Statistica Sinica **34** (2024), 2043-2064
doi:https://doi.org/10.5705/ss.202021.0273

A SIMPLE METHOD FOR ESTIMATING GAUSSIAN GRAPHICAL MODELS

Yiyi Yin, Yang Song and Hui Zou*

University of Minnesota

Abstract: The penalized likelihood estimator is the state-of-the-art method for estimating a Gaussian graphical model, because it delivers a symmetric graph and is efficient to compute, owing to the graphical lasso implementation. However, the estimator requires a stringent irrepresentability condition in order to achieve consistent recovery of the underlying graph. Another popular method, neighborhood selection, does not offer a symmetric solution by itself, and also requires a set of irrepresentability conditions for exact recovery. In this paper, we propose a new method, called the simple graph maker, for estimating an underlying graph. The simple graph maker produces a symmetric estimator by using a simple ℓ_1-penalized quadratic problem, which is easily computed by coordinate descent. Furthermore, it is shown to recover the underlying graph with overwhelming probability, without assuming additional structure conditions on the variables. The rates of convergence under various matrix norms are also established. The new method is shown to exhibit excellent performance on simulated and real data.

Key words and phrases: Coordinate descent, exact recovery, Gaussian graphical model, graphical Lasso, irrepresentable conditions, sparsity.

1. Introduction

In this study, we examine the problem of constructing a Gaussian graphical model from n independent and identically distributed observations (i.i.d.) from a multivariate Gaussian distribution. Suppose that $\mathbf{X} = (X_1, X_2, \ldots, X_p)^T$ follows a multivariate Gaussian distribution $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}^*)$. Let $\boldsymbol{\Theta}^* = (\theta^*_{ij})$ and $\boldsymbol{\Sigma}^* = (\boldsymbol{\Theta}^*)^{-1}$ denote the precision matrix and the covariance matrix, respectively. It is known that the (i, j) element of $\boldsymbol{\Theta}^*$ is zero if and only if variables X_i and X_j are conditional independent, given all the other variables (Lauritzen (1996)). Thus, data analysts often use the sparsity pattern of an estimated sparse precision matrix to construct a Gaussian graphical model that describes the dependence relationships between variables. As a result, the problem of estimating a large sparse precision matrix has received increased attention in the past decade, for a comprehensive review, see Chapter 9 of Fan et al. (2020), and the references therein. Currently, the two most popular methods are neighborhood selection (Meinshausen and Bühlmann (2006)) and the penalized likelihood estimator (i.e.,

*Corresponding author.

the graphical lasso)(Yuan and Lin (2007); Rothman et al. (2008); Friedman, Hastie and Tibshirani (2008); Ravikumar et al. (2011).)

In this paper, we propose a new method for estimating an underlying graph. In order to motivate our proposal, we first discuss the strengths and weaknesses of the two most popular existing methods. Neighborhood selection was proposed prior to the penalized likelihood estimator. It is a column-wise recovery method, in the sense that it estimates the columns of $\mathbf{\Theta}^*$ one by one. As a result, the matrix estimation problem is cast into p separate vector estimation problems, making the computation easy by running a lasso linear regression. However, the solution is usually not symmetric, and hence a post-processing step is necessary to make the estimator symmetric. This was the major motivation for researchers to study the penalized likelihood estimator. The graphical lasso delivers a sparse symmetric precision matrix estimator by following the penalized likelihood principle. In addition, the graphical lasso can be solved efficiently (Friedman, Hastie and Tibshirani (2008)), making it the first choice for many users when a sparse precision matrix estimator is needed. Theoretically, neighborhood selection requires an irrepresentability condition (Zhao and Yu (2006); Zou (2006)) in order to estimate each column of $\mathbf{\Theta}^*$. A similar matrix-version of the irrepresentability condition is required for the graphical lasso (Ravikumar et al. (2011)). Because these conditions are so stringent, theoretical support for the two methods is not strong. Note that these issues cannot be removed by replacing the lasso penalty with the adaptive lasso penalty or the concave penalty, because the likelihood function or the "loss" function in neighborhood selection is a key factor in creating these theoretical obstacles. See the discussion in Section 2.3 for details.

Based on the above discussion, we develop a new method for estimating a sparse precision matrix that has three desirable properties:

1. the proposed method yields a symmetric matrix estimator, as in the case of the graphical lasso;
2. the proposed method is computationally efficient;
3. the theoretical justification for the proposed method does not require the irrepresentability condition or other strong structure conditions.

In other words, the proposed method enjoys the advantages of existing methods, but avoids their major drawbacks.

In Section 2, we present the technical details of the proposed method, which we call the simple graph maker (SGM). The SGM estimator is symmetric and easy to compute. In Section 3, we prove its sparse recovery property and establish its rates of convergence under several common matrix norms. In Section 4, we present a simulation study and real-data examples to demonstrate the performance of the proposed method, and compare it with Glasso, Galasso

(Fan, Feng and Wu (2009)), CLIME (Cai, Liu and Luo (2011)) and Dtrace (Zhang and Zou (2014)) methods. Technical proofs are relegated to the appendix.

2. Methodology

2.1. Notation

Here, we introduce the notation and definitions used throughout this paper. For a vector $\mathbf{v}$, $||\mathbf{v}||_{\max} = \max_i |v_i|$, $||\mathbf{v}||_{\min} = \min_i |v_i|$, and $|\mathbf{v}|_1 = \sum_i |v_i|$. We use $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ to denote the largest and smallest eigenvalues, respectively, of a matrix $\mathbf{A}$. Denote by $\mathrm{tr}(\mathbf{A})$ the trace of a square matrix $\mathbf{A}$. For a real matrix $\mathbf{A} = (a_{ij})$, $||\mathbf{A}||_{\max} = \max_{i,j} |a_{ij}|$, $||\mathbf{A}||_{\min} = \min_{i,j} |a_{ij}|$, $||\mathbf{A}||_1 = \sum_{i,j} |a_{ij}|$, $||\mathbf{A}||_{\ell_\infty} = \max_i \sum_j |a_{ij}|$, $||\mathbf{A}||_{\ell_1} = \max_j \sum_i |a_{ij}|$, $||\mathbf{A}||_F = \sqrt{\sum_{i,j} |a_{ij}|^2}$, and $||\mathbf{A}||_2 = \sqrt{\lambda_{max}(\mathbf{A}^T\mathbf{A})}$. We use $\mathbf{A} \succ 0$ to indicate that $\mathbf{A}$ is a positive-definite matrix. We write $\mathbf{A}_1 \succeq \mathbf{A}_2$ when $\mathbf{A}_1 - \mathbf{A}_2$ is a positive semidefinite matrix. We use $\mathrm{vec}(\mathbf{A})$ to denote the vectorization of $\mathbf{A}$ in the column by column order. Let $\boldsymbol{e}_i$ be the ith column of the p-dimensional identity matrix. We use $\mathbf{A} \circ \mathbf{B}$ to denote the Hadamard product of matrices $\mathbf{A}$ and $\mathbf{B}$. Define $\boldsymbol{\Gamma}(\boldsymbol{\Sigma}) = (1/2)(\boldsymbol{\Sigma} \otimes \mathbf{I} + \mathbf{I} \otimes \boldsymbol{\Sigma})$, where $\otimes$ is the Kronecker product. It is easy to see that $\boldsymbol{\Gamma}(\boldsymbol{\Sigma})$ is positive definite when $\boldsymbol{\Sigma}$ is positive definite. Let $S = \{(i,j) | \theta^*_{ij} \neq 0\}$ denote the support set of $\boldsymbol{\Theta}^*$, and S^c the complement of S. For each j, let $S_j = \{(i,j) | \theta^*_{ij} \neq 0\}$ be the support set of the jth column of $\boldsymbol{\Theta}^*$, and let S_j^c be the complement of S_j. Let $d = \max_j |S_j|$ and $s = |S|$.

