A Completely Tuning-Free and Robust Approach to Sparse Precision Matrix Estimation

Chau Tran [1] **Guo Yu** [1]

Abstract

Despite the vast literature on sparse Gaussian graphical models, current methods either are asymptotically tuning-free (which still require fine-tuning in practice) or hinge on computationally expensive methods (e.g., cross-validation) to determine the proper level of regularization. We propose a *completely tuning-free* approach for estimating sparse Gaussian graphical models. Our method uses model-agnostic regularization parameters to estimate each column of the target precision matrix and enjoys several desirable properties. Computationally, our estimator can be computed efficiently by linear programming. Theoretically, the proposed estimator achieves minimax optimal convergence rates under various norms. We further propose a second-stage enhancement with non-convex penalties which possesses the strong oracle property. Through comprehensive numerical studies, our methods demonstrate favorable statistical performance. Remarkably, our methods exhibit strong robustness to the violation of the Gaussian assumption and significantly outperform competing methods in the heavy-tailed settings.

1. Introduction

Undirected graphical models are ubiquitous in the general field of machine learning. Learning the edge of an undirected graph G with nodes $X_1, \ldots, X_d$ is equivalent to estimating the dependence structure among these d random variables. Specifically, if (j, k) is an edge in the graph G, then X_j and X_k are dependent conditioned on the rest of variables. In Gaussian graphical models, where $X = (X_1, \ldots, X_d) \sim N_d(\mathbf{0}, \mathbf{\Sigma})$, the conditional dependence structure is encoded in the sparsity pattern of the precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$: $\mathbf{\Omega}_{jk} = 0$ when (j, k) is not an edge in G (Lauritzen, 1996). Also known as the covariance selection problem (Dempster, 1972), we are primarily interested in estimating $\mathbf{\Omega}$ using n observations of the d-dimensional random vector X.

It is well-known that this problem becomes challenging in the high dimensional settings where $d > n$. Regularization becomes a common strategy for feasible estimation of high-dimensional Gaussian graphical models. Inducing sparsity is an especially favorable choice of regularization since non-zero entries in $\mathbf{\Omega}$ correspond to the edges in G.

In literature, there are essentially two types of sparsity-inducing estimators in Gaussian graphical models. One type of methods is based on penalized likelihood estimation, which includes the graphical lasso (GLasso) (Yuan & Lin, 2007; Friedman et al., 2007; Banerjee et al., 2008) as an example. Likelihood-based methods are usually less favorable in terms of theoretical properties (Rothman et al., 2008; Ravikumar et al., 2011; Mazumder & Hastie, 2012; Yu & Bien, 2017). An alternative type of approach that is more amenable to theoretical analysis estimates $\mathbf{\Omega}$ in a column-by-column fashion, where each column is estimated by a regularized linear regression problem. For example, to estimate each column of $\mathbf{\Omega}$, Meinshausen & Bühlmann (2006) use Lasso (Tibshirani, 1996), Yuan (2010) use the Dantzig selector (Candes & Tao, 2007), Sun & Zhang (2013) use the scaled Lasso (Sun & Zhang, 2012), Liu & Wang (2017) use the SQRT-Lasso (Belloni et al., 2011).

The optimal performance of these methods typically depends on choosing the proper value of regularization parameter, which usually relies on unknown population quantities. In practice, determining the level of regularization involves computationally intensive procedures, such as cross-validation. The only exceptions, as far as we know, are the strongly related TIGER (Liu & Wang, 2017) and the scaled Lasso (Sun & Zhang, 2013). Although both methods greatly simplify the tuning procedure, the claimed tuning-free property only holds **asymptotically**. The computational caveats of these methods include (1) enforcing the same tuning parameter value to be used for estimating all columns of $\mathbf{\Omega}$, and (2) the common tuning parameter value includes a con-

[1]Department of Statistics and Applied Probability, University of California, Santa Barbara, CA, USA. Correspondence to: Chau Tran <chautran@ucsb.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

stant that still requires fine-tuning. These limitations call for the development of a completely tuning-free estimator whose level of regularization can be determined without any tuning and is fully adaptive to each column problem of $\mathbf{\Omega}$.

Furthermore, in many applications such as finance, neuroscience, and genetics, the underlying data generating distribution is often heavy-tailed. Methods that assume normality conditions usually lead to less satisfying performance (Finegold & Drton, 2011; de Miranda Cardoso et al., 2021).

Contributions: In this paper, we propose a completely tuning-free method in high-dimensional Gaussian graphical models. Our estimator possesses the *completely pivotal* property, so the regularization parameter for each column problem does not depend on any unknown parameters and can easily be computed. Theoretically, our method achieves the minimax optimal rate of convergence for a well-studied matrix class under different norms. We further propose a second-stage enhancement using non-convex penalties which enjoys the oracle property. Through comprehensive numerical studies, we demonstrate the favorable performance of the proposed methods and illustrate their robustness to the violation of the Gaussian assumption.

Notation: For the rest of the paper, we let operator $|\cdot|$ denote absolute value for a scalar and cardinal number of a set. For a vector $\boldsymbol{\alpha} \in \mathbb{R}^d$, $\boldsymbol{\alpha}_i$ denotes its i-th element. We define the ℓ_p norm of a vector as $\|\boldsymbol{\alpha}\|_p = (\sum_{i=1}^d |\boldsymbol{\alpha}_i|^p)^{1/p}$ for $0 < p < \infty$, and $\|\boldsymbol{\alpha}\|_\infty = \max_i |\boldsymbol{\alpha}_i|$. For a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{A}_{jk}$ denotes its (j,k) entry, $\mathbf{A}_{*,j}$ denotes the j-th column of $\mathbf{A}$, and $\mathbf{A}_{*,-j}$ denotes the submatrix of $\mathbf{A}$ with j-th column removed. We denote the matrix ℓ_p norm as $\|\mathbf{A}\|_p = \max_{\|\mathbf{v}\|_p=1} \|\mathbf{A}\mathbf{v}\|_p$, and matrix Frobenius norm as $\|\mathbf{A}\|_F = (\sum_{j,k} |\mathbf{A}_{j,k}|^2)^{1/2}$. Finally, $\mathbf{A} \succ 0$ denotes that the matrix $\mathbf{A}$ is positive definite.

2. Method

2.1. Column-by-column estimation of Gaussian graphical models

Consider a d-dimensional multivariate Gaussian random vector $X = (X_1, \ldots, X_d) \sim N(\mathbf{0}, \mathbf{\Sigma})$. Given a data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, where each row of $\mathbf{X}$ is assumed to be independent and following the same distribution as X, we are interested in estimating the precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$.

It is well known that for each $j = 1, \ldots, d$, the joint normality implies the following conditional distribution $X_j | X_{-j} \sim N_{d-1}(\mathbf{\Sigma}_{j,-j}[\mathbf{\Sigma}_{-j,-j}]^{-1}X_{-j}, \mathbf{\Sigma}_{j,j} - \mathbf{\Sigma}_{j,-j}[\mathbf{\Sigma}_{-j,-j}]^{-1}\mathbf{\Sigma}_{-j,j})$, which is equivalent to the following linear model (by implicitly conditioning on X_{-j}):

$$X_j = X_{-j}^T \boldsymbol{\beta}^{(j)} + \epsilon_j, \tag{1}$$

where $\boldsymbol{\beta}^{(j)} = [\mathbf{\Sigma}_{-j,-j}]^{-1}\mathbf{\Sigma}_{-j,j}$ and $\epsilon_j \sim N\left(0, \sigma_j^2\right)$ with $\sigma_j^2 = \mathbf{\Sigma}_{j,j} - \mathbf{\Sigma}_{j,-j}[\mathbf{\Sigma}_{-j,-j}]^{-1}\mathbf{\Sigma}_{-j,j}$. By the block matrix inversion formula, we have

$$\mathbf{\Omega}_{j,j} = \sigma_j^{-2}, \qquad \text{and} \qquad \mathbf{\Omega}_{-j,j} = -\sigma_j^{-2}\boldsymbol{\beta}^{(j)}. \tag{2}$$

It suggests that an estimate of the j-th column of $\mathbf{\Omega}$ can be obtained by estimating the regression coefficients $\boldsymbol{\beta}^{(j)}$ and the error variance σ_j^2 of the linear model in (1). Thus, the problem of estimating $\mathbf{\Omega}$ can be formulated as a series of d regression problems, each of which estimates one column of $\mathbf{\Omega}$.

Note from (2) that the sparsity pattern in an estimate of $\boldsymbol{\beta}^{(j)}$ is equivalent to the sparsity pattern of the estimated j-th column of $\mathbf{\Omega}$ under joint normality. This observation drives many recently proposed methods, most of which are built upon various regularized regression techniques. However, these methods either require computationally intensive procedures (e.g., cross-validation) to carefully choose the proper level of regularization, which depends on certain unknown population parameters, or are tuning-free asymptotically and still require tuning in finite samples.

2.2. Our proposed method: gRankLasso

In this paper, we propose the graphical Rank Lasso estimator (gRankLasso) of Gaussian graphical models, where each column of $\mathbf{\Omega}$ is estimated using Rank Lasso (Wang et al., 2020). Specifically, to estimate the j-th column of $\mathbf{\Omega}$ using the data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, we use the following rank loss function

$$Q_j(\boldsymbol{\beta}) = [n(n-1)]^{-1} \sum_{k=1}^{n} \sum_{m\neq k} \times |(\mathbf{X}_{kj} - \mathbf{X}_{mj}) - (\mathbf{X}_{k,-j} - \mathbf{X}_{m,-j})\boldsymbol{\beta}|, \tag{3}$$

which is the summation of absolute pairwise difference (among the n observations) of the linear model predictions when X_j is regressed on all other variables X_{-j}. In nonparametric regression, this loss is equivalent to, up to a constant, the Jaeckel's dispersion function with Wilcoxon scores (Jaeckel, 1972; Hettmansperger & McKean, 2010). Then the estimate of the j-th column of $\mathbf{\Omega}$ can be obtained by

$$\begin{aligned} \hat{\boldsymbol{\beta}}^{(j)} &= \underset{\boldsymbol{\beta}\in\mathbb{R}^{d-1}}{\operatorname{argmin}}\{Q_j(\boldsymbol{\beta}) + \lambda_j\|\boldsymbol{\beta}\|_1\}, \\ \hat{\sigma}_j^2 &= n^{-1}\|\mathbf{X}_{*,j} - \mathbf{X}_{*,-j}\hat{\boldsymbol{\beta}}^{(j)}\|_2^2, \\ \hat{\mathbf{\Omega}}_{jj} &= 1/\hat{\sigma}_j^2, \qquad \hat{\mathbf{\Omega}}_{-j,j} = -\hat{\mathbf{\Omega}}_{jj}\hat{\boldsymbol{\beta}}^{(j)}. \end{aligned} \tag{4}$$

