A Fast Algorithm for Graph Learning under Attractive Gaussian Markov Random Fields

Jiaxi Ying, José Vinícius de M. Cardoso, and Daniel P. Palomar
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
Email: {jx.ying, jvdmc}@connect.ust.hk, palomar@ust.hk

2021 55th Asilomar Conference on Signals, Systems, and Computers | 978-1-6654-5828-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IEEECONF53345.2021.9723090

***Abstract*—We consider the problem of graph learning under Gaussian Markov random fields, where all partial correlations are nonnegative. Such model is called attractive Gaussian Markov random fields, and has received considerable attention in recent years. The graph learning problem under this model can be formulated as the ℓ_1-norm regularized Gaussian maximum likelihood estimation of the precision matrix under sign constraints. In this paper, we propose a projected Newton-like algorithm, which is computationally efficient. By exploiting the structure of the Gaussian maximum likelihood estimation problem, the proposed algorithm significantly reduces the computational cost in computing the approximate Newton direction. Then we prove that the proposed method can recover the graph edges correctly under the irrepresentability condition. Numerical results on synthetic and financial time-series data sets demonstrate the effectiveness of the proposed method.**

I. Introduction

We consider the problem of learning graphs under attractive Gaussian Markov random fields, where all the partial correlations are nonnegative. Such property is also known as multivariate totally positive of order 2 (MTP_2), and its applications include actuarial sciences [1], taxonomic reasoning [2], financial markets [3], [4], factor analysis in psychometrics [5], and graph signal processing [6], [7]. For example, in financial markets, instruments usually have positive dependencies as a result of the market factor [3], [8]. Graph learning under the attractive Gaussian Markov random fields can be formulated as estimating the precision matrices under MTP_2 constraints.

Recent works reported that MTP_2 constraints can reduce the sample complexity to make the maximum likelihood estimator exist [2], [5], [9], [10]. One interesting result provided in [2], [5] showed that the maximum likelihood estimator under MTP_2 constraints exists if the sample size satisfies $n \geq 2$, independent of the underlying dimension p. This result leads to a significant reduction from $n \geq p$, which is necessary for the maximum likelihood estimator to exist under unconstrained Gaussian graphical models. Aside from Gaussian distributions, the advantages of MTP_2 constraints have also been explored in the binary exponential family, showing that the maximum likelihood estimator may exist with only $n = p$ observations [10], while $n \geq 2^p$ is required if there are no MTP_2 constraints. Such advantages of MTP_2 constraints are significant in the high-dimensional regime, where the samples are usually limited compared with the dimension.

Graph learning under Gaussian Markov random fields has been widely studied. One well-known method is the graphical lasso [11], [12], [13], which is formulated as the ℓ_1-norm penalized Gaussian maximum likelihood estimation. Various extensions of graphical lasso and their theoretical properties have also been studied [14], [15], [16], [17], [18], [19], [20], [21], [22]. For the case of attractive Gaussian Markov random fields, the authors in [2], [23] proposed the Gaussian maximum likelihood estimation method under MTP_2 constraints, and developed a block-coordinate descent (BCD) algorithm. The authors in [6] incorporated the ℓ_1-norm regularization and connectivity constraints, and developed a similar scheme by cyclically updating each column/row. Note that BCD algorithms are usually time-consuming in high-dimensional problems. In another direction, one may consider second-order methods. However, they usually need to compute the (approximate) inverse Hessian, which leads to computational inefficiency. In addition, it is still unknown whether the ℓ_1-norm approaches can succeed to recover the underlying graph edges under attractive Gaussian Markov random fields.

The main contributions of this paper are threefold:

- We propose a computationally efficient projected Newton-like algorithm to solve the ℓ_1-norm regularized Gaussian maximum likelihood estimation under MTP_2 constraints. By exploiting the structure of the Gaussian maximum likelihood estimation, the proposed algorithm significantly reduces the computational cost in computing the approximate Newton direction.
- We prove that the ℓ_1-norm regularized Gaussian maximum likelihood estimation method can recover the graph edges correctly under irrepresentability condition.
- Numerical experiments on synthetic and financial time-series data demonstrate the effectiveness of the proposed algorithm. It is observed that the proposed algorithm takes less computational time than the state-of-the-art methods.

The remainder of the paper is organized as follows. Preliminaries and related work are provided in Section II. We propose a novel algorithm, and present the theoretical results on the successful edge recovery guarantees in Section III. Experimental results are provided in Section IV, and conclusions are made in Section V.