2.2. The SGM

Let $\hat{\boldsymbol{\Sigma}}$ be the sample covariance matrix, and define $\tilde{\boldsymbol{\Sigma}} = \hat{\boldsymbol{\Sigma}} + a\sqrt{\log p/n}\mathbf{I}$, where a is a positive constant. If $\hat{\boldsymbol{\Sigma}}$ is positive definite, we can set $a = 0$. The perturbation term $a\sqrt{\log p/n}\mathbf{I}$ is primarily used to improve the numeric stability of the estimator when $\hat{\boldsymbol{\Sigma}}$ has a zero or near zero eigenvalue. The theoretical upper bound on a is given in the next section. In practice, we use a small a, such as $a = 0.05$.

The SGM $\hat{\boldsymbol{\Theta}}^{\mathrm{SGM}}$ is defined as follows:

$$\hat{\boldsymbol{\Theta}}^{\mathrm{SGM}} = \underset{\boldsymbol{\Theta}^T = \boldsymbol{\Theta}}{\operatorname{argmin}} \frac{1}{2}\mathrm{tr}(\boldsymbol{\Theta}^T \tilde{\boldsymbol{\Sigma}} \boldsymbol{\Theta}) - \mathrm{tr}(\boldsymbol{\Theta}) + \lambda_1 ||\mathbf{W} \circ \boldsymbol{\Theta}||_1, \tag{2.1}$$

where λ_1 is a penalization parameter, and the adaptive weight matrix $\mathbf{W} = (w_{ij})$ is computed from

$$w_{ij} = \begin{cases} (\min\{|\hat{\theta}^0_{ij}|, |\hat{\theta}^0_{ji}|\} + u(n,p))^{-1} & \text{for } i \neq j \\ 0 & \text{for } i = j, \end{cases}$$

where $(\hat{\boldsymbol{\Theta}}_0)_{ij} = \hat{\theta}^0_{ij}$ is a pilot estimator of $\boldsymbol{\Theta}^*$, and $u(n,p)$ is a positive-valued

function of n and p. In theory, we can let $u(n,p)=0$ and set the weight $w_{ij}=\infty$ if dividing by zero occurs. When $w_{ij}=\infty$, it automatically implies that $\hat{\theta}_{ij}^{\mathrm{SGM}}=0$. Any $u(n,p)$ below a theoretical upper bound is good, in theory. In practice, we use a small, but positive $u(n,p)$ to avoid zero division, for example, $u(n,p)=(np)^{-2}$.

Given the weight matrix, it is easy to solve the optimization problem in (2.1). In order to handle the symmetry constraint, we parametrize $\mathbf{\Theta}=(\theta_{ij})$ with $\theta_{ij}=\theta_{ji}$. Then we recast the constrained optimization problem as an unconstrained ℓ_1 penalization problem in which the unknowns are θ_{ii}, for $1\le i\le p$, and θ_{ij}, for $j>i$. Note that the objective function in (2.1) is a quadratic function of the unknowns plus the weighted ℓ_1-penalty term. Following Friedman, Hastie and Tibshirani (2010), we use the coordinate descent algorithm and computational tricks, such as active set update and warm start, to solve (2.1) for a grid of λ_1 values.

We now discuss the pilot estimator from which we compute the weight matrix. The primary goal is to ensure that the SGM estimator recovers the true graph with probability going to one as the sample size and the dimension grow together. Our analysis of the SGM estimator reveals a sufficient condition for the weight matrix under which the exact recovery property of the SGM estimator holds. Based on that analysis, we design a pilot estimator $\hat{\mathbf{\Theta}}_0$ as follows:

$$\hat{\mathbf{\Theta}}_0=\operatorname*{argmin}_{\mathbf{\Theta}}\frac{1}{2}\operatorname{tr}(\mathbf{\Theta}^T\tilde{\mathbf{\Sigma}}\mathbf{\Theta})-\operatorname{tr}(\mathbf{\Theta})+\lambda_0||\mathbf{\Theta}||_1, \tag{2.2}$$

where λ_0 is a tuning parameter.

Remark 1. A seemingly natural pilot estimator is

$$\tilde{\mathbf{\Theta}}_0=\operatorname*{argmin}_{\mathbf{\Theta}^T=\mathbf{\Theta}}\frac{1}{2}\operatorname{tr}(\mathbf{\Theta}^T\tilde{\mathbf{\Sigma}}\mathbf{\Theta})-\operatorname{tr}(\mathbf{\Theta})+\lambda_0||\mathbf{\Theta}||_1. \tag{2.3}$$

Although we do not deny the legitimacy of $\tilde{\mathbf{\Theta}}_0$ as a pilot estimator for the SGM estimator, we prefer to use $\hat{\mathbf{\Theta}}_0$, for computational convenience. We can use the coordinate descent algorithm for solving (2.1) to solve (2.3). It turns out that (2.2) is even easier to compute, owing to the removal of the symmetry constraint. Let $\boldsymbol{\theta}_j$ denote the jth column of $\mathbf{\Theta}$. Observe that

$$\frac{1}{2}\operatorname{tr}(\mathbf{\Theta}^T\tilde{\mathbf{\Sigma}}\mathbf{\Theta})-\operatorname{tr}(\mathbf{\Theta})+\lambda_0||\mathbf{\Theta}||_1=\sum_{j=1}^{p}\left(\frac{1}{2}\boldsymbol{\theta}_j^T\tilde{\mathbf{\Sigma}}\boldsymbol{\theta}_j-\theta_{jj}+\lambda_0|\boldsymbol{\theta}_j|_1\right).$$

Therefore, if $\hat{\theta}_j$ is the minimizer of $(1/2)\boldsymbol{\theta}_j^T\tilde{\mathbf{\Sigma}}\boldsymbol{\theta}_j-\theta_{jj}+\lambda_0|\boldsymbol{\theta}_j|_1$, then $\hat{\mathbf{\Theta}}=[\hat{\theta}_1\cdots\hat{\theta}_p]$ is the minimizer of $(1/2)\operatorname{tr}(\mathbf{\Theta}^T\tilde{\mathbf{\Sigma}}\mathbf{\Theta})-\operatorname{tr}(\mathbf{\Theta})+\lambda_0||\mathbf{\Theta}||_1$. Hence, we can solve (2.2) by solving p ℓ_1-penalized quadratic problems in parallel.

