Precision Matrix Estimation in High Dimensional Gaussian Graphical Models with Faster Rates

Lingxiao Wang
University of Virginia

Xiang Ren
UIUC

Quanquan Gu
University of Virginia

Abstract

We present a new estimator for precision matrix in high dimensional Gaussian graphical models. At the core of the proposed estimator is a collection of node-wise linear regression with nonconvex penalty. In contrast to existing estimators for Gaussian graphical models with $O(s\sqrt{\log d/n})$ estimation error bound in terms of spectral norm, where s is the maximum degree of a graph, the proposed estimator could attain $O(s/\sqrt{n}+\sqrt{\log d/n})$ spectral norm based convergence rate in the best case, and it is no worse than exiting estimators in general. In addition, our proposed estimator enjoys the oracle property under a milder condition than existing estimators. We show through extensive experiments on both synthetic and real datasets that our estimator outperforms the state-of-the-art estimators.

1 INTRODUCTION

In high dimensional statistical learning, the problem of estimating the sparse inverse covariance matrix (precision matrix) in the Gaussian graphical model has attracted increasing attention in recent years [2, 11, 20, 22]. In Gaussian graphical models, a d-dimensional random vector $\boldsymbol{X} = (X_1, \dots, X_d)^\top$ follows a multivariate normal distribution $N_d(\mathbf{0}, \boldsymbol{\Sigma}^*)$. It corresponds to the vertex set $V = \{1, \dots, d\}$ of an undirected graph $G = (V, E)$, where the edge set E describes the conditional independence relationships between $X_1, \dots, X_d$. It is well-known that the graph G is encoded by the sparsity pattern of the precision matrix $\boldsymbol{\Theta}^* = \boldsymbol{\Sigma}^{*-1}$. More specifically, no edge connects X_i and X_j if and only if $\Theta^*_{ij} = 0$. Consequently, estimation of the precision matrix $\boldsymbol{\Theta}^*$ corresponds to parameter estimation, and specifying the non-zero set of $\boldsymbol{\Theta}^*$ corresponds to graphical model selection [4]. Note that in Gaussian graphical models, a graph with maximal degree s corresponds to the precision matrix with at most s non-zero elements per row, and we call this precision matrix is s-sparse.

A large body of literature has studied the problem of estimating the precision matrix in the high dimensional setting. For example, Meinshausen and Bühlmann [20] proposed a neighborhood pursuit approach for estimating Gaussian graphical models by solving a collection of sparse regression problems using Lasso in parallel. Yuan and Lin [30], Friedman et al. [8], Banerjee et al. [1] developed a ℓ_1-penalized likelihood approach to directly estimate the precision matrix, namely graphical Lasso (GLasso). Rothman et al. [24], Ravikumar et al. [22] studied the theoretical properties of GLasso under different assumptions. For example, Rothman et al. [24] derived a rate in terms of Frobenius norm under mild conditions on the eigenvalues of $\boldsymbol{\Theta}^*$. More recently, Yuan [29], Cai et al. [2] proposed the graphical Dantzig selector and CLIME, respectively. Both of these methods can be solved by linear programming and have more favorable theoretical properties than GLasso. In particular, they are able to attain $O(s\sqrt{\log d/n})$ estimator error rate in terms of spectral norm. Very recently, Ren et al. [23] developed a novel estimator based on scaled Lasso [25] to estimate each column of the precision matrix. One draw back of their method is that they need to solve a large number of regression problems. In their study, they proved the estimation error bound of their estimator matches the minimax lower bound, and provided a confidence interval for each entry of the precision matrix. However, all above estimators are based on the convex penalty, which incurs bias in estimation. Recent studies [5, 10, 12, 31, 33] have shown that the nonconvex regularization is able to correct intrinsic estimation bias. Motivated by advantages of the nonconvex regularization, Lam and Fan [16] proposed the nonconvex penalized likelihood approach for the s-parse precision matrix estimation problem, where they replaced the ℓ_1-penalty with the nonconvex penalty

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

SCAD [5] in GLasso. Nevertheless, they failed to get a sharper convergence rate in their analysis. Recently, Loh and Wainwright [19] also studied Gaussian graphical model selection with non-convex regularizers, but they did not consider the magnitude of the nonzero entries in the precision matrix and failed to provide a faster convergence rate. In another recent work, Fan et al. [7] proved the oracle property of their estimator using the nonconvex regularization in GLasso. However, in the general case, they did not produce a sharper convergence rate either.

Another line of research is to scale up Gaussian graphical model algorithms to large scale data, from the computational perspective. For example, Hsieh et al. [14, 15] provided an algorithm QUIC and its extension to solve the GLasso with millions of dimensions. Wang et al. [27] developed a large scale distributed algorithm to estimate the sparse precision matrix with extra large dimension using CLIME. Although all these algorithms can solve extremely high dimensional problems, the statistical rate of their estimators are no better than the original GLasso and CLIME estimators.

In this paper, we propose a new estimator for the precision matrix $\mathbf{\Theta}^*$ in the Gaussian Graphical model based on a collection of node-wise linear regression. More specifically, our method estimates each column of the precision matrix $\mathbf{\Theta}^*$ by solving a sparse linear regression problem using nonconvex penalty. The idea of estimating the precision matrix in a column-by-column fashion has been previously used in many other methods, such as neighborhood selection [20], graph Dantzig selector [29], and CLIME [2]. We will show that our estimator outperforms existing estimators both theoretically and empirically. In particular, we derive a sharper convergence rate for our estimator under mild conditions. We show that when the s-sparse precision matrix $\mathbf{\Theta}^*$ enjoys the large entry magnitude property, our estimator attains $O(s/\sqrt{n}+\sqrt{\log d/n})$ convergence rate in terms of spectral norm, which is much faster than the spectral norm based estimation error bound $O(s\sqrt{\log d/n})$ of existing estimators [20, 22, 29, 2]. In addition, we also prove the oracle property of our estimator under this large magnitude assumption. Even if the large entry magnitude assumption does not hold, the convergence rate of our estimator is $O(\sqrt{s}\sqrt{s_1/n}+\sqrt{s}\sqrt{s_2\log d/n}+\sqrt{\log d/n})$ in terms of spectral norm, which is also faster than existing estimators. Here s_1 corresponds to the precision matrix $\mathbf{\Theta}^*$ with at least s_1 elements per row which enjoy the large entry magnitude property, and $s_2 = s - s_1$.

