the edges so that the variables X_i, X_j are conditionally independent given the other variables if and only if the (i, j)-th element $\theta_{i,j}$ in $\mathbf{\Theta}$ is 0:

$$\{i, j\} \notin E \iff \theta_{i,j} = 0 \iff X_i \perp\!\!\!\perp X_j | X_{V \setminus \{i,j\}},$$

where each edge is expressed as a set of two elements in $\{1, \ldots, p\}$. In this sense, constructing a Gaussian graphical model is equivalent to estimating a precision matrix.

Suppose that we estimate the undirected graph from data consisting of n tuples of p variables and that dimension p is much higher than sample size n. For example, if we have expression data of $p = 20,000$ genes for $n = 100$ case/control patients, how can we construct a gene regulatory network structure from the data? It is almost impossible to estimate the locations of the nonzero elements in $\mathbf{\Theta}$ by obtaining the inverse of sample covariance matrix $\boldsymbol{S} \in \mathbb{R}^{p \times p}$, which is the unbiased estimator of $\mathbf{\Sigma}$. In fact, if $p > n$, then no inverse $\boldsymbol{S}^{-1}$ exists because the rank of $\boldsymbol{S} \in \mathbb{R}^{p \times p}$ is, at most, n.

In order to address this situation, two directions are suggested:

1. Sequentially find the variables on which each variable depends via regression so that the quasilikelihood is maximized [2].

2. Find the locations in $\mathbf{\Theta}$, the values of which are zeros, so that the ℓ_1 regularized log-likelihood is maximized [3–6].

We follow the second approach because we assume Gaussian variables, also known as graphical lasso (GL). The ℓ_1 regularized log-likelihood is defined by:

$$\underset{\mathbf{\Theta}}{\text{maximize}}\{\log\det\mathbf{\Theta} - \text{trace}(\boldsymbol{S}\mathbf{\Theta}) - \lambda||\mathbf{\Theta}||_1\}, \tag{1}$$

where tuning parameter λ controls the amount of sparsity, and $||\mathbf{\Theta}||_1$ denotes the sum of the absolute value of the off-diagonal elements in $\mathbf{\Theta}$. Several optimization techniques [4,7–12] have been studied for the optimization problem of (1).

In particular, we consider a generalized version of the abovementioned GL. For example, suppose that the gene regulatory networks of thirty case and seventy control patients are different. One might construct a gene regulatory network separately for each of the two categories. However, estimating each on its own does not provide an advantage if a common structure is shared. Instead, we use 100 samples to construct two networks simultaneously. Intuitively speaking, using both types of data improves the reliability of the estimation by increasing the sample size for the genes that show similar values between case and control patients, while using only one type of data leads to a more accurate estimate for genes that show significantly different values. Ref. [13] proposed a joint graphical lasso (JGL) model by including an additional convex penalty (fused or group lasso penalty) to the graphical lasso objective function for K classes. For example, K is equal to two for the case/control patients in the example. JGL includes fused graphical lasso with fused lasso penalty, which encourages sparsity and the similarity of the value of edges across K classes, and group graphical lasso with group lasso penalty, which promotes similar sparsity structure across K graphs. Although there are several approaches to handling the multiple graphical models, such as those of [14–17], the JGL is considered the most promising.

The main topic of this paper is efficiency improvement in terms of solving the JGL problem. For the GL, relatively efficient solving procedures exist. If we differentiate the ℓ_1 regularized log-likelihood (1) by $\mathbf{\Theta}$, then we have an equation to solve [4]. Moreover, several improvements have been considered for the GL, such as proximal Newton [12] and proximal gradient [10] procedures. However, for the JGL, even if we derive such an equation, we have no efficient way of handling it.

Instead, the alternating direction method of multipliers (ADMM) [18], which is a procedure for solving convex optimization problems for general purposes, has been the main approach taken [13,19–21]. However, ADMM does not scale well concerning the feature dimension p and number of classes K. It usually takes time to converge to a high-accuracy solution [22].

For efficient procedures to solve the JGL problem, ref. [23] proposed a method based on the proximal Newton method only when the penalty term is expressed by fused lasso. The existing method requires expensive computations for the Hessian matrix and Newton directions, which means that it would be expensive to use for high-dimensional problems.

In this paper, we propose efficient proximal-gradient-based algorithms to solve the JGL problem by extending the procedure in [10] and employing the step-size selection strategy proposed in [24]. Moreover, we provide the theoretical analysis of both methods for the JGL problem.

In our proximal gradient methods for the JGL problem, the proximal operator in each iteration is quite simple, which eases the implementation process and requires very little computation and memory at each step. Simulation experiments are used to justify our proposed methods over the existing ones.

Our main contributions are as follows:

- We propose efficient algorithms based on the proximal gradient method to solve the JGL problem. The algorithms are first-order methods and quite simple, and the subproblems can be solved efficiently with a closed-form solution. The numerical

results indicate that the methods can achieve high accuracy and precision, and the computational time is competitive with state-of-art algorithms.
- We provide the boundedness for the solution to the JGL problem and the iterates in algorithms, which is related to the convergence rate of the algorithms. With the boundedness, we can guarantee that our proposed method converges linearly.

Table 1 summarizes the relationship between the proposed and existing methods.

Table 1. Efficient JGL procedures.

Model	ADMM	Proximal Newton	Proximal Gradient
GL [4]	[8]	[12]	[10]
JGL [13]	[13]	[23] (for fused penalty)	Current Paper (for fused and group penalties)

The remaining parts of this paper are as follows. In Section 2, we first provide the background of our proposed methods and introduce the joint graphical lasso problem. In Section 3, we illustrate the detailed content of the proposed algorithms and provide a theoretical analysis. In Section 4, we report some numerical results of the proposed approaches, including comparisons with efficient methods and performance evaluations. Finally, we draw some conclusions in Section 5.

Notation: In this paper, $||\boldsymbol{x}||_p$ denotes the ℓ_p norm of a vector $\boldsymbol{x} \in \mathbb{R}^d$, $||\boldsymbol{x}||_p := (\sum_{i=1}^d |x_i|^p)^{\frac{1}{p}}$ for $p \in [1, \infty)$, and $||\boldsymbol{x}||_\infty := \max_i |x_i|$. For a matrix $\boldsymbol{X} \in \mathbb{R}^{p \times q}$, $||\boldsymbol{X}||_F$ denotes the Frobenius norm, $||\boldsymbol{X}||_2$ denotes the spectral norm, $||\boldsymbol{X}||_\infty := \max_{i,j} |x_{i,j}|$, and $||\boldsymbol{X}||_1 := \sum_{i=1}^p \sum_{j=1}^q |x_{i,j}|$ if not specified. The inner product is defined by $\langle \boldsymbol{X}, \boldsymbol{X} \rangle := \mathrm{trace}(\boldsymbol{X}^T \boldsymbol{X})$.

2. Preliminaries

This section first reviews the graphical lasso (GL) problem and introduces the graphical iterative shrinkage-thresholding algorithm (G-ISTA) [10] to solve it. Then, we introduce the step-size selection strategy that we apply to the joint graphical lasso (JGL) in Section 3.2.

2.1. Graphical Lasso

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ be $n \geq 1$ observations of dimension $p \geq 1$ that follow the Gaussian distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^p$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$, where without loss of generality, we assume $\boldsymbol{\mu} = \boldsymbol{0}$. Let $\boldsymbol{\Theta} := \boldsymbol{\Sigma}^{-1}$, and the empirical covariance matrix $\boldsymbol{S} := \frac{1}{n} \sum_{i=1}^n \boldsymbol{x}_i^T \boldsymbol{x}_i$. Given penalty parameter $\lambda > 0$, the graphical lasso (GL) is the procedure to find the positive definite $\boldsymbol{\Theta} \in \mathbb{R}^{p \times p}$ such that:

$$\underset{\boldsymbol{\Theta}}{\text{minimize}} \{ -\log\det \boldsymbol{\Theta} + \mathrm{trace}(\boldsymbol{S}\boldsymbol{\Theta}) + \lambda \|\boldsymbol{\Theta}\|_1 \}, \tag{2}$$

where $||\boldsymbol{\Theta}||_1 = \sum_{j \neq k} |\theta_{j,k}|$. If we regard $V := \{1, \ldots, p\}$ as a vertex set, then we can construct an undirected graph with edge set $\{\{j,k\} | \theta_{j,k} \neq 0\}$, where set $\{j,k\}$ denotes an undirected edge that connects the nodes $j, k \in V$.

If we take the subgradient of (2), then we find that the optimal solution $\boldsymbol{\Theta}_*$ satisfies the condition:

$$-\boldsymbol{\Theta}_*^{-1} + \boldsymbol{S} + \lambda \boldsymbol{\Phi} \ni 0, \tag{3}$$

where $\boldsymbol{\Phi} = (\Phi_{j,k})$ is

$$\Phi_{j,k} = \begin{cases} 1, & \theta^*_{j,k} > 0 \\ [-1, 1], & \theta^*_{j,k} = 0 \\ -1, & \theta^*_{j,k} < 0 \end{cases}.$$

2.2. ISTA for Graphical Lasso

In this subsection, we introduce the method for solving the GL problem (2) by the iterative shrinkage-thresholding algorithm (ISTA) proposed by [10], which is a proximal gradient method usually employed in dealing with nondifferentiable composite optimization problems.

Specifically, the general ISTA solves the following composite optimization problem:

$$\underset{\boldsymbol{x}}{\text{minimize}}\ F(\boldsymbol{x}) := f(\boldsymbol{x}) + g(\boldsymbol{x}), \tag{4}$$

where f and g are convex, with f differentiable and g possibly being nondifferentiable.

For the GL problem (2), we denote $f, g : \mathbb{R}^{p\times p} \to \mathbb{R}$ as

$$f(\boldsymbol{\Theta}) := -\log\det\boldsymbol{\Theta} + \operatorname{trace}(\boldsymbol{S}\boldsymbol{\Theta}),$$

and

$$g(\boldsymbol{\Theta}) := \lambda\|\boldsymbol{\Theta}\|_1.$$

If we define the quadratic approximation $Q_\eta : \mathbb{R}^{p\times p} \times \mathbb{R}^{p\times p} \to \mathbb{R}$ w.r.t. $f(\boldsymbol{\Theta})$ and $\eta > 0$ by

$$Q_\eta(\boldsymbol{\Theta}', \boldsymbol{\Theta}) := f(\boldsymbol{\Theta}) + \langle \boldsymbol{\Theta}' - \boldsymbol{\Theta}, \nabla f(\boldsymbol{\Theta})\rangle + \frac{1}{2\eta}||\boldsymbol{\Theta}' - \boldsymbol{\Theta}||_F^2, \tag{5}$$

then we can describe the ISTA as a procedure that iterates

$$\boldsymbol{\Theta}_{t+1} = \arg\min_{\boldsymbol{\Theta}}\{Q_{\eta_t}(\boldsymbol{\Theta}, \boldsymbol{\Theta}_t) + g(\boldsymbol{\Theta})\} \tag{6}$$

$$= \operatorname{prox}_{\eta_t g}(\boldsymbol{\Theta}_t - \eta_t\nabla f(\boldsymbol{\Theta}_t)), \tag{7}$$

given initial value $\boldsymbol{\Theta}_0$, where the value of step size $\eta_t > 0$ may change at each iteration $t = 1, 2, \ldots$, for efficient convergence purpose, and we use the proximal operator:

$$\operatorname{prox}_g(\boldsymbol{z}) := \arg\min_{\boldsymbol{\theta}}\{\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{\theta}\|_2^2 + g(\boldsymbol{\theta})\}. \tag{8}$$

Note that the proximal operator of function $g = \lambda||\boldsymbol{\Theta}||_1$ is the soft-thresholding operator: the absolute value $|\theta_{i,j}|$ of each off-diagonal element $\theta_{i,j}$ with $i \neq j$ becoming either $\theta_{i,j} - \operatorname{sgn}(\theta_{i,j})\lambda$ or zero (if $|\theta_{i,j}| < \lambda$). We use the following function for the operator in Section 3:

$$[\mathcal{S}_\lambda(\boldsymbol{\Theta})]_{i,j} = \operatorname{sgn}(\theta_{i,j})(|\theta_{i,j}| - \lambda)_+ \tag{9}$$

where $(x)_+ := \max(x, 0)$.