The construction of the SGM estimator is traced back to the penalized Dtrace loss estimator (Zhang and Zou (2014)),

$$\min_{\boldsymbol{\Theta} \succeq \epsilon \boldsymbol{I}} L_D(\boldsymbol{\Theta}, \hat{\boldsymbol{\Sigma}}) + \lambda ||\boldsymbol{\Theta}||_1, \tag{2.4}$$

where the loss function $L_D(\boldsymbol{\Theta}, \boldsymbol{\Sigma}) = (1/2)\,\mathrm{tr}(\boldsymbol{\Theta}^T \boldsymbol{\Sigma} \boldsymbol{\Theta}) - \mathrm{tr}(\boldsymbol{\Theta})$ is called the Dtrace loss. Note that the graphical lasso estimator is $\min_{\boldsymbol{\Theta} \succ 0} L_G(\boldsymbol{\Theta}, \hat{\boldsymbol{\Sigma}}) + \lambda ||\boldsymbol{\Theta}||_1$, with $L_G(\boldsymbol{\Theta}, \boldsymbol{\Sigma}) = \mathrm{tr}(\boldsymbol{\Sigma}\boldsymbol{\Theta}) - \mathrm{logdet}(\boldsymbol{\Theta})$. The L_G loss function is essentially the negative log-likelihood function (up to a scale factor and a constant term). The Dtrace loss was originally proposed as a nonlikelihood-based approach to estimate a large precision matrix. However, the penalized Dtrace loss estimator also requires a kind of irrepresentability condition in order to recover the true graph consistently. That motivated us to use an adaptive lasso penalty (Zou (2006)) to replace the lasso penalty in (2.4). Further, if we aim to recover the true graph, we need only have the symmetry constraint, and can be free with the eigenvalue constraint. Thus, we remove this constraint to explore the fact that L_D is a quadratic function of $\boldsymbol{\Theta}$. As a result, we can use the coordinate descent algorithm to compute a solution path of the SGM estimator. If we choose to keep the eigenvalue constraint, the state-of-the-art algorithm for (2.4) with an adaptive lasso penalty is the alternating direction method of multipliers (ADMM) (Boyd et al. (2011)). We need to run the ADMM algorithm for each penalization parameter. Thus, it is computationally much more expensive than the SGM estimator.

Remark 2. We comment on the tuning of the SGM estimator. Suppose that we have a training set and a validation set. Denote by $\hat{\boldsymbol{\Theta}}_0^{tr}(\lambda_0)$ the pilot estimator, with λ_0 as its penalization parameter. Let $\hat{\boldsymbol{\Sigma}}_v$ be the sample covariance matrix from the validation data. Then, the validation error is defined as $\mathrm{ValErr}(\lambda_0) = L_D(\hat{\boldsymbol{\Theta}}_0^{tr}(\lambda_0), \hat{\boldsymbol{\Sigma}}_v)$, which we can use to compute the cross-validation (CV) error, if necessary. After computing the solution path of the pilot estimator for a grid of λ_0 values, we can pick the one yielding the smallest validation (or CV) error. Then, we fix λ_0 (and hence the pilot estimator and the weight matrix) when selecting λ_1 in (2.1). Likewise, let $\hat{\boldsymbol{\Theta}}_{tr}^{\mathrm{SGM}}(\lambda_1)$ be the SGM estimator with λ_1 as its penalization parameter. Then, its validation error is defined as $\mathrm{ValErr}(\lambda_1) = L_D(\hat{\boldsymbol{\Theta}}_{tr}^{\mathrm{SGM}}(\lambda_1), \hat{\boldsymbol{\Sigma}}_v)$, which we can use to compute the CV error, if necessary. After computing the SGM estimator for a grid of λ_1 values, we pick the one with the smallest validation (or CV) error. The procedure is similar to the tuning of the graphical lasso, in which L_G (instead of L_D) is used to compute the validation (or CV) error.

2.3. Comparison with related estimators

In this section, we discuss several other related estimators. As noted earlier, neighborhood selection and the graphical lasso require the irrepresentability condition in order to be consistent in terms of recovering the true graph. The

irrepresentability condition is caused by the lasso penalty. A natural remedy is to use the adaptive lasso penalty in these two methods. For neighborhood selection, we can first fit the lasso regression, and then fit an adaptive lasso regression to estimate the support of each column of the precision matrix. For $j = 1, 2, \ldots, p$,

(N1). First solve $\min_{\boldsymbol{\beta}} \sum_{i=1}^{n}(X_{i,j} - \sum_{l \neq j} X_{il}\beta_l)^2 + \lambda_0 \sum_{l \neq j} |\beta_l|$ and let $w_l = (|\hat{\beta}_l| + u(n,p))^{-1}$.

(N2). Then, solve $\min_{\boldsymbol{\beta}} \sum_{i=1}^{n}(X_{i,j} - \sum_{l \neq j} X_{il}\beta_l)^2 + \lambda_1 \sum_{l \neq j} w_l|\beta_l|$.

For the graphical lasso, the modified procedure is as follows:

(G1). First solve $\min_{\boldsymbol{\Theta} \succ 0} L_G(\boldsymbol{\Theta}, \hat{\boldsymbol{\Sigma}}) + \lambda_0 ||\boldsymbol{\Theta}||_1$ and let $w_{ij} = (|\hat{\theta}_{ij}| + u(n,p))^{-1}$.

(G2). Then, solve $\min_{\boldsymbol{\Theta} \succ 0} L_G(\boldsymbol{\Theta}, \hat{\boldsymbol{\Sigma}}) + \lambda_1 ||\mathbf{W} \circ \boldsymbol{\Theta}||_1$.

For their theoretical justification, we need to show that the estimator is good enough such that the next step delivers the right solution, in theory. For the lasso regression, the rate of convergence of $\boldsymbol{\beta}$ can be established without using the irrepresentability condition. Still, we need to assume other conditions on the Gram matrix, such as the restricted eigenvalue condition or the compatibility condition, that remain difficult to satisfy in practice; see Bühlmann and van de Geer (2011) and Fan et al. (2020). For a more general lasso problem, such conditions are imposed on the Hessian of the loss function. In other words, the two estimators still require some structure assumptions in addition to the sparsity assumption of $\boldsymbol{\Theta}^*$.