This work was supported by the Hong Kong GRF 16207820 research grant.

Notations: Lower case bold letters denote vectors and upper case bold letters denote matrices. Both X_{ij} and $[\boldsymbol{X}]_{ij}$ denote the (i,j)-th entry of $\boldsymbol{X}$. Let $\otimes$ be the Kronecker product, and $\mathrm{supp}(\boldsymbol{X}) = \{(i,j)|X_{ij} \neq 0\}$. $[p]$ denotes the set $\{1,\ldots,p\}$. $\boldsymbol{X}^\top$ denotes transpose of $\boldsymbol{X}$. $\|\boldsymbol{x}\|$, $\|\boldsymbol{X}\|_{\mathrm{F}}$ and $\|\boldsymbol{X}\|_2$ denote Euclidean norm, Frobenius norm and operator norm, respectively. $|||\boldsymbol{X}|||_\infty$ denotes the ℓ_∞/ℓ_∞-operator norm given by $|||\boldsymbol{X}|||_\infty := \max_{i=1,\ldots,p} \sum_{j=1}^p |X_{ij}|$. Let $\|\boldsymbol{x}\|_{\max} = \max_i |x_i|$ and $\|\boldsymbol{x}\|_{\min} = \min_i |x_i|$. $\mathbb{S}_+^p$ and $\mathbb{S}_{++}^p$ denote the sets of positive semi-definite and positive definite matrices with size $p \times p$, respectively.

II. Preliminaries and Related Work

In this section, we first introduce the attractive Gaussian Markov random fields, then present related works.

A. Attractive Gaussian Markov Random Fields

We denote an undirected graph by $\mathcal{G} = (V, E)$, where $V = \{1,\ldots,p\}$ is the vertex set, and E is the edge set. Let $\boldsymbol{y} = (y_1,\ldots,y_p)$ be a zero-mean p-dimensional random vector following $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. We associate the random variables $y_1,\ldots,y_p$ with the vertex set V. Then, the random vector $\boldsymbol{y}$ forms a Gaussian Markov random field with respect to a graph $\mathcal{G} = (V, E)$, where

$$\begin{aligned} \Theta_{ij} \neq 0 &\iff (i,j) \in E \;\forall\, i \neq j, \\ \Theta_{ij} = 0 &\iff y_i \perp\!\!\!\perp y_j \mid \boldsymbol{y}_{[p]\setminus\{i,j\}}, \end{aligned} \tag{1}$$

where $\boldsymbol{\Theta} := \boldsymbol{\Sigma}^{-1}$ is called precision matrix.

In this paper, we focus on the *attractive* Gaussian Markov random fields, where the precision matrix $\boldsymbol{\Theta}$ is a symmetric *M-matrix* [24], *i.e.*, $\left[\Theta_{ij}\right] \leq 0$, for any $i \neq j$. In other words, all the partial correlations are nonnegative in attractive Gaussian Markov random fields.

We aim to estimate a sparse precision matrix under attractive Gaussian Markov random fields given independent and identically distributed observations $\boldsymbol{y}^{(1)},\ldots,\boldsymbol{y}^{(n)} \in \mathbb{R}^p$. Let $\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}^{(i)}(\boldsymbol{y}^{(i)})^\top$ be the sample covariance matrix. The sparse precision matrix estimation under the attractive Gaussian Markov random fields can be formulated as the ℓ_1-norm regularized maximum likelihood estimation under the MTP_2 constraints,

$$\boldsymbol{X}^\star = \arg\min_{\boldsymbol{X} \in \mathcal{M}^p} -\log\det(\boldsymbol{X}) + \mathrm{tr}\,(\boldsymbol{X}\boldsymbol{S}) + \lambda \sum_{i \neq j} |X_{ij}|\,, \tag{2}$$

where $\mathcal{M}^p$ is the set of all p-dimensional, symmetric, non-singular *M-matrices*, which is defined by

$$\mathcal{M}^p := \left\{ \boldsymbol{X} \in \mathbb{S}_{++}^p \,|\, X_{ij} \leq 0, \forall\, i \neq j \right\}. \tag{3}$$

We propose a second-order algorithm in Section III to solve this problem.