Definition 1. *A differentiable function $f : \mathbb{R}^{n\times p} \to \mathbb{R}$ is said to have a Lipschitz-continuous gradient if there exists $L > 0$ (Lipschitz constant) such that*

$$||\nabla f(\boldsymbol{X}) - \nabla f(\boldsymbol{Y})||_F \leq L||\boldsymbol{X} - \boldsymbol{Y}||_F, \quad \forall \boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{n\times p}. \tag{10}$$

It is known that if we choose $\eta_t = \frac{1}{L}$ for each step in the ISTA that minimizes $F(\cdot)$, then the convergence rate is, at most, as follows [25]:

$$F(\boldsymbol{\Theta}_t) - F(\boldsymbol{\Theta}_*) = O(\frac{1}{t}) \tag{11}$$

However, for the GL problem (2), we know neither the exact value of the Lipschitz constant L nor any nontrivial upper bound. [10] implement a backtracking line search option in Step 1 of Algorithm 1 below to handle this issue.

The backtracking line search enables us to compute the η_t value for each time $t = 1, 2, \ldots$ by repeatedly multiplying η_t by a constant $c \in (0, 1)$ until $\boldsymbol{\Theta}_{t+1} \succ 0$ ($\boldsymbol{\Theta}$ is positive definite) and

$$f(\boldsymbol{\Theta}_{t+1}) \leq Q_{\eta_t}(\boldsymbol{\Theta}_{t+1}, \boldsymbol{\Theta}_t), \tag{12}$$

for the $\boldsymbol{\Theta}_{t+1}$ in (7). Additionally, (12) is a sufficient condition for (11), which was derived in [25] (see the relationship between Lemma 2.3 and Theorem 3.1 in [25]).

The whole procedure is given in Algorithm 1.

Algorithm 1 G-ISTA for problem (2).

Input: S, tolerance $\epsilon > 0$, backtracking constant $0 < c < 1$, initial value η_0, $\boldsymbol{\Theta}_0$, $t = 0$.
While $t < t_{\max}$ (until convergence) **do**

1: Backtracking line search: Continue to multiply η_t by c until

$$\boldsymbol{\Theta}_{t+1} \succ 0 \quad \text{and} \quad f(\boldsymbol{\Theta}_{t+1}) \leq Q_{\eta_t}(\boldsymbol{\Theta}_{t+1}, \boldsymbol{\Theta}_t)$$

for $\boldsymbol{\Theta}_{t+1} := prox_{\eta_t g}(\boldsymbol{\Theta}_t - \eta_t \nabla f(\boldsymbol{\Theta}_t))$.

2: Update iterate: $\boldsymbol{\Theta}_{t+1} \leftarrow prox_{\eta_t g}(\boldsymbol{\Theta}_t - \eta_t \nabla f(\boldsymbol{\Theta}_t))$.

3: Set next initial step size η_{t+1} by the Barzilai—Borwein method.

4: $t \leftarrow t + 1$

end
Output: ϵ-optimal solution to problem (2), $\boldsymbol{\Theta}_* = \boldsymbol{\Theta}_{t+1}$.

2.3. Composite Self-Concordant Minimization

The notion of the self-concordant function was proposed in [26–28]. In the following, we say a convex function f is self-concordant with parameter $M \geq 0$ if

$$|f'''(\boldsymbol{x})| \leq M f''(\boldsymbol{x})^{3/2}, \text{ for all } \boldsymbol{x} \in \text{dom } f.$$

where dom f is the domain of f.

Reference [24] considered a composite version of self-concordant function minimization and provided a way to efficiently calculate the step size for the proximal gradient method for the GL problem without relying on the Lipschitz gradient assumption in (10). They proved that

$$f(\boldsymbol{\Theta}) := -\log\det\boldsymbol{\Theta} + \text{trace}(S\boldsymbol{\Theta})$$

in (2) is self-concordant and considers the following minimization:

$$F^* := \underset{\boldsymbol{x}}{\text{minimize}} \{F(\boldsymbol{x}) := f(\boldsymbol{x}) + g(\boldsymbol{x})\},$$

where f is convex, differentiable, and self-concordant, and g is convex and possibly non-differentiable. As for Algorithm 1, without using the backtracking line search, we can compute direction $\boldsymbol{d}_t$ with initial step size η_t as follows:

$$\boldsymbol{d}_t := prox_{\eta_t g}(\boldsymbol{\Theta}_t - \eta_t \nabla f(\boldsymbol{\Theta}_t)) - \boldsymbol{\Theta}_t, \tag{13}$$

where the operator $prox$ is defined by (8). Then, we use the modified step size α_t to update $\boldsymbol{\Theta}_{t+1} := \boldsymbol{\Theta}_t + \alpha_t \boldsymbol{d}_t$, which can be determined by the direction $\boldsymbol{d}_t$. After defining two parameters related to the direction: $\beta_t := \eta_t^{-1} ||\boldsymbol{d}_t||_F^2$ and $\lambda_t := \left(\langle \nabla^2 f(\boldsymbol{\Theta}_t)\boldsymbol{d}_t, \boldsymbol{d}_t\rangle\right)^{1/2}$, the modified step size can be obtained by

$$\alpha_t := \frac{\beta_t}{\lambda_t(\lambda_t + \beta_t)}. \tag{14}$$

By Lemma 12 in [24], if the modified step size $\alpha_t \in (0, 1]$, then it can ensure a decrease in the objective function and guarantee convergence in the proximal gradient scheme.

From (14), if $\lambda_t \geq 1$, then the condition $\alpha_t \in (0, 1]$ is satisfied. Therefore, we only need to check the case when $\lambda_t < 1$. If the condition $\alpha_t \in (0, 1]$ does not hold, we can change the value of the initial η_t (such as the bisection method) to influence the value of $\boldsymbol{d}_t$ in (13) until the condition is satisfied.

2.4. Joint Graphical Lasso

Let $N \geq 1, p \geq 1, K \geq 2$, and $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \in \mathbb{R}^p \times \{1, \ldots, K\}$, where each $\boldsymbol{x}_i$ is a row vector. Let n_k be the number of occurrences in $y_1, \ldots, y_N$ such that $y_i = k$, so that $\sum_{k=1}^{K} n_k = N$.

For each $k = 1, \ldots, K$, we define the empirical covariance matrix $\boldsymbol{S}^{(k)} \in \mathbb{R}^{p \times p}$ of the data $\boldsymbol{x}_i$ as follows:

$$\boldsymbol{S}^{(k)} := \frac{1}{n_k} \sum_{i:y_i=k} \boldsymbol{x}_i^T \boldsymbol{x}_i.$$

Given the penalty parameters $\lambda_1 > 0$ and $\lambda_2 > 0$, the joint graphical lasso (JGL) is the procedure to find the positive definite matrix $\boldsymbol{\Theta}^{(k)} \in \mathbb{R}^{p \times p}$ for $k = 1, \ldots, K$, such that:

$$\underset{\boldsymbol{\Theta}}{\text{minimize}} \left\{ -\sum_{k=1}^{K} n_k \{\log\det \boldsymbol{\Theta}^{(k)} - \text{trace}(\boldsymbol{S}^{(k)}\boldsymbol{\Theta}^{(k)})\} + \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\theta_{k,i,j}| + P(\boldsymbol{\Theta}) \right\}, \quad (15)$$

where $P(\boldsymbol{\Theta})$ penalizes $\boldsymbol{\Theta} := [\boldsymbol{\Theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(K)}]^T$. For example, ref. [13] suggested the following fused and group lasso penalties:

$$P_F(\boldsymbol{\Theta}) := \lambda_2 \sum_{k<l} \sum_{i,j} |\theta_{k,i,j} - \theta_{l,i,j}|$$

and

$$P_G(\boldsymbol{\Theta}) := \lambda_2 \sum_{i \neq j} \left\{ \sum_{k=1}^{K} \theta_{k,i,j}^2 \right\}^{1/2},$$

where $\theta_{k,i,j}$ is the (i, j)-th element of $\boldsymbol{\Theta}^{(k)} \in \mathbb{R}^{p \times p}$ for $k = 1, \ldots, K$.

Unfortunately, there is no equation like (3) for the JGL to find the optimum $\boldsymbol{\Theta}_*$. [13] considered the ADMM to solve the JGL problem. However, ADMM is quite time consuming for large-scale problems.

3. The Proposed Methods

In this section, we propose two efficient algorithms for solving the JGL problem. One is an extended ISTA based on the G-ISTA in Section 2.2, and the other is based on the step-size selection strategy introduced in Section 2.3.

3.1. ISTA for the JGL Problem

To describe the JGL problem, we define $f, g : \mathbb{R}^{K \times p \times p} \to \mathbb{R}$ by

$$f(\boldsymbol{\Theta}) := -\sum_{k=1}^{K} n_k \left\{ \log\det \boldsymbol{\Theta}^{(k)} - \text{trace}(\boldsymbol{S}^{(k)}\boldsymbol{\Theta}^{(k)}) \right\}, \quad (16)$$

$$g(\boldsymbol{\Theta}) := \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\theta_{k,i,j}| + P(\boldsymbol{\Theta}). \quad (17)$$

Then, the problem (15) reduces to:

$$\underset{\boldsymbol{\Theta}}{\text{minimize}}\, F(\boldsymbol{\Theta}) := f(\boldsymbol{\Theta}) + g(\boldsymbol{\Theta}),$$

where the function f is convex and differentiable, and g is convex and nondifferentiable. Therefore, the ISTA is available for solving the JGL problem (15).

The main difference between the G-ISTA and the proposed method is that the latter needs to simultaneously consider K categories of graphical models in the JGL problem (15). What is more, there are two combined penalties in $g(\mathbf{\Theta})$, which complicate the proximal operator in the ISTA procedure. Consequently, the operator for the proposed method is not a simple soft thresholding operator, as it is for the G-ISTA method.