In practice, the modified neighborhood selection estimator is still not symmetric. Thus, the modified graphical lasso procedure is preferred, which we refer to as Galasso, and include it in our numerical study.

An alternative is to use the folded concave penalty (Fan and Li (2001)) in step (N2) and step (G2) (Fan, Feng and Wu (2009)). The theory for folded concave penalized estimation also requires a reasonably good estimator (Fan, Xue and Zou (2014)). There is no fundamental difference in theory between using the folded concave penalty and the adaptive lasso penalty. We must deal with the nonconvexity problem when using the folded concave penalty. When applicable, the coordinate descent algorithm often finds a suboptimal local solution of the folded concave penalized problem, as shown by examples in Fan, Xue and Zou (2014). A better algorithm is the local linear approximation algorithm (Zou and Li (2008)), which is shown to find the oracle solution within two iterations with a high probability under ultrahigh dimensions (Fan, Xue and Zou (2014)). Each iteration is an adaptive lasso penalized problem.

3. Theory

Our analysis uses the following well-known proposition, which is shown under the sub-Gaussian assumption for the distribution of $\mathbf{X} = (X_1, X_2, \ldots, X_p)^T$.

Proposition 1. *For any* $0 < \epsilon < 1$, *there exists some* $c_0 > 0$, *such that*

$$\mathrm{P}(||\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}^*||_{\max} > \epsilon) \le p^2 \exp(-c_0 n\epsilon^2).$$

Proposition 1 is obtained by the union bound and the bound on $|\hat{\Sigma}_{ij} - \Sigma^*_{ij}|$ by the sub-Gaussian assumption (Ravikumar et al. (2011)).

First, we show the validity of the pilot estimator. The pilot estimator is only used to compute the weight matrix in the SGM estimator. We do not need to worry about whether it can recover the true graph with a high probability. Because the weight matrix is defined entrywise from the pilot estimator, the analysis of the SGM estimator shows that it is sufficient to require the pilot estimator to be close to the true precision matrix under the matrix max norm. This property of the pilot estimator is established in Theorem 1.

Theorem 1. *Let* $M = ||\mathbf{\Theta}^*||_{\ell_1}$, *and take* $0 < a \le \lambda_0/(4M)$. *With probability at least* $1 - p^2 \exp(-c_0 n\lambda_0^2/(16M^2))$,

$$||\hat{\mathbf{\Theta}}_0 - \mathbf{\Theta}^*||_{\max} \le \frac{5}{2}\lambda_0 M.$$

Based on Theorem 1, we can set $\lambda_0 = c_1\sqrt{\log p/n}$ where $c_1 > \sqrt{(32 + 16t_0)/c_0}$ M, and $t_0 > 0$ is a constant. Pick any $0 < a \le c_1/(4M)$. Then, with probability at least $1 - p^{-t_0}$, we have $||\hat{\mathbf{\Theta}}_0 - \mathbf{\Theta}^*||_{\max} \le (5/2)c_1 M\sqrt{\log p/n}$.

The next theorem concerns the exact recovery property of the SGM estimator and its rates of convergence under some matrix norms.

Theorem 2. *Let* $\Psi = \min_{(i,j)\in S} |\theta^*_{ij}|$, $G = ||(\mathbf{\Gamma}^*_{SS})^{-1}||_{\ell_\infty}$ *and* $H = ||\mathbf{\Gamma}^*_{S^cS}(\mathbf{\Gamma}^*_{SS})^{-1}||_{\ell_\infty}$, *where* $\mathbf{\Gamma}^* = \mathbf{\Gamma}(\mathbf{\Sigma}^*)$. *Take* $\lambda_0 \le (1/(5M))[\{\Psi/(2H + 1 + dG\Psi(1 + H))\} - 2u(n,p)]$ *where* $u(n,p) < \Psi/\{2(2H + 1 + dG\Psi(1 + H))\}$, $\lambda_1 < \min\{1/(2dG), \Psi^2/(2G(2+dG\Psi)), \lambda_{\min}(\mathbf{\Theta}^*)/(2\min\{\sqrt{s}, d\}((2/\Psi) + dG)G)\}$, *and* $0 < a \le \min\{\lambda_0/(4M), \lambda_1/2\}\sqrt{n/\log p}$. *Then, with probability at least* $1 - p^2\exp(-c_0 n\min\{\lambda_0^2/(16M^2), \lambda_1^2/4\})$, $\hat{\mathbf{\Theta}}^{\mathrm{SGM}}$ *is positive definite and recovers the true graph, that is,* $\{(i,j)|\hat{\theta}_{ij} \ne 0\} = S$. *Furthermore, we have*

$$||\hat{\mathbf{\Theta}}^{\mathrm{SGM}} - \mathbf{\Theta}^*||_{\max} < 2\left(\frac{2}{\Psi} + dG\right)G\lambda_1,$$

$$||\hat{\mathbf{\Theta}}^{\mathrm{SGM}} - \mathbf{\Theta}^*||_F < 2\sqrt{s}\left(\frac{2}{\Psi} + dG\right)G\lambda_1,$$

$$||\hat{\mathbf{\Theta}}^{\mathrm{SGM}} - \mathbf{\Theta}^*||_2 < 2\min\{\sqrt{s}, d\}\left(\frac{2}{\Psi} + dG\right)G\lambda_1.$$

Remark 3. Based on Theorem 2, we can take $\lambda_0 = c_1\sqrt{\log p/n}, \lambda_1 = c_2\sqrt{\log p/n}$, where

$$\sqrt{\frac{32 + 16t_0}{c_0}}M < c_1 < \sqrt{\frac{n}{\log p}}\frac{1}{10M}\frac{\Psi}{2H + 1 + dG\Psi(1 + H)}$$

and

$$\sqrt{\frac{8+4t_0}{c_0}} < c_2 < \sqrt{\frac{n}{\log p}} \min\left\{ \frac{1}{2dG}, \frac{\Psi^2}{2G(2+dG\Psi)}, \frac{\lambda_{\min}(\mathbf{\Theta}^*)}{2\min\{\sqrt{s}, d\}((2/\Psi)+dG)G} \right\}.$$

Further, let the small perturbations a and $u(n,p)$ satisfy $0 < a \leq \min\{c_1/(4M), c_2/2\}$ and $u(n,p) < \Psi/(4(2H+1+dG\Psi(1+H)))$, respectively. Then, with probability at least $1-p^{-t_0}$, $\hat{\mathbf{\Theta}}^{\mathrm{SGM}}$ is positive definite and recovers the true graph, with matrix bounds $||\hat{\mathbf{\Theta}}^{\mathrm{SGM}} - \mathbf{\Theta}^*||_{\max} < 2((2/\Psi)+dG)Gc_2\sqrt{\log p/n}$, $||\hat{\mathbf{\Theta}}^{\mathrm{SGM}} - \mathbf{\Theta}^*||_F < 2\sqrt{s}(2/\Psi+dG)Gc_2\sqrt{\log p/n}$, and $||\hat{\mathbf{\Theta}}^{\mathrm{SGM}} - \mathbf{\Theta}^*||_2 < 2\min\{\sqrt{s}, d\}(2/\Psi+dG)Gc_2\sqrt{\log p/n}$. Comparing these with the results for the graphical lasso in Ravikumar et al. (2011) under the irrepresentable condition, the SGM and the graphical lasso have similar asymptotic rates of convergence under different matrix norms.