In terms of implementation, our method is attractive because of its simplicity and scalability in the high dimensional setting. In order to obtain our estimator, we only need to solve a bunch of linear regression problems, where each problem estimates one column of the precision matrix. As a result, our estimator can be easily paralleled or implemented in a distributed manner.

The remainder of this paper is organized as follows: Section 2 introduces notations used in this paper and also gives some necessary backgrounds. In Section 3, we summarize our proposed method in general. Section 4 presents our main results, as well as comparisons with some existing methods. Section 5 provides numerical results, for our method and a number of other methods, of some simulated data sets and a real example on breast cancer. Section 6 concludes with discussion.

2 NOTATION AND BACKGROUND

In this section, we introduce notations to be used throughout the paper, and describe nonconvex penalty functions we use in our method.

2.1 Notation

Let $\mathbf{x} = (x_1, \ldots, x_d)^\top \in \mathbb{R}^d$ be a d-dimensional vector. For $0 \le q \le \infty$, we define the ℓ_0, ℓ_q and ℓ_∞ vector norms as $\|\mathbf{x}\|_0 = \sum_{i=1}^d \mathbb{I}(x_i \neq 0), \|\mathbf{x}\|_q = \Big(\sum_{i=1}^d |x_i|^q\Big)^{\frac{1}{q}}, \|\mathbf{x}\|_\infty = \max_{1\le i\le d}|x_i|$, where $\mathbb{I}(\cdot)$ denotes the indicator function. Let $\mathbf{A} = (A_{ij}) \in \mathbb{R}^{d\times d}$ be a $d \times d$ matrix. We denote $\mathbf{A}_{*j} = (\boldsymbol{A}_{1j}, \ldots, \boldsymbol{A}_{dj})^\top$ to be the j^{th} column vector of $\mathbf{A}$ and $\mathbf{A}_{*\backslash j}$ to be the submatrix of $\mathbf{A}$ with the j^{th} column $\mathbf{A}_{*j}$ removed. We use the following notation for the matrix ℓ_q, $\ell_{\max}$ and ℓ_F norms

$$\|\mathbf{A}\|_q = \max_{\|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\|_q, \quad \|\mathbf{A}\|_\infty = \max_{ij} |A_{ij}|,$$
$$\|\mathbf{A}\|_1 = \max_j \|\mathbf{A}_{*j}\|_1, \quad \|\mathbf{A}\|_F = \Big(\sum_{ij} |A_{ij}|^2\Big)^{\frac{1}{2}}.$$

It is easy to see that when $q = 2$, $\|\mathbf{A}\|_2$ is the spectral norm of matrix $\mathbf{A}$. We also denote by $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ the largest and smallest eigenvalues of matrix $\mathbf{A}$, respectively. In this paper, the set $\mathcal{S}_{++}^d = \{\mathbf{A} \in \mathbb{R}^{d\times d} \mid \mathbf{A} = \mathbf{A}^\top, \mathbf{A} \succ 0\}$ denotes all symmetric positive definite matrices in d dimensions, where $\mathbf{A} \succ 0$ means $\mathbf{A}$ is positive definite. Furthermore, for a matrix $\mathbf{\Theta}$ and a set of tuples S, $\mathbf{\Theta}_S$ denotes the set of numbers $(\Theta_{jk})_{(j,k)\in S}$. Finally, as mentioned in introduction, we define the maximum degree of a graph or row cardinality as $s = \max_{1\le i\le n} |\{j \in V \mid \Theta^*_{ij} \neq 0\}|$, where $V = \{1, \ldots, d\}$ is the vertex set mentioned before.

2.2 Nonconvex Penalty Functions

One important component of our method is the nonconvex penalty. Throughout this paper, we consider the decomposable nonconvex penalty function, *i.e.*, $\mathcal{G}_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^{d-1} g_\lambda(\beta_i)$. Examples of these nonconvex penalties include the smoothly clipped absolute deviation (SCAD) penalty [5] and minimax concave penalty MCP [31]. These estimators can eliminate the estimation bias and attain more refined statistical rates of convergence [28]. For example, MCP penalty is defined as follows

$$g_\lambda(\beta) = \lambda \int_0^{|\beta|} \big(1 - \frac{z}{\lambda b}\big)_+ dz,$$

where $b > 0$, $\lambda > 0$ are fixed regularization parameters. In particular, the nonconvex penalty $g_\lambda(\beta)$ can be equivalently rewritten into the sum of the ℓ_1 penalty and a concave function $h_\lambda(\beta)$, *i.e.*, $g_\lambda(\beta) = \lambda|\beta| + h_\lambda(\beta)$. In the case that $g_\lambda(\beta)$ is MCP penalty, $h_\lambda(\beta)$ has the form as follows

$$h_\lambda(\beta) = -\frac{\beta^2}{2b}\mathbb{I}(|\beta| \le b\lambda) + \Big(\frac{b\lambda^2}{2} - \lambda|\beta|\Big)\mathbb{I}(|\beta| > b\lambda).$$

In fact, our method is not limited to specific nonconvex penalties, like MCP and SCAD. More generally, we only require some common regularity conditions, which will be described later, on $g_\lambda(\beta)$ and its concave component $h_\lambda(\beta)$.

3 NEIGHBORHOOD SELECTION WITH NONCONVEX PENALTIES

In this section, we will introduce our proposed estimator, which is based on neighborhood selection [20].

3.1 Properties of Gaussian Graphical Models

It is well known that if $\boldsymbol{X}$ follows a multivariate normal distribution: $\boldsymbol{X} \sim N_d(\mathbf{0}, \boldsymbol{\Sigma}^*)$, then the conditional distribution of X_j given $\boldsymbol{X}_{\setminus j}$ remains normally distributed as $X_j|\boldsymbol{X}_{\setminus j} \sim N\big(\boldsymbol{\Sigma}^*_{j,\setminus j}\boldsymbol{\Sigma}^{*-1}_{\setminus j,\setminus j}\boldsymbol{X}_{\setminus j},\ \Sigma^*_{jj} - \boldsymbol{\Sigma}^*_{j,\setminus j}\boldsymbol{\Sigma}^{*-1}_{\setminus j,\setminus j}\boldsymbol{\Sigma}^*_{\setminus j,j}\big)$, where $\setminus j = \{1, \ldots, j-1, j+1, \ldots, d\}$. This implies that $X_j = \boldsymbol{X}_{\setminus j}^\top \boldsymbol{\beta}_j^* + \epsilon_j$, where $\boldsymbol{\beta}_j^* = \boldsymbol{\Sigma}^{*-1}_{\setminus j,\setminus j}\boldsymbol{\Sigma}^*_{\setminus j,j}$ is a $(d-1)$-dimensional vector, and $\epsilon_j \sim N(0, \Sigma^*_{jj} - \boldsymbol{\Sigma}^*_{j,\setminus j}\boldsymbol{\Sigma}^{*-1}_{\setminus j,\setminus j}\boldsymbol{\Sigma}^*_{\setminus j,j})$. Thus, $\sigma_j^2 = \mathrm{Var}(\epsilon_j) = \Sigma^*_{jj} - \boldsymbol{\Sigma}^*_{j,\setminus j}\boldsymbol{\Sigma}^{*-1}_{\setminus j,\setminus j}\boldsymbol{\Sigma}^*_{\setminus j,j}$. By the inverse formula for block matrices [9], we can show that