Wang et al. (2020) show that the subgradient of $Q_j(\boldsymbol{\beta})$ evaluated at the true $\boldsymbol{\beta}^{(j)}$ is

$$\mathbf{L}_j = \left.\frac{\partial Q_j(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(j)}}$$

$$= -2[n(n-1)]^{-1}\sum_{k=1}^{n}\mathbf{X}_{k,-j}^{T}(\sum_{m\neq k}\text{sign}(\boldsymbol{\epsilon}_k^{(j)}-\boldsymbol{\epsilon}_m^{(j)})),$$

where $\text{sign}(a)$ is the sign of a scalar a. It follows that $\sum_{m=1,m\neq k}^{n}\text{sign}(\boldsymbol{\epsilon}_k^{(j)}-\boldsymbol{\epsilon}_m^{(j)}) = 2\cdot\text{rank}(\boldsymbol{\epsilon}_k^{(j)})-(n+1)$, where $\text{rank}(\boldsymbol{\epsilon}_k^{(j)})$ denotes the rank of $\boldsymbol{\epsilon}_k^{(j)}$ among $\{\boldsymbol{\epsilon}_1^{(j)},\ldots,\boldsymbol{\epsilon}_n^{(j)}\}$, which are i.i.d. Thus $\{\text{rank}(\boldsymbol{\epsilon}_1^{(j)}),\ldots,\text{rank}(\boldsymbol{\epsilon}_n^{(j)})\}$ follows a uniform distribution on the random permutations of integers $\{1,2,\ldots,n\}$. As suggested in Wang et al. (2020), for $c>1$ and $\alpha\in[0,1]$, we choose the regularization parameter λ_j in (4) so that

$$\lambda_j = c\cdot Q^{-1}_{\|\mathbf{L}_j\|_\infty}(1-\alpha),$$

where $Q^{-1}_{\|\mathbf{L}_j\|_\infty}(1-\alpha)$ denotes the $(1-\alpha)-$quantile of the distribution of $\|\mathbf{L}_j\|_\infty$. This choice of λ_j can easily be simulated using Algorithm 1. Furthermore, this λ_j satisfies the subgradient condition (Bickel et al., 2009; Bühlmann & Van De Geer, 2011) of the Rank Lasso objective function in (4) for each column problem with high probability (Wang et al., 2020). The values of α and c are theoretical necessities, and $\alpha=0.1$ and $c=1.01$ work well in practice. Moreover, the optimization problem in (4) can be formulated as a linear programming (LP), which can be solved efficiently using any standard LP solver.

Algorithm 1 Simulate λ_j, for $j=1,\ldots,d$

Input: $\mathbf{X}\in\mathbb{R}^{n\times d}$, $\alpha\in[0,1]$, $c>1$, $B\in\mathbb{N}$
for k in $1:B$ **do**
  for j in $1:d$ **do**
    $\boldsymbol{r}\leftarrow$ a random permutation of the integers $1:n$
    $\boldsymbol{\phi}\leftarrow 2\cdot\boldsymbol{r}-(n+1)$
    $L_j[k]\leftarrow c\cdot\|2[n(n-1)]^{-1}(\mathbf{X}_{*,-j})^T\boldsymbol{\phi}\|_\infty$
  end for
end for
Output: $\lambda_j\leftarrow\text{Quantile}(L_j,1-\alpha)$, for $j=1,\ldots,d$

Although recently there have been numerous tuning-free methods in high-dimensional linear models (Wang, 2013; Lederer & Müller, 2015; Chichignoud et al., 2016; Belloni et al., 2017; Yu & Bien, 2019), we argue that the Rank Lasso is an especially attractive candidate for column-by-column estimation of $\boldsymbol{\Omega}$. First of all, the Rank Lasso enjoys the completely pivotal property, which means that the theoretically optimal regularization parameter does not depend on any unknown model parameters and adjusts to both the distribution of random errors and the structure of the design matrix. Hence, we allow regularization parameters λ_j to be different for estimating different columns. Furthermore, the regularization parameters λ_j can be easily simulated from data, which is extremely efficient to compute without any fine tuning. Thirdly, among other regression methods that share similar properties, the Rank Lasso is significantly more efficient in Gaussian settings (Wang et al., 2020). Finally, it was also noted that the Rank Lasso estimator is robust to heavy-tailed errors. This auxiliary property makes it appealing for many data applications where the stringent multivariate Gaussian assumption is not guaranteed.

2.3. A second-stage improvement

The ℓ_1 penalty used in (4), while being computationally friendly, is known to induce estimation bias (Tibshirani, 2011). Therefore, many non-convex penalties (Fan & Li, 2001; Zhang, 2010, among others) have been proposed to circumvent this issue. We present a second-stage improvement with non-convex penalties (Wang et al., 2020) using gRankLasso as an initial estimator. Specifically, for $1\le j\le d$,

$$\tilde{\boldsymbol{\beta}}^{(j)} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{d-1}}{\text{argmin}}\{Q_j(\boldsymbol{\beta})+\sum_{i=1}^{d}p'_\eta(|\hat{\boldsymbol{\beta}}_i^{(j)}|)|\boldsymbol{\beta}_i|\}, \quad (5)$$
$$\tilde{\sigma}_j^2 = n^{-1}\|\mathbf{X}_{*,j}-\mathbf{X}_{*,-j}\tilde{\boldsymbol{\beta}}^{(j)}\|_2^2,$$
$$\tilde{\boldsymbol{\Omega}}_{jj} = 1/\tilde{\sigma}_j^2, \qquad \tilde{\boldsymbol{\Omega}}_{-j,j} = -\tilde{\boldsymbol{\Omega}}_{jj}\tilde{\boldsymbol{\beta}}^{(j)},$$

where $\hat{\boldsymbol{\beta}}^{(j)}$ is obtained in (4) and $p'_\eta(\cdot)$ denotes the derivative of a non-convex penalty function $p_\eta(\cdot)$ with a tuning parameter $\eta>0$. The second-stage improvement applies to a general class of non-convex penalties, which will be described in Section 3. The optimization problem in (5) can also be solved efficiently using LP standard solver.

Note that a tuning parameter η is required in the general non-convex penalty in (5), which needs light tuning. At a higher computational cost, we show in Section 3 that with a proper choice of η, the second-stage enhancement achieves stronger theoretical guarantees than gRankLasso. Particularly, the second-stage enhancement enjoys the oracle property, meaning that it performs as if one knows the support of the true $\boldsymbol{\Omega}$.

Practically, to ensure that the estimate of $\boldsymbol{\Omega}$ is symmetric, we set $\hat{\boldsymbol{\Omega}}_{ij}^{\text{sym}}=\hat{\boldsymbol{\Omega}}_{ji}^{\text{sym}}=\min\{\hat{\boldsymbol{\Omega}}_{ij},\hat{\boldsymbol{\Omega}}_{ji}\}$ and $\tilde{\boldsymbol{\Omega}}_{ij}^{\text{sym}}=\tilde{\boldsymbol{\Omega}}_{ji}^{\text{sym}}=\min\{\tilde{\boldsymbol{\Omega}}_{ij},\tilde{\boldsymbol{\Omega}}_{ji}\}$ for $i\neq j$. This additional symmetrization step does not affect the theoretical analysis as shown in Cai et al. (2011).

3. Theoretical analysis

In this section, we study the theoretical properties of the proposed estimators. Let $S_j=\{i:i\neq j,\boldsymbol{\Omega}_{ij}\neq 0\}$ be the support of the off-diagonal part of the j-th column of $\boldsymbol{\Omega}$. We define the matrix class $\mathcal{M}(s,M_d)=\{\boldsymbol{\Omega}=\boldsymbol{\Omega}^T\in\mathbb{R}^{d\times d}:\boldsymbol{\Omega}\succ 0,\xi^{-1}\le\Lambda_{\min}(\boldsymbol{\Omega})\le\Lambda_{\max}(\boldsymbol{\Omega})\le\xi,\max_{1\le j\le d}|S_j|\le s,\|\boldsymbol{\Omega}\|_1\le M_d\}$, where ξ is a positive constant, $\Lambda_{\min}(\boldsymbol{\Omega})$ and $\Lambda_{\max}(\boldsymbol{\Omega})$ are minimum and maximum eigenvalues of $\boldsymbol{\Omega}$, and M_d may scale with d. We

assume the following conditions:
(C1) $\mathbf{\Omega} \in \mathcal{M}(s, M_d)$,
(C2) $s^2 \log d = o(n)$.
Condition (C1) requires that the true precision matrix has a bounded minimum and maximum eigenvalues, and is sparse column-wise. Condition (C2) allows the maximum degree of the graph encoded by true $\mathbf{\Omega}$ to grow with d in a certain rate.

3.1. Main theorems

Theorem 3.1. *(Matrix ℓ_1 norm and spectral norm rates) With the adaptive choice of $\lambda_j, j = 1, \ldots, d$ from Algorithm 1, under assumptions (C1) and (C2), we have*

$$\|\hat{\mathbf{\Omega}} - \mathbf{\Omega}\|_2 \le \|\hat{\mathbf{\Omega}} - \mathbf{\Omega}\|_1 \le C s M_d \sqrt{\frac{\log d}{n}}$$

with probability at least $1 - O(1/d)$, where C is an absolute constant.

Theorem 3.1 shows the convergence rate of matrix estimation under the matrix ℓ_1 and spectral norms. This is the minimax optimal rate of convergence for the matrix class $\mathcal{M}(s, M_d)$ (Theorem 4 in Yuan, 2010).

Corollary 3.2. *(Frobenius norm rate) With the adaptive choice of λ_j for $j = 1, \ldots, d$ from Algorithm 1, under assumptions (C1) and (C2), we have*

$$\|\hat{\mathbf{\Omega}} - \mathbf{\Omega}\|_F \le C s M_d \sqrt{\frac{d \log d}{n}}$$

with probability at least $1 - O(1/d)$, where C is an absolute constant.

The convergence rate under the Frobenius norm is worse than the minimax optimal rate by a factor of $\sqrt{s}$ (Rothman et al., 2008; Cai et al., 2016; Liu & Wang, 2017). Whether gRankLasso could achieve the minimax optimal rate under the elementwise max-norm as well as the Frobenius norm is an interesting future research (Cai et al., 2011; Kelner et al., 2020).

Next, we show the strong oracle property and the faster convergence rate using the second-stage enhancement. We assume the following conditions on the general non-convex penalty functions:

1. $p_\eta(t)$ is increasing and concave for $t \in [0, +\infty)$, and has a continuous derivative $p'_\eta(t)$ on $(0, +\infty)$.
2. $p_\eta(t)$ has a singularity at the origin, *i.e.* $p'_\eta(0+) > 0$, which can be standardized so that $p'_\eta(0+) = \eta$.
3. There exist constants $a_1 > 0$ and $a_2 > 1$ such that $p'_\eta(t) \ge a_1 \eta$ for all $0 < t < a_2 \eta$; and $p'_\eta(t) = 0$ for all $t > a_2 \eta$.