Remark 4. Although the SGM estimator is positive definite with overwhelming probability, it is not guaranteed to be positive definite for every data set. In all of our numerical examples, we have checked that the computed SGM estimator is positive definite. If the user only cares about recovering the graph, then this is not important, as for neighborhood selection. On the other hand, if the application demands using a positive-definite matrix estimator, and $\hat{\mathbf{\Theta}}^{\mathrm{SGM}}$ happens to have a zero or negative eigenvalue, we can perform an additional optimization by adding an eigenvalue constraint, as follows:

$$\hat{\mathbf{\Theta}}_+^{\mathrm{SGM}} = \underset{\mathbf{\Theta} \succeq 10^{-5}\boldsymbol{I}}{\operatorname{argmin}} \frac{1}{2}\operatorname{tr}(\mathbf{\Theta}^T\tilde{\mathbf{\Sigma}}\mathbf{\Theta}) - \operatorname{tr}(\mathbf{\Theta}) + \lambda_1||\mathbf{W} \circ \mathbf{\Theta}||_1, \tag{3.1}$$

which can be solved efficiently using the ADMM algorithm in Zhang and Zou (2014). Note too that we only solve (3.1) after tuning the SGM estimator, which means that λ_1 is the chosen penalization parameter in the final SGM estimator and the weight matrix is given too. Thus, we run the ADMM algorithm only once.

4. Numerical Results

4.1. Simulations

In the simulation study, we generate n i.i.d. samples from $N_p(\mathbf{0}, \mathbf{\Sigma}^*)$ under four different $\mathbf{\Theta}^*$ generation processes:

Model 1: $\mathbf{\Theta}^*$ is fixed, with $\theta^*_{ii} = 1$, and $\theta^*_{ij} = 0.3$ for $|i-j| = 1$, and $\theta^*_{ij} = 0$ otherwise.

Model 2: $\mathbf{\Theta}^*$ is fixed, with $\theta^*_{ii} = 1$, and $\theta^*_{ij} = 0.4$ for $|i-j| = 1, \theta^*_{ij} = 0.3$ for

$|i-j|=2, \theta^*_{ij}=0.2$ for $|i-j|=3$, and $\theta^*_{ij}=0$ otherwise.

Model 3: $\mathbf{\Theta}^*$ is generated randomly. First, let $\mathbf{B}=(b_{i,j})$ be a $p\times p$ matrix, such that $b_{j,i}=b_{i,j} \overset{i.i.d.}{\sim} \text{Bernoulli}(q), \forall i>j$. The diagonal elements of $\mathbf{B}$ are zero. Next, select $\delta\in\mathbb{R}$ such that $\mathbf{M}=\mathbf{B}+\delta\mathbf{I}$ is positive definite and the condition number of $\mathbf{M}$ equals to p. Finally, select $a>0$ and let the precision matrix $\mathbf{\Theta}^*=a\mathbf{M}$ such that the diagonal elements of $\mathbf{\Theta}^*$ are equal to 1. We set $q=0.05$ in the simulations.

Model 4: $\mathbf{\Theta}^*$ is generated randomly. Let $\mathbf{\Theta}^1,\mathbf{\Theta}^2,\ldots,\mathbf{\Theta}^5$ be five $(p/5)\times(p/5)$ matrices generated independently by model 3, with $q=0.25$. Then, the precision matrix $\mathbf{\Theta}^*=\text{diag}\{\mathbf{\Theta}^1,\ldots,\mathbf{\Theta}^5\}$.

Model 1 and model 2 are commonly used for precision matrix estimator comparisons (Zhang and Zou (2014)). The generation process of model 3 is based on that of model 2 in Cai, Liu and Luo (2011). Model 4 is the block-diagonal version of model 3, corresponding to a graph with five unconnected parts, with denser connections within each. We set $q=0.05$ and 0.25 for model 3 and model 4, respectively, such that the overall sparsity levels of the precision matrices are the same. In each model, we use three n,p combinations: (i) $n=400, p=100$; (ii) $n=400, p=500$; and (iii) $n=100, p=500$. We compare theSGM with Glasso, Galasso, the Dtrace estimator (Zhang and Zou (2014)), and CLIME (Cai, Liu and Luo (2011)). Glasso and CLIME are implemented using the R packages glasso and clime, respectively. Dtrace is implemented using the code from Zhang and Zou (2014). The performance of each estimator is evaluated by the following measures:

- Frobenius risk $E||\hat{\mathbf{\Theta}}-\mathbf{\Theta}^*||_F$; spectral risk $E||\hat{\mathbf{\Theta}}-\mathbf{\Theta}^*||_2$; the ℓ_1 risk $E||\hat{\mathbf{\Theta}}-\mathbf{\Theta}^*||_{\ell_1}$; the max risk $E\|\hat{\mathbf{\Theta}}-\mathbf{\Theta}^*\|_{max}$
- Sensitivity $=TP/(TP+FN)$, and Specificity $=TN/(TN+FP)$, where TP, FP, TN and FN denote the numbers of true positives, false positives, true negatives, and false negatives, respectively.

The results are summarized in Tables 1–12, where we report the mean and standard error of each metric based on 100 independent repetitions. We also report the running time of each method, in seconds. Note that when $p=500$, the code for CLIME gives an error message or does not finish the computation within one hour. For these cases, we record NA for CLIME.

Several observations can be made from these tables. For the quality of the estimates, under model 1 with $n=400$ and $p=100$ or $n=400$ and $p=500$, the SGM performs similarly to Galasso, and both outperform the other methods. For model 2 with $n=400$, and $p=100$ or $n=400$ and $p=500$, the SGM is a clear winner among all the methods. When the precision matrices are generated randomly, in the $n=400$ and $p=100$ scenario (table 7 and 10), the SGM gives

the best estimates, measured by all the matrix norms, sensitivity, and specificity. When $n = 400$ and $p = 500$, the SGM is the best measured by the l_1-norm for model 3, and the best measured by the spectral norm and the l_1-norm for model 4. In the scenario of $n = 100$ and $p = 500$, under all four data-generating models, the SGM is among the best when measured by the l_1-norm, and has a slightly larger l_2-norm. Overall, the simulation results provide numerical results that confirm or complement the theoretical bounds for the SGM.