$$\begin{aligned}\Theta^*_{jj} &= (\Sigma^*_{jj} - \boldsymbol{\Sigma}^*_{j,\setminus j}\boldsymbol{\Sigma}^{*-1}_{\setminus j,\setminus j}\boldsymbol{\Sigma}^*_{\setminus j,j})^{-1}\\ &= (\Sigma^*_{jj} - 2\boldsymbol{\beta}_j^{*\top}\boldsymbol{\Sigma}^*_{\setminus j,j} + \boldsymbol{\beta}_j^{*\top}\boldsymbol{\Sigma}^*_{\setminus j,\setminus j}\boldsymbol{\beta}_j^*)^{-1},\\ \boldsymbol{\Theta}^*_{\setminus j,j} &= -\Theta^*_{jj}\boldsymbol{\beta}_j^*.\end{aligned}$$

This immediately yields

$$\Theta^*_{jj} = (\sigma_j^2)^{-1}, \quad \boldsymbol{\Theta}^*_{\setminus j,j} = -(\sigma_j^2)^{-1}\boldsymbol{\beta}_j^*. \qquad (3.1)$$

Now, we can recover each column of $\boldsymbol{\Theta}^*$, and the problem turns into estimating $\boldsymbol{\beta}_j^*$ for each variable X_j given $\boldsymbol{X}_{\setminus j}$. Note that in our paper, we denote the support set of the vector $\boldsymbol{\beta}_j^* = (\beta_1^*, \ldots, \beta_{d-1}^*)$ as $S_j = \{i \mid \beta_i^* \neq 0\}$ for $j = 1, \ldots, d$.

3.2 The Proposed Estimator

Given n i.i.d. observations $\mathbf{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n]^\top$, our proposed estimator is as follows, where $\lambda > 0$ is a tuning parameter.

1. Estimation: For $j = 1, \ldots, d$, calculate

$$\widehat{\boldsymbol{\beta}}_j = \mathop{\mathrm{argmin}}_{\boldsymbol{\beta}} \frac{1}{2n}\|\mathbf{X}_{*j} - \mathbf{X}_{*\setminus j}\boldsymbol{\beta}\|_2^2 + \mathcal{G}_\lambda(\boldsymbol{\beta}), \qquad (3.2)$$

$$\begin{aligned}&\widehat{\sigma}_j^2 = \frac{1}{n}\|\mathbf{X}_{*j} - \mathbf{X}_{*\setminus j}\widehat{\boldsymbol{\beta}}_j\|_2^2, \quad \widehat{\Theta}^1_{jj} = 1/\widehat{\sigma}_j^2,\\ &\widehat{\boldsymbol{\Theta}}^1_{\setminus jj} = -\widehat{\Theta}^1_{jj}\widehat{\boldsymbol{\beta}}_j. \qquad (3.3)\end{aligned}$$

2. Symmetrization:

$$\widehat{\boldsymbol{\Theta}} = \mathop{\mathrm{argmin}}_{\boldsymbol{\Theta} \in \mathcal{S}^d_{++}} \|\boldsymbol{\Theta} - \widehat{\boldsymbol{\Theta}}^1\|_1. \qquad (3.4)$$

Note that the nonconvex penalty term in (3.2) can avoid over-penalization when the magnitude is very large, which allows us to get a sharper convergence rate. Although $\|\mathbf{X}_{*j} - \mathbf{X}_{*\setminus j}\boldsymbol{\beta}\|_2^2$ is quadratic and $\mathcal{G}_\lambda(\boldsymbol{\beta})$ is nonconvex, it can be shown that, under certain conditions, the problem in (3.2) is strongly convex, and therefore has a unique global solution. For computational purpose, the regression problem (3.2) can be solved by proximal-gradient descent algorithm with iteration complexity $O(\log(1/\epsilon))$, where ϵ is the optimization precision. And the symmetrization procedure in (3.4) can be solved by the projected gradient descent method. Based on the strongly convex property of the problem (3.2), we will show the theoretical results of our estimator in next section.

4 THEORETICAL RESULTS

This section presents our main results, and discusses connections with some related works. We start by stating some assumptions, which are required in our analysis.

4.1 Assumptions

The first assumption is about regularity conditions on the nonconvex penalty $g_\lambda(\beta)$ introduced in Section 2.2. Recall that $g_\lambda(\beta)$ can be formulated as $g_\lambda(\beta) = \lambda|\beta| + h_\lambda(\beta)$.

Assumption 4.1. We provide following regularity conditions on $g_\lambda(\beta)$ and its concave component $h_\lambda(\beta)$:

(a) $g'_\lambda(\beta) = 0$, for $|\beta| \geq \nu > 0$.

(b) $h'_\lambda(\beta)$ is monotone, and Lipschitz continuous, *i.e.*, for $\beta' \geq \beta$, there exists a constant $\zeta_- \geq 0$ such that $-\zeta_-(\beta' - \beta) \leq h'_\lambda(\beta') - h'_\lambda(\beta)$.

(c) $h_\lambda(\beta)$ and $h'_\lambda(\beta)$ pass through the origin, *i.e.*, $h_\lambda(0) = h'_\lambda(0) = 0$.

(d) $h'_\lambda(\beta)$ is bounded, *i.e.*, $|h'_\lambda(\beta)| \leq \lambda$ for any β.

The above conditions have been made in Loh and Wainwright [18], Wang et al. [28] and hold for a variety of nonconvex penalty functions including MCP penalty and SCAD penalty. In particular, MCP penalty satisfies the above conditions with $\nu = b\lambda$ and $\zeta_- = 1/b$.