These general conditions hold for many non-convex penalty functions, including the two popular choices SCAD (Fan & Li, 2001) and MCP (Zhang, 2010). We show that the second-stage improvement performs as if one knows the sparsity pattern of the true $\mathbf{\Omega}$. Specifically, let $\check{\mathbf{\Omega}}$ be the oracle estimator of $\mathbf{\Omega}$ defined as follows: For $i \le j \le d$

$$\begin{aligned}
&\check{\boldsymbol{\beta}}^{(j)} = \underset{\text{supp}(\boldsymbol{\beta}) \subset S_j}{\text{argmin}} \; Q_j(\boldsymbol{\beta}), \\
&\check{\sigma}_j^2 = n^{-1} \|\mathbf{X}_{*,j} - \mathbf{X}_{*,-j} \check{\boldsymbol{\beta}}^{(j)}\|_2^2, \\
&\check{\mathbf{\Omega}}_{jj} = 1/\check{\sigma}_j^2, \qquad \check{\mathbf{\Omega}}_{-j,j} = -\check{\mathbf{\Omega}}_{jj} \check{\boldsymbol{\beta}}^{(j)}.
\end{aligned}$$

That is, $\check{\boldsymbol{\beta}}^{(j)}$ is the minimizer of the rank loss function $Q_j(\boldsymbol{\beta})$ in (3) when the support of the j-th column of $\mathbf{\Omega}$ is known. Using a non-convex penalty such as SCAD or MCP, we can show the oracle property of our second-stage estimator.

Theorem 3.3. *Let $\tilde{\mathbf{\Omega}}$ be the second-stage estimator of $\mathbf{\Omega}$ using gRankLasso $\hat{\mathbf{\Omega}}$ as an initial estimator. Suppose the conditions in Theorem 3.1 are satisfied and the non-convex penalty function satisfies the general conditions above. Furthermore, suppose $s = O(n^{a_1})$, $\eta = O(n^{-(1-a_2)/2})$, $\log d = n^{a_3}$, and nonzero entries of the true $\mathbf{\Omega}$ satisfies*

$$\min_{i \ne j} |\mathbf{\Omega}_{ij}| \ge b n^{-(1-a_4)/2} \tag{6}$$

where a_1, a_2, a_3, a_4, b are positive constants such that $2a_1 < a_2 < a_4 \le 1$ and $a_1 + a_3 < a_2$, then we have

$$\tilde{\mathbf{\Omega}} = \check{\mathbf{\Omega}}, \qquad \text{and}$$

$$\|\tilde{\mathbf{\Omega}} - \mathbf{\Omega}\|_2 \le \|\tilde{\mathbf{\Omega}} - \mathbf{\Omega}\|_1 \le C_1 M_d \frac{s}{\sqrt{n}} + C_2 M_d \sqrt{\frac{\log d}{n}}$$

with probability at least $1 - O(1/d)$, where C_1, C_2 are absolute constants.

Theorem 3.3 states that under the minimal signal strength condition (6), the second-stage improvement recovers the true support as the oracle estimator does, i.e., $\tilde{\mathbf{\Omega}} = \check{\mathbf{\Omega}}$. Furthermore, it achieves a significantly faster convergence rate. The minimum condition on the magnitude of the true non-zero entries (6) is mild and standard to prove support recovery results and the oracle property (Meinshausen & Bühlmann, 2006; Cai et al., 2011; Wang et al., 2016). Remarkably, we do not need to impose the stringent irrepresentable condition (IC, Zhao & Yu, 2006; Ravikumar et al., 2011). Intuitively, the conditions on the values of a_1 to a_4 provides a theoretical trade-off among the sparsity level (a_1), the shrinkage effect of the non-convex penalties (a_2), the ambient dimension (a_3), and the minimal signal strength (a_4) in obtaining the oracle property: the denser the true precision matrix is (a_1 larger), the stronger minimal signal strength is required (a_4 larger), the larger value of η (a_2 larger) is necessary for the non-convex penalties, or a smaller dimension can be handled (a_3 smaller).

Table 1. Theoretical properties of different ℓ_1-regularized methods.

METHOD	$\|\cdot\|_1$ MINIMAX OPTIMAL	$\|\cdot\|_F$ MINIMAX OPTIMAL	TUNING-FREE
CLIME	×	×	×
GLASSO	WITH IC	WITH IC	×
SCIO	WITH IC	WITH IC	×
GRAPHICAL DANTZIG SELECTOR	√	×	×
TIGER	√	√	ASYMPTOTIC
GRANKLASSO	√	×	COMPLETE

3.2. Comparison to existing methods

In Table 1, we summarize theoretical properties of different ℓ_1-regularized estimators. Specifically, SCIO (Liu & Luo, 2015) achieves the same convergence rate as gRankLasso, but the irrepresentable condition (IC) is required. It is still unclear if SCIO can achieve the optimal minimax rate without the irrepresentable condition.

Similar to SCIO, the GLasso estimator also assumes the irrepresentable condition to obtain $O_p(sM_d\sqrt{(\log d)/n})$ rate of convergence under the spectral norm (Ravikumar et al., 2011). Fan et al. (2014) prove that the SCAD-penalized maximum likelihood estimator achieves the oracle property, but with a slower convergence rate than our second-stage enhancement.

CLIME (Cai et al., 2011) and TIGER (Liu & Wang, 2017) consider a larger matrix class where the conditional number (instead of the minimal and maximal eigenvalues) of the precision matrix is bounded. TIGER achieves the minimax optimal rate for this larger matrix class, but the convergence rate of CLIME under the spectral norm is $O_p(sM_d^2\sqrt{(\log d)/n})$, which is slower.

Wang et al. (2016) estimate each column of $\mathbf{\Omega}$ using a nonconvex penalty and also achieves a faster convergence rate under the spectral norm. However, their method requires heavy tuning, while our second-stage estimator only needs light tuning with the high-dimensional Bayesian information criteria (HBIC) as suggested in Wang et al. (2020).

4. Simulation studies

We consider MCP penalty in the second-stage enhancement and denote our method as gRankMCP. In this section, we compare the performances of gRankLasso and gRankMCP with GLasso, CLIME, and TIGER in terms of precision matrix estimation. All numerical experiments are implemented in `R` (R Core Team, 2021). The CLIME estimator is computed using the R package `flare` (Li et al., 2015); the TIGER and GLasso estimators are computed using the R package `huge` (Zhao et al., 2012).

4.1. General Comparison

We consider 3 types of graph: random, band, and cluster graph as described in Liu & Wang (2017) to determine the sparsity pattern in the final Gaussian graphical models. Particularly,

1. Erdős–Rényi random graph: Each pair of nodes is connected by an edge with probability 0.05 independently.
2. Band graph (with bandwidth 3): Two nodes i, j are connected by an edge if $|i-j| \leq 3$.
3. Cluster graph: The d nodes are partitioned into $\lceil d/20 \rceil$ disjoint groups. The subgraph of each group is a random graph with edge probability 0.2.

From each generated graph, we further generate an adjacency matrix $\mathbf{A}$ by setting the non-zero off-diagonal elements to be 0.3 and the diagonal elements to be 0. Let $\Lambda_{\min}(\mathbf{A})$ be the smallest eigenvalue of $\mathbf{A}$. The precision matrix is then generated by

$$\mathbf{\Omega} = \mathbf{D}[\mathbf{A} + (|\Lambda_{\min}(\mathbf{A})| + 0.2) \cdot \mathbf{I}_d]\mathbf{D}, \tag{7}$$

where $\mathbf{D} \in \mathbb{R}^{d\times d}$ is a diagonal matrix with $\mathbf{D}_{jj} = 1$ for $j = 1, \ldots, d/2$ and $\mathbf{D}_{jj} = 1.5$ for $j = d/2+1, \ldots, d$. Finally, n i.i.d. observations are sampled from the multivariate Gaussian distribution $N_d(\mathbf{0}, \mathbf{\Omega}^{-1})$. For each type of graph, we set $n = 100$ and $d \in \{25, 50, 100, 200, 400\}$ and repeat the simulation 50 times. For CLIME and GLasso, the optimal tuning parameter values are chosen using a validation set approach. Specifically, for each tuning parameter, CLIME and GLasso estimate the precision matrix $\hat{\mathbf{\Omega}}$ using the training data, and the optimal tuning parameter is chosen so that it minimizes the negative log-likelihood loss $L(\hat{\mathbf{\Omega}}) = \text{trace}(\hat{\mathbf{\Omega}}\hat{\mathbf{\Sigma}}) - \log\det(\hat{\mathbf{\Omega}})$ on the validation set, where $\hat{\mathbf{\Sigma}}$ is the sample covariance matrix. We use the regularization parameter $\lambda = \sqrt{(\log d)/n}$ for TIGER as suggested in Liu & Wang (2017) instead of fine-tuning. While gRankLasso is completely tuning-free, gRankMCP requires some light tuning. We use the HBIC (Wang et al., 2020) to select the best value of η in (5).

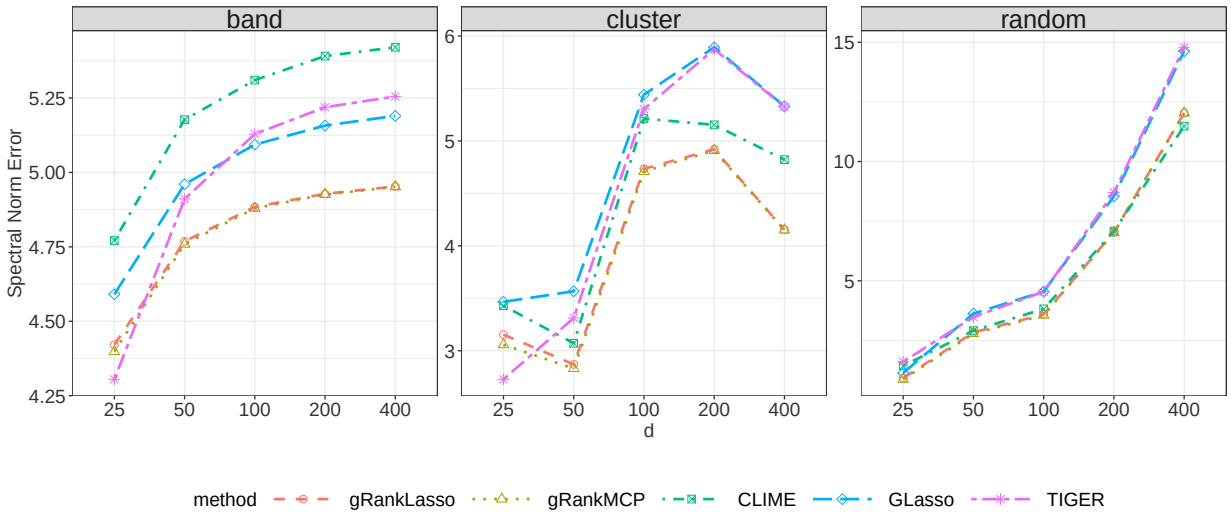


Figure 1. Comparison of estimation performance (in terms of the spectral norm error $\|\hat{\mathbf{\Omega}} - \mathbf{\Omega}\|_2$, averaged over 50 replications) of various methods in the three graph models with $d \in \{25, 50, 100, 200, 400\}$.