If we define the quadratic approximation $Q_{\eta_t} : \mathbb{R}^{K\times p\times p} \to \mathbb{R}$ of $f(\mathbf{\Theta}_t)$ by:

$$Q_{\eta_t}(\mathbf{\Theta}, \mathbf{\Theta}_t) := f(\mathbf{\Theta}_t) + \sum_{k=1}^{K} \left\langle \mathbf{\Theta}^{(k)} - \mathbf{\Theta}_t^{(k)}, \nabla f(\mathbf{\Theta}_t^{(k)}) \right\rangle + \frac{1}{2\eta_t} \sum_{k=1}^{K} ||\mathbf{\Theta}^{(k)} - \mathbf{\Theta}_t^{(k)}||_F^2,$$

then the update iteration is simplified as:

$$\begin{aligned} \mathbf{\Theta}_{t+1} &= \underset{\mathbf{\Theta}}{\operatorname{argmin}} \left\{ Q_{\eta_t}(\mathbf{\Theta}, \mathbf{\Theta}_t) + g(\mathbf{\Theta}) \right\} \\ &= \operatorname{prox}_{\eta_t g}(\mathbf{\Theta}_t - \eta_t \nabla f(\mathbf{\Theta}_t)). \end{aligned}$$

Nevertheless, the Lipschitz gradient constant of $f(\mathbf{\Theta})$ is unknown over the whole domain in the JGL problem. Therefore, our approach needs a backtracking line search to calculate step size η_t. We show the details in Algorithm 2.

Algorithm 2 ISTA for problem (15).

Input: S, tolerance $\epsilon > 0$, backtracking constant $0 < c < 1$, initial step size η_0, initial iterate $\mathbf{\Theta}_0$.

For $t = 0, 1, \cdots$, (until convergence) **do**

1: Backtracking line search: Continue to multiply η_t by c until

$$f(\mathbf{\Theta}_{t+1}) \le Q_{\eta_t}(\mathbf{\Theta}_{t+1}, \mathbf{\Theta}_t) \text{ and } \mathbf{\Theta}_{t+1}^{(k)} \succ 0 \text{ for } k = 1, \cdots, K.$$

for $\mathbf{\Theta}_{t+1} := \operatorname{prox}_{\eta_t g}(\mathbf{\Theta}_t - \eta_t \nabla f(\mathbf{\Theta}_t))$.

2: Update iterate: $\mathbf{\Theta}_{t+1} \leftarrow \operatorname{prox}_{\eta_t g}(\mathbf{\Theta}_t - \eta_t \nabla f(\mathbf{\Theta}_t))$.

3: Set next initial step size η_{t+1}. See details in Section 3.3.

end

Output: optimal solution to problem (15), $\mathbf{\Theta}_* = \mathbf{\Theta}_{t+1}$.

In the update of $\mathbf{\Theta}_{t+1}$, we need to compute the proximal operators for the fused and group lasso penalties. In the following, for each of them, the problem can be divided into the fused lasso problems [29] and group lasso problems [30,31] for $\boldsymbol{\theta}_{i,j} \in \mathbb{R}^K, i, j = 1, \ldots, p$. We apply the solutions given by (20) and (21) below.

3.1.1. Fused Lasso Penalty P_F

By the definition of the proximal operator in the update step, we have:

$$\begin{aligned} \mathbf{\Theta}_{t+1} = \arg\min_{\mathbf{\Theta}} \Bigg\{ \frac{1}{2} \sum_{k=1}^{K} ||\mathbf{\Theta}^{(k)} - \mathbf{\Theta}_t^{(k)} + \eta_t \nabla f(\mathbf{\Theta}_t^{(k)})||_F^2 + \eta_t \lambda_1 \sum_{k=1}^{K} \sum_{i\neq j} |\theta_{k,i,j}| \\ + \eta_t \lambda_2 \sum_{k<l} \sum_{i,j} |\theta_{k,i,j} - \theta_{l,i,j}| \Bigg\}. \end{aligned} \tag{18}$$

Problem (18) is separable with respect to the elements $\theta_{k,i,j}$ in $\mathbf{\Theta}^{(k)} \in \mathbb{R}^{p\times p}$; hence, the proximal operator can be computed in a componentwise manner: Let $\boldsymbol{A} = \mathbf{\Theta}_t - \eta_t \nabla f(\mathbf{\Theta}_t)$; then, problem (18) reduces to the following for $i = 1, \cdots, p$, $j = 1, \cdots, p$:

$$\underset{\theta_{1,i,j},\cdots,\theta_{K,i,j}}{\operatorname{argmin}} \left\{ \frac{1}{2}\sum_{k=1}^{K}(\theta_{k,i,j} - a_{k,i,j})^2 + \eta_t\lambda_1 1_{i\neq j}\sum_{k=1}^{K}|\theta_{k,i,j}| + \eta_t\lambda_2\sum_{k<l}|\theta_{k,i,j} - \theta_{l,i,j}|\} \right\}, \tag{19}$$

where $1_{i\neq j}$ is an indicator function, the value of which is 1 only when $i \neq j$.

The problem (19) is known as the fused lasso problem [29,32] given $a_{k,i,j}$ for $k = 1, \ldots, K$. In particular, let $\alpha := \eta_t\lambda_1 1_{i\neq j}$ and $\beta := \eta_t\lambda_2$. When $i \neq j$, $\alpha \neq 0$ and $\beta > 0$, the solution to (19) can be obtained through the soft thresholding operator based on the solution when $\alpha = 0$ by the following Lemma.

Lemma 1. *([33]) Denote the solution to parameters α and β as $\theta_i(\alpha, \beta)$, and then the solution $\theta_i(\alpha, \beta)$ of the fused lasso problem:*

$$\frac{1}{2}\sum_{i=1}^{n}(y_i - \theta_i)^2 + \alpha\sum_{i=1}^{n}|\theta_i| + \beta\sum_{i=1}^{n-1}|\theta_i - \theta_{i+1}| \tag{20}$$

is given by $[S_\alpha(\theta(0, \beta))]_i$ when $y_1, \cdots, y_n \in \mathbb{R}$ are given for $n \geq 1$.

Additionally, rather efficient algorithms are available for solving the fused lasso problem (20) when $\alpha = 0$ (i.e., $\theta(0, \beta)$) such as [32,34,35].

3.1.2. Group Lasso Penalty P_G

By definition, the update of $\mathbf{\Theta}_{t+1}$ for the group lasso penalty $P_G(\mathbf{\Theta})$ is as follows:

$$\mathbf{\Theta}_{t+1} = \arg\min_{\mathbf{\Theta}}\left\{ \frac{1}{2}\sum_{k=1}^{K}||\mathbf{\Theta}^{(k)} - \mathbf{\Theta}_t^{(k)} + \eta_t\nabla f(\mathbf{\Theta}_t^{(k)})||_F^2 + \eta_t\lambda_1\sum_{k=1}^{K}\sum_{i\neq j}|\theta_{k,i,j}| \right.$$
$$\left. +\eta_t\lambda_2\sum_{i\neq j}(\sum_{k=1}^{K}\theta_{k,i,j}^2)^{1/2} \right\}.$$

Similarly, let $\boldsymbol{A} = \mathbf{\Theta}_t - \eta_t\nabla f(\mathbf{\Theta}_t)$; then, the problem becomes the following for $i = 1, \cdots, p$, $j = 1, \cdots, p$:

$$\underset{\theta_{1,i,j},\cdots,\theta_{K,i,j}}{\operatorname{argmin}} \left\{ \frac{1}{2}\sum_{k=1}^{K}(\theta_{k,i,j} - a_{k,i,j})^2 + \eta_t\lambda_1 1_{i\neq j}\sum_{k=1}^{K}|\theta_{k,i,j}| + \eta_t\lambda_2 1_{i\neq j}(\sum_{k=1}^{K}\theta_{k,i,j}^2)^{1/2} \right\}.$$

We have $\theta_{k,i,j} = a_{k,i,j}$ for $i = j$. In addition, for $i \neq j$, the solution [31,36,37] is given by

$$\theta_{k,i,j} = \mathcal{S}_{\eta_t\lambda_1}(a_{k,i,j})\left(1 - \frac{\eta_t\lambda_2}{\sqrt{\sum_{k=1}^{K}\mathcal{S}_{\eta_t\lambda_1}(a_{k,i,j})^2}}\right)_+. \tag{21}$$

3.2. Modified ISTA for JGL

Thus far, we have seen that $f(\mathbf{\Theta})$ in the JGL problem (15) is not globally Lipschitz gradient continuous. The ISTA may not be efficient enough for the JGL case because it includes the backtracking line search procedure for this case, which needs to evaluate the objective function and the positive definiteness of $\mathbf{\Theta}_{t+1}$ in Step 1 of Algorithm 2 and is inefficient when the evaluation is expensive.

In this section, we modify Algorithm 2 to Algorithm 3 based on the step-size selection strategy in Section 2.3, which takes advantage of the properties of the self-concordant

function. The self-concordant function does not rely on the Lipschitz gradient assumption on the function $f(\boldsymbol{\Theta})$ [24], and we can eliminate the need for the backtracking line search.

Lemma 2. *([38]) Self-concordance is preserved by scaling and addition: if f is a self-concordant function and a constant $a \leq 1$, then af is self-concordant. If f_1, f_2 are self-concordant, then $f_1 + f_2$ is self-concordant.*

By Lemma 2, the function $f(\boldsymbol{\Theta})$ in (16) is self-concordant. In Algorithm 3, for the initial step size of η_t in each iteration, we use the Barzilai–Borwein method [39]. We apply the step-size mechanism in Section 2.3, which is employed in Steps 3–5 of Algorithm 3.

Algorithm 3 Modified ISTA (M-ISTA).

Input: S, tolerance $\epsilon > 0$, initial step size η_0, initial iterate $\boldsymbol{\Theta}_0$.
For $t = 0, 1, \cdots,$ (until convergence) **do**

1: Initialize η_t.
2: Compute

$$\boldsymbol{d}_t := \text{prox}_{\eta_t g}(\boldsymbol{\Theta}_t - \eta_t \nabla f(\boldsymbol{\Theta}_t)) - \boldsymbol{\Theta}_t.$$

3: Compute

$$\beta_t := \eta_t^{-1} ||\boldsymbol{d}_t||_F^2$$

and

$$\lambda_t := \sum_{k=1}^{K} \sqrt{n_k} ||(\boldsymbol{\Theta}_t^{(k)})^{-1} \boldsymbol{d}_t^{(k)}||_F.$$

4: Determine the step size $\alpha_t := \frac{\beta_t}{\lambda_t(\lambda_t + \beta_t)}$.
5: If $\alpha_t > 1$, then set $\eta_t := \eta_t / 2$ and go back to Step 2.
6: Update $\boldsymbol{\Theta}_{t+1} := \boldsymbol{\Theta}_t + \alpha_t \boldsymbol{d}_t$.

end
Output: optimal solution to problem (15), $\boldsymbol{\Theta}_* = \boldsymbol{\Theta}_{t+1}$.

There is no backtracking procedure in this algorithm that guarantees the positive definiteness of $\boldsymbol{\Theta}_{t+1}$, as in Step 1 of Algorithm 2. We next show how to ensure the positive definiteness of $\boldsymbol{\Theta}_{t+1}$ in the iterations of Algorithm 3.

Lemma 3. *([40], Theorem 2.1.1) Let f be a self-concordant function, and let $\boldsymbol{x} \in \text{dom } f$. Additionally, if*

$$W(\boldsymbol{x}) = \{\boldsymbol{y} | \Big(\langle \nabla^2 f(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \Big)^{1/2} \leq 1\},$$

then $W(\boldsymbol{x}) \subset dom f$.

