University of California
Santa Barbara

Statistical Guarantees of Tuning-Free Methods for Gaussian Graphical Models

A thesis submitted in partial satisfaction
of the requirements for the degree

Master of Arts
in
Statistics

by

Chau Bao Tran

Committee in charge:

Professor Alexander Petersen, Co-Chair
Professor Sang-Yun Oh, Co-Chair
Professor Guo Yu

June 2022

The Thesis of Chau Bao Tran is approved.

Professor Guo Yu

Professor Sang-Yun Oh, Committee Co-Chair

Professor Alexander Petersen, Committee Co-Chair

May 2022

Statistical Guarantees of Tuning-Free Methods for Gaussian Graphical Models

by

Chau Bao Tran

This thesis is dedicated to my family.

Acknowledgements

First and foremost, I would like to thank my advisors Professors Alexander Petersen, Sang-Yun Oh, and Guo Yu. Without their supports and patience, I would not be able to achieve any accomplishment. I am incredibly lucky to have them as my advisors, and they inspire me to pursue a career as an academic statistician.

I am also grateful to my professors at UC Santa Barbara for their guidance during my studies, with a special mention to Professors Alexander Franks, Wendy Meiring, S. Rao Jammalamadaka, Tomoyuki Ichiba, and Yekaterina Kharitonova. I also thank my excellent collaborators Pedro Cisneros-Velarde and Chao Zhang, from whom I learn a lot from our research discussions.

I would like to thank the administrative staff from the Department of Statistics and Applied Probability. I also thank my friends and fellow students from the PSTAT Graduate Student Committee for their dedication to creating an encouraging community.

Finally, my deepest gratitude goes to my parents for their unconditional love and constant encouragement.

Curriculum Vitæ

Chau Bao Tran

Education

2022	M.A. in Statistics, University of California, Santa Barbara.
2020	B.A. in Statistics and Data Science, University of California, Santa Barbara.

Publications

- C. Tran and G. Yu. "A Completely Tuning-free and Robust Approach to Sparse Precision Matrix Estimation". *39th International Conference on Machine Learning*.
- C. Tran, P. Cisneros-Velarde, S. Oh, and A. Petersen. "Family-wise Error Rate control in Gaussian Graphical Model Selection via Distributionally Robust Optimization". To appear in *Stat*.

Abstract

Statistical Guarantees of Tuning-Free Methods for Gaussian Graphical Models

by

Chau Bao Tran

The majority of methods for sparse precision matrix estimation rely on computationally expensive procedures, such as cross-validation, to determine the proper level of regularization. Recently, a special case of precision matrix estimation based on a distributionally robust optimization (DRO) framework has been shown to be equivalent to the graphical lasso. From this formulation, a method for choosing the regularization term, i.e., for graphical model selection, without tuning was proposed. In Chapter 2 of this thesis, we establish a theoretical connection between the confidence level of graphical model selection via the DRO formulation and the asymptotic family-wise error rate of estimating false edges. Simulation experiments and real data analyses illustrate the utility of the asymptotic family-wise error rate control behavior even in finite samples.

Next, we propose a completely tuning-free approach to estimating sparse precision matrix based on linear regression in Chapter 3. Theoretically, the proposed estimator is minimax optimal under various norms. In addition, we propose a second-stage enhancement with non-convex penalties, which possesses strong oracle properties. We assessed our proposed methods through comprehensive simulations and real data application on human gene network analysis.

Contents

Chapter 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 $X = (X_1, \ldots, X_d) \sim N_d(\mathbf{0}, \Sigma)$, the conditional dependence structure is encoded in the sparsity pattern of the precision matrix $\Omega = \Sigma^{-1}$: $\Omega_{jk} = 0$ when (j, k) is not an edge in G [1]. This conditional dependency make Gaussian Graphical Model a useful tool for network analysis in many applications such as finance, neuroscience, and genetics [2, 3, 4, 5].

Also known as the covariance selection problem [6], we are primarily interested in estimating Ω using n observations of the d-dimensional random vector X. In high-dimensional setting where $n < d$, this problem becomes challenging, so regularization becomes a common strategy for graph selection and estimation. Inducing sparsity is an especially favorable choice of regularization since the sparsity pattern in Ω encodes the conditional independence structures among $X_1, \ldots, X_d$.

In literature, there are essentially two types of sparsity-inducing estimators in Gaus-

sian graphical models. One type of method is based on penalized likelihood estimation with the well-studied graphical lasso estimator [7, 8, 9] being an example of this type. Subsequently, theoretical properties of likelihood-based methods are established [10, 11, 12, 13], and penalized likelihood estimation methods with non-convex penalties are proposed [14, 15]. An alternative type of approach that is more amenable to theoretical analysis estimates Ω in a column-by-column fashion, where each column is estimated in a regularized regression problem [16, 17, 18, 19]. These pseudo-likelihood methods are also more flexible and less computationally challenging compared to the full likelihood estimators [20]. The optimal performance of estimators from both types 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. Therefore, providing statistical guarantees for tuning-free methods the level of regularization of which can be determined without any tuning is an interesting research direction.

Recently, a distributionally robust formulation for inverse covariance matrix was proposed [21], resulting in a class of ℓ_p-regularized estimators, with the graphical lasso with $p = 1$ is a special case. Additionally, they utilizing the Robust Wasserstein Profile function [22] for this formulation, [21] proposed the Robust Selection criterion with a fast bootstrap-based algorithm for estimating the regularization parameter for graphical lasso. In Chapter 2, we provide a theoretical connection between the Robust Selection criterion and asymptotic family-wise error rate control in Gaussian Graphical model selection.

For penalized linear regression, a tuning-free estimator referred to as Rank Lasso has been proposed [23]. The optimal regularization level for this estimator does not depend on any unknown quantities and automatically adjusts for design matrix and random error distribution. As a result, the regularization parameter can easily be simulated from data. In Chapter 3, we proposed a novel estimator for high-dimensional Gaussian

graphical models based on the Rank Lasso and provide the convergence rate for this estimator. We further propose a second-stage enhancement using non-convex penalties, which enjoys oracle properties.

Chapter 2

Family-wise Error Rate control for Graphical Lasso

2.1 Introduction

Distributionally robust optimization (DRO) as an estimation framework seeks parameters that minimize the worst expected risk over the uncertainty set of distributions (often called *ambiguity set* in DRO terminology) [24]. Leveraging the DRO framework, [21] showed that for a fixed $\rho \geq 1$ and $p \in [1, \infty]$, their DRO formulation of regularized inverse covariance estimation is equivalent to the following expression:

$$\min_{K \in \mathbb{S}_{\mathrm{d}}^{++}} \left\{ \mathrm{Tr}(KA_n) - \log |K| + \delta^{1/\rho} \|\mathbf{vec}(K)\|_p \right\}, \tag{2.1}$$

where A_n is the empirical covariance matrix, $\mathbb{S}_{\mathrm{d}}^{++}$ denotes the set of $d \times d$ positive definite matrices, and δ is the radius of ambiguity set, which is constructed as a ball in the Wasserstein space of distributions, centered at the empirical measure of the data. Note that the graphical lasso objective function is a special case of (2.1) when $p = 1$ and

$\rho = 1$. Constants p and ρ specify the Wasserstein distance metric between two probability distributions. Remarkably, the regularization parameter of graphical lasso corresponds to the ambiguity set radius δ despite the differing premise between DRO and maximum likelihood estimator. Intuitively, an increase in ambiguity set radius δ (i.e., an increased robustness in DRO) corresponds to an increased amount of regularization in graphical lasso (which results in conservative selection of non-zeros).

Using the *Robust Wasserstein Profile (RWP) function* R_n introduced by [22], [21] derived the RWP function for the graphical lasso, $R_n(K) = \|\mathbf{vec}(A_n - K)\|_\infty$, and characterized its asymptotic distribution. The distribution is used to determine δ (equivalently, the regularization parameter λ in graphical lasso from [8]) given the user specified error tolerance level α:

$$\begin{aligned}\lambda = \delta &:= \inf\{\delta > 0 \mid \mathbb{P}_0(R_n(\Omega) \leq \delta)\} \\ &= \inf\{\delta > 0 \mid \mathbb{P}_0(\|\mathbf{vec}(A_n - \Sigma)\|_\infty \leq \delta) \geq 1 - \alpha\}, \end{aligned} \tag{2.2}$$

where $\mathbb{P}_0$ denotes the true underlying distribution of the data. This graphical model selection procedure is called *RobSel* in [21]. Then, by Corollary 3.3 of [21], $n^{1/2}\delta$ tends to the $1-\alpha$ quantile of R_n, $r_{1-\alpha}$, and the corresponding δ can be determined from an order statistic in finite samples. The asymptotic result also motivates the approximation of the RWP function through a bootstrap procedure in Algorithm 1 to determine the regularization parameter λ, given significance level α.

2.2 Family-wise error rate control with RobSel

In this section, we provide results for the interpretation of α and its relation to Type I error control in graphical model selection. Recall that equation (2.1) shows that the DRO

Algorithm 1 RobSel algorithm for estimation of the regularization parameter λ [21]

Input: n observations, $X_1, \ldots, X_n$.
Set parameters $\alpha \in (0,1)$ and $B \in \mathbb{N}$.
Compute empirical covariance A_n.
for $b = 1, ..., B$ **do**
Obtain a bootstrap sample $X^*_{1b}, \ldots, X^*_{nb}$ by sampling uniformly and with replacement from the data
Compute empirical covariance $A^*_{n,b}$ from the bootstrap sample.
$R^*_{n,b} \leftarrow \|A^*_{n,b} - A_n\|_\infty$
end for
Set λ to be the bootstrap order statistic $R^*_{n,((B+1)(1-\alpha))}$.

estimator is equivalent to the ℓ_1-penalized estimator in graphical lasso, which produces a sparse estimator of Ω, denoted $\hat{\Omega}^\delta$. Given equation (2.2), a natural question is how to interpret error tolerance α, which was not addressed in [21]. The following result directly connects the parameter α in RobSel and the asymptotic FWER of the corresponding obtained estimator.