B. Related Work

Sparse precision matrix estimation under Gaussian Markov random fields has been extensively studied in the literature. One popular approach is the ℓ_1-norm regularized Gaussian maximum likelihood estimation [11], [12], [13], and numerous algorithms have been proposed for solving this problem. A representative, yet not exhaustive, list of works available in the literature include: block coordinate ascent method [11], [25], Nesterov's smooth gradient method [12], projected gradient method [26], projected quasi-Newton [27], augmented Lagrangian method [28], inexact interior point method [29], primal proximal-point with Newton-CG method [30], and Newton's method with quadratic approximation [31], [32], [33]. However, all the methods mentioned above cannot be directly extended to estimate precision matrices in our problem because of MTP_2 constraints.

To estimate precision matrices under MTP_2 constraints, one option is using the Gaussian maximum likelihood estimator [5], [34], and the sign constraint can promote the sparsity of the solution implicitly. To solve this problem, a primal algorithm [2] and a dual algorithm [23] were proposed based on block coordinate descent. The authors in [6] proposed the ℓ_1-norm regularized Gaussian maximum likelihood estimation, and updated each column/row of the variable by solving a nonnegative quadratic program. In addition, there is growing interest in learning graphs under the generalized graph Laplacian models [6], [23], [35], where the precision matrices satisfy MTP_2 constraints. In this paper, we aim to propose an efficient algorithm for estimating sparse precision matrices under MTP_2 constraints, and establish the successful edge recovery guarantees.

III. Proposed Algorithm and Theoretical Results

In this section, we first derive a second-order algorithm to solve the ℓ_1-norm regularized Gaussian maximum likelihood estimation under MTP_2 constraints, then provide theoretical results on successful recovery of graph edges.

A. Proposed Algorithm

To solve Problem (2), we propose a projected Newton-like algorithm. In each iteration, we partition the algorithm variables into two sets, *i.e.*, *free* and *restricted* sets, and only update the variables in the *free* set while fixing the variables in the *restricted* set. The *restricted* set of variables is defined by

$$\mathcal{I}_k := \left\{ (i,j) \in [p]^2 \;\middle|\; [\boldsymbol{X}_k]_{ij} = 0, [\nabla f\,(\boldsymbol{X}_k)]_{ij} < 0 \right\}, \tag{4}$$

where f is the objective function of Problem (2), and the *free* set is the complement of the *restricted* set, *i.e.*, $\mathcal{I}_k^c$.

Now we construct the projected Newton-like step as follows,

$$\boldsymbol{X}_{k+1} = \mathcal{P}_\Omega\big(\boldsymbol{X}_k - \gamma_k \boldsymbol{P}_k\big), \tag{5}$$

where γ_k is the step size in the k-th iteration, $\mathcal{P}_\Omega$ is the projection onto the set $\Omega := \{\boldsymbol{X} \,|\, X_{ij} \leq 0, \forall i \neq j\}$, and $\boldsymbol{P}_k$ is the approximate Newton direction defined by

$$\mathrm{pvec}_k(\boldsymbol{P}_k) = \boldsymbol{Q}_k\, \mathrm{pvec}_k(\nabla f\,(\boldsymbol{X}_k)), \tag{6}$$

where $\boldsymbol{Q}_k$ is the gradient scaling matrix, which is an approximate inverse Hessian. We define $\mathrm{pvec}_k(\boldsymbol{X})$ as follows,

$$\mathrm{pvec}_k(\boldsymbol{X}) := \begin{bmatrix} [\boldsymbol{X}]_{\mathcal{I}_k^c} \\ [\boldsymbol{X}]_{\mathcal{I}_k} \end{bmatrix}, \tag{7}$$

where $[\boldsymbol{X}]_{\mathcal{I}_k} \in \mathbb{R}^{|\mathcal{I}_k|}$ and $[\boldsymbol{X}]_{\mathcal{I}_k^c} \in \mathbb{R}^{|\mathcal{I}_k^c|}$ denote the two vectors containing all the elements of $\boldsymbol{X}$ in the sets $\mathcal{I}_k$ and $\mathcal{I}_k^c$, respectively. According to (7), we can see that $\mathrm{pvec}_k(\boldsymbol{X})$ stacks the elements of $\boldsymbol{X}$ into a single vector, which is similar to $\mathrm{vec}(\boldsymbol{X})$, but it places the elements of $\boldsymbol{X}$ in the sets $\mathcal{I}_k^c$ and $\mathcal{I}_k$ in order.