In Algorithm 3, because we know $\alpha_t := \frac{\beta_t}{\lambda_t(\lambda_t + \beta_t)} < 1$ with $\beta_t > 0$ and $\lambda_t > 0$ by Steps 3–5. Thus, we have $\alpha_t \lambda_t < 1$:

$$\alpha_t \lambda_t := \alpha_t \Big(\langle \nabla^2 f(\boldsymbol{\Theta}_t) \boldsymbol{d}_t, \boldsymbol{d}_t \rangle \Big)^{1/2} < 1,$$

which implies,

$$\Big(\langle \nabla^2 f(\boldsymbol{\Theta}_t)(\boldsymbol{\Theta}_{t+1} - \boldsymbol{\Theta}_t), \boldsymbol{\Theta}_{t+1} - \boldsymbol{\Theta}_t \rangle \Big)^{1/2} < 1.$$

Hence, from Lemma 3, we see that $\boldsymbol{\Theta}_{t+1}$ stays in the domain and maintains positive definiteness.

3.3. Theoretical Analysis

For multiple Gaussian graphical models, Honorio and Samaras [14] and Hara and Washio [17] provided lower and upper bounds for the optimal solution $\mathbf{\Theta}_*$. However, the models they considered are different than the JGL. To the best of our knowledge, no related research has provided the bounds of the optimal solution $\mathbf{\Theta}_*$ for the JGL problem (15).

In the following section, we show the bounds of the optimal solution $\mathbf{\Theta}_*$ for the JGL and the iterates $\mathbf{\Theta}_t$ generated by Algorithms 2 and 3, which are applied to both fused and group lasso-type penalties.

Proposition 1. *The optimal solution $\mathbf{\Theta}_*$ of the problem (15) satisfies*

$$\max_{1\le k\le K} \frac{n_k}{p\lambda_c + n_k||\boldsymbol{S}^{(k)}||_2} \le ||\mathbf{\Theta}_*^{(k)}||_2 \le \frac{Np}{\lambda_1} + \sum_{k=1}^{K}\sum_{i=1}^{p}(s_{k,i,i})^{-1},$$

where $\lambda_c := \sqrt{K\lambda_1^2 + 2K\lambda_1\lambda_2 + \lambda_2^2}$, and $s_{k,i,i}$ is the i-th diagonal element of $\boldsymbol{S}^{(k)}$.

For the proof, see Appendix A.1.

Note that the objective function value $F(\mathbf{\Theta})$ always decreases with the increase in iteration in both algorithms due to [25] (Remark 3.1) and Lemma 12 in [24]. Therefore, the following inequality holds for Algorithms 2 and 3:

$$F(\mathbf{\Theta}_{t+1}) \le F(\mathbf{\Theta}_t) \qquad \text{for } t = 0, 1, \ldots. \tag{22}$$

Then, based on the condition (22), we provide the explicit bounds of iterates $\{\mathbf{\Theta}_t\}_{t=0,1\ldots}$ in Algorithms 2 and 3 for the JGL problem (15).

Proposition 2. *Sequence $\{\mathbf{\Theta}_t\}_{t=0,1,\cdots}$, generated by Algorithms 2 and 3 can be bounded:*

$$m \le ||\mathbf{\Theta}_t||_2 \le M,$$

where $M := ||\mathbf{\Theta}_0||_F + \frac{2Np}{\lambda_1} + 2\sum_{k=1}^{K}\sum_{i=1}^{p} {s_{k,i,i}}^{-1}$, $m := e^{-\frac{C_1}{n_m}} M^{(1-Kp)}$, $n_m = \max_k n_k$, and constant $C_1 := F(\mathbf{\Theta}_0)$.

For the proof, see Appendix A.2.

With the help of Propositions 1 and 2, and the following Lemma, we can obtain the range of the step size that ensures the linear convergence rate of Algorithm 2.

Lemma 4. *Let $\mathbf{\Theta}_t$ be t-th iterate in Algorithm 2. Denote λ_{min} and λ_{max} as the minimum and maximum eigenvalues of the corresponding matrix, respectively. Define*

$$a_k := \min\{\lambda_{\min}(\mathbf{\Theta}_t^{(k)}), \lambda_{\min}(\mathbf{\Theta}_*^{(k)})\},\ b_k := \max\{\lambda_{\max}(\mathbf{\Theta}_t^{(k)}), \lambda_{\max}(\mathbf{\Theta}_*^{(k)})\}$$

and $n_l = \min_{k=1\ldots,K} n_k$, $n_m = \max_{k=1\ldots,K} n_k$, $a_l = \min_{k=1\ldots,K} a^{(k)}$, and $b_m = \max_{k=1\ldots,K} b^{(k)}$. The sequence $\{\mathbf{\Theta}_t\}_{t=0,1,\ldots}$ generated by Algorithm 2 satisfies

$$||\mathbf{\Theta}_{t+1} - \mathbf{\Theta}_*||_F \le \gamma_t ||\mathbf{\Theta}_t - \mathbf{\Theta}_*||_F$$

with the convergence rate $\gamma_t := \max\{\frac{\eta_t n_m}{a_l^2} - 1, 1 - \frac{\eta_t n_l}{b_m^2}\}$.

Proof. It can be easily extended by Lemma 3 in [10]. □

Lemma 4 implies that to obtain the convergence rate $\gamma_t < 1$, we require:

$$0 < \eta_t < \frac{2a_l^2}{n_m}. \tag{23}$$

After using Propositions 1 and 2, we can obtain the bounds of a_l. Further, we can obtain the step size η_t that satisfies (23) and guarantee s the linear convergence rate $(\gamma_t < 1)$. However, the step size is quite conservative in practice. Hence, we consider the Barzilai–Borwein method for implementation and regard the step size η_t that satisfies (23) as a safe choice. When the number of backtracking iterations in Step 1 of Algorithm 2 exceeds the given maximum number to fulfill the backtracking line search condition, we can use the safe step size η_t for the subsequent calculations. In Section 4.2.3, we confirm the linear convergence rate of the proposed ISTA by experiment.

4. Experiments

In this section, we evaluate the performance of the proposed methods on both synthetic and real datasets, and we compare the following algorithms:

- ADMM: the general ADMM method proposed by [13].
- FMGL: the proximal Newton-type method proposed by [23].
- ISTA: the proposed method in Algorithm 2.
- M-ISTA: the proposed method in Algorithm 3.

We perform all the tests in R Studio on a Macbook Air with 1.6 GHz Intel Core i5 and 8 GB memory. The wall times are recorded as the run times for the four algorithms.

4.1. Stopping Criteria and Model Selection

In the experiments, we consider two stopping criteria for the algorithms.

1. Relative error stopping criterion:

$$\frac{\sum_{k=1}^{K} ||\mathbf{\Theta}_{t+1}^{(k)} - \mathbf{\Theta}_t^{(k)}||_F}{\max\{\sum_{k=1}^{K} ||\mathbf{\Theta}_t^{(k)}||_F, 1\}} \leq \epsilon.$$

2. Objective error stopping criterion:

$$F(\mathbf{\Theta}_t) - F(\mathbf{\Theta}_*) \leq \epsilon.$$

ϵ is a given accuracy tolerance; we terminate the algorithm if the above error is smaller than ϵ or the maximum number of iterations exceeds 1000. We use the objective error for convergence rate analysis and the relative error for the time comparison.

The JGL model is affected by regularized parameters λ_1 and λ_2. For selecting the parameters, we use the V-fold crossvalidation method. First, the dataset is randomly split into V segments of equal size, a single subset (test data), estimated by the other $V-1$ subsets (training data), is evaluated, and the subset is changed for the test to repeat V times so that each subset is used.

Let $\mathbf{S}_v^{(k)}$ be the sample covariance matrix of the v-th ($v = 1, \ldots, V$) segment for class $k = 1, \ldots, K$. We estimate the inverse covariance matrix by the remaining $V-1$ subsets $\hat{\mathbf{\Theta}}_{\lambda,-v}^{(k)}$ and choose λ_1 and λ_2, which minimize the average predictive negative log-likelihood as follows:

$$CV(\lambda_1, \lambda_2) = \sum_{v=1}^{V} \sum_{k=1}^{K} \left\{ n_k \text{trace}(\mathbf{S}_v^{(k)} \hat{\mathbf{\Theta}}_{\lambda,-v}^{(k)}) - \text{logdet}\hat{\mathbf{\Theta}}_{\lambda,-v}^{(k)} \right\}$$

4.2. Synthetic Data

The performance of the proposed methods was assessed on synthetic data in terms of the number of iterations, the execution time, the squared error, and the receiver operating

characteristic (ROC) curve. We follow the data generation mechanism described in [41] with some modifications for the JGL model. We put the details in Appendix B.

4.2.1. Time Comparison Experiments

We vary p, N, K and λ_1 to compare the execution time of our proposed methods with that of the existing methods. We consider only the fused penalty in our proposed method for a fair comparison in the experiments because the FMGL algorithm applies only to the fused penalty. First, we compare the performance among different algorithms under various dimensions p, which are shown in Figure 1.

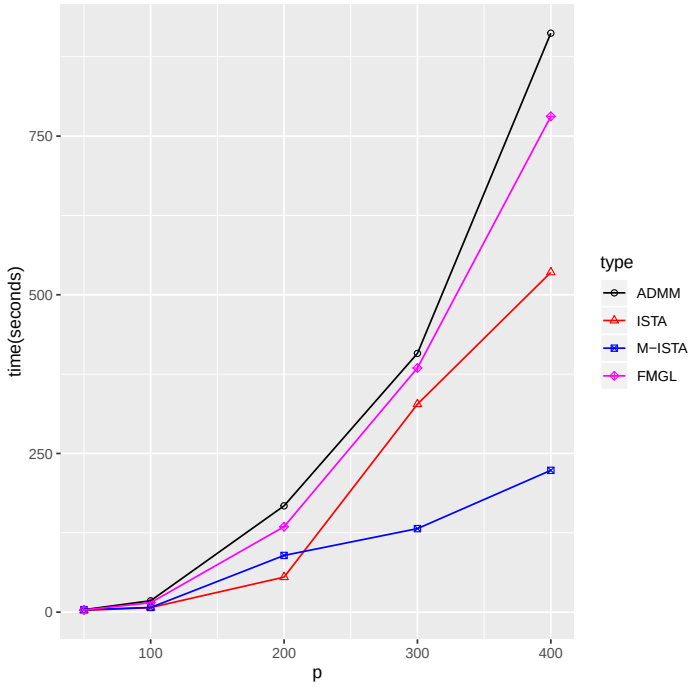


Figure 1. Plot of time comparison under different p. Setting $\lambda_1 = 0.1$, $\lambda_2 = 0.05$, $K = 2$, and $N = 200$.

Figure 1 shows that the execution time of the FMGL and ADMM increases rapidly as p increases. In particular, we observe that the M-ISTA significantly outperforms when p exceeds 200. The ISTA shows better performance than the three methods when p is less than 200, but it requires more time as p grows, compared to the M-ISTA. It is reasonable to consider that evaluating the objective function in the backtracking line search at every iteration increases the computational burden, especially when p increases, which means that the M-ISTA is a good choice for these cases. Furthermore, the ISTA can be a good candidate when the evaluation is inexpensive.