Theorem 2.2.1 (FWER of graphical lasso) *Let $\Xi = \{(i,j) : \Omega_{ij} = 0\}$ be the indices corresponding to zero entries of Ω. For a fixed α, let δ satisfy* (2.2) *and let $\hat{\Omega}^\delta$ be the unique solution to optimization problem* (2.1) *with $\rho = 1$. Then*

$$\lim_{n\to\infty} \mathbb{P}(\hat{\Omega}^\delta_{ij} \neq 0 \text{ for some } (i,j) \in \Xi) \leq \alpha. \tag{2.3}$$

Proof: In this proof, let $\mathbb{S}_\mathrm{d}$ be the set of $d \times d$ symmetric matrices. Recall that $n^{1/2}\delta \to r_{1-\alpha}$, where $r_{1-\alpha}$ is the $1-\alpha$ quantile of the distribution in Corollary 3.3 and Remark 3.5 of [21]. Then, by Theorem 1 of [25], we have that $n^{1/2}(\hat{\Omega}^\delta - \Omega)$ converges in distribution to U^*, the minimizer of

$$\arg\min_{U=U'} \ \mathrm{Tr}(U\Sigma U\Sigma) + \mathrm{Tr}(UH) + r_{1-\alpha} \sum_{i\neq j} \left\{ u_{ij}\mathrm{sign}(\Omega_{ij})\mathbf{1}(\Omega_{ij} \neq 0) + |u_{ij}|\mathbf{1}(\Omega_{ij} = 0) \right\},$$

where $H \in \mathbb{S}_{\mathrm{d}}$ is a matrix of jointly Gaussian random variables with zero mean such that $\mathrm{Cov}\left(h_{ij}, h_{k\ell}\right) = E\left[x_i x_j x_k x_\ell\right] - \Sigma_{ij}\Sigma_{k\ell}$. By the convex nature of the above optimization problem, using the first optimality criterion using subdifferentials (Corollary 2.7 of [26]), it follows that there exists some $Z \in \mathbb{S}_{\mathrm{d}}$ satisfying

$$Z_{ij} = \begin{cases} 0, & i = j, \\ \mathrm{sign}(\Omega_{ij}), & i \neq j, \Omega_{ij} \neq 0, \\ \mathrm{sign}(u_{ij}), & i \neq j, \Omega_{ij} = 0, u_{ij} \neq 0, \\ \in [-1, 1], & i \neq j, \Omega_{ij} = u_{ij} = 0. \end{cases}$$

for which $H + 2\Sigma U^* \Sigma + r_{1-\alpha} Z = 0$. Letting $\otimes$ denote the matrix Kronecker product and $\Gamma = \Sigma \otimes \Sigma$, it follows that

$$\mathbf{vec}(U^*) = -\frac{1}{2}\Gamma^{-1}\left\{\mathbf{vec}(H) + r_{1-\alpha}\mathbf{vec}(Z)\right\}.$$

Finally, let $\hat{\Omega}^{\delta}_{\Xi}$ denote the vector of elements of $\hat{\Omega}^{\delta}$ whose indices are in Ξ, Ω_{Ξ} denote the vector of elements of Ω whose indices are in Ξ (so it is the zero vector), and U^*_{Ξ} denote the vector of elements of U^* whose indices are in Ξ. Then one concludes that

$$\begin{aligned} &\lim_{n\to\infty} \mathbb{P}(\hat{\Omega}^{\delta}_{ij} \neq 0 \text{ for some } (i,j) \in \Xi) \\ &= \lim_{n\to\infty} \mathbb{P}(\sqrt{n}(\hat{\Omega}^{\delta}_{\Xi} - \Omega_{\Xi}) \neq 0) = \mathbb{P}(U^*_{\Xi} \neq 0) \\ &\leq \mathbb{P}(U^* \neq 0) = 1 - \mathbb{P}(H \neq -r_{1-\alpha}Z) \leq 1 - \mathbb{P}(\|\mathbf{vec}(H)\|_\infty \leq r_{1-\alpha}) \\ &= \alpha. \end{aligned}$$

■

Using the estimated regularization parameter $\lambda(\alpha)$ from RobSel for graphical lasso, Theorem 2.2.1 states that the asymptotic probability that the estimated graph includes

a false edge (false non-zero estimated in $\hat{\Omega}^{\delta}$) is bounded by α. This interpretation is equivalent to having the FWER bounded by α in hypothesis testing-based graphical model selection in [27]. As a result, Theorem 2.2.1 implies that RobSel can also serve as a tool for controlling graphical lasso's FWER at some chosen significance level α with similar to using a hypothesis testing-based graphical model selection.

Concretely, testing $d(d-1)/2$ null hypotheses that each pairwise partial correlation is zero can serve as an alternative way to construct a graphical model, where the partial correlation between variables i and j is defined as $\rho_{ij\cdot\text{rest}} = -\Omega_{ij}/\sqrt{\Omega_{ii}\Omega_{jj}}$ and $i, j = 1, 2, \ldots, d$. The unadjusted p-value π_{ij} for each null hypothesis is be obtained by

$$\pi_{ij} = 2[1 - \Phi(\sqrt{n-d-1} \cdot |z_{ij\cdot\text{rest}}|)], \tag{2.4}$$

where Φ is the CDF of standard normal distribution, $z_{ij\cdot\text{rest}} = \text{arctanh}(r_{ij\cdot\text{rest}})$ is the Fisher z transformed sample partial correlation $r_{ij\cdot\text{rest}}$ for population partial correlation $\rho_{ij\cdot\text{rest}}$. To account for multiple comparison, a p-value correction is needed to achieve a desired FWER characteristic. One of the multiple testing correction methods given in [27] controls the FWER based on Holm's approach for p-value adjustment:

$$\pi_{a\uparrow}^{\text{Holm}} = \max_{b=1,\ldots,a} \left[\min \left\{ \left(\binom{d}{2} - b + 1 \right) \pi_{b\uparrow}, 1 \right\} \right], \text{ for } 1 \leq a \leq \binom{d}{2}. \tag{2.5}$$

where $\pi_{1\uparrow} \leq \pi_{2\uparrow} \leq ... \leq \pi_{d(d-1)/2\uparrow}$ are the ordered p-values from (2.4). This approach will be referred to as the Holm-corrected testing method for graphical model selection in our numerical experiments. Other multiple testing correction approaches discussed in [27] include Bonferroni and Šidák adjustments. For the remainder of our work, we compare RobSel with the Holm-corrected testing method for its simplicity (compared to the Šidák-based approach) and better power characteristic (compared to the Bonferroni-

based approach). We emphasize that the distinct advantage of graphical lasso is that it can perform model selection and parameter estimation of Ω simultaneously, whereas any testing-based approach can only identify the zeros/non-zero locations of Ω.

2.3 Numerical results

In this section, analyses of simulated and real data illustrate the usefulness of RobSel's asymptotic FWER property in finite samples and compare to the Holm-based multiple testing approach for Gaussian graphical model selection. Furthermore, RobSel is used to analyze real datasets from genomics.

To carry out our numerical experiments, we used packages `CVglasso` for cross validation, `qgraph` for the extended Bayesian information criterion, and `robsel` for Robust Selection. These packages are from CRAN, and they use package `glasso` to estimate the sparse inverse covariance matrix. Robust Selection algorithm is also available as a Python package, `robust-selection`, at `https://pypi.org/project/robust-selection/`. The codes to reproduce the numerical results is available at `https://github.com/cbtran/robsel-reproducible`.

2.3.1 Simulation experiments

In applications, the finite sample behavior of the FWER characteristic whose asymptotic properties are given in Theorem 2.2.1 is of practical interest. In this section, simulation studies are used to verify the FWER of graph reconstruction when using RobSel with graphical lasso. Furthermore, the FWER of a testing-based graphical model selection from [27] is given as a comparison.

The true precision matrix $\Omega \in \mathbb{S}_{\mathrm{d}}^{++}$ used to generate the simulated data has been constructed as follows. First, generate an adjacency matrix of an undirected Erdős-Renyi

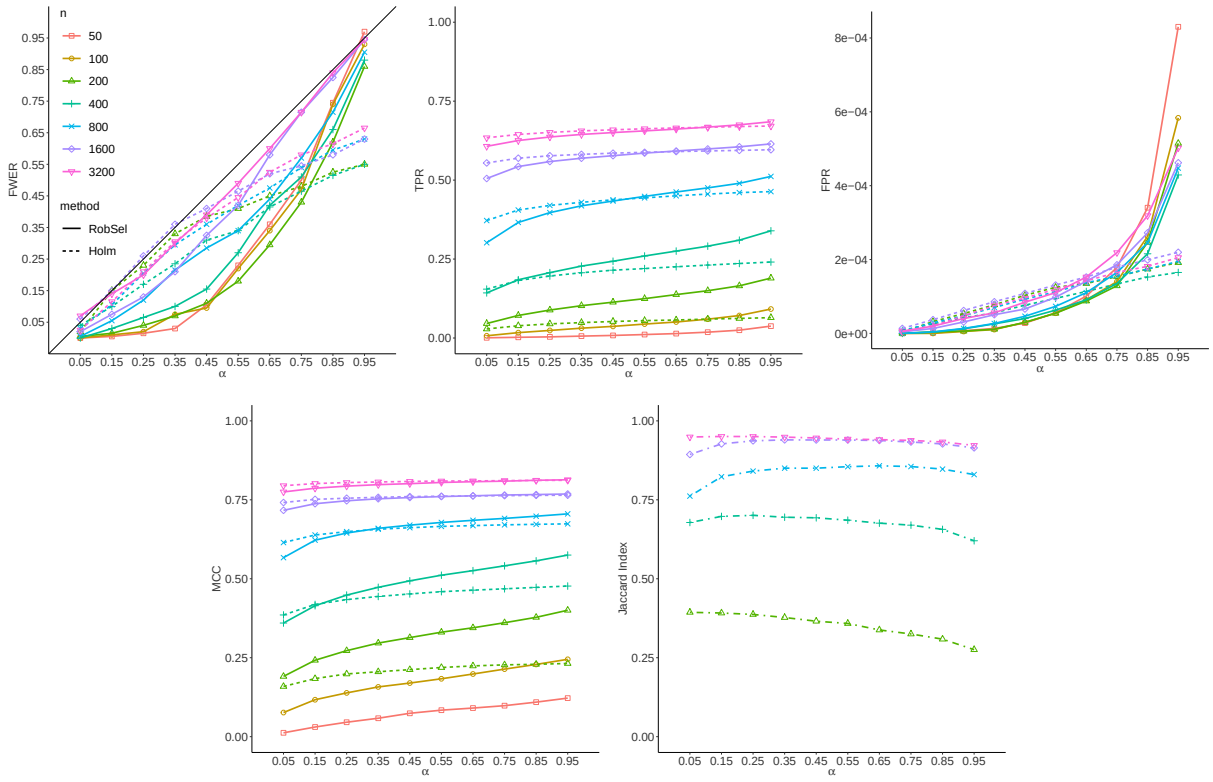


Figure 2.1: Observed family-wise error rate (top-left), True Positive Rate (top-middle), False Positive Rate (top-right), Matthews Correlation Coefficient (bottom-left), and Jaccard index of similarity (bottom-right) evaluated from graphs estimated with RobSel with graphical lasso and Holm-based multiple testing method. Note that Holm-based method is not applicable when $n \leq d = 100$. All traces represent average quantities over 200 datasets.