Next, we impose an important eigenvalue condition on the population covariance matrix.

Assumption 4.2. We have $1/\kappa \leq \lambda_{\min}(\mathbf{\Sigma}^*) \leq \lambda_{\max}(\mathbf{\Sigma}^*) \leq \kappa$, where κ is a constant.

From Assumption 4.2, we can exclude singular or nearly singular covariance matrices, thus guarantee the uniqueness of $\mathbf{\Theta}^*$.

In this paper, we consider the precision matrix $\mathbf{\Theta}^*$ which belongs to the class of matrices $\mathcal{U}(0, s)$, *i.e.*, $\mathcal{U}(0, s) = \big\{\mathbf{\Omega} \in \mathbb{R}^{p\times p} \mid \mathbf{\Omega} \succ 0, \|\mathbf{\Omega}\|_1 \leq M, \max_{1\leq i\leq p}\sum_{j=1}^{p} \mathbb{1}(\Omega_{ij} \neq 0) \leq s\big\}$. Note that this sparse precision matrix class has been previously considered in Cai et al. [2], Liu and Wang [17], Zhao and Liu [32]. In addition, it immediately implies that $\|\mathbf{\Theta}^*_{*j}\|_1 \leq \|\mathbf{\Theta}^*\|_1 \leq M$, where $\mathbf{\Theta}^*_{*j}$ is the jth column vector of $\mathbf{\Theta}^*$.

In order to prove the oracle property of our estimator, we first introduce the definition of an oracle estimator, denoted by $\widehat{\mathbf{\Theta}}_O$.

Definition 4.3 (Oracle Estimator)**.** The oracle estimator $\widehat{\mathbf{\Theta}}_O$ is constructed as follows:

1. Estimation: Recall (3.3), we construct $\widehat{\mathbf{\Theta}}_O$ by the same method

$$\widehat{\boldsymbol{\beta}}_{O,j} = \mathop{\mathrm{argmin}}_{\mathrm{supp}(\boldsymbol{\beta})\subseteq S_j} \mathcal{L}(\boldsymbol{\beta}), \quad \text{for} \quad j = 1, \ldots, d, \tag{4.1}$$

$$\widehat{\sigma}^2_{O,j} = \frac{1}{n}\|\mathbf{X}_{*j} - \mathbf{X}_{*\setminus j}\boldsymbol{\beta}_{O,j}\|_2^2, \quad \widehat{\Theta}_{jj} = 1/\widehat{\sigma}^2_{O,j},$$
$$\widehat{\mathbf{\Theta}}_{\setminus jj} = -\widehat{\Theta}_{jj}\widehat{\boldsymbol{\beta}}_{O,j}, \tag{4.2}$$

where $\mathcal{L}(\boldsymbol{\beta}) = 1/(2n)\|\mathbf{X}_{*j} - \mathbf{X}_{*\setminus j}\boldsymbol{\beta}\|_2^2$. We call $\widehat{\boldsymbol{\beta}}_O$ the oracle estimator of $\boldsymbol{\beta}^*$.

2. Symmetrization:

$$\widehat{\mathbf{\Theta}}_O = \mathop{\mathrm{argmin}}_{\mathbf{\Theta}\in\mathcal{S}^d_{++}} \|\mathbf{\Theta} - \widehat{\mathbf{\Theta}}\|_1. \tag{4.3}$$

The oracle estimator $\widehat{\mathbf{\Theta}}_O$ is not a practical estimator, since we do not know the true support in practice. The oracle estimator is introduced as a reference estimator for our analysis.

4.2 Main Theory

We present two main results of our estimator. The first one shows that under a large magnitude condition on elements of the true precision matrix $\mathbf{\Theta}^*$, our estimator can exactly recover the support of $\mathbf{\Theta}^*$. With this condition, we also provide a significant faster convergence rate for our estimator. The second one shows that even without this condition, our estimator can also obtain a sharper convergence rate than existing estimators.

The following theorem presents the advantage of our estimator when all the nonzero entries in the precision matrix have large magnitude: it not only enjoys the oracle property but also attains a faster convergence rate.

Theorem 4.4 (Large Entry Magnitude)**.** Suppose the nonconvex penalty $\mathcal{G}_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^{d-1} g_\lambda(\beta_i)$ satisfies conditions in Assumption 4.1. If nonzero entries of $\mathbf{\Theta}^*$ satisfies

$$\min_{i\neq j} |\Theta^*_{ij}| \geq \nu/\lambda_{\max}(\mathbf{\Sigma}^*) + C\sqrt{\log s/n},$$

and as stated in Assumption 4.2 that $\min|\Theta^*_{jj}| \geq 1/\lambda_{\max}(\mathbf{\Sigma}^*)$. For the estimator $\widehat{\mathbf{\Theta}}$ in (3.4) with the regularization parameter satisfying $\lambda = C\sqrt{\log d/n}$ and $\zeta_- \leq \lambda_{\min}(\mathbf{\Sigma}^*)/64$, we have following results that hold with probability at least $1 - C/d$:

1. $\widehat{\mathbf{\Theta}} = \widehat{\mathbf{\Theta}}_O$, which further implies $\mathrm{supp}(\widehat{\mathbf{\Theta}}) = \mathrm{supp}(\widehat{\mathbf{\Theta}}_O) = \mathrm{supp}(\mathbf{\Theta}^*)$.

2. The convergence rate of $\widehat{\mathbf{\Theta}}$ is

$$\|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}^*\|_2 \leq C_1\frac{s}{\sqrt{n}} + C_2\sqrt{\frac{\log d}{n}},$$

where C_1 and C_2 are absolute constants.

Theorem 4.4 indicates that in order to recover the support of the precision matrix $\mathbf{\Theta}^*$, we need to impose a large entry magnitude assumption. Actually, this assumption is widely assumed in the Gaussian graphical model estimator for model selection. For example, with MCP or SCAD penalty, we require $\min_{i\neq j}|\Theta^*_{ij}| \geq C\sqrt{\log d/n}$ to guarantee oracle property of the proposed estimator, which is similar to the magnitude assumption for model selection of CLIME estimator. In addition, we can see that the convergence rate of our method for large entry magnitude is $O(s/\sqrt{n} + \sqrt{\log d/n})$, which is significant faster than existing methods.