In Figure 1, we present the estimation error (averaged over 50 replications) under the spectral norm $\|\hat{\mathbf{\Omega}} - \mathbf{\Omega}\|_2$ for the three graph models. Evidently, gRankLasso and gRankMCP both outperform other methods in all three types of graphs. The performance advantage is especially pronounced in the high-dimensional setting where $d = 400$. The performance of CLIME and GLasso, the two methods that require tuning, are sensitive to the underlying graph type. In particular, CLIME has a higher estimation error than GLasso and TIGER for band graph, but achieves a lower estimation error for the other graph types. Remarkably, gRankLasso outperforms TIGER when both methods do not use tuning. This could be due to the difference in the completely tuning-free property of gRankLasso and the asymptotically tuning-free property of TIGER. With fine tuning, TIGER could potentially achieve an improved estimation performance, at a cost of more expensive computation. In Section 4.2, we further investigate the performance difference between gRankLasso and TIGER in various settings.

4.2. Sensitivity of tuning-free methods

In this section, we further illustrate the advantage of the completely tuning-free property of gRankLasso. To this end, we focus on the random graph model and study the performance difference between gRankLasso and TIGER. We consider various settings of the diagonal matrix $\mathbf{D}$ in (7). In particular, we set $\mathbf{D}_{jj} = 1$ for $j = 1, \ldots, d/2$ and $\mathbf{D}_{jj} = \tau$ for $j = d/2+1, \ldots, d$ with $\tau \in \{1, 1.5, 2, 2.5, 3\}$. Intuitively, the optimal level of regularization for estimating each column then falls into one of the two categories ($\mathbf{D}_{jj} = 1$ versus $\mathbf{D}_{jj} = \tau$). Thus τ gives a simplified characterization of the difference in the optimal level of regularization in estimating different columns of $\mathbf{\Omega}$. As the value of τ increases, it is expected that a method like TIGER, which enforces the regularization parameter λ_j to be the same across all column problems, will have a deteriorating performance.

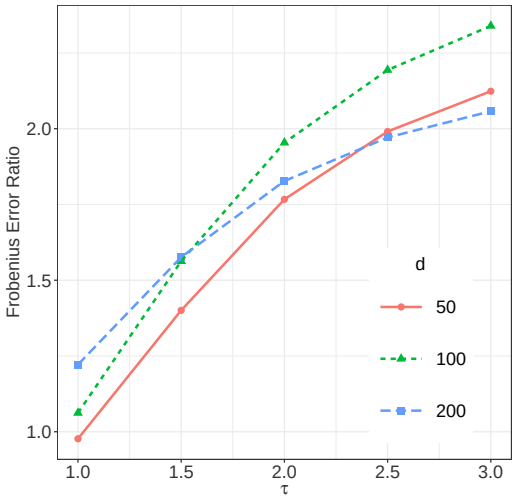


Figure 2. Frobenius error ratio $\|\hat{\mathbf{\Omega}}_{\text{gRankLasso}} - \mathbf{\Omega}\|_F^{-1} \|\hat{\mathbf{\Omega}}_{\text{TIGER}} - \mathbf{\Omega}\|_F$ (averaged over 50 replications) in the random graph model.

For each generated precision matrix, we follow the same paradigm to generate $n = 100$ observations of dimension $d \in \{50, 100, 200\}$ from the multivariate Gaussian distribution $\mathbf{X} \sim N_d(\mathbf{0}, \mathbf{\Omega}^{-1})$. Figure 2 shows the Frobenius errors ratio (averaged over 50 replications) $\|\mathbf{\Omega}_{\text{gRankLasso}} -$

$\|\boldsymbol{\Omega}\|_F^{-1}\|\boldsymbol{\Omega}_{\text{TIGER}} - \boldsymbol{\Omega}\|_F$. As expected, we observe that with an increasing value of τ, the performance advantage of gRankLasso over TIGER becomes more pronounced. This demonstrates a setting where the completely tuning-free property of gRankLasso is favored, and the asymptotically tuning-free property might fall short. We also note that this pattern holds for all three values of d, which covers the entire spectrum of the $n > d$, $n = d$, and $n < d$ settings.

4.3. Benefit of the second-stage enhancement

It is almost impossible to identify the difference in performance between gRankLasso and gRankMCP in Figure 1. To better understand the benefit of the second-stage enhancement in practice, we consider a more challenging setting with a denser true precision matrix: $\boldsymbol{\Omega}_{ij} = 0.6^{|i-j|}$, for $1 \le i, j \le d$, which is also considered in Cai et al. (2011). We then generate $n = 100$ observations with dimension $d \in \{25, 50, 100, 200, 400\}$ from $N_d(\mathbf{0}, \boldsymbol{\Omega}^{-1})$.

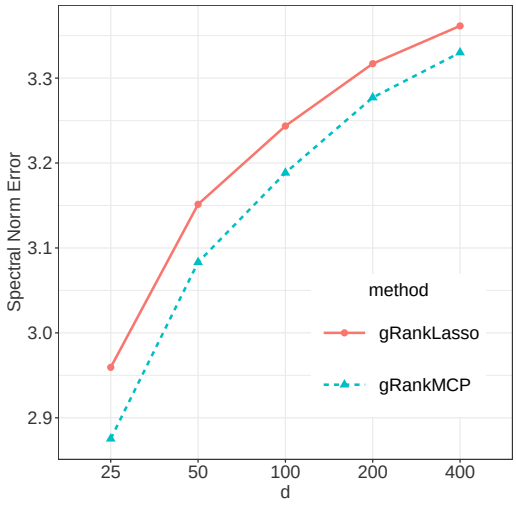


Figure 3. Comparisons between gRankLasso and gRankMCP on a decay graph model in terms of the spectral norm error $\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}\|_2$ (averaged over 50 replications).

Figure 3 shows the spectral norm error (averaged over 50 replications) $\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}\|_2$ of gRankLasso and gRankMCP for 5 values of d. Unsurprisingly, in a more challenging scenario, the efficiency gain of gRankMCP becomes more obvious. However, as mentioned above, this comes at a cost of additional tuning. It is then up to the practitioners' discretion to choose between the tuning-free gRankLasso and its second-stage enhancement based on the trade-off of budgets on statistical error and computational resources.

4.4. Heavy-tailed setting

Finally, as mentioned in Section 2, one potentially useful property of our proposed methods is the robustness against the violation of the underlying joint normality assumption. In this section, we evaluate performance of our methods in heavy-tailed setting in comparison with other methods. We consider the same setting of the random graph model as in Section 4.1. Instead of the Gaussian distribution, we generate observations from a multivariate t-distribution $t_\nu(\mathbf{0}, \boldsymbol{\Omega}^{-1})$ of dimension $d \in \{25, 50, 100, 200, 400\}$ with degrees of freedom $\nu \in \{3, 5, 10\}$.

Figure 4 shows the Frobenius norm error (averaged over the 50 replications) $\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}\|_F$. Across different settings, gRankLasso and gRankMCP still achieve the most favorable performance among all competing methods. In the most extreme case when $\nu = 3$, while all methods suffer, gRankLasso and gRankMCP clearly outperform competitors in the challenging high-dimensional case ($d = 400$). When $\nu = 10$, we see a similar performance to the Gaussian setting for all methods, which again shows the efficiency advantage of using the rank loss in (3).

5. Data example: Human gene network

We apply our proposed methods to reconstruct the interaction network from human gene expression data in the `BDgraph` R package (Mohammadi & Wit, 2019), which was previously studied by Bhadra & Mallick (2013); Mohammadi & Wit (2015); Liu & Wang (2017). This dataset consists of $n = 60$ individuals of Northern and Western European ancestry from Utah, whose genotypes are available online at the Sanger Institute website[1]. We use $d = 100$ variables in the dataset that are the 100 most variable probes corresponding to different Illumina TargetID transcripts, and were selected from the previous study of Bhadra & Mallick (2013) and the subsequent study of Mohammadi & Wit (2015).

The goal of this analysis is to learn the significant associations among the 100 chosen traits. As shown in Mohammadi & Wit (2019), all chosen traits are continuous but not Gaussian, so the assumption of joint normality is hardly satisfied. For the sake of comparison, we first use the Bayesian approach from Mohammadi & Wit (2015) to estimate the posterior probabilities of all possible edges, which leads to 124 significant edges (interaction with the estimated posterior probability greater than 0.6), and use these recovered edges as the baseline as if they were the truth. We then use the following methods to estimate the underlying graph: GLasso (the optimal tuning parameter selected using a 5-fold cross-validation), TIGER (with the regularization parameter set as $\lambda = \sqrt{(\log d)/n}$), gRankLasso, and gRankMCP (light tune with the HBIC).

Table 2 shows the `precision`, which is the ratio between `True` and `Total`, where `True` is the number of recovered

[1] ftp://ftp.sanger.ac.uk/pub/genevar

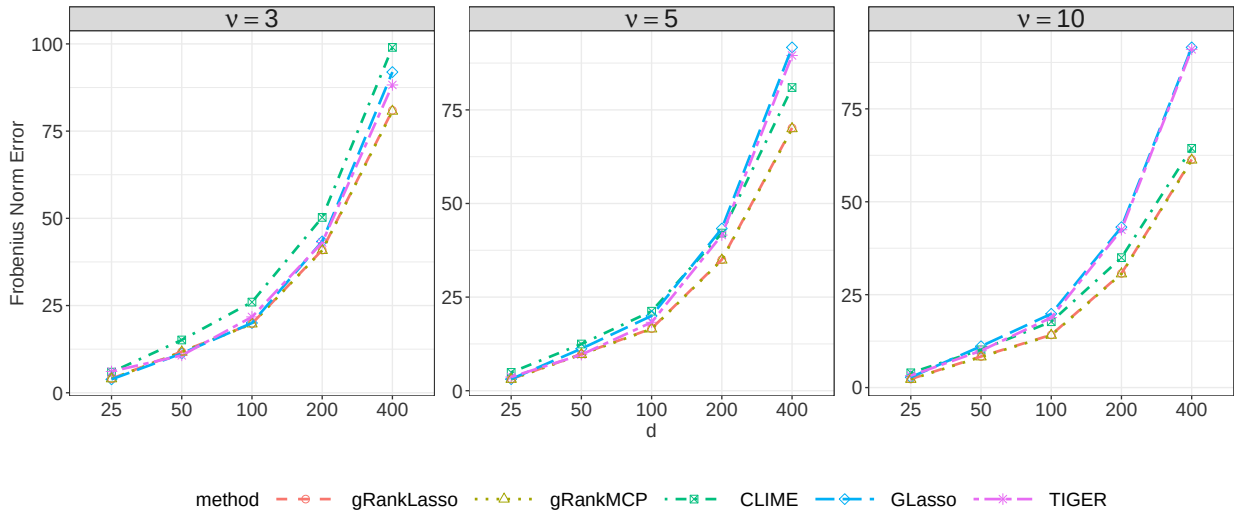
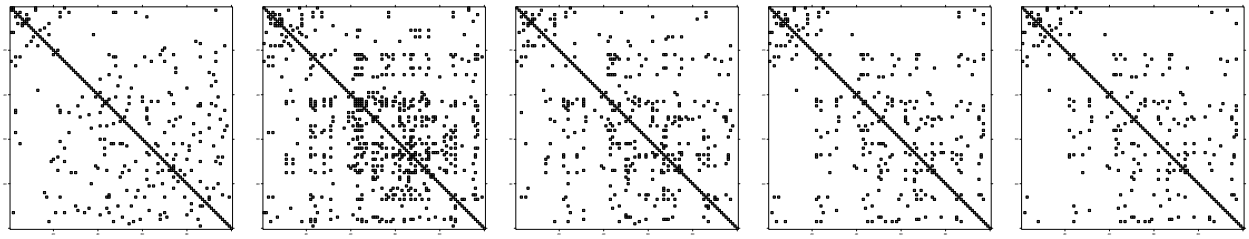


Figure 4. Comparison of estimation performance (in terms of the Frobenius norm error $\|\hat{\mathbf{\Omega}} - \mathbf{\Omega}\|_F$, averaged over 50 replications) of various methods in the three graph models when data are drawn from a multivariate t-distribution with degrees of freedom $\nu \in \{3, 5, 10\}$.