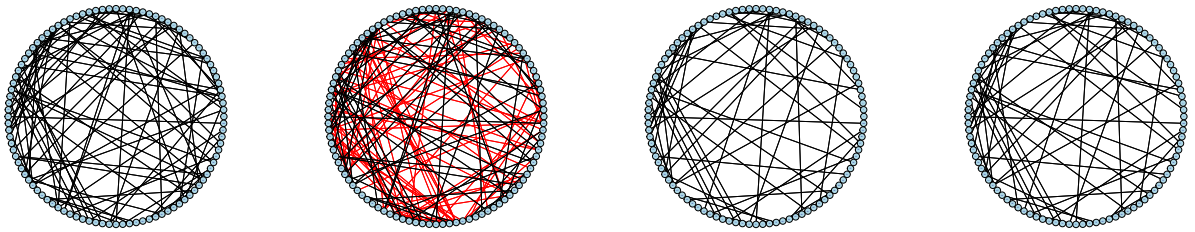
graph with equal edge probability of 0.02 discarding any self-loops. Then, the weight of each edge (the magnitude of the non-zero element) is sampled uniformly between $[0.5, 1]$, and the sign of each non-zero element is set to be positive or negative with equal probability of 0.5. The resulting matrix is made diagonally dominant by following a procedure described in [28], which ensures that the resulting matrix Ω is positive definite with ones on the diagonal. Finally, the diagonal entries of Ω are resampled uniformly between $[1, 1.5]$. Throughout this numerical study section, one randomly generated instance of sparse matrix Ω with $d = 100$ variables is fixed. Using this Ω, a total of $N = 200$ datasets for each sample size $n \in \{50, 100, 200, 400, 800, 1600, 3200\}$ were generated independently

from a multivariate zero-mean Gaussian distribution, i.e., $\mathcal{N}(\mathbb{0}_d, \Omega^{-1})$.

To evaluate the selected models, family-wise error rate (FWER), true positive rate (TPR), false positive rate (FPR), Matthews correlation coefficient (MCC), and Jaccard index were used as performance metrics. These metrics are derived from elements in the confusion matrix, true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), where a positive indicates an estimated presence of an edge (two non-zero entries in Ω). In this setting, family-wise error rate is the probability of any false edge detection: $FWER = \mathbf{1}(FP > 0)$. True positive rate is the proportion of edges in true graph G that are correctly identified in the estimated graph: $TPR = \frac{TP}{TP+FN}$. False positive rate is the proportion of nonedges in true graph G that are incorrectly identified as edges in the estimated graph: $FPR = \frac{FP}{FP+TN}$. Matthews correlation coefficient summarizes all count in confusion matrix to measure quality of graph recovery performance: $MCC = \frac{TP \cdot TN - FP \cdot FN}{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}$. Jaccard index measure the similarity between two edge sets E_A and E_B: $J(E_A, E_B) = \frac{|E_A \cap E_B|}{|E_A \cup E_B|}$, and, by convention, Jaccard index of two empty sets is defined to be one, i.e., $J(\emptyset, \emptyset) = 1$.

Figure 2.1 shows the FWER, TPR, FPR, MCC, and Jaccard index of the estimated graphs from both Holm's multiple testing method and the graphical lasso with RobSel criterion. TPR increases as sample size increases; however, for each sample size, both method have similar TPR, but RobSel appears to be more conservative at small significant levels since it tends to have smaller TPR and FWER. For larger α, RobSel is less conservative with higher TPR while its FWER still bounded by α. Figure 2.1 also show the average Jaccard index from 200 simulations at 5 different sample size and 10 different levels α. It can be seen that Jaccard index increases as sample size increases indicating the estimated graphs from both RobSel and Holm-based multiple testing method become increasingly similar.

Figure 2.2 illustrates a striking similarity between graphical lasso tuned with RobSel



(a) Ground Truth

(b) Graphical lasso tuned with cross-validation

(c) Graphical lasso tuned with RobSel ($\alpha = 0.05$)

(d) Holm's multiple testing ($\alpha = 0.05$)

Figure 2.2: True and three estimated graphs from a dataset with $n = 3200$. Red edges denote False Positive edges.

and testing-based graphs for large n. Most edges appear in both graphs and both graphs do not contain any false positive edge owing to the stringent significance level. On the other hands, graphical lasso tuned with cross-validation have many false positive edges. These qualitative observations were typical in our numerical simulations when data were generated from multivariate normal distributions across a wide range of sample sizes we considered.

2.3.2 Application to gene regulatory network reconstruction

Here, we infer gene regulatory networks from real datasets provided for the DREAM5 transcriptional network inference challenge from [29]. We reconstructed the networks of interactions among transcription factors (TF). TF-encoding genes usually act as hub-genes with large numbers of interactions with other genes [30]. Thus, identifying interactions between TFs may help researchers better understand the relationships between different groups of genes. The *in silico* dataset contains $d = 195$ transcription factors on $n = 805$ arrays. The *Escherichia coli* (E. coli) dataset contains $d = 334$ transcription factors on $n = 805$ arrays. The *Saccharomyces cerevisiae* (S. cerevisiae) dataset contains $d = 333$ transcription factors on $n = 536$ arrays. To evaluate the inferred networks,

we validated the edges in estimated graphical models against experimentally validated interactions given in [29].

Graphical models were constructed using graphical lasso tuned with three different regularization parameter selection approaches as well as the using the Holm-corrected testing method described in Section 2.2. The regularization parameter tuning approaches we considered were as follows. The first is *Robust Selection (RobSel)*, with $B = 200$ sets of bootstrap samples. The second is 5-*fold cross-validation (CV)* procedure, where the performance on the validation set is the evaluation of the graphical loss function under the empirical measure of the samples on the training set. The third is extended Bayesian information criterion (EBIC) proposed in [31]. CV and EBIC are evaluated on the same grid of λ, which are ten logarithmically spaced values in the interval $(0.05s_{\max}, s_{\max}]$ with $s_{\max}$ being the minimal value of regularization that gives an empty graph: i.e., setting $\lambda = s_{\max}$ for graphical lasso returning a diagonal matrix Ω. Note that increasing the number of λ values on the grid increases computational time.

Because DRO framework minimizes worst case expected loss, specifying a small error tolerance α for RobSel often results in a graph with very few edges being estimated especially when analyzing a real dataset. In practice, a larger α might be beneficial in order to estimate graphs with more edges. Note, however, that setting a λ corresponding to a large α when using graphical lasso would still return a very sparse graph. In our analyses, RobSel was specified with $\alpha = 0.9$, EBIC with parameter $\gamma = 0.5$, and 5-fold for cross-validation. EBIC criterion has the following form:

$$\text{EBIC}_\gamma(E) = -2\mathcal{L}(\hat{\Omega}(E)) + |E| \log n + \gamma 4|E| \log d, \tag{2.6}$$

where E is the edge set of a candidate graph implied by $\hat{\Omega}$, and $\mathcal{L}(\hat{\Omega}(E))$ denotes the maximized log-likelihood function of the associated model.

Dataset	Method	# Estimated	# Validated	Precision	Time(s)
In silico	Holm	289	63	0.2184	0.088
	RobSel	693	89	0.1284	0.467
	EBIC	1237	108	0.0873	1.566
	CV	7241	168	0.0232	8.611
E. coli	Holm	269	14	0.0520	0.166
	RobSel	3479	22	0.0063	3.355
	EBIC	6599	37	0.0056	10.46
	CV	10770	43	0.0040	52.92
S. cerevisiae	Holm	56	3	0.0536	0.149
	RobSel	4259	46	0.0108	2.728
	EBIC	7731	70	0.0091	17.80
	CV	11367	93	0.0082	85.64

Table 2.1: Graph recovery results and computational times in seconds from the DREAM5 datasets for three methods, Holm's testing procedure with $\alpha = 0.9$, RobSel with $\alpha = 0.9$, extended BIC (EBIC) with $\gamma = 0.5$, and 5-fold cross-validation (CV).

Table 2.1 show the number of edges in the estimated graph, number of validated edges (interactions found in [29]), precision (the ratio of validated edge counts to total edge counts), and the wall clock times. In our results, an estimated edge (i.e. gene interaction) is a true positive if it is experimentally validated interaction in the database, i.e. in [29]. We can see that for all three data sets, RobSel appears to be faster than EBIC and CV with similar precisions. Between E. coli and S. cerevisiae data sets, computational time for RobSel decreases when sample size decreases, but computational times of both EBIC and CV increase. Even though we used RobSel with $\alpha = 0.9$ to get a denser graph, the estimated graph by RobSel are still much sparser than EBIC and CV.

2.4 Discussion

We made a theoretical connection between significant level α from RobSel and family-wise error rate of estimating any false positive edges when RobSel is used to tune graphical lasso. Furthermore, the asymptotic FWER control property is tested in finite sample us-

ing simulation experiments. The similarity between Holm-testing method and RobSel tuned graphical lasso solutions when using the same significance level α give users practical insight about the behavior of graphical lasso: graphical lasso regularization can be chosen according to a user specified FWER level.

Chapter 3

A Completely Tuning-Free Approach to Precision Matrix Estimation

3.1 Introduction

In this chapter, we consider the pseudo-likelihood approach to Gaussian Graphical Model estimation. Given a d-dimensional multivariate Gaussian random vector $X = (X_1, \ldots, X_d) \sim N(0, \Sigma)$, we are interested in estimating the precision matrix $\Omega = \Sigma^{-1}$.