Next, we turn to the general case, where the nonzero entries of $\boldsymbol{\Theta}^*$ have both large and small magnitude. In this case, for $j = 1, \ldots, d$, we define $S_j = \{i \in V \mid \Theta^*_{ij} \neq 0\}$, $S^1_j = \{i \in V \mid |\Theta^*_{ij}| \geq \nu/\lambda_{\max}(\boldsymbol{\Sigma}^*)\}$, and $S^2_j = \{i \in V \mid 0 < |\Theta^*_{ij}| < \nu/\lambda_{\max}(\boldsymbol{\Sigma}^*)\}$. We also define that $s = \max_{j\in V} |S_j|$, $s_1 = \min_{j\in V} |S^1_j|$, and $s_2 = s - s_1$. Note that here s is also the row cardinality, as defined in Section 2.1, for the symmetric matrix $\boldsymbol{\Theta}^*$.

Theorem 4.5 (Arbitrary Entry Magnitude)**.** Suppose the nonconvex penalty $\mathcal{G}_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^{d-1} g_\lambda(\beta_i)$ satisfies conditions in Assumption 4.1. For the estimator $\widehat{\boldsymbol{\Theta}}$ in (3.4) with the regularization parameter satisfying $\lambda = C\sqrt{\log d/n}$ and $\zeta_- \leq \lambda_{\min}(\boldsymbol{\Sigma}^*)/128$, with probability at least $1 - C/d$, we have

$$\|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*\|_2 \leq C_1\sqrt{s}\sqrt{\frac{s_1}{n}} + C_2\sqrt{s}\sqrt{\frac{s_2 \log d}{n}} + C_3\sqrt{\frac{\log d}{n}},$$

where C_1, C_2 and C_3 are absolute constants.

Theorem 4.5 suggests that, for those entries with large magnitude, *i.e.*, $S^1_j \subset S_j$ for all $j \in V$, the error of our estimator is $O(s/\sqrt{n} + \sqrt{\log d/n})$, which is the same as Theorem 4.4. On the other hand, for those entries with small magnitude, *i.e.*, $S^2_j \subset S_j$ for all $j \in V$, our estimation error is $O(s\sqrt{\log d/n})$, which is identical to existing methods such as GLasso and CLIME. Using nonconvex penalty regularization, our method provides a refined rate of convergence, which is better than the rate obtained by existing estimators.

4.3 Comparisons with Existing Estimators

In this subsection, we will compare our estimator with existing estimators in more details.

First, the most relevant estimator to ours is neighborhood selection, which was originally proposed in Meinshausen and Bühlmann [20]. Neighborhood s-election can recover the support of $\boldsymbol{\Theta}^*$ and obtain $O(s\sqrt{\log d/n})$ convergence rate in terms of spectral norm [17]. As we have seen, the rate of our estimator is sharper than theirs. To guarantee the model selection consistency, they imposed incoherence condition on the covariance matrix: $\|\boldsymbol{\Sigma}_{S^cS}(\boldsymbol{\Sigma}^{-1}_{SS})\|_1 \leq 1 - \alpha$, where $\alpha \in (0, 1]$. In contrast, our Theorem 4.4 only requires the Large magnitude entry assumption, which is a mild condition.

Second, we compare our results with GLasso [30, 24, 22]. For example, Ravikumar et al. [22] obtained $O(s\sqrt{\log d/n})$ rate of convergence in spectral norm. In addition, Ravikumar et al. [22] proved the model selection consistency for GLasso. However, both the spectral norm based convergence rate and the consistency result obtained by Ravikumar et al. [22] require the stringent incoherence condition on the covariance matrix. Our proposed estimator achieves a sharper convergence rate in terms of spectral norm and oracle property under the Large magnitude assumption, which is milder than incoherence condition. The reason why non-convex regularizers can lead this improvement is that for parameters with large absolute values, the non-convex regularizers impose smaller penalization than ℓ_1 penalty. Thus it will not over penalized our problem, and is able to remove bias introduced by over penalization. We note that Lam and Fan [16] studied GLasso with nonconvex penalty such as SCAD [5]. However, their estimation error rate is the same as GLasso, *i.e.*, $O(s\sqrt{\log d/n})$. In addition, Fan et al. [7] proved the oracle property of GLasso with nonconvex penalty. Nevertheless, they failed to improve the convergence rate either.

Finally, we compare our results with CLIME [2] and graph Dantzig selector [29]. These two estimators also attain $O(s\sqrt{\log d/n})$ rate of convergence in spectral norm. Yuan [29], Cai et al. [3] prove that the minimax optimal rate for estimating the sparse precision matrix is $O(s\sqrt{\log d/n})$. The reason why our estimator can have a faster rate than this minimax rate is that our theory considers the magnitude information of the precision matrix, while the minimax lower bound in Yuan [29], Cai et al. [3] does not take into account the entry magnitude information.