The SGM and Galasso have comparable computing time, and both are much faster than Dtrace and CLIME. The ratio of the timing between the SGM and Glasso is about five and the ratio of the timing between the SGM and Galasso is about two. The ratio stays stable when p increases from 100 to 500, suggesting that the SGM can scale as well as Glasso or Galasso for practical applications. In addition, we observe that the SGM is the one of the most stable methods. In some runs, CLIME, Glasso, and Galasso either report error messages or cannot finish the computation within one hour. When an error message occurred, we do not use that run to compute the average and standard error for that method. In contrast, the SGM exhibits no such an issue in our simulations.

4.2. Real-data examples

We examine the performance of the SGM on two gene expression data sets. Data set 1 contains data on prostate cancer, studied by Singh et al. (2002). It contain 52 prostate tumor samples and 50 nontumor prostate samples, with 12,600 gene expression levels. Data set 2 contains data on breast cancer, analyzed by Hess et al. (2006), and consists of 22,283 gene expressions of 133 subjects, among which, 34 have pathological complete response and 99 have residual disease. First, we randomly split each data set into training, validation, and test sets of almost equal sizes. The splits are done in a stratified way, such that the class proportions are preserved in each set. Then, using the training and validation sets, we preprocess the data by screening the genes (Fan and Fan, 2008; Fan, Feng and Wu, 2009)) down to a subset of size p_s, containing the most significant genes, according to the two-sample t-tests between the two classes, and standardizing the gene expressions.

To estimate the precision matrices, for each method, we fit it using the training set on a grid of regularization parameter values, and choose the best estimate by minimizing a loss function on the validation set. Here, the Dtrace loss is used for the SGM and Dtrace, and the graphical lasso loss is used for Glasso, Galasso, and CLIME. We report the ratios of nonzero entries in the estimated precision matrices. A sparser estimate is usually more favorable, for ease of interpretation. Because the true precision matrices are unknown, we examine and compare the quality of the precision matrix estimates by using a linear discriminant analysis (LDA) in which the resulting precision matrix estimator can be used to fit the LDA rule. The rationale is that a better precision

Table 1. Model 1 with $n = 400$ and $p = 100$.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	1.170	0.302	0.419	0.206	1.000	0.995	1.701
	(0.007)	(0.004)	(0.006)	(0.003)	(0.000)	(0.000)	(0.002)
Glasso	1.808	0.409	0.710	0.213	1.000	0.905	0.432
	(0.007)	(0.002)	(0.006)	(0.002)	(0.000)	(0.002)	(0.002)
Galasso	1.161	0.297	0.417	0.192	1.000	0.993	0.949
	(0.006)	(0.004)	(0.005)	(0.002)	(0.000)	(0.000)	(0.004)
CLIME	1.581	0.311	0.443	0.217	1.000	0.967	353.795
	(0.005)	(0.003)	(0.004)	(0.002)	(0.000)	(0.001)	(0.165)
Dtrace	1.620	0.366	0.626	0.202	1.000	0.921	2.942
	(0.006)	(0.003)	(0.006)	(0.002)	(0.000)	(0.001)	(0.016)

Table 2. Model 1 with $n = 400$ and $p = 500$. The code for CLIME gives an error message or does not finish the computation within one hour.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	2.863	0.374	0.538	0.256	1.000	0.998	197.519
	(0.007)	(0.003)	(0.005)	(0.003)	(0.000)	(0.000)	(0.962)
Glasso	4.853	0.492	0.947	0.258	1.000	0.972	44.053
	(0.007)	(0.002)	(0.008)	(0.001)	(0.000)	(0.000)	(0.234)
Galasso	2.789	0.370	0.538	0.249	1.000	0.980	78.001
	(0.007)	(0.003)	(0.006)	(0.003)	(0.000)	(0.000)	(0.406)
CLIME	NA	NA	NA	NA	NA	NA	NA
Dtrace	4.227	0.444	0.742	0.247	1.000	0.980	257.417
	(0.006)	(0.002)	(0.004)	(0.002)	(0.000)	(0.000)	(3.967)

Table 3. Model 1 with $n = 100$ and $p = 500$. The code for CLIME gives an error message or does not finish the computation within one hour.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	8.208	0.839	1.346	0.499	0.738	0.998	273.517
	(0.014)	(0.009)	(0.018)	(0.009)	(0.002)	(0.000)	(1.875)
Glasso	8.434	0.776	1.670	0.381	0.896	0.976	98.188
	(0.010)	(0.001)	(0.010)	(0.003)	(0.002)	(0.000)	(0.549)
Galasso	7.968	0.788	1.319	0.416	0.773	0.996	138.960
	(0.013)	(0.005)	(0.011)	(0.006)	(0.002)	(0.000)	(0.653)
CLIME	NA	NA	NA	NA	NA	NA	NA
Dtrace	8.287	0.766	1.310	0.473	0.860	0.990	437.238
	(0.011)	(0.004)	(0.015)	(0.008)	(0.002)	(0.000)	(0.574)

Table 4. Model 2 with $n = 400$ and $p = 100$.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	1.939	0.580	0.958	0.216	0.991	0.978	1.848
	(0.011)	(0.007)	(0.011)	(0.002)	(0.000)	(0.000)	(0.004)
Glasso	4.503	1.242	2.049	0.290	0.998	0.689	0.434
	(0.02)	(0.006)	(0.008)	(0.002)	(0.000)	(0.003)	(0.002)
Galasso	2.759	0.832	1.333	0.242	0.989	0.934	1.021
	(0.018)	(0.007)	(0.011)	(0.003)	(0.001)	(0.001)	(0.004)
CLIME	3.830	0.751	1.458	0.384	0.991	0.848	374.404
	(0.011)	(0.006)	(0.009)	(0.004)	(0.001)	(0.003)	(0.445)
Dtrace	3.108	0.888	1.582	0.236	0.999	0.771	3.017
	(0.016)	(0.007)	(0.011)	(0.002)	(0.000)	(0.004)	(0.017)

Table 5. Model 2 with $n = 400$ and $p = 500$. The code for CLIME gives an error message or does not finish the computation within one hour.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	5.800	0.857	1.402	0.297	0.963	0.992	231.802
	(0.017)	(0.006)	(0.009)	(0.002)	(0.000)	(0.000)	(2.340)
Glasso	13.397	1.630	2.758	0.368	0.975	0.897	46.049
	(0.034)	(0.004)	(0.014)	(0.001)	(0.001)	(0.002)	(0.623)
Galasso	9.230	1.247	2.000	0.345	0.932	0.980	87.763
	(0.035)	(0.005)	(0.010)	(0.004)	(0.001)	(0.000)	(1.064)
CLIME	NA	NA	NA	NA	NA	NA	NA
Dtrace	9.678	1.243	2.036	0.315	0.991	0.938	218.368
	(0.039)	(0.006)	(0.011)	(0.002)	(0.000)	(0.001)	(0.137)