(a) Significant edges (b) GLasso (c) TIGER (d) gRankLasso (e) gRankMCP

Figure 5. The sparsity pattern of estimated graphs from TIGER, gRankLasso, and gRankMCP on human gene network data. The plot 5(a) shows 124 significant edges whose estimated posterior probabilities are greater than 0.6, and is considered to be the comparison baseline.

Table 2. Comparison of gRankLasso, gRankMCP, and TIGER on human gene expression data in terms of the number of True recovery, Total recovery, and Precision.

METHOD	TRUE	TOTAL	PRECISION
GLASSO	77	301	0.255
TIGER	66	179	0.368
GRANKLASSO	62	136	0.456
GRANKMCP	56	108	0.518

edges that are significant (in the sense of recovery by Mohammadi & Wit (2015)), and `Total` is the total number of recovered edges from each method. The sparsity patterns of the recovered graphs are shown in Figure 5. While the graph estimated by gRankLasso and gRankMCP are sparser, which is a favorable feature in terms of interpretability, they both achieve higher precision than TIGER and GLasso.

6. Discussion

We present gRankLasso, a completely tuning-free method for estimating Gaussian graphical models, which does not require tuning in finite samples. Its minimax optimal rates of convergence under the matrix ℓ_1 norm and the spectral norm are established. Our proposed method is accompanied by a second-stage enhancement that improves statistical efficiency due to the reduction of estimation bias. Under mild conditions, the second-stage estimator achieves faster convergence rates and enjoys the oracle property. Both proposed estimators can be computed very efficiently by linear programming. Favorable finite sample performance of our methods are illustrated through extensive numerical simulations and a real data application.

As we mentioned above, it is theoretically interesting to investigate whether gRankLasso can achieve the minimax optimal rate under the Frobenius norm and the element-wise max-norm over the matrix class $\mathcal{M}(s, M_d)$. Another

potential improvement is showing graph recovery results of gRankLasso under weaker assumptions (Kelner et al., 2020).

Recently, inference in Gaussian graphical models has drawn more attention (Fan et al., 2019; Li & Maathuis, 2021). One could formulate the debiased version (Javanmard & Montanari, 2018; Fan et al., 2020) of gRankLasso, and establish certain inferential results using the framework of Janková & van de Geer (2018). Additionally, estimation in high-dimensional Network Granger causal models (Basu et al., 2015) is another possible application of our methods. Their observed robustness to the heavy-tailed contamination makes them attractive candidates in estimating vector autoregressive models, especially in the presence of heavy-tailed and/or heteroscedastic noise. It will be interesting to evaluate performances of our proposed methods in these potential extensions.

References

Banerjee, O., Ghaoui, L. E., and d'Aspremont, A. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.*, 9:485–516, 2008. ISSN 1532-4435.

Basu, S., Shojaie, A., and Michailidis, G. Network granger causality with inherent grouping structure. *The Journal of Machine Learning Research*, 16(1):417–453, 2015.

Belloni, A., Chernozhukov, V., and Wang, L. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

Belloni, A., Kaul, A., and Rosenbaum, M. Pivotal estimation via self-normalization for high-dimensional linear models with error in variables. *arXiv preprint arXiv:1708.08353*, 2017.

Bhadra, A. and Mallick, B. Joint high-dimensional bayesian variable and covariance selection with an application to eqtl analysis. *Biometrics*, 69, 04 2013. doi: 10.1111/biom.12021.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. Simultaneous analysis of lasso and dantzig selector. *The Annals of statistics*, 37(4):1705–1732, 2009.

Bühlmann, P. and Van De Geer, S. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

Cai, T., Liu, W., and Luo, X. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, jun 2011. doi: 10.1198/jasa.2011.tm10155.

Cai, T. T., Liu, W., and Zhou, H. H. Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *The Annals of Statistics*, 44(2): 455 – 488, 2016. doi: 10.1214/13-AOS1171. URL https://doi.org/10.1214/13-AOS1171.

Candes, E. and Tao, T. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, pp. 2313–2351, 2007.

Chichignoud, M., Lederer, J., and Wainwright, M. J. A practical scheme and fast algorithm to tune the lasso with optimality guarantees. *The Journal of Machine Learning Research*, 17(1):8162–8181, 2016.

de Miranda Cardoso, J. V., Ying, J., and Palomar, D. Graphical models in heavy-tailed markets. *Advances in Neural Information Processing Systems*, 34, 2021.

Dempster, A. P. Covariance selection. *Biometrics*, pp. 157–175, 1972.

Fan, J. and Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001. doi: 10.1198/016214501753382273. URL https://doi.org/10.1198/016214501753382273.

Fan, J., Xue, L., and Zou, H. Strong oracle optimality of folded concave penalized estimation. *The Annals of Statistics*, 42(3):819 – 849, 2014. doi: 10.1214/13-AOS1198. URL https://doi.org/10.1214/13-AOS1198.

Fan, J., Ma, C., and Wang, K. Comment on "a tuning-free robust and efficient approach to high-dimensional regression". *Journal of the American Statistical Association*, 115(532):1720–1725, 2020. doi: 10.1080/01621459.2020.1837138. URL https://doi.org/10.1080/01621459.2020.1837138.

Fan, Y., Demirkaya, E., Li, G., and Lv, J. Rank: large-scale inference with graphical nonlinear knockoffs. *Journal of the American Statistical Association*, 2019.

Finegold, M. and Drton, M. Robust graphical modeling of gene networks using classical and alternative t-distributions. *The Annals of Applied Statistics*, 5(2A): 1057 – 1080, 2011. doi: 10.1214/10-AOAS410. URL https://doi.org/10.1214/10-AOAS410.

Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 12 2007. ISSN 1465-4644. doi: 10.1093/biostatistics/kxm045.

Hettmansperger, T. P. and McKean, J. W. *Robust nonparametric statistical methods*. CRC Press, 2010.

Jaeckel, L. A. Estimating regression coefficients by minimizing the dispersion of the residuals. *The Annals of Mathematical Statistics*, pp. 1449–1458, 1972.

Janková, J. and van de Geer, S. Inference in high-dimensional graphical models. In *Handbook of graphical models*, pp. 325–350. CRC Press, 2018.

Javanmard, A. and Montanari, A. Debiasing the lasso: Optimal sample size for gaussian designs. *The Annals of Statistics*, 46(6A):2593–2622, 2018.

Johnstone, I. M. Chi-square oracle inequalities. *Lecture Notes-Monograph Series*, pp. 399–418, 2001.

Kelner, J., Koehler, F., Meka, R., and Moitra, A. Learning some popular gaussian graphical models without condition number bounds. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 10986–10998. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/7cc980b0f894bd0cf05c37c246f215f3-Paper.pdf.

Laurent, B. and Massart, P. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302 – 1338, 2000. doi: 10.1214/aos/1015957395. URL https://doi.org/10.1214/aos/1015957395.

Lauritzen, S. L. *Graphical models*, volume 17. Clarendon Press, 1996.

Lederer, J. and Müller, C. Don't fall for tuning parameters: tuning-free variable selection in high dimensions with the trex. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Li, J. and Maathuis, M. H. Ggm knockoff filter: False discovery rate control for gaussian graphical models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(3):534–558, 2021.

Li, X., Zhao, T., Yuan, X., and Liu, H. The flare package for high dimensional linear regression and precision matrix estimation in r. *Journal of Machine Learning Research*, 16(18):553–557, 2015. URL http://jmlr.org/papers/v16/li15a.html.

Liu, H. and Wang, L. TIGER: A tuning-insensitive approach for optimally estimating Gaussian graphical models. *Electronic Journal of Statistics*, 11(1):241 – 294, 2017. doi: 10.1214/16-EJS1195. URL https://doi.org/10.1214/16-EJS1195.

Liu, W. and Luo, X. Fast and adaptive sparse precision matrix estimation in high dimensions. *Journal of Multivariate Analysis*, 135:153–162, 2015. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2014.11.005. URL https://www.sciencedirect.com/science/article/pii/S0047259X14002607.

Mazumder, R. and Hastie, T. Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, 13(27):781–794, 2012. URL http://jmlr.org/papers/v13/mazumder12a.html.

Meinshausen, N. and Bühlmann, P. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436 – 1462, 2006. doi: 10.1214/009053606000000281. URL https://doi.org/10.1214/009053606000000281.

Mohammadi, A. and Wit, E. C. Bayesian Structure Learning in Sparse Gaussian Graphical Models. *Bayesian Analysis*, 10(1):109 – 138, 2015. doi: 10.1214/14-BA889. URL https://doi.org/10.1214/14-BA889.

Mohammadi, R. and Wit, E. C. Bdgraph: An r package for bayesian structure learning in graphical models. *Journal of Statistical Software*, 89(3):1–30, 2019. doi: 10.18637/jss.v089.i03. URL https://www.jstatsoft.org/index.php/jss/article/view/v089i03.

R Core Team. R: A language and environment for statistical computing. r foundation for statistical computing, 2021, 2021.

Raskutti, G., Wainwright, M. J., and Yu, B. Restricted eigenvalue properties for correlated gaussian designs. *Journal of Machine Learning Research*, 11(78):2241–2259, 2010. URL http://jmlr.org/papers/v11/raskutti10a.html.

Ravikumar, P., Wainwright, M. J., Raskutti, G., and Yu, B. High-dimensional covariance estimation by minimizing ℓ1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5(none):935 – 980, 2011. doi: 10.1214/11-EJS631. URL https://doi.org/10.1214/11-EJS631.

Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. Sparse permutation invariant covariance estimation. *Electron. J. Statist.*, 2:494–515, 2008. doi: 10.1214/08-EJS176.

Sun, T. and Zhang, C.-H. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012.

Sun, T. and Zhang, C.-H. Sparse matrix inversion with scaled lasso. *The Journal of Machine Learning Research*, 14(1):3385–3418, 2013.

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

Tibshirani, R. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011.

Wainwright, M. J. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_l -constrained quadratic programming (lasso). *IEEE transactions on information theory*, 55(5):2183–2202, 2009.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Wang, L. The l1 penalized lad estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 120:135–151, 2013. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2013.04.001. URL https://www.sciencedirect.com/science/article/pii/S0047259X1300047X.

Wang, L., Ren, X., and Gu, Q. Precision matrix estimation in high dimensional gaussian graphical models with faster rates. In Gretton, A. and Robert, C. C. (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 177–185, Cadiz, Spain, 09–11 May 2016. PMLR. URL https://proceedings.mlr.press/v51/wang16a.html.

Wang, L., Peng, B., Bradic, J., Li, R., and Wu, Y. A tuning-free robust and efficient approach to high-dimensional regression. *Journal of the American Statistical Association*, 115(532):1700–1714, 2020. doi: 10.1080/01621459.2020.1840989. URL https://doi.org/10.1080/01621459.2020.1840989.

Yu, G. and Bien, J. Learning local dependence in ordered data. *The Journal of Machine Learning Research*, 18(1):1354–1413, 2017.

Yu, G. and Bien, J. Estimating the error variance in a high-dimensional linear model. *Biometrika*, 106(3):533–546, 2019.

Yuan, M. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11(79):2261–2286, 2010. URL http://jmlr.org/papers/v11/yuan10b.html.

Yuan, M. and Lin, Y. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 03 2007. ISSN 0006-3444. doi: 10.1093/biomet/asm018. URL https://doi.org/10.1093/biomet/asm018.

Zhang, C.-H. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894 – 942, 2010. doi: 10.1214/09-AOS729. URL https://doi.org/10.1214/09-AOS729.

Zhao, P. and Yu, B. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.

Zhao, T., Liu, H., Roeder, K., Lafferty, J., and Wasserman, L. The huge package for high-dimensional undirected graph estimation in r. *Journal of Machine Learning Research*, 13(37):1059–1062, 2012. URL http://jmlr.org/papers/v13/zhao12a.html.

A. Proofs

A.1. Preliminaries

We follow the framework in Yuan (2010) and Liu & Wang (2017) to prove convergence results. First, we provide some preliminaries. For a constant $c > 1$, define $\bar{c} = \frac{c+1}{c-1}$ and consider the cone set

$$\mathbf{\Gamma}^d = \{\boldsymbol{\gamma} \in \mathbb{R}^d : \|\boldsymbol{\gamma}_{S^c}\|_1 \leq \bar{c}\|\boldsymbol{\gamma}_S\|_1, S \subset \{1, 2, \ldots, d\}, \|S\|_0 \leq s\}.$$

Let S_j be the support of the j-th column of $\mathbf{\Omega}$, recall the s-sparse matrix class

$$\mathcal{M}(s, M_d) = \{\mathbf{\Omega} = \mathbf{\Omega}^T \in \mathbb{R}^{d\times d} : \mathbf{\Omega} \succ 0, \xi^{-1} \leq \Lambda_{\min}(\mathbf{\Omega}) \leq \Lambda_{\max}(\mathbf{\Omega}) \leq \xi, \max_{1\leq j\leq d} |S_j| \leq s, \|\mathbf{\Omega}\|_1 \leq M_d\},$$

We assume that the following conditions are satisfied:
(C1) $\mathbf{\Omega} \in \mathcal{M}(s, M_d)$,
(C2) $s^2 \log d = o(n)$.

For a non-convex penalty function, we assume that some general conditions are satisfied:

1. $p_\eta(t)$ is increasing and concave for $t \in [0, +\infty)$, with a continuous derivative $p'_\eta(t)$ on $(0, +\infty)$.
2. $p_\eta(t)$ has a singularity at the origin, *i.e.* $p'_\eta(0+) > 0$.
3. There exist constants $a_1 > 0$ and $a_2 > 1$ such that $p'_\eta(t) \geq a_1\eta$ for all $0 < t < a_2\eta$; and $p'_\eta(t) = 0$ for all $t > a_2\eta$.

Note that we use capital letter C to denotes an absolute constant, which can be different in different equations.

A.2. Technical lemmas

Lemma A.1. *Let $Y \sim \chi^2_d$. We have*

$$\mathbb{P}(|Y - d| > dt) \leq \exp\left(\frac{-3}{16}dt^2\right), \forall t \in [0, 1/2),$$
$$\mathbb{P}(Y \leq (1 - t)d) \leq \exp\left(\frac{-1}{4}dt^2\right), \forall t \in [0, 1/2).$$

Lemma A.2. *Let $\boldsymbol{\epsilon}^{(j)} \in \mathbb{R}^n$ such that $\boldsymbol{\epsilon}^{(j)} \sim N(0, \sigma_j^2 I_n)$. Then*

$$\max_{1\leq j\leq d} \left|\frac{\|\boldsymbol{\epsilon}^{(j)}\|_2^2}{n\sigma_j^2} - 1\right| \leq 3.5\sqrt{\frac{\log d}{n}}$$

holds with probability at least $1 - 1/d$.

Lemmas A.1 and A.2 are taken from Johnstone (2001); Laurent & Massart (2000); Liu & Wang (2017).

Lemma A.3. **ℓ_1 Restricted Eigenvalue condition**: *Let $\hat{\mathbf{\Sigma}} = \mathbf{X}^T\mathbf{X}/n$. Suppose $s\log(d) = o(n)$, then there exist constants c_1, c_2 such that*

$$\inf_{\boldsymbol{\gamma}\in\mathbf{\Gamma}^d} \frac{\sqrt{\boldsymbol{\gamma}\hat{\mathbf{\Sigma}}\boldsymbol{\gamma}}}{\|\boldsymbol{\gamma}\|_2} \geq \frac{1}{5\xi^{1/2}}$$

holds with probability at least $1 - c_1\exp(-c_2 n)$.

Proof. For any $S \subset \{1, 2, \ldots, n\}$ with $|S| \leq s$, we have, for any $\boldsymbol{\gamma} \in \mathbf{\Gamma}^d$,

$$\|\boldsymbol{\gamma}\|_1 \leq (1 + \bar{c})\|\boldsymbol{\gamma}_S\|_1 \leq (1 + \bar{c})\sqrt{s}\|\boldsymbol{\gamma}_S\|_2 \leq (1 + \bar{c})\sqrt{s}\|\boldsymbol{\gamma}\|_2,$$

and

$$\boldsymbol{\gamma}\boldsymbol{\Sigma}\boldsymbol{\gamma} \geq \Lambda_{\min}(\boldsymbol{\Sigma})\|\boldsymbol{\gamma}\|_2^2 \geq \Lambda_{\min}(\boldsymbol{\Sigma})\|\boldsymbol{\gamma}_S\|_2^2 \geq \Lambda_{\min}(\boldsymbol{\Sigma})\frac{\|\boldsymbol{\gamma}\|_1^2}{s(1+\bar{c})^2}.$$

Consider a random matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, in which each row is drawn i.i.d. from a $N(0, \boldsymbol{\Sigma})$. From Raskutti et al. (2010), there exists two positive constants c_1, c_2 such that

$$\mathbb{P}\left(\sqrt{\boldsymbol{\gamma}\hat{\boldsymbol{\Sigma}}\boldsymbol{\gamma}} \geq \frac{1}{4}\sqrt{\boldsymbol{\gamma}\boldsymbol{\Sigma}\boldsymbol{\gamma}} - 9\max_{1\leq j\leq d}\sqrt{\boldsymbol{\Sigma}_{jj}}\sqrt{\frac{\log d}{n}}\|\boldsymbol{\gamma}\|_1, \forall \boldsymbol{\gamma}\in\mathbb{R}^d\right) \geq 1 - c_1\exp(-c_2 n)$$

By definition, we have

$$\Lambda_{\max}(\boldsymbol{\Sigma}) \geq \max_{1\leq j\leq d}\boldsymbol{\Sigma}_{jj} \geq \min_{1\leq j\leq d}\boldsymbol{\Sigma}_{jj} \geq \Lambda_{\min}(\boldsymbol{\Sigma}).$$

It follows that,

$$\mathbb{P}\left(\sqrt{\boldsymbol{\gamma}\hat{\boldsymbol{\Sigma}}\boldsymbol{\gamma}} \geq \frac{1}{4}\sqrt{\Lambda_{\min}(\boldsymbol{\Sigma})}\|\boldsymbol{\gamma}\|_2 - 9(1+\bar{c})\sqrt{\Lambda_{\max}(\boldsymbol{\Sigma})}\sqrt{\frac{s\log d}{n}}\|\boldsymbol{\gamma}\|_2, \forall\boldsymbol{\gamma}\in\mathbb{R}^d\right) \geq 1 - c_1\exp(-c_2 n).$$

Thus

$$\mathbb{P}\left(\inf_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}}\frac{\sqrt{\boldsymbol{\gamma}\hat{\boldsymbol{\Sigma}}\boldsymbol{\gamma}}}{\|\boldsymbol{\gamma}\|_2} \geq \frac{1}{4}\sqrt{\Lambda_{\min}(\boldsymbol{\Sigma})} - 9(1+\bar{c})\sqrt{\Lambda_{\max}(\boldsymbol{\Sigma})}\sqrt{\frac{s\log d}{n}}, \forall\boldsymbol{\gamma}\in\mathbb{R}^d\right) \geq 1 - c_1\exp(-c_2 n).$$

Since we assume $s\log d = o(n)$, for n large enough, we have

$$\begin{aligned}\frac{1}{4}\sqrt{\Lambda_{\min}(\boldsymbol{\Sigma})} - 9(1+\bar{c})\sqrt{\Lambda_{\max}(\boldsymbol{\Sigma})}\sqrt{\frac{s\log d}{n}} &\geq \frac{1}{4\xi^{1/2}} - 9\xi^{1/2}(1+\bar{c})\sqrt{\frac{s\log d}{n}} \\ &\geq \frac{1}{5\xi^{1/2}}\end{aligned}$$

□

Lemma A.4. Prediction error bound of first-stage estimator: *Let $\hat{\boldsymbol{\beta}}^{(j)}$ be the Rank Lasso estimator of $\boldsymbol{\beta}^{(j)}$. Then*

$$\max_{1\leq j\leq d}\|\mathbf{X}_{*,-j}(\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)})\|_2 \leq C\sqrt{s\log d},$$

holds with probability at least $1 - O(1/d)$.