Table 2 summarizes the performance of the four algorithms under different parameter settings to achieve a given precision, ϵ, of the relative error. The results presented in Table 2 reveal that when we increase the number of classes K, all the algorithms require more time than usual. Moreover, the execution time of ADMM becomes huge among them. When we vary λ_1, the algorithms become more efficient as the value increases. For most instances, the M-ISTA and ISTA outperform the existing methods, such as ADMM and FMGL. For the exceptional cases ($p = 20, k = 2, N = 60, \lambda_1 = 0.1$ and $\lambda_2 = 0.05$), the M-ISTA and ISTA are still comparable with the FMGL and faster than ADMM.

Table 2. Computational time under different settings.

Parameters Setting						Computational Time			
p	K	N	λ_1	λ_2	precision ϵ	ADMM	FMGL	ISTA	M-ISTA
20	2	60	0.1	0.05	0.00001	10.506 s	**1.158 s**	2.174 s	1.742 s
	3					1.879 min	4.267 s	**3.357 s**	3.668 s
	5		1	0.5		1.123 min	10.556 s	4.216 s	**2.874 s**
30	2	120	0.1	0.05	0.0001	10.095 s	5.259 s	**2.690 s**	4.857 s
	3					2.014 min	38.562 s	**14.722 s**	31.870 s
	5		1	0.5		2.447 min	15.819 s	22.431 s	**12.113 s**
50	2	600	0.02	0.005	0.0001	6.427 s	10.228 s	7.213 s	**4.625 s**
			0.03			6.240 s	8.925 s	6.645 s	**4.023 s**
			0.04			7.025 s	9.381 s	6.144 s	**3.993 s**
200	2	400	0.09	0.05	0.0001	4.050 min	1.874 min	2.289 min	**35.038 s**
			0.1			4.569 min	1.137 min	1.340 min	**24.852 s**
			0.12			3.848 min	1.881 min	1.443 min	**18.367 s**

4.2.2. Algorithm Assessment

We generate the simulation data as described in Appendix B and regard the synthetic inverse covariance matrices $\mathbf{\Theta}^{(k)}$ as the true values for our assessment experiments.

First, we assessed our proposed method by drawing an ROC curve, which displays the number of true positive edges (i.e., TP edges) selected compared to the number of false positive edges (i.e., FP edges) selected. We say that an edge (i, j) in the k-th class is selected in estimate $\hat{\mathbf{\Theta}}^{(k)}$ if element $\hat{\theta}_{k,i,j} \neq 0$, and the edges are true positive edges selected if the precision matrix element $\theta_{k,i,j} \neq 0$ and false positive edges selected if the precision matrix element $\theta_{k,i,j} = 0$, where the two quantities are defined by

$$TP = \sum_{k=1}^{K} \sum_{i,j} 1(\theta_{k,i,j} \neq 0) \cdot 1(\hat{\theta}_{k,i,j} \neq 0)$$

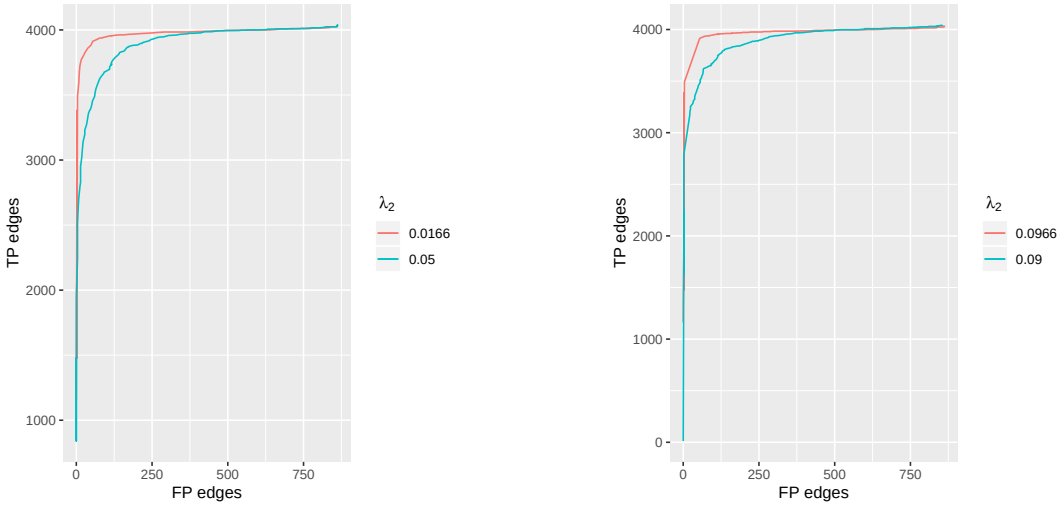
and

$$FP = \sum_{k=1}^{K} \sum_{i,j} 1(\theta_{k,i,j} = 0) \cdot 1(\hat{\theta}_{k,i,j} \neq 0)\,,$$

where $1(\cdot)$ is the indicator function.

To confirm the validity of the proposed methods, we compare the ROC figures of the fused penalty and group penalty. We fix the parameters λ_2 for each curve and change the λ_1 value to obtain various numbers of selected edges because the sparsity penalty parameter λ_1 can control the number of selected total edges.

We show the ROC curves for fused and group lasso penalties in Figure 2a,b respectively. From the figures, we observe that both penalties show highly accurate predictions for the edge selections. The result of $\lambda_2 = 0.0166$ in the fused penalty case is better than that in $\lambda_2 = 0.05$. Additionally, the result of $\lambda_2 = 0.0966$ in the group penalty case is better than that in $\lambda_2 = 0.09$, which means that if we select the tuning parameters properly, then we can obtain precise results while simultaneously meeting our different model demands.



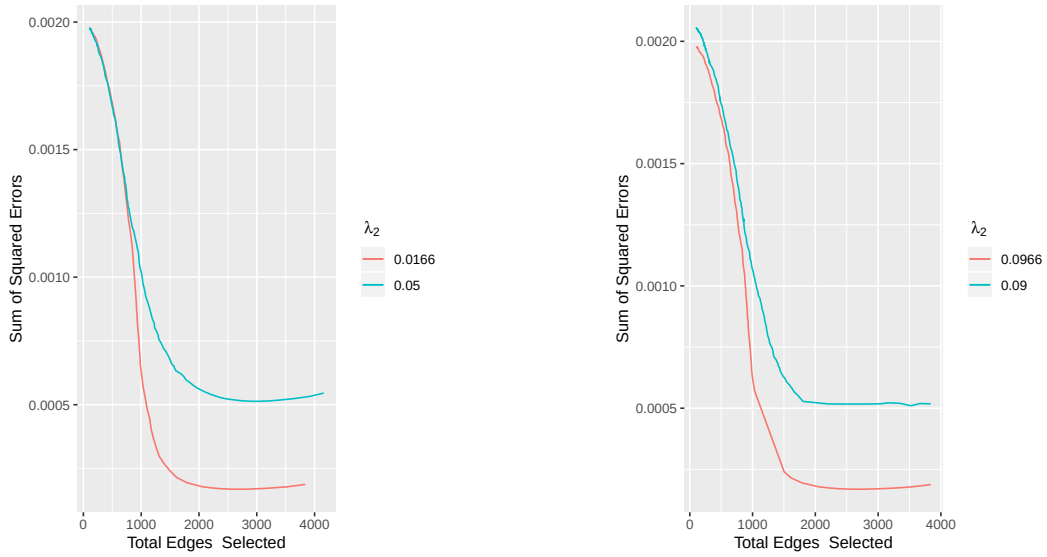
(**a**) (**b**)

Figure 2. Plot of true positive edges vs. false positive edges selected. Setting $p = 50$, $K = 2$. (**a**) The fused penalty; (**b**) The group penalty.

Then, Figure 3a,b display the mean squared error (MSE) between the estimated values and true values.

$$MSE = \frac{2}{Kp(p-1)} \sum_{k=1}^{K} \sum_{i<j} (\hat{\theta}_{k,i,j} - \theta_{k,i,j})^2,$$

where $\hat{\theta}_{k,i,j}$ is the value estimated by the proposed method, and $\theta_{k,i,j}$ is the true precision matrix value we used in the data generation.



(**a**) (**b**)

Figure 3. Plot of the mean squared errors vs. total edges selected. Setting $p = 50$, $K = 2$. (**a**) The fused penalty; (**b**) The group penalty.

The figures illustrate that when the total number of edges selected increases, the errors decrease and finally achieve relatively low values.

Overall, the proposed method shows competitive efficiency not only in computational time but also in accuracy.

4.2.3. Convergence Rate

This section shows the convergence rate of the ISTA for solving the JGL problem (15) in practice, with $\lambda_1 = 0.1$, 0.09 and 0.08. We recorded the number of iterations to achieve the different tolerance of $F(\mathbf{\Theta}_t) - F(\mathbf{\Theta}_*)$ in Figure 4 and ran it on a synthetic dataset, with $p = 200$, $K = 2$, $\lambda_2 = 0.05$, and $N = 400$. The figure reveals that as λ_1 decreases, more iterations are needed to converge to the specified tolerance. Moreover, the figure shows the linear convergence rate of the proposed ISTA method, which corroborate the theoretical analysis in Section 3.3.

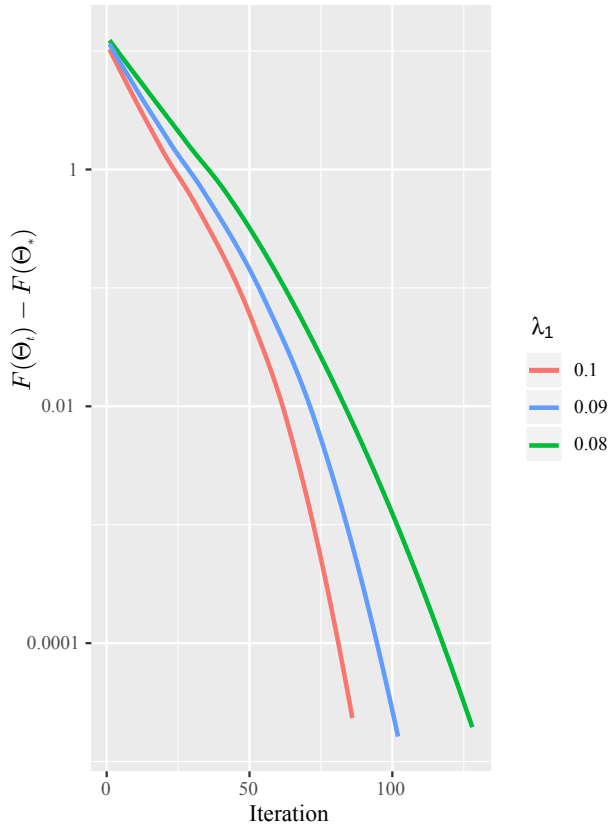


Figure 4. Plot of $\log(F(\mathbf{\Theta}_t) - F(\mathbf{\Theta}_*))$ vs. the number of iterations with different λ_1 values. Setting $p = 200, N = 400, K = 2$ and $\lambda_2 = 0.05$.

4.3. Real Data

In this section, we use two different real datasets to demonstrate the performance of our proposed method and visualize the result.