The approximate Newton direction in (6) may involve computing the inverse of the Hessian matrix with the dimension $p^2 \times p^2$ or solving a system of linear equations of the same dimension, which is computationally challenging. In what follows, we aim to simplify the computation in (6). We construct the gradient scaling matrix $\boldsymbol{Q}_k$ as follows,

$$\boldsymbol{Q}_k = \begin{bmatrix} \left[\boldsymbol{H}_k^{-1}\right]_{\mathcal{I}_k^c\mathcal{I}_k^c} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_k \end{bmatrix},$$

where $\boldsymbol{D}_k$ is a positive definite diagonal matrix with the dimension $|\mathcal{I}_k| \times |\mathcal{I}_k|$, and $\left[\boldsymbol{H}_k^{-1}\right]_{\mathcal{I}_k^c\mathcal{I}_k^c}$ is a principal submatrix of $\boldsymbol{H}_k^{-1}$ keeping rows and columns indexed by $\mathcal{I}_k^c$, in which $\boldsymbol{H}_k$ is the Hessian matrix. By calculation, we obtain $\boldsymbol{H}_k = \boldsymbol{X}_k^{-1} \otimes \boldsymbol{X}_k^{-1}$, and thus

$$\boldsymbol{H}_k^{-1} = \boldsymbol{X}_k \otimes \boldsymbol{X}_k. \tag{8}$$

Following from the property of Kronecker product that $\mathrm{vec}\left(\boldsymbol{ABC}\right) = \left(\boldsymbol{C}^\top \otimes \boldsymbol{A}\right)\mathrm{vec}(\boldsymbol{B})$, we obtain

$$\boldsymbol{H}_k^{-1}\mathrm{vec}\left(\nabla f\left(\boldsymbol{X}_k\right)\right) = \mathrm{vec}\left(\boldsymbol{X}_k \nabla f\left(\boldsymbol{X}_k\right)\boldsymbol{X}_k\right). \tag{9}$$

As a result, we can get

$$\begin{aligned}\boldsymbol{Q}_k\,\mathrm{pvec}_k(\nabla f\left(\boldsymbol{X}_k\right)) &= \begin{bmatrix} \left[\boldsymbol{H}_k^{-1}\right]_{\mathcal{I}_k^c\mathcal{I}_k^c} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_k \end{bmatrix}\begin{bmatrix} \left[\nabla f\left(\boldsymbol{X}_k\right)\right]_{\mathcal{I}_k^c} \\ \left[\nabla f\left(\boldsymbol{X}_k\right)\right]_{\mathcal{I}_k} \end{bmatrix} \\ &= \begin{bmatrix} \left[\boldsymbol{X}_k \mathcal{P}_{\mathcal{I}_k^c}\left(\nabla f\left(\boldsymbol{X}_k\right)\right)\boldsymbol{X}_k\right]_{\mathcal{I}_k^c} \\ \boldsymbol{D}_k\left[\nabla f\left(\boldsymbol{X}_k\right)\right]_{\mathcal{I}_k} \end{bmatrix},\end{aligned}$$

where $\mathcal{P}_{\mathcal{I}_k^c}(\boldsymbol{A})$ is defined as follows,

$$\left[\mathcal{P}_{\mathcal{I}_k^c}(\boldsymbol{A})\right]_{ij} = \begin{cases} A_{ij} & \text{if } (i,j) \in \mathcal{I}_k^c, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, we have

$$\left[\boldsymbol{P}_k\right]_{\mathcal{I}_k^c} = \left[\boldsymbol{X}_k \mathcal{P}_{\mathcal{I}_k^c}\left(\nabla f\left(\boldsymbol{X}_k\right)\right)\boldsymbol{X}_k\right]_{\mathcal{I}_k^c}.$$

In each iteration, we only update the variables in the *free* set, *i.e.*, $[\boldsymbol{X}]_{\mathcal{I}_k^c}$, and set the remaining variables in the *restricted* set to be zero. Finally, we can rewrite the projected Newton-like iteration in (5) as

$$\left[\boldsymbol{X}_{k+1}\right]_{\mathcal{I}_k^c} = \mathcal{P}_\Omega\Big(\left[\boldsymbol{X}_k\right]_{\mathcal{I}_k^c} - \gamma_k\left[\boldsymbol{X}_k \mathcal{P}_{\mathcal{I}_k^c}\left(\nabla f\left(\boldsymbol{X}_k\right)\right)\boldsymbol{X}_k\right]_{\mathcal{I}_k^c}\Big),$$

and

$$[\boldsymbol{X}_{k+1}]_{\mathcal{I}_k} = \boldsymbol{0}.$$

Note that the Newton-type methods for solving our problem (2) are usually computationally expensive, because of the computations of the (approximate) inverse of the Hessian matrices with the dimension $p^2 \times p^2$. However, it is observed that our constructed iterates as shown above only involves the matrix multiplication with the dimension $p \times p$ and gradient computation. The step size can be computed by the Armijo step-size rule.