Proof. We have

$$\inf_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}^d}\frac{\sqrt{\boldsymbol{\gamma}\hat{\boldsymbol{\Sigma}}\boldsymbol{\gamma}}}{\|\boldsymbol{\gamma}\|_2} = \inf_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}^d}\frac{\|\mathbf{X}\boldsymbol{\gamma}\|_2}{\sqrt{n}\|\boldsymbol{\gamma}\|_2} \geq \frac{1}{5\xi^{1/2}},$$

then,

$$\min_{1\leq j\leq d}\inf_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}^{d-1}}\frac{\|\mathbf{X}_{*,-j}\boldsymbol{\gamma}\|_2}{\sqrt{n}\|\boldsymbol{\gamma}\|_2} \geq \inf_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}^d}\frac{\|\mathbf{X}\boldsymbol{\gamma}\|_2}{\sqrt{n}\|\boldsymbol{\gamma}\|_2} \geq \frac{1}{5\xi^{1/2}}.$$

Thus the ℓ_1-RE condition holds for all Rank Lasso subproblem from each column. From Lemma 2 of Wang et al. (2020), using a simulated λ_j from Algorithm 1, we have $\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)} \in \boldsymbol{\Gamma}^{d-1}$. From Lemma 9 of (Wainwright, 2009), we have

$$\mathbb{P}\left[\|\mathbf{X}^T\mathbf{X}/n\|_2 \leq \Lambda_{\max}(\boldsymbol{\Sigma})(1+\delta(n,d,t))\right] \geq 1 - 2\exp\left(-nt^2/2\right),$$

where $\delta(n,d,t) = 2(\sqrt{d/n}+t) + (\sqrt{d/n}+t)^2$. By setting $t = \sqrt{\dfrac{2\log d}{n}}$, we have $\delta(n,d,t) \leq 8$, and

$$\mathbb{P}\left[\|\mathbf{X}^T\mathbf{X}/n\|_2 \leq 9\Lambda_{\max}(\boldsymbol{\Sigma})\right] \geq 1 - 2/d.$$

From Theorem 1 of Wang et al. (2020) we have

$$\mathbb{P}\left[\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_2 \leq 5\xi^{1/2} C_0 \sqrt{\frac{s\log d}{n}}\right] \geq 1 - \alpha - 1/d^2.$$

Thus

$$\begin{aligned}
\max_{1\leq j\leq d} \frac{\|\mathbf{X}_{*,-j}(\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)})\|_2}{\sqrt{n}} &\leq \max_{1\leq j\leq d} \|\mathbf{X}_{*,-j}^T \mathbf{X}_{*,-j}/n\|_2^{1/2} \|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_2 \\
&\leq \|\mathbf{X}^T\mathbf{X}/n\|_2^{1/2} \max_{1\leq j\leq d} \|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_2 \\
&\leq 3\Lambda_{\max}^{1/2}(\boldsymbol{\Sigma}) 5\xi^{1/2} C_0 \sqrt{\frac{s\log d}{n}} \\
&= C\sqrt{\frac{s\log d}{n}},
\end{aligned}$$

with $C = 15\xi C_0$. We get the desired result by choosing $\alpha = 0$. □

Lemma A.5. Prediction error bound for oracle estimator: *Let $\check{\boldsymbol{\beta}}^{(j)}$ be the oracle estimator of $\boldsymbol{\beta}^{(j)}$. Then*

$$\max_{1\leq j\leq d} \|\mathbf{X}_{*,-j}(\check{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)})\|_2 \leq C\sqrt{s},$$

holds with probability at least $1 - O(1/d)$.

Proof. From Lemma 3 of Wang et al. (2020), we have

$$\|\check{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_2 = O_P(\sqrt{s/n}).$$

Then similar to Lemma A.4

$$\begin{aligned}
\max_{1\leq j\leq d} \frac{\|\mathbf{X}_{*,-j}(\check{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)})\|_2}{\sqrt{n}} &\leq \max_{1\leq j\leq d} \|\mathbf{X}_{*,-j}^T \mathbf{X}_{*,-j}/n\|_2^{1/2} \|\check{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_2 \\
&\leq \max_{1\leq j\leq d} \|\mathbf{X}^T\mathbf{X}/n\|_2^{1/2} \|\check{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_2 \\
&\leq 3\Lambda_{\max}^{1/2}(\boldsymbol{\Sigma}) C_0 \sqrt{\frac{s}{n}} \leq 3\xi^{1/2} C_0 \sqrt{\frac{s}{n}} = C\sqrt{\frac{s}{n}},
\end{aligned}$$

with $C = 3\xi^{1/2} C_0$. □

A.3. Main lemmas

Lemma A.6. Analyzing the diagonal elements of the gRankLasso estimator

$$\max_{1\leq j\leq d} \left|\hat{\boldsymbol{\Omega}}_{jj} - \boldsymbol{\Omega}_{jj}\right| \leq C\|\boldsymbol{\Omega}\|_2 \sqrt{\frac{\log d}{n}}$$

Proof. We have

$$\begin{aligned}
\left|(\hat{\boldsymbol{\Omega}}_{jj})^{-1} - (\boldsymbol{\Omega}_{jj})^{-1}\right| &= \left|\frac{\|\mathbf{X}_{*,j} - \mathbf{X}_{*,-j}\hat{\boldsymbol{\beta}}^{(j)}\|_2^2}{n} - \sigma_j^2\right| \\
&= \left|\frac{\|\mathbf{X}_{*,-j}(\boldsymbol{\beta}^{(j)} - \hat{\boldsymbol{\beta}}^{(j)}) + \boldsymbol{\epsilon}^{(j)}\|_2^2}{n} - \sigma_j^2\right| \\
&\leq \left|\frac{\|\boldsymbol{\epsilon}^{(j)}\|_2^2}{n} - \sigma_j^2\right| + \frac{\|\mathbf{X}_{*,-j}(\boldsymbol{\beta}^{(j)} - \hat{\boldsymbol{\beta}}^{(j)})\|_2^2}{n} + 2\frac{\left|(\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)})^T \mathbf{X}_{*,-j}^T \boldsymbol{\epsilon}^{(j)}\right|}{n}
\end{aligned}$$

From Lemmas A.1, A.2, A.4, we have

$$\left|\frac{\|\boldsymbol{\epsilon}^{(j)}\|_2^2}{n} - \sigma_j^2\right| \leq 3.5\sigma_j^2\sqrt{(\log d)/n},$$
$$\|\mathbf{X}_{*,-j}(\boldsymbol{\beta}^{(j)} - \hat{\boldsymbol{\beta}}^{(j)})\|_2 \leq C\sqrt{s\log d}.$$

From standard Gaussian tail bounds in Wainwright (2019), we also have for all $\delta > 0$

$$\mathbb{P}\left[\|\frac{\mathbf{X}_{*,-j}^T\boldsymbol{\epsilon}^{(j)}}{n}\|_\infty \leq C\sigma_j\left(\sqrt{(2\log d)/n} + \delta\right)\right] \geq 1 - 2\exp\big(-n\delta^2/2\big).$$

By setting $\delta = \sqrt{\frac{2\log d}{n}}$, we have

$$\mathbb{P}\left[\|\frac{\mathbf{X}_{*,-j}^T\boldsymbol{\epsilon}^{(j)}}{n}\|_\infty \leq C\sigma_j\left(2\sqrt{2}\sqrt{(\log d)/n}\right)\right] \geq 1 - 2/d.$$

It follows that

$$\begin{aligned}
2\frac{\left|(\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)})^T\mathbf{X}_{*,-j}^T\boldsymbol{\epsilon}^{(j)}\right|}{n} &\leq 2\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_1\|\frac{\mathbf{X}_{*,-j}^T\boldsymbol{\epsilon}^{(j)}}{n}\|_\infty \\
&\leq 2(1+\bar{c})\sqrt{s}\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_2 C\sigma_j\left(2\sqrt{2}\sqrt{\frac{\log d}{n}}\right) \\
&\leq 4\sqrt{2}(1+\bar{c})C\sigma_j\sqrt{s}\sqrt{\frac{s\log d}{n}}\left(\sqrt{\frac{\log d}{n}}\right) \\
&= 4\sqrt{2}(1+\bar{c})C\sigma_j s\frac{\log d}{n}.
\end{aligned}$$

Therefore,

$$\left|(\hat{\boldsymbol{\Omega}}_{jj})^{-1} - (\boldsymbol{\Omega}_{jj})^{-1}\right| \leq 3.5\sigma_j^2\sqrt{\frac{\log d}{n}} + C^2\frac{s\log d}{n} + 4\sqrt{2}(1+\bar{c})C\sigma_j s\frac{\log d}{n}.$$

Note that $s\sqrt{\frac{\log d}{n}} = o(1)$, so there exists a constant C such that, for large enough n

$$\left|(\hat{\boldsymbol{\Omega}}_{jj})^{-1} - (\boldsymbol{\Omega}_{jj})^{-1}\right| \leq C\sigma_j^2\sqrt{\frac{\log d}{n}}.$$

The rest of the proof follow Liu & Wang (2017). Since $\boldsymbol{\Omega}_{jj} = 1/\sigma_j^2$, we have

$$\left|\frac{\boldsymbol{\Omega}_{jj}}{\hat{\boldsymbol{\Omega}}_{jj}} - 1\right| \leq C\sqrt{\frac{\log d}{n}}.$$

This implies that

$$\left(1 + C\sqrt{\frac{\log d}{n}}\right)^{-1} \leq \frac{\hat{\boldsymbol{\Omega}}_{jj}}{\boldsymbol{\Omega}_{jj}} \leq \left(1 - C\sqrt{\frac{\log d}{n}}\right)^{-1}.$$

Then, for large enough n

$$1 - C\sqrt{\frac{\log d}{n}} \leq \left(1 + C\sqrt{\frac{\log d}{n}}\right)^{-1} \quad \text{and} \quad \left(1 - C\sqrt{\frac{\log d}{n}}\right)^{-1} \leq 1 + 2C\sqrt{\frac{\log d}{n}},$$

we have

$$\left(1 - C\sqrt{\frac{\log d}{n}}\right) \leq \frac{\hat{\mathbf{\Omega}}_{jj}}{\mathbf{\Omega}_{jj}} \leq \left(1 + C\sqrt{\frac{\log d}{n}}\right).$$

Thus

$$\max_{1\leq j\leq d}\left|\hat{\mathbf{\Omega}}_{jj} - \mathbf{\Omega}_{jj}\right| \leq C \max_{1\leq j\leq d}\mathbf{\Omega}_{jj}\sqrt{\frac{\log d}{n}} \leq C\|\mathbf{\Omega}\|_2\sqrt{\frac{\log d}{n}}.$$

□

Lemma A.7. Analyzing the off-diagonal elements in ℓ_1-norm error of the gRankLasso estimator

$$\max_{1\leq j\leq d}\|\hat{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}\|_1 \leq C(\|\mathbf{\Omega}\|_2 s + \|\mathbf{\Omega}\|_1)\sqrt{\frac{\log d}{n}}$$