Firstly, we used the presidential speeches dataset in [42] for the experiment to jointly estimate common links across graphs and show the common structure. The dataset contains 75 most-used words (features) from several big speeches of the 44 US presidents (samples). In addition, we used the clustering result in [42], where the authors split the 44 samples into two groups with similar features, and then we obtained two classes of samples $(K = 2)$.

We used Cytoscape [43] to visualize the results when $\lambda_1 = 1.9$ and $\lambda_2 = 0.16$. We chose these relatively large tuning parameters for better interpretation of the network figure. Figure 5 shows the relationship network graph of the high-frequency words identified by the JGL model with the proposed method. As shown in the figure, each node represents a word, and the edges demonstrate the relationships between words.

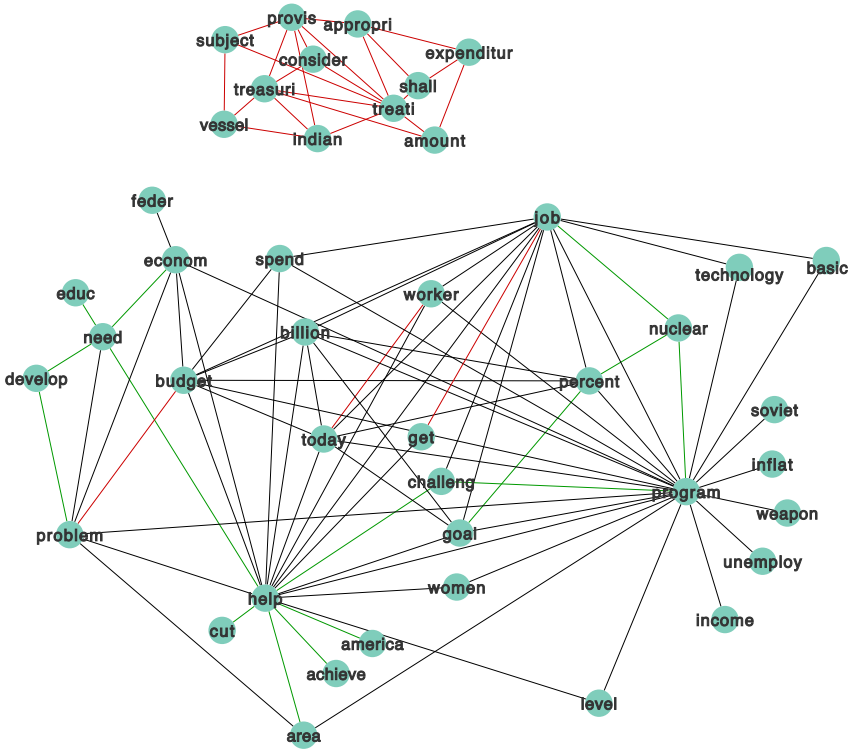


Figure 5. Network figure of the words in president speeches dataset.

We use different colors to show various structures. The black edges are a common structure between the two classes, the red edges are the specific structures for the first class ($k = 1$), and the green edges are for the second class ($k = 2$). Figure 5 shows a subnetwork on the top with red edges, meaning there are relationships among those words, and the connections only exist in the first group.

We compared the time cost among four algorithms and show the results in Table 3. We used the crossvalidation method ($V = 6$) described in Section 4.1 to select the optimal tuning parameters ($\lambda_1 = 0.1$, $\lambda_2 = 0.05$). In addition, we manually chose the other two pairs of parameters for more comparisons.

Table 3. Time comparison result of two real datasets.

Dataset	Parameters Setting			Computational Time			
	λ_1	λ_2	Precision ϵ	ADMM	FMGL	ISTA	M-ISTA
Speeches	0.1	0.05	0.0001	19.969 s	4.977 min	**11.829 s**	12.867 s
	0.2	0.1		4.661 min	3.209 min	**11.560 s**	12.682 s
	0.5	0.25		5.669 min	1.490 min	**11.043 s**	12.788 s
Breast cancer	0.1	0.0166	0.0001	3.809 min	7.937 min	1.305 min	**1.158 min**
	0.2	0.02		6.031 min	5.198 min	1.503 min	**1.230 min**
	0.3	0.03		5.499 min	2.265 min	1.188 min	**1.061 min**

Table 3 shows that ISTA outperforms the other three algorithms, and our proposed methods offer stable performance when varying the parameters, while ADMM is the slowest in most cases.

Secondly, we use a breast cancer dataset [44] for time comparison. There are 250 samples and 1000 genes in the dataset, with 192 control samples and 58 case samples ($K = 2$). Furthermore, we extract 200 genes with the highest variances among the original genes. The tuning parameter pair ($\lambda_1 = 0.01$, $\lambda_2 = 0.0166$) was chosen by the crossvalidation

method. Table 3 exhibits that our proposed methods (ISTA and M-ISTA) outperform ADMM and FMGL, and M-ISTA shows the best performance in the breast cancer dataset.

5. Discussion

We propose two efficient proximal gradient descent procedures with and without the backtracking line search option for the joint graphical lasso. The first (Algorithm 2) does not require extra variables, unlike ADMM, which needs manual tuning the Lagrangian penalty parameters ρ in [13] and storing and calculating dual variables. Moreover, we reduce the update iterate step to subproblems that can be solved efficiently and precisely by lasso-type problems. Based on Algorithm 2, we modified the step-size selection by extending the strategy in [24] to the second one (Algorithm 3), which does not rely on the Lipschitz assumption. Additionally, the second does not require a backtracking line search, significantly reducing the computation time needed to evaluate objective functions.

From the theoretical perspective, we reach the linear convergence rate for the ISTA. Furthermore, we derive the lower and upper bounds of the solution to the JGL problem and the iterates in the algorithms, guaranteeing that the ISTA converges linearly. Numerically, the methods are demonstrated on simulated and real datasets to illustrate their robust and efficient performance over state-of-the-art algorithms.

For further computational improvement, the most expensive step in the algorithms is to calculate the inversion of matrices required by the gradient of $f(\mathbf{\Theta})$. Both algorithms have a complexity of $O(Kp^3)$ per iteration. Moreover, we can solve the matrix inversion problem with more efficient algorithms with lower complexity. In addition, we can also use the faster computation procedure in [13] to decompose the optimization problem for the proposed methods and regard it as preprocessing. Overall, the proposed methods are highly efficient for the joint graphical lasso problem.

Author Contributions: Conceptualization, J.C., R.S. and J.S.; methodology, J.C., R.S. and J.S.; software, J.C. and R.S.; validation, J.C., R.S. and J.S.; formal analysis, J.C., R.S. and J.S.; writing—original draft preparation, J.C. and J.S.; writing—review and editing, J.C., R.S. and J.S.; visualization, J.C.; supervision, J.S.; project administration, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Grant-in-Aid for Scientific Research (KAKENHI) C, Grant number: 18K11192.

Data Availability Statement: Publicly available datasets were analyzed in this paper. Presidential speeches dataset: https://www.presidency.ucsb.edu, accessed on 5 November 2021; Breast cancer dataset: https://www.rdocumentation.org/packages/doBy/versions/4.5-15/topics/breastcancer, accessed on 5 November 2021.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADMM	alternating direction method of multipliers
FMGL	fused multiple graphical lasso algorithm
FP	false positive
G-ISTA	graphical iterative shrinkage-thresholding algorithm
GL	graphical lasso
ISTA	iterative shrinkage-thresholding algorithm
JGL	joint graphical lasso
M-ISTA	modified iterative shrinkage-thresholding algorithm
MSE	mean squared error
ROC	receiver operating characteristic
TP	true positive

Appendix A. Proofs of Propositions

Appendix A.1. Proof of Proposition 1

We first introduce the Lagrange dual problem of (15). By introducing the auxiliary variables $\mathbf{Z} = \{\mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(K)}\}$, we can rewrite the problem as follows:

$$\min_{\boldsymbol{\Theta}} f(\boldsymbol{\Theta}) + g(\boldsymbol{\Theta})$$
$$\text{subject to } \mathbf{Z} = \boldsymbol{\Theta}$$

Then, the Lagrange function of the above is given by:

$$L(\boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Lambda}) = f(\boldsymbol{\Theta}) + g(\mathbf{Z}) + \sum_{k=1}^{K} \langle \boldsymbol{\Lambda}^{(k)}, \boldsymbol{\Theta}^{(k)} - \mathbf{Z}^{(k)} \rangle,$$

where $\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}^{(1)}, \ldots, \boldsymbol{\Lambda}^{(K)}\}, \boldsymbol{\Lambda}^{(k)} \in \mathbb{R}^{p \times p}$ are dual variables. To obtain the dual problem, we minimize the primal variables as follows:

$$\begin{aligned}
\min_{\boldsymbol{\Lambda}, \mathbf{Z}} L(\boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Lambda}) &= \min_{\boldsymbol{\Theta}} \{f(\boldsymbol{\Theta}) + \sum_{k=1}^{K} \langle \boldsymbol{\Lambda}^{(k)}, \boldsymbol{\Theta}^{(k)} \rangle\} - \max_{\mathbf{Z}} \{-g(\mathbf{Z}) - \sum_{k=1}^{K} \langle \boldsymbol{\Lambda}^{(k)}, -\mathbf{Z}^{(k)} \rangle\} \\
&= \min_{\boldsymbol{\Theta}} \{f(\boldsymbol{\Theta}) + \sum_{k=1}^{K} \langle \boldsymbol{\Lambda}^{(k)}, \boldsymbol{\Theta}^{(k)} \rangle\} - g^*(\boldsymbol{\Lambda}) \\
&= \min_{\boldsymbol{\Theta}} \left\{ \sum_{k=1}^{K} \langle \boldsymbol{\Lambda}^{(k)} + n_k \mathbf{S}^{(k)}, \boldsymbol{\Theta}^{(k)} \rangle - \sum_{k=1}^{K} n_k \text{logdet} \boldsymbol{\Theta}^{(k)} \right\} - g^*(\boldsymbol{\Lambda}).
\end{aligned}$$

Taking derivative of the function:

$$L_1 := \sum_{k=1}^{K} \langle \boldsymbol{\Lambda}^{(k)} + n_k \mathbf{S}^{(k)}, \boldsymbol{\Theta}^{(k)} \rangle - \sum_{k=1}^{K} n_k \text{logdet} \boldsymbol{\Theta}^{(k)},$$

$$\nabla_{\boldsymbol{\Theta}^{(k)}} L_1 = 0,$$

We obtain

$$n_k \mathbf{S}^{(k)} + \boldsymbol{\Lambda}^{(k)} = n_k (\boldsymbol{\Theta}^{(k)})^{-1} \tag{A1}$$

for $k = 1, \cdots, K$. Substitute the Equation (A1) into the dual problem $\min_{\boldsymbol{\Theta}, \mathbf{Z}} L(\boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Lambda})$, then it becomes:

$$\min_{\boldsymbol{\Theta}, \mathbf{Z}} L(\boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Lambda}) = \sum_{k=1}^{K} n_k p + \sum_{k=1}^{K} n_k \text{logdet}(\mathbf{S}^{(k)} + \frac{1}{n_k} \boldsymbol{\Lambda}^{(k)}) - g^*(\boldsymbol{\Lambda}).$$

Hence, we can obtain the duality gap [38] (the primal problem minus the dual problem) as follows:

$$\begin{aligned}
&f(\boldsymbol{\Theta}) + g(\mathbf{Z}) - \sum_{k=1}^{K} n_k p - \sum_{k=1}^{K} n_k \{\text{logdet} \boldsymbol{\Theta}^{(k)}\} + g^*(\boldsymbol{\Lambda}) \\
&\quad = \sum_{k=1}^{K} n_k \text{trace}(\mathbf{S}^{(k)} \boldsymbol{\Theta}^{(k)}) + g(\mathbf{Z}) - \sum_{k=1}^{K} n_k p + g^*(\boldsymbol{\Lambda}),
\end{aligned}$$

when the gap value is 0, the optimal solution is found. Because the conjugate function $g^*(\boldsymbol{\Lambda})$ is the indicator function, the value is hence 0 for the optimal solution.