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

$$X_j = X_{-j}^T \beta^{(j)} + \epsilon_j, \tag{3.1}$$

where $\beta^{(j)} = [\Sigma_{-j,-j}]^{-1}\Sigma_{-j,j}$ and $\epsilon_j \sim N\left(0, \sigma_j^2\right)$ with $\sigma_j^2 = \Sigma_{j,j} - \Sigma_{j,-j}[\Sigma_{-j,-j}]^{-1}\Sigma_{-j,j}$.

By the block matrix inversion formula, we have

$$\Omega_{j,j} = \sigma_j^{-2}, \qquad \text{and} \qquad \Omega_{-j,j} = -\sigma_j^{-2}\beta^{(j)}. \tag{3.2}$$

It suggests that an estimate of the j-th column of Ω can be obtained by estimating the regression coefficients $\beta^{(j)}$ and the error variance σ_j^2 of the linear model (3.1). Thus, the problem of estimating Ω can be formulated as a series of d regression problems, each of which estimates one column of Ω.

Note from (3.2) that the sparsity pattern in an estimate of $\beta^{(j)}$ is equivalent to the sparsity pattern of the estimated j-th column of Ω under the joint normality. This observation drives many recently proposed methods, most of which are built upon various regularized regression techniques. For example, to estimate each column of Ω, neighborhood selection [16] use lasso [32], [17] use the Dantzig selector [33], [18] use scaled lasso [34], [19] use square-root lasso [35]. 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. The only exceptions, as far as the we know, are the strongly related TIGER [19] and the scaled lasso [18]. Although both methods greatly simplify the tuning procedure, the claimed tuning-free property only holds asymptotically. In practice, the computational caveats of these methods include (1) enforcing the same tuning parameter value to be used for estimating all columns of Ω, and (2) the common tuning parameter value includes a constant that still requires fine-tuning. These limitations call for the development of an estimator of Ω that is completely tuning-free, where the level of regularization can be determined without any tuning and is fully adaptive to each column problem separately.

Contributions: In this chapter, 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 be easily 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 properties. Through comprehensive numerical studies, we demonstrate that the favorable performance of the proposed methods, and illustrate their robustness to the violation of the Gaussian assumptions.

First, we provide detail on the notation. For the rest of the section, we let operator $|\cdot|$ denote absolute value for a scalar and cardinal number of a set. For a vector $\beta \in \mathbb{R}^d$, β_i denotes its i-th element. We define the ℓ_p norm of a vector as $\|\beta\|_p = (\sum_{i=1}^d |\beta_i|^p)^{1/p}$ for $0 < p < \infty$, and $\|\beta\|_\infty = \max_i |\beta_i|$. For a matrix $A \in \mathbb{R}^{n\times d}$, A_{jk} denotes its (j,k) entry, $A_{*,j}$ denotes the j-th column of A, and $A_{*,-j}$ denotes the submatrix of A with j-th column removed. We denote the matrix L_p norm as $\|A\|_p = \max_{\|\mathbf{v}\|_p=1} \|Av\|_p$, and matrix Frobenius norm as $\|A\|_F = (\sum_{j,k} |A_{j,k}|^2)^{1/2}$. Finally, $A \succ 0$ denotes that the matrix A is positive definite. We use capital letter C to denotes an absolute constant which may change for each line of equations.

3.2 Methods

3.2.1 Our proposed method: gRankLasso

In this section, we propose the graph rank lasso estimator (gRankLasso) of Gaussian graphical models, where each column of Ω is estimated using rank lasso [23]. Specifically, to estimate the j-th column of Ω using the data matrix $X \in \mathbb{R}^{n\times d}$, we use the following

rank loss function

$$Q_j(\beta) = [n(n-1)]^{-1} \sum_{k=1}^{n} \sum_{m \neq k} |(X_{kj} - X_{mj}) - (X_{k,-j} - X_{m,-j})\beta|, \tag{3.3}$$

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 non-parametric regression, this loss is equivalent to, up to a constant, the Jaeckel's dispersion function with Wilcoxon scores [36, 37]. Then the estimate of the j-th column of Ω can be obtained by

$$\begin{aligned} &\hat{\beta}^{(j)} = \underset{\beta \in \mathbb{R}^{d-1}}{\operatorname{argmin}} \{Q_j(\beta) + \lambda_j \|\beta\|_1\}, \\ &\hat{\sigma}_j^2 = n^{-1} \|X_{*,j} - X_{*,-j}\hat{\beta}^{(j)}\|_2^2, \\ &\hat{\Omega}_{jj} = 1/\hat{\sigma}_j^2, \qquad \hat{\Omega}_{-j,j} = -\hat{\Omega}_{jj}\hat{\beta}^{(j)}. \end{aligned} \tag{3.4}$$

Using Algorithm 2, we can easily simulate the regularization parameter λ_j in (3.4) that satisfies the subgradient condition with high probability [23, 38, 39]. The value of α and c are theoretical necessities. Setting $\alpha = 0.1$ and $c = 1.01$ works well in practice. Moreover, the optimization problem (3.4) can be formulated as a linear programming (LP), which can be solved efficiently using a standard solver.

While recently there are numerous tuning-free methods in high-dimensional linear models [40, 41, 42, 43], we argue that rank lasso loss (3.3) is an especially attractive candidate in each column estimation problem. First of all, 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 design matrix. Hence, we allow regularization parameters λ_j to be different for different j. Second, the regularization parameters λ_j

Algorithm 2 Simulate λ_j, for $j = 1, ..., d$

Input: $X \in \mathbb{R}^{n\times d}$, $\alpha \in [0, 1], c > 1, B \in \mathbb{N}$
for k in $1 : B$ **do**
  Generate random perturbation for $1 : n$, denotes as r
  $\phi = 2 \cdot r - (n + 1)$
  for j in $1 : d$ **do**
    $L_j[k] = c \cdot \|2[n(n-1)]^{-1}(X_{*,-j})^T\phi\|_\infty$
  end for
end for
Output: $\lambda_j = \text{Quantile}(L_j, 1 - \alpha/d)$, for $j = 1, ..., d$

can be easily simulated from data, which is extremely efficient to compute and requires absolutely no fine tuning. Third, among other regression methods that share similar properties, the rank lasso is significantly more efficient in Gaussian settings [23]. Finally, it was also noted that the rank lasso estimator is robust to heavy-tailed error contamination. This bonus property makes it attractive for many data applications where the stringent multivariate Gaussian assumption is not guaranteed.

3.2.2 A second-stage improvement

The ℓ_1 penalty used in (3.4), while being computationally friendly, is known to induce large estimation bias [44]. Therefore, many non-convex penalties have been proposed to circumvent this issue [45, 46]. Next, we present a second-stage improvement with non-convex penalties [23] using gRankLasso as an initial estimator. Specifically, for $1 \le j \le d$,

$$\tilde{\beta}^{(j)} = \underset{\beta\in\mathbb{R}^{d-1}}{\text{argmin}}\{Q_j(\beta) + \sum_{i=1}^{d} p'_\eta(|\hat{\beta}_i^{(j)}|)|\beta_i|\}, \tag{3.5}$$

$$\tilde{\sigma}_j^2 = n^{-1}\|X_{*,j} - X_{*,-j}\tilde{\beta}^{(j)}\|_2^2,$$

$$\tilde{\Omega}_{jj} = 1/\tilde{\sigma}_j^2, \qquad \tilde{\Omega}_{-j,j} = -\tilde{\Omega}_{jj}\tilde{\beta}^{(j)},$$

where $\hat{\beta}^{(j)}$ is obtained in (3.4) 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 is motivated by the local linear approximation algorithm [47, 23] and applies to a general class of non-convex penalties, which will be described in Section 3.3. The optimization problem (3.5) can also be formulated as a LP, which can be solved efficiently. As mentioned in [47], this one-step estimate provides dramatically computational speed-up without losing statistical efficiency. Note that a tuning parameter η is required in the general non-convex penalty in (3.5), which requires light tuning. At a cost of higher computational cost, we show in Section 3.3 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 true Ω.

Practically, to ensure that the estimate of Ω is symmetric, we set $\hat{\Omega}_{ij}^{\text{sym}} = \hat{\Omega}_{ji}^{\text{sym}} = \min\{\hat{\Omega}_{ij}, \hat{\Omega}_{ji}\}$ and $\tilde{\Omega}_{ij}^{\text{sym}} = \tilde{\Omega}_{ji}^{\text{sym}} = \min\{\tilde{\Omega}_{ij}, \tilde{\Omega}_{ji}\}$ for $i \neq j$. This additional symmetrization step does not affect the theoretical analysis shown in [48].

3.3 Theoretical analysis

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

(C1) $\Omega \in \mathcal{M}(s, M_d)$,

(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 the true Ω to grow with d.

3.3.1 Main theorems

Theorem 3.3.1 *(Matrix L_1 and spectral norm rates) With the adaptive choice of $\lambda_j, j = 1, ..., d$ from Algorithm 2, under assumptions (C1) and (C2), we have*

$$\|\hat{\Omega} - \Omega\|_2 \leq \|\hat{\Omega} - \Omega\|_1 \leq CsM_d\sqrt{\frac{\log d}{n}}$$

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

Theorem 3.3.1 shows the convergence rate of matrix estimation under 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 [17]).

Corollary 3.3.2 *(Frobenius norm rate) With the adaptive choice of $\lambda_j, j = 1, ..., d$ from Algorithm 2, under assumption (C1) and (C2), we have*

$$\|\hat{\Omega} - \Omega\|_F \leq CsM_d\sqrt{\frac{d\log d}{n}}$$

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

The Frobenius norm bound is worse than the minimax optimal rate by a factor of $\sqrt{s}$ [10, 49, 19]. Whether gRankLasso could achieve the minimax Frobenius norm rate is an interesting future research direction.

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 function:

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 $p'_\eta(t) \geq a_1\eta$ 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 [45] and MCP [46]. We show that the second-stage improvement performs as if one knows the sparsity pattern of the true Ω. Specifically, let $\check{\Omega}$ be the oracle estimator of Ω defined as follows: For $i \leq j \leq d$

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

That is, $\check{\beta}^{(j)}$ is the minimizer of the rank loss function Q_j in (3.3) when the support of the j-th column of Ω is known. Using a non-convex penalty such as SCAD or MCP for the second-stage estimator, we can show the oracle property of the precision matrix estimation.