Proof. Recall that

$$\mathbf{\Omega}_{-j,j} = -\mathbf{\Omega}_{jj}\boldsymbol{\beta}^{(j)},$$

Then

$$\begin{aligned}
\|\hat{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}\|_1 &= \|\hat{\sigma}_j^{-2}\hat{\boldsymbol{\beta}}^{(j)} - \sigma_j^{-2}\boldsymbol{\beta}^{(j)}\|_1 \\
&= \|\hat{\mathbf{\Omega}}_{jj}\hat{\boldsymbol{\beta}}^{(j)} + \hat{\mathbf{\Omega}}_{jj}\boldsymbol{\beta}^{(j)} - \hat{\mathbf{\Omega}}_{jj}\boldsymbol{\beta}^{(j)} - \mathbf{\Omega}_{jj}\boldsymbol{\beta}^{(j)}\|_1 \\
&\leq \left|\hat{\mathbf{\Omega}}_{jj}\right|\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_1 + \left|\hat{\mathbf{\Omega}}_{jj} - \mathbf{\Omega}_{jj}\right|\|\boldsymbol{\beta}^{(j)}\|_1 \\
&= \left|\hat{\mathbf{\Omega}}_{jj}\right|\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_1 + \left|\hat{\mathbf{\Omega}}_{jj} - \mathbf{\Omega}_{jj}\right|\|\mathbf{\Omega}_{-j,j}\mathbf{\Omega}_{jj}^{-1}\|_1 \\
&\leq \left|\hat{\mathbf{\Omega}}_{jj}\right|\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_1 + \left|\frac{\hat{\mathbf{\Omega}}_{jj}}{\mathbf{\Omega}_{jj}} - 1\right|\|\mathbf{\Omega}_{-j,j}\|_1
\end{aligned}$$

From lemmas A.4, A.6, we have

$$\begin{aligned}
&\hat{\mathbf{\Omega}}_{jj} \leq \left(1 + C\sqrt{\frac{\log d}{n}}\right)\mathbf{\Omega}_{jj} \leq 2\|\mathbf{\Omega}\|_2, \\
&\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_1 \leq (1+\bar{c})\sqrt{s}\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_2 \leq C(1+\bar{c})s\sqrt{\frac{\log d}{n}}, \\
&\left|\frac{\hat{\mathbf{\Omega}}_{jj}}{\mathbf{\Omega}_{jj}} - 1\right| \leq C\sqrt{\frac{\log d}{n}}.
\end{aligned}$$

Thus

$$\|\hat{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}\|_1 \leq C\|\mathbf{\Omega}\|_2 s\sqrt{\frac{\log d}{n}} + C\|\mathbf{\Omega}\|_1\sqrt{\frac{\log d}{n}}.$$

□

Lemma A.8. Analyzing the diagonal elements of second-stage estimator

$$\max_{1\leq j\leq d}\left|\tilde{\mathbf{\Omega}}_{jj} - \mathbf{\Omega}_{jj}\right| \leq C\|\mathbf{\Omega}\|_2\sqrt{\frac{\log d}{n}}$$

Proof. Let $\tilde{\boldsymbol{\beta}}^{(j)}$ be the second-stage estimator of $\boldsymbol{\beta}^{(j)}$, $\check{\boldsymbol{\beta}}^{(j)}$ be the oracle estimator of $\boldsymbol{\beta}^{(j)}$. From Theorem 2 of Wang et al. (2020), we have for α from algorithm 1

$$\mathbb{P}(\tilde{\boldsymbol{\beta}}^{(j)} = \check{\boldsymbol{\beta}}^{(j)}) \geq 1 - \alpha - h_n,$$

where $h_n \to 0$ as $n \to \infty$. It follows that

$$\mathbb{P}(\tilde{\sigma}_j = \check{\sigma}_j) \geq 1 - \alpha - h_n.$$

With $\alpha = 0$, we then obtain the strong oracle property by union bound.
Follow similar arguments from A.6, we have

$$\left|(\tilde{\mathbf{\Omega}}_{jj})^{-1} - (\mathbf{\Omega}_{jj})^{-1}\right| \leq \left|\frac{\|\boldsymbol{\epsilon}^{(j)}\|_2^2}{n} - \sigma_j^2\right| + \frac{\|\mathbf{X}_{*,-j}(\boldsymbol{\beta}^{(j)} - \tilde{\boldsymbol{\beta}}^{(j)})\|_2^2}{n} + 2\frac{\left|(\tilde{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)})^T \mathbf{X}_{*,-j}^T \boldsymbol{\epsilon}^{(j)}\right|}{n}.$$

From Lemmas A.1, A.2, A.5, A.6, we have

$$\begin{aligned}
&\left|\frac{\|\boldsymbol{\epsilon}^{(j)}\|_2^2}{n} - \sigma_j^2\right| \leq 3.5\sigma_j^2\sqrt{(\log d)/n}, \\
&\|\mathbf{X}_{*,-j}(\boldsymbol{\beta}^{(j)} - \tilde{\boldsymbol{\beta}}^{(j)})\|_2 \leq C\sqrt{s}, \\
&\frac{\left|(\tilde{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)})^T \mathbf{X}_{*,-j}^T \boldsymbol{\epsilon}^{(j)}\right|}{n} \leq 2\sqrt{2}(1+\bar{c})C\sigma_j s\sqrt{\frac{1}{n}}\left(\sqrt{\frac{\log d}{n}}\right).
\end{aligned}$$

Therefore,

$$\left|(\tilde{\mathbf{\Omega}}_{jj})^{-1} - (\mathbf{\Omega}_{jj})^{-1}\right| \leq 3.5\sigma_j^2\sqrt{\frac{\log d}{n}} + C^2\frac{s}{n} + 4\sqrt{2}(1+\bar{c})C\sigma_j\frac{s}{\sqrt{n}}\left(\sqrt{\frac{\log d}{n}}\right).$$

Since $s\sqrt{\frac{\log d}{n}} = o(1)$, there exists a constant C such that, for large enough n

$$\left|(\tilde{\mathbf{\Omega}}_{jj})^{-1} - (\mathbf{\Omega}_{jj})^{-1}\right| \leq C\sigma_j^2\sqrt{\frac{\log d}{n}}.$$

The rest of the proof follow Lemma A.6. We have

$$\max_{1\leq j\leq d}\left|\tilde{\mathbf{\Omega}}_{jj} - \mathbf{\Omega}_{jj}\right| \leq C\max_{1\leq j\leq d}\mathbf{\Omega}_{jj}\sqrt{\frac{\log d}{n}} \leq C\|\mathbf{\Omega}\|_2\sqrt{\frac{\log d}{n}}.$$

□

Lemma A.9. Analyzing the off-diagonal elements in ℓ_1-norm error of second-stage estimator

$$\max_{1\leq j\leq d}\|\tilde{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}\|_1 \leq C_1\|\mathbf{\Omega}\|_2\frac{s}{\sqrt{n}} + C_2\|\mathbf{\Omega}\|_1\sqrt{\frac{\log d}{n}}.$$

Proof. Follow similar arguments of Lemma A.7, we have

$$\|\tilde{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}\|_1 \leq \left|\tilde{\mathbf{\Omega}}_{jj}\right|\|\tilde{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_1 + \left|\frac{\tilde{\mathbf{\Omega}}_{jj}}{\mathbf{\Omega}_{jj}} - 1\right|\|\mathbf{\Omega}_{-j,j}\|_1$$

From Lemmas A.5, A.8, we have

$$\begin{aligned}
&\tilde{\mathbf{\Omega}}_{jj} \leq \left(1 + C\sqrt{\frac{\log d}{n}}\right)\mathbf{\Omega}_{jj} \leq 2\|\mathbf{\Omega}\|_2, \\
&\|\tilde{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_1 \leq (1+\bar{c})\sqrt{s}\|\tilde{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\|_2 \leq C(1+\bar{c})\frac{s}{\sqrt{n}}, \\
&\left|\frac{\tilde{\mathbf{\Omega}}_{jj}}{\mathbf{\Omega}_{jj}} - 1\right| \leq C\sqrt{\frac{\log d}{n}}.
\end{aligned}$$

Thus

$$\|\tilde{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}\|_1 \leq C_1\|\mathbf{\Omega}\|_2\frac{s}{\sqrt{n}} + C_2\|\mathbf{\Omega}\|_1\sqrt{\frac{\log d}{n}}.$$

□

A.4. Main theorems

Theorem 3.1

Proof. The proof is identical to Liu & Wang (2017). From lemmas A.6 and A.7, we have

$$\begin{aligned}
\|\hat{\mathbf{\Omega}} - \mathbf{\Omega}\|_1 &= \max_{1\le j\le d} \|\hat{\mathbf{\Omega}}_{*,j} - \mathbf{\Omega}_{*,j}\|_1 \\
&\le \max_{1\le j\le d} \left|\hat{\mathbf{\Omega}}_{jj} - \mathbf{\Omega}_{jj}\right| + \max_{1\le j\le d} \|\hat{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}\|_1 \\
&\le C\|\mathbf{\Omega}\|_2\sqrt{\frac{\log d}{n}} + C(\|\mathbf{\Omega}\|_2 s + \|\mathbf{\Omega}\|_1)\sqrt{\frac{\log d}{n}} \\
&\le C(\|\mathbf{\Omega}\|_2 s + \|\mathbf{\Omega}\|_1)\sqrt{\frac{\log d}{n}} \\
&\le C\left(s\|\mathbf{\Omega}\|_1\sqrt{\frac{\log d}{n}}\right) \\
&\le C\left(sM_d\sqrt{\frac{\log d}{n}}\right).
\end{aligned}$$

□

Theorem 3.3

Proof. From lemmas A.8 and A.9, we have

$$\begin{aligned}
\|\tilde{\mathbf{\Omega}} - \mathbf{\Omega}\|_1 &= \max_{1\le j\le d} \|\tilde{\mathbf{\Omega}}_{*,j} - \mathbf{\Omega}_{*,j}\|_1 \\
&\le \max_{1\le j\le d} \left|\tilde{\mathbf{\Omega}}_{jj} - \mathbf{\Omega}_{jj}\right| + \max_{1\le j\le d} \|\tilde{\mathbf{\Omega}}_{-j,j} - \mathbf{\Omega}_{-j,j}\|_1 \\
&\le C_0\|\mathbf{\Omega}\|_2\sqrt{\frac{\log d}{n}} + C_1\|\mathbf{\Omega}\|_2\frac{s}{\sqrt{n}} + C_2\|\mathbf{\Omega}\|_1\sqrt{\frac{\log d}{n}} \\
&\le C_1\|\mathbf{\Omega}\|_1\frac{s}{\sqrt{n}} + C_2\|\mathbf{\Omega}\|_1\sqrt{\frac{\log d}{n}} \\
&\le C_1 M_d\frac{s}{\sqrt{n}} + C_2 M_d\sqrt{\frac{\log d}{n}}
\end{aligned}$$

□