Firstly, for the group penalty $P_G(\boldsymbol{\Theta})$, the duality gap is

$$\sum_{k=1}^{K} [n_k \text{trace}(\mathbf{S}^{(k)} \boldsymbol{\Theta}_*^{(k)}) - n_k p] + \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\theta_{*k,i,j}| + \lambda_2 \sum_{i \neq j} \sqrt{\sum_{k=1}^{K} {\theta_{*k,i,j}}^2} = 0. \tag{A2}$$

From Equation (A2), we obtain

$$\lambda_1||\mathbf{\Theta}_*||_1 = -\sum_{k=1}^{K} n_k \text{trace}(\boldsymbol{S}^{(k)}\mathbf{\Theta}_*^{(k)}) - \lambda_2 \sum_{i\neq j}\sqrt{\sum_{k=1}^{K}{\theta_*}_{k,i,j}{}^2} + \sum_{k=1}^{K} n_k p + \sum_{k=1}^{K}\sum_{i=1}^{p}\lambda_1|\theta_{*k,i,i}|$$
$$\leq \sum_{k=1}^{K} n_k p + \sum_{k=1}^{K}\sum_{i=1}^{p}\lambda_1|\theta_{*k,i,i}|.$$

From Equation (A1), we have the following relationship of diagonal elements,

$$\theta_{k,i,i}{}^* = diag\left(\boldsymbol{S}^{(k)} + \frac{1}{n_k}\mathbf{\Lambda}_*^{(k)}\right)^{-1},$$

and due to dual variable $\Lambda^*_{k,i,i} > 0$, for $k = 1, \ldots, K$. Hence,

$$||\mathbf{\Theta}_*||_1 \leq \frac{1}{\lambda_1}\sum_{k=1}^{K} n_k p + \sum_{k=1}^{K}\sum_{i=1}^{p} diag\left(\boldsymbol{S}^{(k)} + \frac{1}{n_k}\mathbf{\Lambda}_*^{(k)}\right)^{-1}$$
$$\leq \frac{1}{\lambda_1}\sum_{k=1}^{K} n_k p + \sum_{k=1}^{K}\sum_{i=1}^{p} diag\left(\boldsymbol{S}^{(k)}\right)^{-1}.$$

By $||\mathbf{\Theta}_*||_2 \leq ||\mathbf{\Theta}_*||_F \leq ||\mathbf{\Theta}_*||_1$, we obtain the upper bound:

$$||\mathbf{\Theta}_*||_2 \leq ||\mathbf{\Theta}_*||_F \leq \frac{1}{\lambda_1}\sum_{k=1}^{K} n_k p + \sum_{k=1}^{K}\sum_{i=1}^{p} {s_{k,i,i}}^{-1}. \tag{A3}$$

The proof is similar for the fused penalty $P_F(\mathbf{\Theta})$; therefore, we omit it here. Next, we continue to prove the lower bound of $\mathbf{\Theta}_*$.

Firstly, for the group penalty $P_G(\mathbf{\Theta})$. Let $\boldsymbol{E}^{(k)}$ be non-negative $p \times p$ matrix satisfying $-E_{k,i,j} \leq \theta_{k,i,j} \leq E_{k,i,j}$. Introducing the Lagrange multipliers $\Gamma^{(k)}$ and $\Gamma_0^{(k)}$ for $k = 1, \ldots, K$. This procedure is similar to the way in [17].

Then, the new Lagrange problem becomes,

$$\max_{\Theta,E}\min_{\Gamma,\Gamma_0}\Bigg\{ f(\mathbf{\Theta}) - \sum_{k=1}^{K}\sum_{i\neq j}\lambda_1 E_{k,i,j} - \lambda_2\sum_{i\neq j}\sqrt{\sum_{k=1}^{K} E_{k,i,j}^2}$$
$$- \sum_{k=1}^{K} tr(\mathbf{\Gamma}^{(k)}\mathbf{\Theta}^{(k)}) - tr(abs(\mathbf{\Gamma}^{(k)})\boldsymbol{E}^{(k)}) - tr(\mathbf{\Gamma}_0^{(k)}\boldsymbol{E}^{(k)})\Bigg\},$$

Taking derivative w.r.t $\mathbf{\Theta}^{(k)}$ and $E_{k,i,j}$, we obtain the following equations:

$$n_k{\mathbf{\Theta}^{(k)}}^{-1} - n_k\boldsymbol{S}^{(k)} - \mathbf{\Gamma}^{(k)} = 0, \tag{A4}$$

$$-\lambda_1 - \lambda_2\frac{E_{k,i,j}}{\sqrt{\sum_{k=1}^{K} E_{k,i,j}^2}} + |\Gamma_{k,i,j}| + \Gamma_0^{(k)} = 0, \text{ for } i \neq j, \tag{A5}$$

$$|\Gamma_{k,i,j}| + \Gamma_0^{(k)} = 0, \text{ for } i = j. \tag{A6}$$

When $i \neq j$, from Equation (A5),

$$|\Gamma_{k,i,j}| \leq \lambda_1 + \lambda_2 \frac{E_{k,i,j}}{\sqrt{\sum_{k=1}^{K} E_{k,i,j}^2}}$$

$$|\Gamma_{k,i,j}|^2 \leq \left(\lambda_1 + \lambda_2 \frac{E_{k,i,j}}{\sqrt{\sum_{k=1}^{K} E_{k,i,j}^2}} \right)^2$$

$$= \lambda_1^2 + 2\lambda_1\lambda_2 \frac{E_{k,i,j}}{\sqrt{\sum_{k=1}^{K} E_{k,i,j}^2}} + \lambda_2^2 \frac{E_{k,i,j}^2}{\sum_{k=1}^{K} E_{k,i,j}^2}$$

$$\leq \lambda_1^2 + 2\lambda_1\lambda_2 + \lambda_2^2 \frac{E_{k,i,j}^2}{\sum_{k=1}^{K} E_{k,i,j}^2}.$$

Taking summation of each k,

$$\sum_{k=1}^{K} |\Gamma_{k,i,j}|^2 \leq K\lambda_1^2 + 2K\lambda_1\lambda_2 + \lambda_2^2.$$

Then,

$$\sqrt{\sum_{k=1}^{K} |\Gamma_{k,i,j}|^2} \leq \sqrt{K\lambda_1^2 + 2K\lambda_1\lambda_2 + \lambda_2^2}. \tag{A7}$$

From (A4) and (A7), we have the following relationship

$$\begin{aligned} ||{\boldsymbol{\Theta}^{(k)}}^{-1}|| \leq ||\frac{1}{n_k}\boldsymbol{\Gamma}^{(k)} + \boldsymbol{S}^{(k)}||_2 &\leq \frac{1}{n_k}||\boldsymbol{\Gamma}^{(k)}||_2 + ||\boldsymbol{S}^{(k)}||_2 \\ &\leq \frac{p}{n_k} \max_{i,j} |\Gamma_{k,i,j}| + ||\boldsymbol{S}^{(k)}||_2 \\ &\leq \frac{p}{n_k} \max_{k} \max_{i,j} |\Gamma_{k,i,j}| + ||\boldsymbol{S}^{(k)}||_2 \\ &\leq \frac{p(\sqrt{K\lambda_1^2 + 2K\lambda_1\lambda_2 + \lambda_2^2})}{n_k} + ||\boldsymbol{S}^{(k)}||_2. \end{aligned}$$

The last equation holds because

$$\max_{k} \max_{i,j} |\Gamma_{k,i,j}| \leq \sqrt{\sum_{k=1}^{K} |\Gamma_{k,i,j}|^2}.$$

We only consider the case when $i \neq j$ for $max_{i,j}|\Gamma_{ij}^{(k)}|$, because from Equations (A5) and (A6), we know $|\Gamma_{ij}^{(k)}| > |\Gamma_{ii}^{(k)}|$. Overall, the lower bound is

$$\frac{n_k}{p\sqrt{K\lambda_1^2 + 2K\lambda_1\lambda_2 + \lambda_2^2} + n_k||\boldsymbol{S}^{(k)}||_2}.$$

The lower bound of fused penalty can be derived in similar way.

Appendix A.2. Proof of Proposition 2

By Equation (22) and convexity of $F(\mathbf{\Theta})$, it is easy to obtain

$$||\mathbf{\Theta}_t - \mathbf{\Theta}_*||_F \leq ||\mathbf{\Theta}_0 - \mathbf{\Theta}_*||_F.$$

Since $||\cdot||_2 \leq ||\cdot||_F$, then

$$\begin{aligned} ||\mathbf{\Theta}_t||_2 - ||\mathbf{\Theta}_*||_2 &\leq ||\mathbf{\Theta}_t - \mathbf{\Theta}_*||_2 \\ &\leq ||\mathbf{\Theta}_t - \mathbf{\Theta}_*||_F \\ &\leq ||\mathbf{\Theta}_0 - \mathbf{\Theta}_*||_F. \end{aligned}$$

Hence,

$$\begin{aligned} ||\mathbf{\Theta}_t||_2 &\leq ||\mathbf{\Theta}_0 - \mathbf{\Theta}_*||_F + ||\mathbf{\Theta}_*||_2 \\ &\leq ||\mathbf{\Theta}_0||_F + 2||\mathbf{\Theta}_*||_F. \end{aligned}$$

Then, by Equation (A3), we can complete the proof of the upper bound.
To prove the lower bound, denote

$$\begin{aligned} a_t^{(k)} &= \lambda_{\min}(\mathbf{\Theta}_t^{(k)}) \\ (a_t)_l &= \min_{k=1,\cdots,K} a_t^{(k)}. \end{aligned}$$

By the definition of the matrix norm, we have

$$||\mathbf{\Theta}_t^{(k)}||_2 \geq a_t^{(k)} \geq (a_t)_l.$$

Denote the upper bound of $||\mathbf{\Theta}_t||_2$ as M, and that of $||\mathbf{\Theta}_t||_2$ as $M^{(k)}$, for $k = 1, \ldots, K$. By definition of tensor norm, we have $M \geq ||\mathbf{\Theta}_t||_2 \geq ||\mathbf{\Theta}_t^{(k)}||_2 \geq (a_t)_l$.
Let constant $C_1 := f(\mathbf{\Theta}_0) + g(\mathbf{\Theta}_0)$. By the Equation (22), we have

$$C_1 \geq f(\mathbf{\Theta}_t) + g(\mathbf{\Theta}_t).$$

Note that $\boldsymbol{S} \succeq 0, \mathbf{\Theta}_t \succ 0$ implies $tr(\boldsymbol{S}\mathbf{\Theta}_t) \geq 0$ and because $g(\mathbf{\Theta}_t) \geq 0$

$$\begin{aligned} C_1 &\geq -\sum_{k=1}^{K} n_k \text{logdet}\mathbf{\Theta}_t^{(k)} \\ &= -\sum_{k=1}^{K} n_k \log(\Pi_{i=1}^{p}\lambda_i). \end{aligned}$$