Theorem 3.3.3 *Let $\tilde{\Omega}$ be the second-stage estimator of Ω using gRankLasso $\hat{\Omega}$ as an initial estimator. Suppose the conditions in Theorem 3.3.1 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 non-zero entries of the true* Ω *satisfies*

$$\min_{i \neq j} |\Omega_{ij}| \geq b n^{-(1-a_4)/2} \tag{3.6}$$

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

$$\tilde{\Omega} = \check{\Omega}, \qquad \textit{and}$$
$$\|\tilde{\Omega} - \Omega\|_2 \leq \|\tilde{\Omega} - \Omega\|_1 \leq C_1 M_d \frac{s}{\sqrt{n}} + C_2 M_d \sqrt{\frac{\log d}{n}}$$

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

Theorem 3.3.3 states that under the minimal signal strength condition (3.6), the second-stage improvement recovers the support of the oracle estimator by noting that $\tilde{\Omega} = \check{\Omega}$. Furthermore, it achieves a significantly faster convergence rate. The minimum condition on the magnitude of true nonzero entries (3.6) is mild and standard for proving support recovery results and oracle property [16, 48, 50]. Remarkably, we do not need to impose the irrepresentable condition [51], which is very stringent.

3.3.2 Comparison to existing methods

We compare the theoretical properties of graphical Rank Lasso with other existing methods for sparse precision matrix estimation.

The matrix class $\mathcal{M}(s, M_d)$ is widely considered in the Gaussian graphical models literature. Specifically, the Graph Dantzig selector [17] can also be formulated as a LP. However, the graphical Dantzig selector requires tuning of regularization parameter while gRankLasso is completely tuning free. Furthermore, graphical Dantzig selector does not have the strong oracle property with a second-stage enhancement.

Another method that utilizes the idea of Dantzig selector [33] is CLIME [48]. While CLIME considers a slightly larger class of matrix where the conditional number (instead of the minimal and maximal eigenvalues) of the precision matrix is bounded, its convergence rate under spectral norm is $O_p(sM_d^2\sqrt{(\log d)/n})$, which is slower. Moreover, similar to the graphical Dantzig selector, CLIME requires tuning of the optimal regularization parameter.

The TIGER method from [19] achieves the minimax optimal rate for the same larger matrix class. However, a fine-tuning procedure is still needed since TIGER is only tuning-insensitive in finite samples. Moreover, TIGER does not have a second-stage enhancement with oracle property and faster convergence rate.

The SCIO estimator [52] requires the irrepresentable condition, which is a very strong condition to achieve the same minimax optimal rate and the support recovery. In contrast, the graphical Rank Lasso only requires the minimal signal strength condition. It is still unclear if SCIO can achive 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 spectral norm [11]. [15] proved the SCAD-penalized maximum likelihood estimator achieves the oracle property. Even so, they can not the improve convergence rate, while our second-stage enhancement with nonconvex penalty can achieve a significantly faster rate.

Under similar conditions as our second-stage estimator, [50] estimate each column of Ω using a non-convex penalty, which achieves the oracle property and a faster convergence rate in spectral norm. However, their method reuquire heavy-tuning while our second-stage estimator only needs light tuning with high-dimensional BIC, whose consistency result is proved in [23].

3.4 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` [53]. The CLIME estimator is computed using R package `flare` [54]; the TIGER and GLasso estimators are computed using R package `huge` [55].

3.4.1 General Comparison

We consider 3 types of graph: random, band, cluster as described in [19] to determine the sparsity pattern in the final Gaussian graphical models. Specifically,

1. Erdős–Rényi random graph: Each pair of nodes are connected by an edge with probability 0.05 independently.

2. Band graph (with bandwidth 3): Two nodes i, j are connected 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 an random graph with edge probability 0.2.

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

$$\Omega = D[A + (|\Lambda_{\min}(A)| + 0.2) \cdot I_d]D, \tag{3.7}$$

where $D \in \mathbb{R}^{d\times d}$ is a diagonal matrix with $D_{jj} = 1$ for $j = 1, ..., d/2$ and $D_{jj} = 1.5$ for $j = d/2 + 1, ..., d$. Finally, n i.i.d. observations are sampled from the multivari-

ate Gaussian distribution $N_d(0, \Omega^{-1})$. 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 $\hat{\Omega}$ using the training data, and the optimal tuning parameter is chosen so that it minimizes the negative log-likelihood loss $L(\hat{\Omega}) = \text{trace}(\hat{\Omega}\hat{\Sigma}) - \log\det(\hat{\Omega})$ on the validation set, where $\hat{\Sigma}$ is the sample covariance matrix. We use the regularization parameter value $\lambda = \sqrt{(\log d)/n}$ for TIGER as suggested in [19] instead of doing a fine tuning. While gRankLasso is completely tuning-free, gRankMCP requires some light tuning. We use the high-dimensional Bayesian information criteria (HBIC) as suggested in [23] to select the best value of η in (3.5).

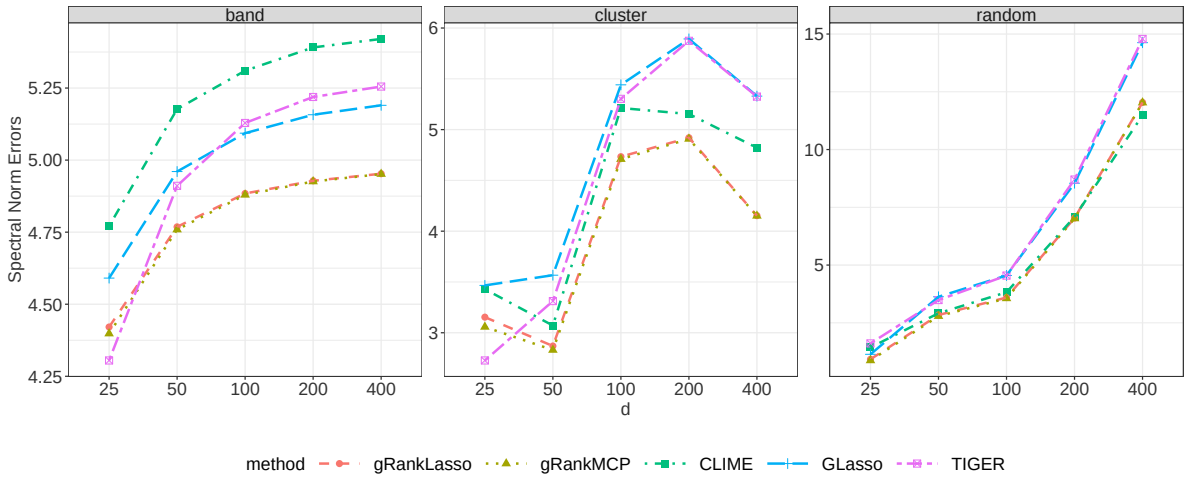


Figure 3.1: Comparison of estimation performance (in terms of spectral norm $\|\hat{\Omega} - \Omega\|_2$, averaged over 50 replications) of various methods in the three graph models with $d \in \{25, 50, 100, 200, 400\}$.

In Figure 3.1 we present the estimation error (averaged over 50 replications) under spectral norm $\|\hat{\Omega} - \Omega\|_2$ for the three graph models. Evidently, gRankLasso and gRankMCP both outperform other methods in all three graph types. The performance advantage is especially pronounced in the high-dimensional setting where $d = 400$. The

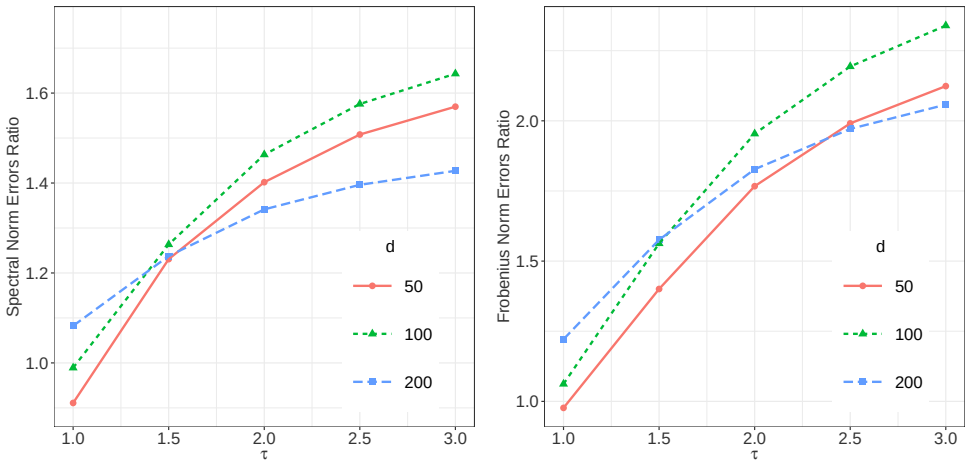


Figure 3.2: Spectral norm ratio $\|\hat{\Omega}_{\text{gRankLasso}} - \Omega\|_2^{-1}\|\hat{\Omega}_{\text{TIGER}} - \Omega\|_2$ and Frobenius norm ratio $\|\hat{\Omega}_{\text{gRankLasso}} - \Omega\|_F^{-1}\|\hat{\Omega}_{\text{TIGER}} - \Omega\|_F$ (averaged over 50 replications) in the random graph model.

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 of the completely tuning free property of gRankLasso and the asymptotic 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 3.4.2, we further investigate the performance difference between gRankLasso and TIGER in various settings.

3.4.2 Sensitivity of tuning-free methods

In this section, we further illustrate the benefit 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 D in (3.7). Specifically, we set $D_{jj} = 1$ for $j = 1, ..., d/2$ and $D_{jj} = \tau$

for $j = d/2 + 1, ..., d$ 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 ($D_{jj} = 1$ versus $D_{jj} = \tau$). Thus the value τ gives a simplified characterization of the difference in the optimal level of regularization in estimating different columns of Ω. 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.

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 $X \sim N_d(0, \Omega^{-1})$. Figure 3.2 shows the ratio (averaged over 50 replications) of the Frobenius norms $\|\Omega_{\text{gRankLasso}} - \Omega\|_F^{-1} \|\Omega_{\text{TIGER}} - \Omega\|_F$. 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 asymptotic tuning-free property might fall short. We also note that this pattern holds for all three values of d, which covers the whole spectrum of the $n > d$, $n = d$, and $n < d$ settings.