Algorithm 1 Projected Newton-like method

1: **Input:** Sample covariance matrix $\boldsymbol{S}$, λ;
2: **while** Stopping criteria not met **do**
3: Compute the *restricted* set by
$$\mathcal{I}^k = \left\{(i,j) \in [p]^2 \,\middle|\, [\boldsymbol{X}_k]_{ij} = 0,\, [\nabla f\left(\boldsymbol{X}_k\right)]_{ij} < 0\right\};$$
4: Compute the approximate Newton direction
$$\left[\boldsymbol{P}_k\right]_{\mathcal{I}_k^c} = \left[\boldsymbol{X}_k \mathcal{P}_{\mathcal{I}_k^c}\left(\nabla f\left(\boldsymbol{X}_k\right)\right)\boldsymbol{X}_k\right]_{\mathcal{I}_k^c};$$
5: Update $\boldsymbol{X}_{k+1}$ by setting $[\boldsymbol{X}_{k+1}]_{\mathcal{I}_k} = \boldsymbol{0}$ and
$$[\boldsymbol{X}_{k+1}]_{\mathcal{I}_k^c} = \mathcal{P}_\Omega\big([\boldsymbol{X}_k]_{\mathcal{I}_k^c} - \gamma_k\left[\boldsymbol{P}_k\right]_{\mathcal{I}_k^c}\big);$$
6: $k \leftarrow k+1$;
7: **end while**
8: **Output:** $\boldsymbol{X}^\star$.

B. Theoretical Results

In this subsection, we provide theoretical results to show that the estimator $\boldsymbol{X}^\star$ defined in (2) can recover the underlying graph edges correctly under irrepresentability condition.

Let $\mathcal{S} := \left\{(i,j) \,|\, \Theta_{ij} \neq 0\right\}$ be the support set of the underlying precision matrix $\boldsymbol{\Theta}$, and d be the maximum number of nonzero elements in any row of $\boldsymbol{\Theta}$. Define

$$K_\Sigma := \max_{i\in\{1,\ldots,p\}} \sum_{j=1}^{p} \Sigma_{ij}, \text{ and } K_H := \left\|\left|\left(\boldsymbol{H}_{\mathcal{SS}}\right)^{-1}\right|\right\|_\infty, \tag{10}$$

where $\boldsymbol{\Sigma}$ is the underlying covariance matrix, and $\boldsymbol{H}_{\mathcal{SS}}$ is the principle submatrix of $\boldsymbol{H}$, with both rows and columns indexed by $\mathcal{S}$, in which $\boldsymbol{H}$ is the Hessian matrix at $\boldsymbol{\Theta}$.

Assumption 1. There exists some $\alpha \in (0,1]$ such that

$$\left\|\left|\boldsymbol{H}_{\mathcal{S}^c\mathcal{S}}\left(\boldsymbol{H}_{\mathcal{SS}}\right)^{-1}\right|\right\|_\infty \leq 1-\alpha. \tag{11}$$

Assumption 2. The nonzero elements of the underlying precision matrix $\boldsymbol{\Theta}$ satisfy

$$\min_{(i,j)\in\mathcal{S}} \left|\Theta_{ij}\right| \geq \left(1+\frac{\alpha}{2}\right)K_H\lambda. \tag{12}$$

Assumption 1 presents the irrepresentability condition that is almost necessary for the ℓ_1-norm approaches to recover the supports correctly [14]. Assumption 2 imposes a lower bound on the minimum absolute value of nonzero elements of $\boldsymbol{\Theta}$.