5 NUMERICAL EXPERIMENTS

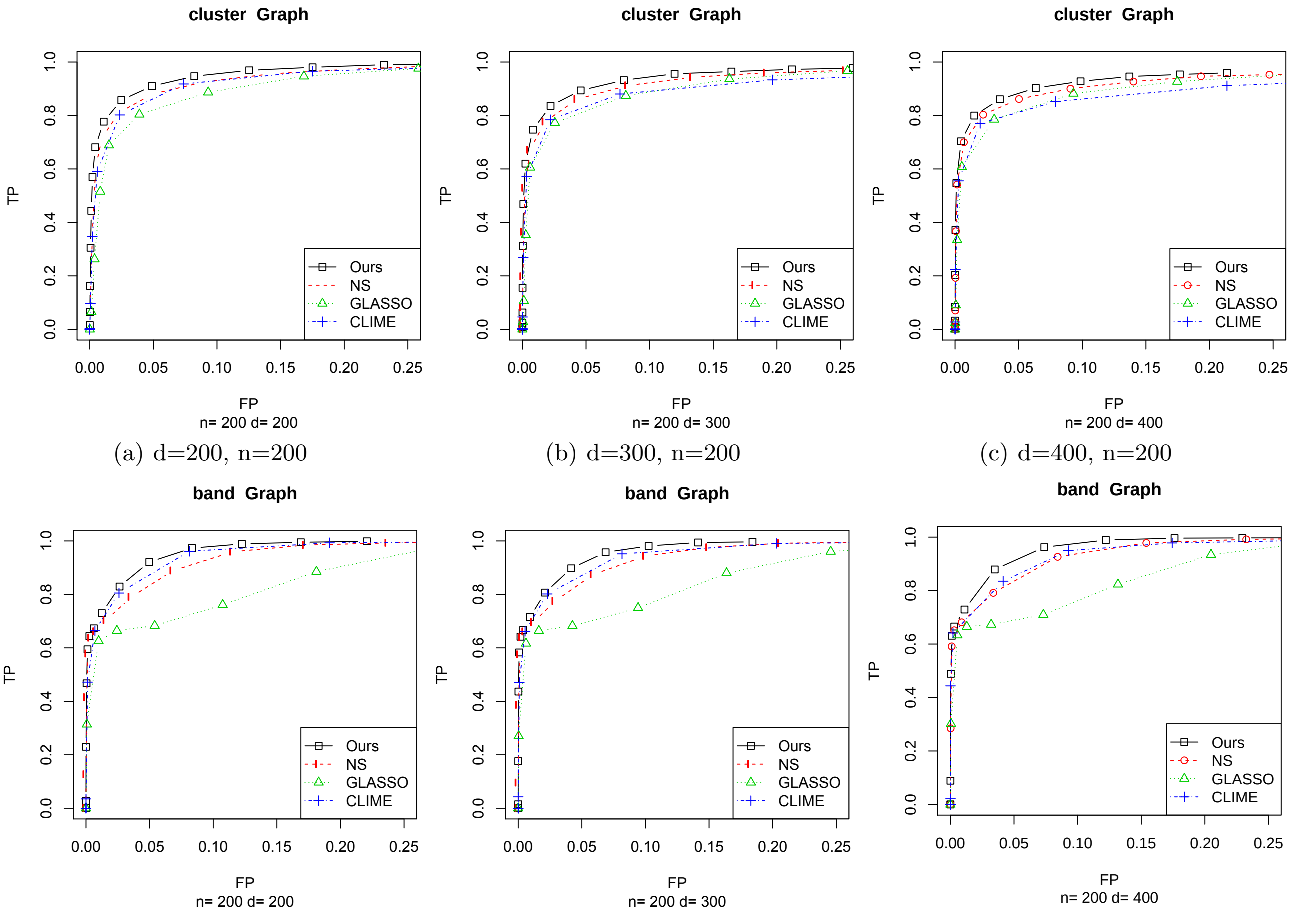
In this section, we use synthetic data to compare our proposed method with others (NS [20], GLasso [30], and CLIME [2]). Our comparisons focus on their performance in graph recovery and parameter estimation. The implementation of the baseline algorithms are from **R** package **huge**[1].

5.1 Numerical Simulation

For the purpose of our comparisons, we consider 3 different settings: (i) $d = 200$, $n = 200$; (ii) $d = 300$, $n = 200$; (iii) $d = 400$, $n = 200$. We use 4 graph models: cluster, band, scale-free, random to generate our graphs and precision matrices. More specifically, the covariance matrices are generated by **huge** package, and the magnitude of correlations is the default value in the huge generator.

First, we use receiver operating characteristic (ROC) curves to compare the overall performance of our proposed method with other methods in model selection over the full paths. The ROC curves for cluster and band graphs are shown in Figure 1 and ROC curves

[1] http://cran.r-project.org/web/packages/huge



(a) d=200, n=200

(b) d=300, n=200

(c) d=400, n=200

(d) d=200, n=200

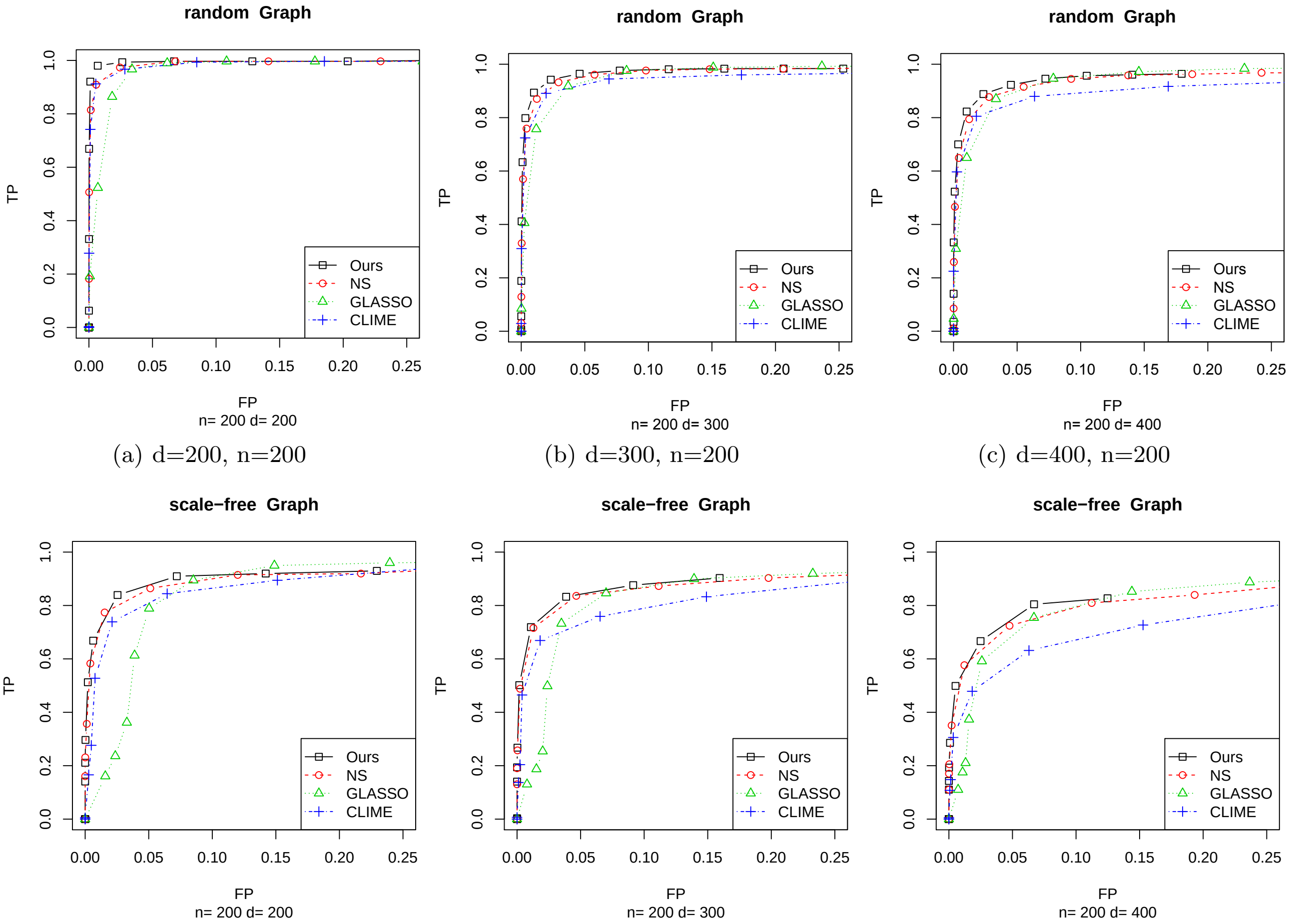
(e) d=300, n=200

(f) d=400, n=200

Figure 1: ROC curves of different methods on cluster and band models. Top (a-c): ROC curves for cluster models. Bottom (d-f): ROC curves for band models