3.4.3 Benefit of the second-stage enhancement

It is almost impossible to identify the difference in performance between gRankLasso and gRankMCP in Figure 3.1. To better understand the benefit of the second-stage enhancement in practice, we consider a more challenging setting with a denser true precision matrix: $\Omega_{ij} = 0.6^{|i-j|}$, for $1 \leq i, j \leq d$, which is also considered in [48]. We then generate $n = 100$ observations with dimension $d \in \{25, 50, 100, 200, 400\}$ from $N_d(0, \Omega^{-1})$.

Figure 3.3 shows the spectral norm error (averaged over 50 replications) $\|\hat{\Omega} - \Omega\|_2$ for gRankLasso and gRankMCP with 5 values of d. Unsurprisingly, in a more challenging sce-

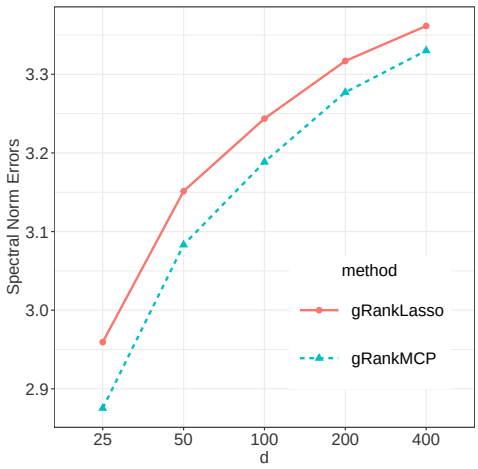


Figure 3.3: Comparisons between gRankLasso and gRankMCP on a decay graph model in terms of spectral norm $\|\hat{\Omega} - \Omega\|_2$ (averaged over 50 replications).

nario, the efficiency gain of gRankMCP becomes more obvious. However, as mentioned, this statistical efficiency gain from gRankMCP 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 computation resources.

3.4.4 Heavy-tailed setting

Finally, as mentioned in Section 3.2.1, one potential bonus 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 set up of the random graph model as in Section 3.4.1. Instead of the Gaussian distribution, we generate observations from a multivariate t-distribution $t_\nu(0, \Omega^{-1})$ of dimension $d \in \{25, 50, 100, 200, 400\}$ with degrees of freedom $\nu \in \{3, 5, 10\}$.

Figure 3.4 shows the estimation error in Frobenius norm (averaged over the 50 replications). Across different settings, gRankLasso and gRankMCP still achieve the most

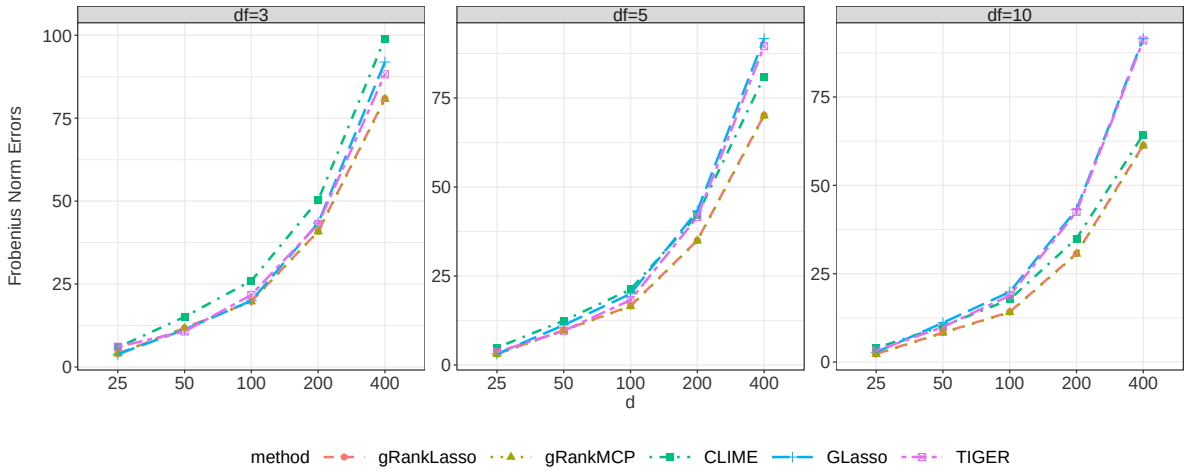


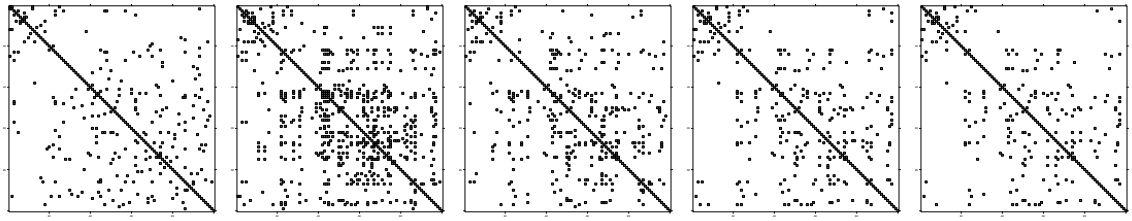
Figure 3.4: Comparison of estimation performance (in terms of Frobenius norm $\|\hat{\Omega} - \Omega\|_F$, averaged over 50 replications) of various methods in the three graph models when data are drawn from a multivariate t-distribution with degrees of freedom $\nu \in \{3, 5, 10\}$.

favorable performance among all competing methods. In the most extreme case when $\nu = 3$, all methods suffer while gRankLasso and gRankMCP clearly win 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 (3.3) [23].

3.5 Data example: Human gene network

We apply our proposed methods to reconstruct the interaction network from human gene expression data in R package `BDgraph` [56], which was previously studied by [57, 58, 19]. 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 they were selected from

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



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

Figure 3.5: The sparsity pattern of estimated graphs from TIGER, gRankLasso, and gRankMCP on human gene network data. The plot 3.5a shows 124 significant edges whose estimated posterior probabilities are greater than 0.6, and is considered to be the comparison baseline.

Method	True	Total	Precision
GLasso	77	301	0.255
TIGER	66	179	0.368
gRankLasso	62	136	0.456
gRankMCP	56	108	0.518

Table 3.1: Comparison of gRankLasso, gRankMCP, and TIGER on the human gene expression data in terms of the number of True recovery, Total recovery, and Precision.

earlier study from [57] and subsequent study from [58].

The goal of this analysis is to learn the significant associations among the 100 chosen traits. As shown in [56], all chosen traits are continuous but not Gaussian, so the joint normality assumption is hardly satisfied. For the sake of comparison, we first use the Bayesian approach from [58] to estimate posterior probabilities of all possible edges, which leads to 124 significant edges (interaction with 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 regularization parameter value set as $\lambda = \sqrt{(\log d)/n}$), gRankLasso, and gRankMCP.

Table 3.5 shows the Precision, which is the ratio between True and Total, where True is the number of recovered edges that are significant (in the sense of recovery by [58]),

and Total is the total number of recovered edges from each method. The sparsity pattern of recovered graphs are showed in Figure 3.5. 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.

3.6 Discussion

We presented gRankLasso, a completely tuning-free method in estimating Gaussian graphical models. This estimator can be efficiently computed using linear programming and requires no tuning in finite samples. Minimax estimation error rates are derived. Our proposed method is accompanied with a second-stage enhancement to reduce estimation bias to improve statistical efficiency with strong oracle properties. Favorable finite sample performance of our methods are illustrated through extensive numerical simulations and a real data application.

Appendix A

Appendix for Chapter 2

A.1 Robust Wasserstein Profile Inference for Neighborhood Selection

Let $X = (X_1, X_2, ..., X_d) \in \mathbb{R}^{n \times d}$ be the data matrix with X_j be the j-th column of X, and X_{-j} be the matrix with j-th column of X removed. Recall that if X follows multivariate Gaussian distribution, then the conditional of X_j given X_{-j} is also Gaussian, and it can be described by the following linear model:

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

where $\beta^{(j)} \in \mathbb{R}^{d-1}$ and $\epsilon_j \sim N\left(0, \sigma_j^2\right)$. Therefore, conditional dependency is reflected on the coefficients vector $\beta^{(j)}$ since $\Omega_{-j,j} = -\sigma_j^{-2}\beta^{(j)}$. Suppose that Ω is sparse, then $\beta^{(j)}$ is also sparse with only few non-zero coefficients, [20] suggested using the square-root

Lasso [35] to estimate $\beta^{(j)}$ and Ω using neighborhood selection [16] as follows:

$$\hat{\beta}^{(j)} = \underset{\beta \in \mathbb{R}^{d-1}}{\operatorname{argmin}} \|X_j - X_{-j}\beta^{(j)}\|_2 + \lambda_j \|\beta\|_1, \tag{A.2}$$

$$\hat{\sigma}_j^2 = n^{-1}\|X_{*,j} - X_{*,-j}\hat{\beta}^{(j)}\|_2^2, \tag{A.3}$$

$$\hat{\Omega}_{jj} = 1/\hat{\sigma}_j^2, \qquad \hat{\Omega}_{-j,j} = -\hat{\Omega}_{jj}\hat{\beta}^{(j)}. \tag{A.4}$$

As shown in [22], we can use a stochastic upper bound for the limit of the RWP function to simulate the regularization parameter of the square-root Lasso:

Algorithm 3 RWP criterion for square-root Lasso regularization with Gaussian errors

Set parameters $\alpha \in (0,1)$, $m \in \mathbb{N}$, $\hat{\Sigma}$
Sample $Z_1, \ldots, Z_m$ independently from $\mathcal{N}(0, \hat{\Sigma})$.
for j = 1,. . . ,d **do**
  for k = 1,. . . ,m **do**
    $a_{jk} \leftarrow \frac{\pi}{\pi-2} \max_{l \neq j} |Z_{kl}|^2$.
  end for
  $\eta^j_{1-\alpha} \leftarrow 1-\alpha$ quantile of $\{a_{j1}, ..., a_{jm}\}$.
  $\lambda_j \leftarrow \sqrt{\frac{\eta^j_{1-\alpha}}{n}}$.
end for
Return $\lambda_1, ..., \lambda_d$.