Theorem 1. *Suppose the sample covariance matrix satisfies* $\|\boldsymbol{S}-\boldsymbol{\Sigma}\|_{\max} \leq \frac{\alpha}{4}\lambda$. *Under Assumptions 1 and 2, if the sample size is lower bounded by* $n \geq cd^2\log p$, *where* $c = \left(3\left(1+\frac{\alpha}{2}\right)c_\lambda K_H K_\Sigma \max\left((1+\frac{2}{\alpha})K_H K_\Sigma^2,\, 1\right)\right)^2$, *then* $\boldsymbol{X}^\star$ *obtained by solving Problem* (2) *with* $\lambda = c_\lambda\sqrt{\frac{\log p}{n}}$ *can recover the support of* $\boldsymbol{\Theta}$ *correctly, i.e.,*

$$\mathrm{supp}\left(\boldsymbol{X}^\star\right) = \mathrm{supp}\left(\boldsymbol{\Theta}\right), \tag{13}$$

where c_λ *is a positive constant.*

Theorem 1 shows that the support of the minimizer $\boldsymbol{X}^\star$ of Problem (2) is the same with that of the underlying precision matrix $\boldsymbol{\Theta}$ under the irrepresentability condition, implying that all the underlying graph edges can be identified correctly through $\boldsymbol{X}^\star$. In addition, the condition $\|\boldsymbol{S}-\boldsymbol{\Sigma}\|_{\max} \leq \frac{\alpha}{4}\lambda$ can hold with overwhelming probability $1-c_1\exp(-c_2\log p)$ for Gaussian observations, where c_1 and c_2 are two positive constants.

IV. Experimental Results

In this section, we present numerical results on synthetic data and financial time-series data to verify the performance of the proposed algorithm. All the experiments are conducted on a PC with a 2.8 GHz Inter Core i7 CPU and 16 GB RAM.

A. Synthetic Data

We consider to learn Barabasi-Albert graphs that are useful in many applications. The graph structure and its weights associated with the edges are generated randomly. We obtain a weighted adjacency matrix $\boldsymbol{W}$, where W_{ij} denotes the graph weight between node i and node j. We construct the underlying precision matrix by $\boldsymbol{\Theta}=\boldsymbol{D}-\boldsymbol{W}$, where $\boldsymbol{D}$ is a diagonal matrix, which is generated to ensure that $\boldsymbol{\Theta}$ is an M-matrix and the irrepresentability condition in Assumption 1 holds.

Given the underlying precision matrix $\boldsymbol{\Theta}\in\mathbb{R}^{p\times p}$, we generate n independent observations $\boldsymbol{y}^{(1)},\ldots,\boldsymbol{y}^{(n)} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{\Theta}^{-1})$, and compute the sample covariance matrix by $\boldsymbol{S}=\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}^{(i)}(\boldsymbol{y}^{(i)})^\top$.

We compare the computational time with the state-of-the-art BCD [2], and GGL [6] in solving the ℓ_1-norm regularized Gaussian maximum likelihood estimation. It is observed in Table I that the three methods lead to the same objective function value, because all of them can obtain the global minimizer. However, our proposed algorithm takes significantly less computational time than BCD and GGL. The results are averaged over 10 Monte Carlo realizations. In addition, GGL can incorporate connectivity constraints in graph learning.

TABLE I: Comparisons of computational time in learning Barabasi-Albert graphs. The "Objective" denotes the objective function value, and the unit of time is seconds.

Methods	$p=500$		$p=1000$		$p=3000$	
	Time	Objective	Time	Objective	Time	Objective
Proposed	**0.49**	495.67	**3.10**	997.07	**92.81**	2997.85
BCD	2.08	495.67	16.34	997.07	427.33	2997.85
GGL	2.41	495.67	17.92	997.07	498.65	2997.85

Next, we show that the estimator defined in (2) can recover the graph edges correctly under irrepresentability condition. We use F-score to measure the performance of edge recovery, which is defined by

$$\text{F-score} := \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}, \tag{14}$$

where TP denotes the number of true positives, FP denotes the number of false positives, and FN denotes the number of false negatives. The F-score takes values in $[0,1]$, and 1 indicates perfect structure recovery.

It is observed in Figure 1 that our algorithm can achieve the F-score to be 1, implying that the graph structure is identified perfectly under the irrepresentability condition, which is consistent with our theoretical results in Theorem 1. We compare our method with the well-known Glasso method [25], which does not impose the MTP_2 constraints. We can observe that our algorithm can obtain a higher F-score than Glasso in the region of small sample size ratios. This is expected because imposing more prior knowledge always helps to improve the estimation performance especially when the samples are limited. The results are averaged over 30 realizations.