Table 1: Quantitative comparisons of Our estimator, NS, GLasso and CLIME on the cluster, scale-free, random, and band models in terms of spectral norm $\|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}\|_2$.

Model	d	GLasso	CLIME	NS	Ours
Cluster	200	7.459(0.025)	6.672(0.355)	6.425(0.163)	3.738(0.194)
	300	8.269(0.080)	7.932(0.264)	7.347(0.132)	4.660(0.132)
	400	7.970(0.055)	7.307(0.174)	6.952(0.130)	4.049(0.291)
Scale-free	200	12.649(0.058)	12.036(0.251)	13.014(0.117)	8.405(0.694)
	300	12.522(0.128)	12.353(0.314)	12.827(0.104)	9.015(0.490)
	400	13.680(0.110)	13.184(0.136)	13.882(0.122)	9.847(0.295)
Random	200	5.696(0.031)	5.588(0.179)	5.040(0.101)	2.887(0.104)
	300	5.441(0.011)	4.932(0.202)	4.640(0.140)	1.843(0.244)
	400	6.760(0.071)	6.179(0.158)	6.100(0.375)	2.447(0.140)
Band	200	6.716(0.011)	6.487(0.031)	5.492(0.233)	4.976(0.091)
	300	6.757(0.015)	6.462(0.007)	5.235(0.018)	4.861(0.141)
	400	6.794(0.026)	6.593(0.081)	5.246(0.059)	4.929(0.076)



(a) d=200, n=200 (b) d=300, n=200 (c) d=400, n=200

(d) d=200, n=200 (e) d=300, n=200 (f) d=400, n=200

Figure 2: ROC curves of different methods on random and scale-free models. Top (a-c): ROC curves for random models. Bottom (d-f): ROC curves for scale-free models

Table 2: Quantitative comparisons of the Our estimator, NS, GLasso and CLIME on the cluster, scale-free, random, and band models in terms of Frobenius norm $\|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_F$.

Model	d	GLasso	CLIME	NS	Ours
Cluster	200	18.924(0.147)	18.199(0.766)	15.612(0.211)	11.264(0.130)
	300	22.624(0.205)	22.479(0.076)	18.256(0.170)	13.611(0.363)
	400	22.761(0.130)	22.592(0.252)	17.758(0.183)	13.805(0.452)
Scale-free	200	14.492(0.061)	14.292(0.214)	13.995(0.145)	9.379(0.695)
	300	16.872(0.134)	16.398(0.281)	16.334(0.107)	11.426(0.455)
	400	16.639(0.122)	15.967(0.177)	15.445(0.136)	11.518(0.242)
Random	200	18.852(0.162)	19.435(0.888)	15.141(0.240)	10.254(0.559)
	300	18.280(0.174)	18.342(0.067)	13.404(0.099)	7.777(0.482)
	400	20.129(0.177)	20.049(0.208)	15.341(0.485)	8.802(0.134)
Band	200	36.577(0.084)	34.805(0.157)	28.991(1.237)	25.996(0.392)
	300	45.477(0.089)	42.455(0.159)	33.920(0.086)	31.199(0.140)
	400	52.962(0.170)	49.914(0.144)	39.351(0.201)	36.741(0.157)

for scale-free and random graphs are presented in Figure 2. From ROC curves, we can see that our method outperforms others in all settings. These results indicate that our method is very competitive in the graph recovery problem. Then, we evaluate performance of our proposed method and other methods in parameter estimation by following procedures. For each model settings mentioned above, we generate data with sample size $n = 200$ as training set, and an independent data with the same size from the same distribution as test set. We choose the tuning parameter λ by grid search based on its performance on training set, and evaluate those estimators on the test set. Here we use spectral norm error $\|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}\|_2$ and Frobenius norm error $\|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}\|_F$ to compare the performance of different methods in parameter estimation. Table 1 and 2 summarizes our results averaged over 50 simulations. We report the spectral norm error in Table 1 and the Frobenius norm error in Table 2. We can see that, the performance of our method in parameter estimation is significantly better than others in all settings.

5.2 Real Data Experiments

In this subsection, we use the breast cancer data[2] , which is analyzed by Hess et al. [13] and later on by Fan et al. [6], Cai et al. [2], Zhao and Liu [32], to illustrate the advantage of our method. This real data set includes 133 subjects with $22,283$ gene expression levels. 34 subjects have obtained pathological complete response (pCR), and others have obtained residual disease (RD). Some studies have pointed out that in the long term, the chance of cancer-free survival of the pCR subjects is higher than the RD subjects. Thus studying the response states of patients (with RD or pCR) to neoadjuvant (preoperative) chemotherapy is of great value. The data are randomly divided into the training set of 107 subjects and the testing set of 26 subjects. The testing set is consists of 6 pCR subjects and 20 RD subjects. By using **R** package **limma**[3] on training set, we select 110 most significant genes, following Fan et al. [6], as predictors. We assume the scaled gene expression data to be normally distributed as $N(\boldsymbol{\mu}_k, \mathbf{\Sigma})$, which means different group has different mean but same covariance (LDA framework). Our discriminant function is

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \widehat{\mathbf{\Theta}} \widehat{\boldsymbol{\mu}}_k - \frac{1}{2} \widehat{\boldsymbol{\mu}}_k^\top \widehat{\mathbf{\Theta}} \widehat{\boldsymbol{\mu}}_k + \log \widehat{\pi}_k,$$

where $\widehat{\pi}_k = n_k/n$. Note that the estimated precision matrices obtained by different methods are applied here, and we classify a sample to class k if the value of $\delta_k(\mathbf{x})$ is largest. Thus the performance of estimated $\mathbf{\Theta}$ corresponds to the performance of classification. And we use test set to evaluate the performance of different $\widehat{\mathbf{\Theta}}$. For the choosing of tuning parameter λ, we adopt 5-fold cross-validation on training set. We use specificity, sensitivity, and Mathews correlation coefficient (MCC) criteria, as defined below, to evaluate the performance of classification.