Algorithm 3 assumes that additive error ϵ_j follows a centered normal distribution, which is the case for Gaussian graphical model. Thus, we can directly use λ_j from algorithm 3 for (A.2). In high-dimensional setting when $n < d$, under standard regularity conditions, λ_j might be chosen as

$$\lambda_j = \frac{\pi}{\pi - 2} \cdot \frac{\Phi^{-1}(1 - \alpha/2d)}{\sqrt{n}},$$

where $\Phi^{-1}(\cdot)$ is the inverse cumulative distribution function of the standard normal distribution and $\alpha \in (0,1)$. Similar to algorithm 3, this formulation is derived from the stochastic upper bound for the RWP limit. As mentioned in [22], this RWP-based ap-

proach for choosing λ_j in high-dimensional settings is in agreement with the theoretical choice that satisfies the subgradient condition of the square-root Lasso ([38, 35, 39]), so asymptotic bounds on estimation errors also hold. Consequently, one can follow framework from [17] to show convergence rate for Ω obtained from (A.4).

Recently, the confidence region and asymptotic normality for DRO estimators have been established in [59]. However, it is unclear if one can provide a Type-I error control in graph selection similar to our result in Theorem 2.2.1.

Appendix B

Appendix for Chapter 3

B.1 Preliminaries

We follow the framework in [17, 19] to prove convergence results. First, we provide some preliminary assumptions. For a constant $c > 1$, define $\bar{c} = \frac{c+1}{c-1}$ and consider the cone set

$$\Gamma^d = \{\gamma \in \mathbb{R}^d : \|\gamma_{S^c}\|_1 \leq \bar{c}\|\gamma_S\|_1, S \subset \{1, 2, ..., d\}, \|S\|_0 \leq s\}.$$

Let S_j be the support of the j-th column of Ω, recall the s-sparse matrix class

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

We assume the following conditions are satisfied:

(C1) $\Omega \in \mathcal{M}(s, M_d)$,

(C2) $s^2 \log d = o(n)$.

For a nonconvex penalty function, we assume 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 $t > a_2\eta$.

B.2 Technical lemmas

Lemma B.2.1 *Let $Y \sim \chi^2_d$. We have*

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

Lemma B.2.2 *Let $\epsilon^{(j)} \in \mathbb{R}^n$ such that $\epsilon^{(j)} \sim N(0, \sigma_j^2 I_n)$. Then*

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

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

Lemmas B.2.1 and B.2.2 are taken from [60, 61, 19].

Lemma B.2.3 ℓ_1 Restricted Eigenvalue condition: *Let $\hat{\Sigma} = X^T X/n$. Suppose $s\log(d) = o(n)$, then there exist constants c_1, c_2 such that*

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

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

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

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

and

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

Consider a random matrix $X \in \mathbb{R}^{n \times d}$, *in which each row is drawn i.i.d. from a* $N(0, \Sigma)$. *From [62], we have there exists two positive constants* c_1, c_2 *such that*

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

By definition, we have

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

It follows that,

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

Thus

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

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

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

■

Lemma B.2.4 Prediction error bound of first-stage estimator: *Let $\hat{\beta}^{(j)}$ be the Rank Lasso estimator of $\beta^{(j)}$. We have*

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

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

Proof: We have

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

then,

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

Thus the ℓ_1-RE condition holds for all Rank Lasso subproblem from each column. From lemma 2 of [23], using a simulated λ_j from 2, we have $\hat{\beta}^{(j)} - \beta^{(j)} \in \Gamma^{d-1}$. Then from

Theorem 1 of [23] and lemma 9 of [63], we have

$$\begin{aligned}\max_{1\le j\le d}\frac{\|X_{*,-j}(\hat{\beta}^{(j)}-\beta^{(j)})\|_2}{\sqrt{n}} &\le \max_{1\le j\le d}\|X_{*,-j}^T X_{*,-j}/n\|_2\|\hat{\beta}^{(j)}-\beta^{(j)}\|_2\\ &\le \max_{1\le j\le d}\|X^T X/n\|_2\|\hat{\beta}^{(j)}-\beta^{(j)}\|_2\\ &\le 9\Lambda_{\max}\frac{C}{\Lambda_{\min}}\sqrt{\frac{s\log d}{n}}\\ &= \frac{9\xi^2 C}{\sqrt{n}}\sqrt{s\log d}.\end{aligned}$$

■

Lemma B.2.5 Prediction error bound for oracle estimator*: Let $\check{\beta}^{(j)}$ be the oracle estimator of $\beta^{(j)}$. Then*

$$\max_{1\le j\le d}\|X_{*,-j}(\check{\beta}^{(j)}-\beta^{(j)})\|_2 \le C\sqrt{s},$$

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

Proof: From Lemma 3 of [23], we have $\|\check{\beta}^{(j)}-\beta^{(j)}\|_2 = O_P(\sqrt{s/n})$. It follows that

$$\begin{aligned}\max_{1\le j\le d}\frac{\|X_{*,-j}(\check{\beta}^{(j)}-\beta^{(j)})\|_2}{\sqrt{n}} &\le \max_{1\le j\le d}\|X_{*,-j}^T X_{*,-j}/n\|_2\|\check{\beta}^{(j)}-\beta^{(j)}\|_2\\ &\le \max_{1\le j\le d}\|X^T X/n\|_2\|\check{\beta}^{(j)}-\beta^{(j)}\|_2\\ &\le 9\Lambda_{\max}C\sqrt{\frac{s}{n}}\\ &\le 9\xi C\sqrt{\frac{s}{n}}.\end{aligned}$$

■

B.3 Proofs of main Lemmas

Lemma B.3.1 *Analyzing the diagonal elements of first-stage estimator*

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

Proof: We have

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

From Lemmas B.2.1, B.2.2, B.2.4, we have

$$\begin{aligned}|\frac{\|\epsilon^{(j)}\|_2^2}{n}-\sigma_j^2|&\leq 3.5\sigma_j^2\sqrt{(\log d)/n}, \\ \|X_{*,-j}(\beta^{(j)}-\hat{\beta}^{(j)})\|_2&\leq C\sqrt{s\log d}.\end{aligned}$$

From standard Gaussian tail bounds in [64], we also have for all $\delta>0$

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

It follows that

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

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

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

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

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

The rest of the proof follow [19]. Since $\Omega_{jj} = 1/\sigma_j^2$*, we have*

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

This implies that

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

Then, for large enough n

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

we have

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

Thus

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

■

Lemma B.3.2 Analyzing the off-diagonal elements in ℓ_1-norm error of first-stage estimator

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

Proof: Recall that

$$\Omega_{-j,j} = -\Omega_{jj}\beta^{(j)},$$

Then

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

From lemmas B.2.4, B.3.1, we have

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

Thus

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

■

Lemma B.3.3 *Analyzing the diagonal elements of oracle estimator*

$$\max_{1\leq j\leq d}|\check{\Omega}_{jj} - \Omega_{jj}| \leq C\|\Omega\|_2\sqrt{\frac{\log d}{n}}$$

Proof: Follow similar arguments from B.3.1, we have

$$|(\check{\Omega}_{jj})^{-1} - (\Omega_{jj})^{-1}| \leq |\frac{\|\epsilon^{(j)}\|_2^2}{n} - \sigma_j^2| + \frac{\|X_{*,-j}(\beta^{(j)} - \check{\beta}^{(j)})\|_2^2}{n} + 2\frac{|(\check{\beta}^{(j)} - \beta^{(j)})^T X_{*,-j}^T \epsilon^{(j)}|}{n}.$$

From Lemmas B.2.1, B.2.2, B.2.5, B.3.1, we have for all $\delta > 0$

$$|\frac{\|\epsilon^{(j)}\|_2^2}{n} - \sigma_j^2| \leq 3.5\sigma_j^2\sqrt{(\log d)/n},$$
$$\|X_{*,-j}(\beta^{(j)} - \check{\beta}^{(j)})\|_2 \leq C\sqrt{s},$$
$$2\frac{|(\check{\beta}^{(j)} - \beta^{(j)})^T X_{*,-j}^T \epsilon^{(j)}|}{n} \leq 4\sqrt{2}(1+\bar{c})C\sigma_j s\sqrt{\frac{1}{n}}\left(\sqrt{\frac{\log d}{n}}\right).$$

Therefore,

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

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

$$|(\check{\Omega}_{jj})^{-1} - (\Omega_{jj})^{-1}| \leq C\sigma_j^2\sqrt{\frac{\log d}{n}}.$$

The rest of the proof follow B.3.1. We have

$$\max_{1\leq j\leq d}|\check{\Omega}_{jj} - \Omega_{jj}| \leq C\max_{1\leq j\leq d}\Omega_{jj}\sqrt{\frac{\log d}{n}} \leq C\|\Omega\|_2\sqrt{\frac{\log d}{n}}.$$

■

Lemma B.3.4 Analyzing the off-diagonal elements in ℓ_1-norm error of oracle estimator

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

Proof: Follow similar arguments of lemma B.3.2, we have

$$\|\check{\Omega}_{-j,j} - \Omega_{-j,j}\|_1 \leq |\check{\Omega}_{jj}|\|\check{\beta}^{(j)} - \beta^{(j)}\|_1 + |\frac{\check{\Omega}_{jj}}{\Omega_{jj}} - 1|\|\Omega_{-j,j}\|_1$$

From lemmas B.2.5, B.3.3, we have

$$\check{\Omega}_{jj} \leq \left(1 + C\sqrt{\frac{\log d}{n}}\right)\Omega_{jj} \leq 2\|\Omega\|_2,$$
$$\|\check{\beta}^{(j)} - \beta^{(j)}\|_1 \leq (1+\bar{c})\sqrt{s}\|\check{\beta}^{(j)} - \beta^{(j)}\|_2 \leq C_1(1+\bar{c})\frac{s}{\sqrt{n}},$$
$$|\frac{\check{\Omega}_{jj}}{\Omega_{jj}} - 1| \leq C_2\sqrt{\frac{\log d}{n}}.$$

Thus

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

■

B.4 Proofs of main Theorems

Theorem 3.3.1

Proof: The proof is identical to [19]. From lemmas B.3.1 and B.3.2, we have

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

■

Theorem 3.3.3

Proof: Let $\tilde{\beta}^{(j)}$ be the second-stage estimator of $\beta^{(j)}$, $\check{\beta}^{(j)}$ be the oracle estimator of $\beta^{(j)}$. From Theorem 2 of [23], we have for α from algorithm 2

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

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

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

Consequently, we get the strong oracle property through union bound.

From lemmas B.3.3 and B.3.4, we have

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

■

Bibliography

[1] S. L. Lauritzen, *Graphical Models*. Oxford University Press, 1996.