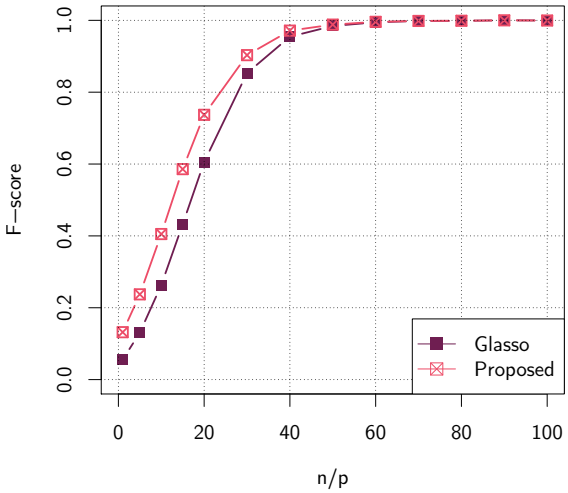


Fig. 1: F-score as a function of the sample size ratio of n/p in learning Barabasi-Albert random graphs. The regularization parameter for both methods is set as $\lambda=0.05$.

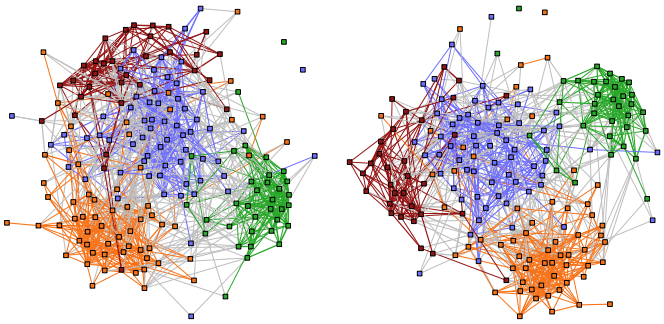
B. Financial Time-series Data

The MTP_2 models are justified well on the stock data since the market factor leads to the positive dependencies among stocks [3]. The data is collected from 189 stocks composing the S&P 500 index during a period from Oct. 1st 2017 to Jan. 1st 2018, resulting in 62 observations per stock, *i.e.*, $p=189$ and $n=62$. We construct the log-returns data matrix by

$$X_{i,j}=\log P_{i,j}-\log P_{i-1,j}, \tag{15}$$

where $P_{i,j}$ denotes the closing price of the j-th stock on the i-th day. The stocks are categorized into 4 sectors: Information Technology, Industrials, Consumer Staples, and Energy.

It is observed in Figure 2 that the performance of our algorithm is better than Glasso, because the latter has more gray-colored connections, which are between stocks from distinct sectors and often spurious from a practical perspective. The modularity values for Glasso and the proposed method are 0.41 and 0.45, respectively. A higher modularity value indicates a better representation of the actual network of stocks.



(a) (b)

Fig. 2: Stock graphs learned via (a) Glasso, and (b) the proposed method.

V. Conclusions

In this paper, we have proposed a projected Newton-like algorithm to solve the ℓ_1-norm regularized Gaussian maximum likelihood estimation under MTP_2 constraints. The proposed algorithm significantly reduces the computational cost in computing the approximate Newton direction. We have proved that our method can recover the graph edges correctly under the irrepresentability condition, which has been verified by numerical results. Experiments have demonstrated that the proposed algorithm takes significantly less computational time than the state-of-the-art methods.

References

[1] M. Denuit, J. Dhaene, M. Goovaerts, and R. Kaas, *Actuarial theory for dependent risks: measures, orders and models*. John Wiley & Sons, 2006.

[2] M. Slawski and M. Hein, "Estimation of positive definite M-matrices and structure learning for attractive Gaussian Markov random fields," *Linear Algebra and its Applications*, vol. 473, pp. 145–179, 2015.

[3] R. Agrawal, U. Roy, and C. Uhler, "Covariance Matrix Estimation under Total Positivity for Portfolio Selection," *Journal of Financial Econometrics*, 09 2020.

[4] Y. Wang, U. Roy, and C. Uhler, "Learning high-dimensional gaussian graphical models under total positivity without adjustment of tuning parameters," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2698–2708.

[5] S. Lauritzen, C. Uhler, P. Zwiernik *et al.*, "Maximum likelihood estimation in Gaussian models under total positivity," *The Annals of Statistics*, vol. 47, no. 4, pp. 1835–1863, 2019.

[6] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constrints," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 825–841, 2017.

[7] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, 2019.

[8] J. V. d. M. Cardoso, J. Ying, and D. P. Palomar, "Algorithms for learning graphs in financial markets," *arXiv preprint arXiv:2012.15410*, 2020.