Table 6. Model 2 with $n = 100$ and $p = 500$. The code for CLIME gives an error message or does not finish the computation within one hour.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	18.136	2.130	2.382	0.541	0.204	0.999	364.245
	(0.012)	(0.001)	(0.004)	(0.002)	(0.001)	(0.000)	(3.469)
Glasso	17.940	2.099	2.773	0.510	0.358	0.984	96.880
	(0.007)	(0.001)	(0.010)	(0.002)	(0.002)	(0.000)	(0.476)
Galasso	18.128	2.126	2.517	0.526	0.212	0.999	140.462
	(0.008)	(0.001)	(0.007)	(0.002)	(0.001)	(0.000)	(0.690)
CLIME	NA	NA	NA	NA	NA	NA	NA
Dtrace	17.962	2.106	2.396	0.521	0.286	0.993	463.535
	(0.009)	(0.001)	(0.004)	(0.002)	(0.002)	(0.000)	(1.098)

Table 7. Model 3 with $n = 400$ and $p = 100$.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	1.424	0.369	0.669	0.210	0.998	0.986	2.353
	(0.009)	(0.005)	(0.009)	(0.003)	(0.000)	(0.000)	(0.017)
Glasso	2.284	0.645	1.353	0.249	1.000	0.811	0.370
	(0.019)	(0.007)	(0.014)	(0.003)	(0.000)	(0.006)	(0.006)
Galasso	1.577	0.443	0.762	0.219	0.998	0.987	0.905
	(0.018)	(0.007)	(0.012)	(0.002)	(0.000)	(0.001)	(0.012)
CLIME	5.251	2.020	3.200	1.581	0.832	0.901	344.889
	(0.092)	(0.044)	(0.074)	(0.068)	(0.026)	(0.008)	(0.491)
Dtrace	1.950	0.534	1.129	0.210	1.000	0.856	9.840
	(0.007)	(0.004)	(0.011)	(0.002)	(0.000)	(0.002)	(0.068)

Table 8. Model 3 with $n = 400$ and $p = 500$. The code for CLIME gives an error message or does not finish the computation within one hour.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	9.001	1.563	3.145	0.309	0.643	0.984	267.299
	(0.018)	(0.008)	(0.014)	(0.003)	(0.003)	(0.000)	(1.779)
Glasso	9.313	1.924	3.847	0.338	0.898	0.866	73.949
(3 NAs)	(0.047)	(0.011)	(0.02)	(0.004)	(0.003)	(0.004)	(1.679)
Galasso	8.518	1.475	3.154	0.280	0.769	0.963	131.181
(3 NAs)	(0.031)	(0.011)	(0.016)	(0.004)	(0.004)	(0.001)	(2.131)
CLIME	NA	NA	NA	NA	NA	NA	NA
Dtrace	8.829	1.863	3.431	0.309	0.842	0.924	1,273.311
	(0.013)	(0.004)	(0.014)	(0.002)	(0.002)	(0.001)	(4.490)

Table 9. Model 3 with $n = 100$ and $p = 500$. The code for CLIME gives an error message or does not finish the computation within one hour.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	14.018	2.861	4.632	0.647	0.113	0.996	517.967
	(0.020)	(0.003)	(0.022)	(0.003)	(0.001)	(0.000)	(8.861)
Glasso	12.684	2.676	4.958	0.480	0.405	0.934	125.451
	(0.024)	(0.005)	(0.021)	(0.003)	(0.003)	(0.001)	(3.055)
Galasso	13.079	2.623	4.723	0.511	0.225	0.984	185.978
	(0.017)	(0.005)	(0.021)	(0.003)	(0.002)	(0.000)	(3.349)
CLIME	NA	NA	NA	NA	NA	NA	NA
Dtrace	13.071	2.795	4.624	0.565	0.269	0.934	1,327.819
	(0.018)	(0.003)	(0.022)	(0.003)	(0.002)	(0.001)	(6.106)

Table 10. Model 4 with $n = 400$ and $p = 100$.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	1.542	0.463	0.824	0.223	0.998	0.986	2.073
	(0.011)	(0.009)	(0.016)	(0.003)	(0.000)	(0.000)	(0.005)
Glasso	2.693	0.916	1.700	0.271	0.999	0.799	0.424
	(0.017)	(0.011)	(0.019)	(0.002)	(0.000)	(0.003)	(0.003)
Galasso	1.747	0.599	1.030	0.234	0.996	0.970	0.915
	(0.016)	(0.011)	(0.019)	(0.002)	(0.000)	(0.001)	(0.004)
CLIME	5.130	1.583	2.511	1.313	0.987	0.901	394.392
	(0.034)	(0.022)	(0.035)	(0.024)	(0.004)	(0.004)	(0.253)
Dtrace	2.202	0.710	1.325	0.222	1.000	0.848	6.912
	(0.011)	(0.009)	(0.016)	(0.003)	(0.000)	(0.001)	(0.048)

Table 11. Model 4 with $n = 400$ and $p = 500$. The code for CLIME gives an error message or does not finish the computation within one hour.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGM	10.745	2.266	4.007	0.336	0.525	0.986	245.181
	(0.016)	(0.008)	(0.023)	(0.003)	(0.002)	(0.000)	(0.590)
Glasso	11.027	2.565	4.670	0.349	0.784	0.870	57.883
	(0.028)	(0.007)	(0.023)	(0.002)	(0.003)	(0.002)	(0.419)
Galasso	10.797	2.330	4.152	0.329	0.590	0.971	111.813
	(0.027)	(0.010)	(0.025)	(0.003)	(0.003)	(0.001)	(0.655)
CLIME	NA	NA	NA	NA	NA	NA	NA
Dtrace	10.669	2.494	4.216	0.331	0.720	0.932	435.302
	(0.011)	(0.006)	(0.022)	(0.002)	(0.002)	(0.000)	(1.566)

Table 12. Model 4 with $n = 100$ and $p = 500$. The code for CLIME gives an error message or does not finish the computation within one hour.

	$\|\cdot\|_F$	$\|\cdot\|_2$	$\|\cdot\|_{l_1}$	$\|\cdot\|_{max}$	Sen.	Spe.	Time(s)
SGB	14.677	3.223	4.806	0.607	0.105	0.997	395.848
	(0.021)	(0.006)	(0.022)	(0.003)	(0.001)	(0.000)	(2.776)
Glasso	13.992	3.144	5.096	0.501	0.322	0.953	96.147
	(0.021)	(0.006)	(0.021)	(0.002)	(0.002)	(0.001)	(0.908)
Galasso	14.137	3.087	4.903	0.516	0.179	0.990	155.107
	(0.018)	(0.006)	(0.021)	(0.002)	(0.001)	(0.000)	(1.029)
CLIME	NA	NA	NA	NA	NA	NA	NA
Dtrace	14.025	3.186	4.768	0.546	0.211	0.945	697.595
	(0.021)	(0.006)	(0.021)	(0.002)	(0.001)	(0.002)	(2.629)

matrix estimator leads to better classification accuracy. Similar comparison methods based on the LDA are used in other work including Fan, Feng and Wu (2009) and Cai, Liu and Luo (2011). Here, we do not repeat the LDA formula. The classification performance is evaluated using the sensitivity, specificity, and Mathews correlation coefficient (MCC) metrics. Let TP, FP, TN and FN denote the numbers of true positives, false positives, true negatives and false negatives, respectively, on the test set. Then, these metrics are defined as Sensitivity $= TP/(TP+FN)$, Specificity $= TN/(TN+FP)$, and MCC $= (TP \times TN - FP \times FN)/\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}$.