[2] M. Drton and M. H. Maathuis, *Structure learning in graphical modeling*, *Annual Review of Statistics and Its Application* **4** (2017) 365–393.

[3] S. M. Smith, K. L. Miller, G. S. Khorshidi, M. A. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich, *Network modelling methods for FMRI*, *NeuroImage* **54** (2011), no. 2 875–891.

[4] O. Stegle, S. A. Teichmann, and J. C. Marioni, *Computational and analytical challenges in single-cell transcriptomics*, *Nature Reviews Genetics* **16** (2015), no. 3 133–145.

[5] S. Na, M. Kolar, and O. Koyejo, *Estimating differential latent variable graphical models with applications to brain connectivity*, *Biometrika* **108** (2021), no. 2 425–442.

[6] A. P. Dempster, *Covariance selection*, *Biometrics* (1972) 157–175.

[7] M. Yuan and Y. Lin, *Model selection and estimation in the gaussian graphical model*, *Biometrika* **94** (03, 2007) 19–35, [https://academic.oup.com/biomet/article-pdf/94/1/19/617853/asm018.pdf].

[8] J. Friedman, T. Hastie, and R. Tibshirani, *Sparse inverse covariance estimation with the graphical lasso*, *Biostatistics* **9** (12, 2007) 432–441.

[9] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, *Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data*, *J. Mach. Learn. Res.* **9** (2008) 485–516.

[10] A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu, *Sparse permutation invariant covariance estimation*, *Electron. J. Statist.* **2** (2008) 494–515.

[11] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu, *High-dimensional covariance estimation by minimizing $\ell 1$-penalized log-determinant divergence*, *Electronic Journal of Statistics* **5** (2011), no. none 935 – 980.

[12] R. Mazumder and T. Hastie, *Exact covariance thresholding into connected components for large-scale graphical lasso*, *Journal of Machine Learning Research* **13** (2012), no. 27 781–794.

[13] G. Yu and J. Bien, *Learning local dependence in ordered data*, *The Journal of Machine Learning Research* **18** (2017), no. 1 1354–1413.

[14] C. Lam and J. Fan, *Sparsistency and rates of convergence in large covariance matrix estimation*, *The Annals of Statistics* **37** (2009), no. 6B 4254 – 4278.

[15] J. Fan, L. Xue, and H. Zou, *Strong oracle optimality of folded concave penalized estimation*, *The Annals of Statistics* **42** (2014), no. 3 819 – 849.

[16] N. Meinshausen and P. Bühlmann, *High-dimensional graphs and variable selection with the lasso*, *The Annals of Statistics* **34** (2006), no. 3 1436 – 1462.

[17] M. Yuan, *High dimensional inverse covariance matrix estimation via linear programming*, *Journal of Machine Learning Research* **11** (2010), no. 79 2261–2286.

[18] T. Sun and C.-H. Zhang, *Sparse matrix inversion with scaled lasso*, *The Journal of Machine Learning Research* **14** (2013), no. 1 3385–3418.

[19] H. Liu and L. Wang, *TIGER: A tuning-insensitive approach for optimally estimating Gaussian graphical models*, *Electronic Journal of Statistics* **11** (2017), no. 1 241 – 294.

[20] J. Janková and S. van de Geer, *Inference in high-dimensional graphical models*, *arXiv preprint arXiv:1801.08512* (2018).

[21] P. Cisneros-Velarde, A. Petersen, and S.-Y. Oh, *Distributionally robust formulation and model selection for the graphical lasso*, in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* (S. Chiappa and R. Calandra, eds.), vol. 108 of *Proceedings of Machine Learning Research*, pp. 756–765, PMLR, 26–28 Aug, 2020.

[22] J. Blanchet, Y. Kang, and K. Murthy, *Robust wasserstein profile inference and applications to machine learning*, *Journal of Applied Probability* **56** (2019), no. 3 830–857.

[23] L. Wang, B. Peng, J. Bradic, R. Li, and Y. Wu, *A tuning-free robust and efficient approach to high-dimensional regression*, *Journal of the American Statistical Association* **115** (2020), no. 532 1700–1714, [https://doi.org/10.1080/01621459.2020.1840989].

[24] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, *Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning*, in *Operations Research & Management Science in the Age of Analytics* (S. Netessine, D. Shier, and H. J. Greenberg, eds.), pp. 130–166. INFORMS, Oct., 2019.

[25] M. Yuan and Y. Lin, *Model selection and estimation in the Gaussian graphical model*, *Biometrika* **94** (2007), no. 1 19–35.

[26] F. H. Clarke, Y. S. Ledyaev, R. J. Stern, and P. R. Wolenski, *Nonsmooth Analysis and Control Theory*. Springer-Verlag New York, 1998.

[27] M. Drton and M. D. Perlman, *Multiple Testing and Error Control in Gaussian Graphical Model Selection*, *Statistical Science* **22** (2007), no. 3 430 – 449.

[28] J. Peng, P. Wang, N. Zhou, and J. Zhu, *Partial correlation estimation by joint sparse regression models*, *Journal of the American Statistical Association* **104** (jun, 2009) 735–746.

[29] D. Marbach, J. Costello, R. Küffner, N. Vega, R. Prill, D. Camacho, K. Allison, A. Aderhold, R. Bonneau, Y. Chen, J. Collins, F. Cordero, M. Crane, F. Dondelinger, M. Drton, R. Esposito, R. Foygel, A. de la Fuente, J. Gertheiss, and R. Zimmer, *Wisdom of crowds for robust gene network inference*, *Nature Methods* **9** (07, 2012) 796–804.

[30] K. M. Tan, P. London, K. Mohan, S.-I. Lee, M. Fazel, and D. Witten, *Learning graphical models with hubs*, *Journal of Machine Learning Research* **15** (2014), no. 95 3297–3331.

[31] R. Foygel and M. Drton, *Extended bayesian information criteria for gaussian graphical models*, in *Advances in Neural Information Processing Systems* (J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds.), vol. 23, Curran Associates, Inc., 2010.

[32] R. Tibshirani, *Regression shrinkage and selection via the lasso*, *Journal of the Royal Statistical Society: Series B (Methodological)* **58** (1996), no. 1 267–288.

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

[34] T. Sun and C.-H. Zhang, *Scaled sparse linear regression*, *Biometrika* **99** (2012), no. 4 879–898.

[35] A. Belloni, V. Chernozhukov, and L. Wang, *Square-root lasso: pivotal recovery of sparse signals via conic programming*, *Biometrika* **98** (2011), no. 4 791–806.

[36] L. A. Jaeckel, *Estimating regression coefficients by minimizing the dispersion of the residuals*, *The Annals of Mathematical Statistics* (1972) 1449–1458.

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

[38] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, *Simultaneous analysis of lasso and dantzig selector*, *The Annals of statistics* **37** (2009), no. 4 1705–1732.

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

[40] L. Wang, *The l1 penalized lad estimator for high dimensional linear regression*, *Journal of Multivariate Analysis* **120** (2013) 135–151.

[41] J. Lederer and C. Müller, *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*, vol. 29, 2015.

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

[43] G. Yu and J. Bien, *Estimating the error variance in a high-dimensional linear model*, *Biometrika* **106** (2019), no. 3 533–546.

[44] R. Tibshirani, *Regression shrinkage and selection via the lasso: a retrospective*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73** (2011), no. 3 273–282.

[45] J. Fan and R. Li, *Variable selection via nonconcave penalized likelihood and its oracle properties*, *Journal of the American Statistical Association* **96** (2001), no. 456 1348–1360, [https://doi.org/10.1198/016214501753382273].

[46] C.-H. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, *The Annals of Statistics* **38** (2010), no. 2 894 – 942.

[47] H. Zou and R. Li, *One-step sparse estimates in nonconcave penalized likelihood models*, *Annals of statistics* **36** (2008), no. 4 1509.

[48] 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** (jun, 2011) 594–607.

[49] T. T. Cai, W. Liu, and H. H. Zhou, *Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation*, *The Annals of Statistics* **44** (2016), no. 2 455 – 488.

[50] L. Wang, X. Ren, and Q. Gu, *Precision matrix estimation in high dimensional gaussian graphical models with faster rates*, in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics* (A. Gretton and C. C. Robert, eds.), vol. 51 of *Proceedings of Machine Learning Research*, (Cadiz, Spain), pp. 177–185, PMLR, 09–11 May, 2016.

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

[52] W. Liu and X. Luo, *Fast and adaptive sparse precision matrix estimation in high dimensions*, *Journal of Multivariate Analysis* **135** (2015) 153–162.

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

[54] X. Li, T. Zhao, X. Yuan, and H. Liu, *The flare package for high dimensional linear regression and precision matrix estimation in r*, *Journal of Machine Learning Research* **16** (2015), no. 18 553–557.

[55] T. Zhao, H. Liu, K. Roeder, J. Lafferty, and L. Wasserman, *The huge package for high-dimensional undirected graph estimation in r*, *Journal of Machine Learning Research* **13** (2012), no. 37 1059–1062.

[56] R. Mohammadi and E. C. Wit, *Bdgraph: An r package for bayesian structure learning in graphical models*, *Journal of Statistical Software* **89** (2019), no. 3 1–30.

[57] A. Bhadra and B. Mallick, *Joint high-dimensional bayesian variable and covariance selection with an application to eqtl analysis*, *Biometrics* **69** (04, 2013).

[58] A. Mohammadi and E. C. Wit, *Bayesian Structure Learning in Sparse Gaussian Graphical Models*, *Bayesian Analysis* **10** (2015), no. 1 109 – 138.

[59] J. Blanchet, K. Murthy, and N. Si, *Confidence regions in wasserstein distributionally robust estimation*, *Biometrika* **109** (2022), no. 2 295–315.

[60] I. M. Johnstone, *Chi-square oracle inequalities*, *Lecture Notes-Monograph Series* (2001) 399–418.

[61] B. Laurent and P. Massart, *Adaptive estimation of a quadratic functional by model selection*, *The Annals of Statistics* **28** (2000), no. 5 1302 – 1338.

[62] G. Raskutti, M. J. Wainwright, and B. Yu, *Restricted eigenvalue properties for correlated gaussian designs*, *Journal of Machine Learning Research* **11** (2010), no. 78 2241–2259.

[63] M. J. Wainwright, *Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_l -constrained quadratic programming (lasso)*, *IEEE transactions on information theory* **55** (2009), no. 5 2183–2202.

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