Let the eigenvalues of $\mathbf{\Theta}_t^{(k)}$ as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_p$. Then $a_t^{(k)} = \lambda_1 \leq \lambda_p \leq M^{(k)}$, hence,

$$\Pi_{i=1}^{p}(\lambda_i) = a_t^{(k)} \cdot \lambda_2 \cdot \cdots \cdot \lambda_p \leq a_t^{(k)} \cdot {M^{(k)}}^{(p-1)}.$$

Then,

$$\sum_{k=1}^{K} n_k \log(\Pi_{i=1}^{p}\lambda_i) \leq \sum_{k=1}^{K} n_k \left[log a_t^{(k)} + (p-1) log M^{(k)} \right].$$

Let the coefficient n_k of the term which contains $(a_t)_l$ in $-\sum_{k=1}^{K} n_k log a_t^{(k)}$ as n_x, then

$$\sum_{k=1}^{K} n_k log a_t^{(k)} = n_x log(a_t)_l + \sum_{k \neq x} n_k log a_t^{(k)}.$$

Because

$$M^{(k)} \leq M,$$

denote $n_m = \max_{i=1\cdots,K} n_k$, then,

$$\sum_{k=1}^{K} n_k log a_t^{(k)} \leq n_m log(a_t)_l + n_m(K-1)logM.$$

Hence,

$$\begin{aligned} C_1 &\geq -\sum_{k=1}^{K} n_k \log(\Pi_{i=1}^{p}(\lambda_i)) \\ &\geq -\sum_{k=1}^{K} n_k \left[log a_t^{(k)} + (p-1) log M^{(k)} \right] \\ &\geq -n_m log(a_t)_l - n_m(K-1) log\ M \\ &\quad - K n_m (p-1) logM. \end{aligned}$$

Then, we can obtain

$$\begin{aligned} log(a_t)_l &\geq -K(p-1)logM - (K-1)logM - \frac{C_1}{n_m} \\ (a_t)_l &\geq e^{(1-Kp)logM - \frac{C_1}{n_m}}. \end{aligned}$$

Hence, the lower bound is proved:

$$||\mathbf{\Theta}_t||_2 \geq ||\mathbf{\Theta}_t^{(k)}||_2 \geq (a_t)_l \geq e^{-\frac{C_1}{n_m}} M^{(1-Kp)}.$$

Appendix B. Data Generation

We generate n_k samples independently and identically distributed observations from a multivariate normal distribution $N\{0, (\hat{\mathbf{\Theta}}^{(k)})^{-1}\}$, where $\mathbf{\Theta}^{(k)}$ is the inverse covariance matrix of the k-th category. Specifically, we generate p points randomly on a unit space and calculate their pairwise distances. Then, we find the m-nearest neighbors point by this distance. We connect any two points that are m-nearest neighbors of each other. The integer m determines for the degree of sparsity of the data, and m values range from 4 to 9 in our experiments.

Additionally, we add heterogeneity to the common structure by building extra individual connections in the following way: we randomly choose a pair of symmetric zero elements, $\theta_{k,i,j} = \theta_{k,j,i} = 0$, and replace them with a value uniformly generated from the $[-1, -0, 5] \cup [0.5, 1]$ interval. This operation is repeated $M/2$ times, where M is the number of off-diagonal nonzero elements in $\mathbf{\Theta}^{(k)}$.

References

1. Lauritzen, S.L. *Graphical Models*; Clarendon Press: Oxford, UK, 1996; Volume 17.
2. Meinshausen, N.; Bühlmann, P. High-dimensional graphs and variable selection with the lasso. *Ann. Stat.* **2006**, *34*, 1436–1462. [CrossRef]
3. Yuan, M.; Lin, Y. Model selection and estimation in the Gaussian graphical model. *Biometrika* **2007**, *94*, 19–35. [CrossRef]
4. Friedman, J.; Hastie, T.; Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **2008**, *9*, 432–441. [CrossRef]
5. Banerjee, O.; El Ghaoui, L.; d'Aspremont, A. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.* **2008**, *9*, 485–516.
6. Rothman, A.J.; Bickel, P.J.; Levina, E.; Zhu, J. Sparse permutation invariant covariance estimation. *Electron. J. Stat.* **2008**, *2*, 494–515. [CrossRef]

7. Banerjee, O.; Ghaoui, L.E.; d'Aspremont, A.; Natsoulis, G. Convex optimization techniques for fitting sparse Gaussian graphical models. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 89–96.
8. Xue, L.; Ma, S.; Zou, H. Positive-definite l1-penalized estimation of large covariance matrices. *J. Am. Stat. Assoc.* **2012**, *107*, 1480–1491. [CrossRef]
9. Mazumder, R.; Hastie, T. The graphical lasso: New insights and alternatives. *Electron. J. Stat.* **2012**, *6*, 2125. [CrossRef]
10. Guillot, D.; Rajaratnam, B.; Rolfs, B.T.; Maleki, A.; Wong, I. Iterative thresholding algorithm for sparse inverse covariance estimation. *arXiv* **2012**, arXiv:1211.2532.
11. d'Aspremont, A.; Banerjee, O.; El Ghaoui, L. First-order methods for sparse covariance selection. *SIAM J. Matrix Anal. Appl.* **2008**, *30*, 56–66. [CrossRef]
12. Hsieh, C.J.; Sustik, M.A.; Dhillon, I.S.; Ravikumar, P. QUIC: Quadratic approximation for sparse inverse covariance estimation. *J. Mach. Learn. Res.* **2014**, *15*, 2911–2947.
13. Danaher, P.; Wang, P.; Witten, D.M. The joint graphical lasso for inverse covariance estimation across multiple classes. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2014**, *76*, 373. [CrossRef]
14. Honorio, J.; Samaras, D. *Multi-Task Learning of Gaussian Graphical Models*; ICML: Baltimore, MA, USA, 2010.
15. Guo, J.; Levina, E.; Michailidis, G.; Zhu, J. Joint estimation of multiple graphical models. *Biometrika* **2011**, *98*, 1–15. [CrossRef] [PubMed]
16. Zhang, B.; Wang, Y. Learning structural changes of Gaussian graphical models in controlled experiments. *arXiv* **2012**, arXiv:1203.3532.
17. Hara, S.; Washio, T. Learning a common substructure of multiple graphical Gaussian models. *Neural Netw.* **2013**, *38*, 23–38. [CrossRef]
18. Glowinski, R.; Marroco, A. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *ESAIM: Math. Model. Numer. Anal.-Modél. Math. Et Anal. Numér.* **1975**, *9*, 41–76. [CrossRef]
19. Tang, Q.; Yang, C.; Peng, J.; Xu, J. Exact hybrid covariance thresholding for joint graphical lasso. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 593–607.
20. Hallac, D.; Park, Y.; Boyd, S.; Leskovec, J. Network inference via the time-varying graphical lasso. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 205–213.
21. Gibberd, A.J.; Nelson, J.D. Regularized estimation of piecewise constant gaussian graphical models: The group-fused graphical lasso. *J. Comput. Graph. Stat.* **2017**, *26*, 623–634. [CrossRef]
22. Boyd, S.; Parikh, N.; Chu, E. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*; Now Publishers Inc.: Norwell, MA, USA, 2011.
23. Yang, S.; Lu, Z.; Shen, X.; Wonka, P.; Ye, J. Fused multiple graphical lasso. *SIAM J. Optim.* **2015**, *25*, 916–943. [CrossRef]
24. Tran-Dinh, Q.; Kyrillidis, A.; Cevher, V. Composite self-concordant minimization. *J. Mach. Learn. Res.* **2015**, *16*, 371–416.
25. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [CrossRef]
26. Nesterov, Y.; Nemirovskii, A. *Interior-Point Polynomial Algorithms in Convex Programming*; SIAM: Philadelphia, PA, USA, 1994.
27. Renegar, J. *A Mathematical View of Interior-Point Methods in Convex Optimization*; SIAM: Philadelphia, PA, USA, 2001.
28. Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*; Springer Science & Business Media: New York, NY, USA, 2003; Volume 87.
29. Tibshirani, R.; Saunders, M.; Rosset, S.; Zhu, J.; Knight, K. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B* **2005**, *67*, 91–108. [CrossRef]
30. Simon, N.; Friedman, J.; Hastie, T.; Tibshirani, R. A sparse-group lasso. *J. Comput. Graph. Stat.* **2013**, *22*, 231–245. [CrossRef]
31. Friedman, J.; Hastie, T.; Tibshirani, R. A note on the group lasso and a sparse group lasso. *arXiv* **2010**, arXiv:1001.0736.
32. Hoefling, H. A path algorithm for the fused lasso signal approximator. *J. Comput. Graph. Stat.* **2010**, *19*, 984–1006. [CrossRef]
33. Friedman, J.; Hastie, T.; Höfling, H.; Tibshirani, R. Pathwise coordinate optimization. *Ann. Appl. Stat.* **2007**, *1*, 302–332. [CrossRef]
34. Tibshirani, R.J.; Taylor, J. The solution path of the generalized lasso. *Ann. Stat.* **2011**, *39*, 1335–1371. [CrossRef]
35. Johnson, N.A. A dynamic programming algorithm for the fused lasso and l 0-segmentation. *J. Comput. Graph. Stat.* **2013**, *22*, 246–260. [CrossRef]
36. Yuan, M.; Lin, Y. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B* **2006**, *68*, 49–67. [CrossRef]
37. Suzuki, J. *Sparse Estimation with Math and R: 100 Exercises for Building Logic*; Springer Nature: Berlin/Heidelberg, Germany 2021.
38. Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
39. Barzilai, J.; Borwein, J.M. Two-point step size gradient methods. *IMA J. Numer. Anal.* **1988**, *8*, 141–148. [CrossRef]
40. Nemirovski, A. Interior point polynomial time methods in convex programming. *Lect. Notes* **2004**, *42*, 3215–3224.
41. Li, H.; Gui, J. Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics* **2006**, *7*, 302–317. [CrossRef]
42. Weylandt, M.; Nagorski, J.; Allen, G.I. Dynamic visualization and fast computation for convex clustering via algorithmic regularization. *J. Comput. Graph. Stat.* **2020**, *29*, 87–96. [CrossRef] [PubMed]

43. Shannon, P.; Markiel, A.; Ozier, O.; Baliga, N.S.; Wang, J.T.; Ramage, D.; Amin, N.; Schwikowski, B.; Ideker, T. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Res.* **2003**, *13*, 2498–2504. [CrossRef] [PubMed]
44. Miller, L.D.; Smeds, J.; George, J.; Vega, V.B.; Vergara, L.; Ploner, A.; Pawitan, Y.; Hall, P.; Klaar, S.; Liu, E.T.; et al. An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival. *Proc. Natl. Acad. Sci. USA* **2005**, *102*, 13550–13555. [CrossRef] [PubMed]