[9] S. Fallat, S. Lauritzen, K. Sadeghi, C. Uhler, N. Wermuth, P. Zwiernik *et al.*, "Total positivity in Markov structures," *The Annals of Statistics*, vol. 45, no. 3, pp. 1152–1184, 2017.

[10] S. Lauritzen, C. Uhler, and P. Zwiernik, "Total positivity in exponential families with application to binary variables," *The Annals of Statistics*, vol. 49, no. 3, pp. 1436–1459, 2021.

[11] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, no. Mar, pp. 485–516, 2008.

[12] A. d'Aspremont, O. Banerjee, and L. El Ghaoui, "First-order methods for sparse covariance selection," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 56–66, 2008.

[13] M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

[14] P. Ravikumar, M. J. Wainwright, G. Raskutti, B. Yu *et al.*, "High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence," *Electronic Journal of Statistics*, vol. 5, pp. 935–980, 2011.

[15] R. Mazumder and T. Hastie, "The graphical lasso: New insights and alternatives," *Electronic Journal of Statistics*, vol. 6, p. 2125, 2012.

[16] C.-J. Hsieh, A. Banerjee, I. S. Dhillon, and P. K. Ravikumar, "A divide-and-conquer method for sparse inverse covariance estimation," in *Advances in Neural Information Processing Systems*, 2012, pp. 2330–2338.

[17] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack, "BIG & QUIC: Sparse inverse covariance estimation for a million variables," in *Advances in Neural Information Processing Systems*, 2013, pp. 3165–3173.

[18] E. Yang, G. Allen, Z. Liu, and P. K. Ravikumar, "Graphical models via generalized linear models," in *Advances in Neural Information Processing Systems*, 2012, pp. 1358–1366.

[19] E. Yang, P. Ravikumar, G. I. Allen, and Z. Liu, "Graphical models via univariate exponential family distributions," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 3813–3847, 2015.

[20] J. Honorio, D. Samaras, I. Rish, and G. Cecchi, "Variable selection for Gaussian graphical models," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 538–546.

[21] C. McCarter and S. Kim, "Large-scale optimization algorithms for sparse conditional Gaussian graphical models," in *International Conference on Artificial Intelligence and Statistics*, 2016, pp. 528–537.

[22] J. Chen, P. Xu, L. Wang, J. Ma, and Q. Gu, "Covariate adjusted precision matrix estimation via nonconvex optimization," in *International Conference on Machine Learning*, 2018, pp. 921–930.

[23] E. Pavez and A. Ortega, "Generalized Laplacian precision matrix estimation for graph signal processing," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6350–6354.

[24] E. Bølviken, "Probability inequalities for the multivariate normal with non-negative partial correlations," *Scandinavian Journal of Statistics*, pp. 49–58, 1982.

[25] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[26] J. Duchi, S. Gould, and D. Koller, "Projected subgradient methods for learning sparse gaussians," in *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, 2008, p. 153–160.

[27] M. Schmidt, E. Berg, M. Friedlander, and K. Murphy, "Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm," in *Artificial Intelligence and Statistics*, 2009, pp. 456–463.

[28] K. Scheinberg, S. Ma, and D. Goldfarb, "Sparse inverse covariance selection via alternating linearization methods," in *Advances in Neural Information Processing Systems*, 2010, pp. 2101–2109.

[29] L. Li and K.-C. Toh, "An inexact interior point method for L1-regularized sparse covariance selection," *Mathematical Programming Computation*, vol. 2, no. 3-4, pp. 291–315, 2010.

[30] C. Wang, D. Sun, and K.-C. Toh, "Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 2994–3013, 2010.

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

[32] F. Oztoprak, J. Nocedal, S. Rennie, and P. A. Olsen, "Newton-like methods for sparse inverse covariance estimation," in *Advances in Neural Information Processing Systems*, 2012, pp. 755–763.

[33] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar, "QUIC: quadratic approximation for sparse inverse covariance estimation," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2911–2947, 2014.

[34] J. A. Soloff, A. Guntuboyina, and M. I. Jordan, "Covariance estimation with nonnegative partial correlations," *arXiv preprint arXiv:2007.15252*, 2020.

[35] E. Pavez, H. E. Egilmez, and A. Ortega, "Learning graphs with monotone topology properties and multiple connected components," *IEEE Transactions on Signal Processing*, vol. 66, no. 9, pp. 2399–2413, 2018.