Table 3: Quantitative comparison of different estimator in the breast cancer data analysis

Method	Specificity	Sensitivity	MCC
Glasso	0.857(0.045)	0.424(0.099)	0.308(0.151)
CLIME	0.862(0.038)	0.453(0.097)	0.340(0.123)
NS	0.889(0.044)	0.442(0.093)	0.372(0.135)
Ours	0.897(0.042)	0.493(0.105)	0.428(0.143)

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}}, \quad \text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}},$$

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP+FP})(\text{TP+FN})(\text{TN+FP})(\text{TN+FN})}},$$

where

$$\text{TP} = \sum_i \mathbb{1}(\widehat{y}_i = 1, y_i^* = 1), \ \text{FP} = \sum_i \mathbb{1}(\widehat{y}_i = 1, y_i^* = 0),$$

$$\text{TN} = \sum_i \mathbb{1}(\widehat{y}_i = 0, y_i^* = 0), \ \text{FN} = \sum_i \mathbb{1}(\widehat{y}_i = 0, y_i^* = 1),$$

where $\widehat{y}_i$'s and y_i^*'s denote true labels and predicted labels repectively. Results over 100 simulations are presented in the Table 3. It can be seen that our proposed estimator outperforms others in all three criteria, especially for MCC. These results further corroborate the superiority of our estimator.

6 Conclusions

In this paper, for the precision matrix estimation in high dimensional Gaussian graphical models, we propose a new estimator based on neighborhood selection method with nonconvex penalty. Theoretically, our estimator not only achieves sharper convergence rates, but also enjoys the oracle property under mild conditions. The results of simulations and real data experiments also demonstrate the outstanding performance of our estimator.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. Research was sponsored by Quanquan Gu's startup funding at Department of Systems and Information Engineering, University of Virginia.

[2] Available on http://bioinformatics.mdanderson.org/

[3] Available on http://bioconductor.org/packages/limma

References

[1] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation. *Journal of Machine Learning Research*, 9(3): 485–516, 2008.

[2] T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106 (494):594–607, 2011.

[3] T Tony Cai, Weidong Liu, and Harrison H Zhou. Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *arXiv preprint arXiv:1212.2882*, 2012.

[4] David Roxbee Cox and Nanny Wermuth. *Multivariate dependencies: Models, analysis and interpretation*, volume 67. CRC Press, 1996.

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

[6] Jianqing Fan, Yang Feng, and Yichao Wu. Network exploration via the adaptive lasso and scad penalties. *The annals of applied statistics*, 3(2):521, 2009.

[7] Jianqing Fan, Lingzhou Xue, and Hui Zou. Strong oracle optimality of folded concave penalized estimation. *Annals of statistics*, 42(3):819, 2014.

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

[9] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0-8018-5414-8.

[10] Quanquan Gu, Zhaoran Wang, and Han Liu. Sparse pca with oracle property. In *Advances in neural information processing systems*, pages 1529–1537, 2014.

[11] Quanquan Gu, Yuan Cao, Yang Ning, and Han Liu. Local and global inference for high dimensional gaussian copula graphical models. *arXiv preprint arXiv:1502.02347*, 2015.

[12] Huan Gui and Quanquan Gu. Towards faster rates and oracle property for low-rank matrix estimation. *arXiv preprint arXiv:1505.04780*, 2015.

[13] Kenneth R Hess, Keith Anderson, W Fraser Symmans, Vicente Valero, Nuhad Ibrahim, Jaime A Mejia, Daniel Booser, Richard L Theriault, Aman U Buzdar, Peter J Dempsey, et al. Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of clinical oncology*, 24(26): 4236–4244, 2006.

[14] Cho-Jui Hsieh, Inderjit S Dhillon, Pradeep K Ravikumar, and Mátyás A Sustik. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems*, pages 2330–2338, 2011.

[15] Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S Dhillon, Pradeep K Ravikumar, and Russell Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. In *Advances in Neural Information Processing Systems*, pages 3165–3173, 2013.

[16] Clifford Lam and Jianqing Fan. Sparsistency and rates of convergence in large covariance matrix estimation. *Annals of statistics*, 37(6B):4254, 2009.

[17] Han Liu and Lie Wang. Tiger: A tuning-insensitive approach for optimally estimating gaussian graphical models. *arXiv preprint arXiv:1209.2437*, 2012.

[18] Po-Ling Loh and Martin J Wainwright. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*, pages 476–484, 2013.

[19] Po-Ling Loh and Martin J Wainwright. Support recovery without incoherence: A case for nonconvex regularization. *arXiv preprint arXiv:1412.5632*, 2014.

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

[21] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Restricted eigenvalue properties for correlated gaussian designs. *The Journal of Machine Learning Research*, 11:2241–2259, 2010.

[22] P. Ravikumar, M. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

[23] Zhao Ren, Tingni Sun, Cun-Hui Zhang, Harrison H Zhou, et al. Asymptotic normality and optimalities in estimation of large gaussian graphical models. *The Annals of Statistics*, 43(3):991–1026, 2015.

[24] A. Rothman, P. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

[25] Tingni Sun and Cun-Hui Zhang. Scaled sparse linear regression. *Biometrika*, page ass043, 2012.

[26] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[27] Huahua Wang, Arindam Banerjee, Cho-Jui Hsieh, Pradeep K Ravikumar, and Inderjit S Dhillon. Large scale distributed sparse precision estimation. In *Advances in Neural Information Processing Systems*, pages 584–592, 2013.

[28] Zhaoran Wang, Han Liu, and Tong Zhang. Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *Annals of statistics*, 42(6):2164, 2014.

[29] M. Yuan. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11(8):2261–2286, 2010.

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

[31] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, pages 894–942, 2010.

[32] Tuo Zhao and Han Liu. Sparse inverse covariance estimation with calibration. In *Advances in Neural Information Processing Systems*, pages 2274–2282, 2013.

[33] Rongda Zhu and Quanquan Gu. Towards a lower sample complexity for robust one-bit compressed sensing. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 739–747, 2015.