In the variable screening step, both Fan, Feng and Wu (2009) and Cai, Liu and Luo (2011) reduce the numbers of variables down to $n+1$, where n is the training sample size. Typically, variable screening reduces the dimension from p to n, $n/\log(n)$, or $2n/\log(n)$ (Fan et al., 2020)). Fan, Feng and Wu (2009) and Cai, Liu and Luo (2011) set the reduced dimension to $n+1$ to emphasize that the sample covariance matrix for the reduced dimension is still singular. Following their practice, we similarly set p_s to be marginally larger than the training sample size, letting $p_s = 35$ for data set 1 and $p_s = 50$ for data set 2. The corresponding training sample sizes are 33 and 44, respectively. Because p_s is larger than the training sample size, we can examine the performance of the methods when the input sample covariance matrices are not invertible. The tumor group of data set 1 and the pathological complete response group of data set 2 are treated as "positive" when computing the classification metrics. We performed 100 repetitions in order to have a more stable comparison. The results are reported in Table 13 (prostate cancer data) and Table 14 (breast cancer data). For the prostate cancer data, the methods perform similarly in terms of classification performance, with the SGM and Dtrace having the highest average MCC scores. The sparsity level of the SGM is significantly better than those of the other methods. For the breast cancer data, the SGM has a significantly higher sensitivity score than Glasso, Galasso, and CLIME, and its specificity score is comparable with those of the other methods. The MCC score of the SGM is the highest among these methods. The SGM again achieves the best sparsity level.

5. Conclusion

In this paper, we have introduced a simple method that we call the SGM for recovering a Gaussian graphical model under ultrahigh dimensions. The SGM is based on a simple quadratic loss function, and hence allows us to use a simple coordinate descent algorithm to achieve excellent computational efficiency compared with that of the graphical lasso, which requires a much more sophisticated algorithm. The SGM can easily handle the symmetry constraint, which is an obvious advantage over methods such as CLIME and neighborhood

Table 13. Performance comparison between SGM, Glasso, Galasso, CLIME, and Dtrace on a prostate cancer data set.

	Sensitivity	Specificity	MCC	Ratio of Nonzero Entries
SGM	0.899	0.920	0.822	0.095
	(0.006)	(0.008)	(0.009)	(0.004)
Glasso	0.900	0.899	0.806	0.330
	(0.008)	(0.009)	(0.009)	(0.004)
Galasso	0.902	0.904	0.812	0.143
	(0.007)	(0.009)	(0.009)	(0.002)
CLIME	0.894	0.904	0.803	0.822
	(0.007)	(0.008)	(0.010)	(0.007)
Dtrace	0.901	0.920	0.823	0.132
	(0.006)	(0.008)	(0.009)	(0.005)

Table 14. Performance comparison between SGM, Glasso, Galasso, CLIME, and Dtrace on a breast cancer data set.

	Sensitivity	Specificity	MCC	Ratio of Nonzero Entries
SGM	0.753	0.736	0.448	0.063
	(0.010)	(0.008)	(0.011)	(0.002)
Glasso	0.543	0.800	0.339	0.266
	(0.014)	(0.008)	(0.014)	(0.003)
Galasso	0.626	0.779	0.387	0.102
	(0.014)	(0.008)	(0.014)	(0.002)
CLIME	0.663	0.766	0.403	0.653
	(0.014)	(0.008)	(0.012)	(0.008)
Dtrace	0.740	0.737	0.437	0.310
	(0.010)	(0.008)	(0.012)	(0.019)

selection. Although the SGM does not guarantee that the resulting precision matrix estimator is positive definite, we provide a simple step to mitigate this issue, and in our numerical experiments we check whether the SGM estimators are positive definite in each run. We compare the rate of convergences of the SGM and the graphical lasso, but the SGM does not require the irrepresentable condition necessary for the graphical lasso. Our simulations confirm that the SGM exhibits excellent and often improved performance over the graphical lasso. Based on our results, we recommend using the SGM to estimate a large Gaussian graphical model.

Acknowledgments

We sincerely thank the editor, associate editor, and referees for their helpful comments. Zou's work was supported, in part, by NSF DMS 1915842 and 2015120.

Appendix

In this appendix we present the proof of the main theorems and the link for downloading the code used in this paper.

A.1. Proofs

Proof of Theorem 1. We bound the difference between $\hat{\mathbf{\Theta}}_0$ and $\mathbf{\Theta}^*$ under element-wise ℓ_∞ norm under the event $||\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}^*||_{\max} \leq \lambda_0/(4M)$. Then

$$||\tilde{\mathbf{\Sigma}} - \mathbf{\Sigma}^*||_{\max} \leq ||\hat{\mathbf{\Sigma}} - \mathbf{\Sigma}^*||_{\max} + a\sqrt{\frac{\log p}{n}} \leq \frac{\lambda_0}{2M}.$$

Firstly, we show $||\tilde{\mathbf{\Sigma}}\hat{\mathbf{\Theta}}_0 - \mathbf{I}||_{\max} \leq \lambda_0$. Since $\hat{\mathbf{\Theta}}_0$ is the optimal solution, it satisfies

$$\tilde{\mathbf{\Sigma}}\hat{\mathbf{\Theta}}_0 - \mathbf{I} + \lambda_0\hat{\mathbf{Z}}_0 = \mathbf{0},$$

where $\hat{\mathbf{Z}}_0 = (\hat{z}^0_{ij})$ is the sub-gradient of $|\hat{\mathbf{\Theta}}_0|$ and

$$\hat{z}^0_{ij} \begin{cases} = \operatorname{sign}(\hat{\theta}^0_{ij}) & \text{if } \hat{\theta}^0_{ij} \neq 0 \\ \in [-1, 1] & \text{if } \hat{\theta}^0_{ij} = 0 \end{cases}$$

where $\hat{\mathbf{\Theta}}_0 = (\hat{\theta}^0_{ij})$. Thus, $||\tilde{\mathbf{\Sigma}}\hat{\mathbf{\Theta}}_0 - \mathbf{I}||_{\max} = || - \lambda_0\hat{\mathbf{Z}}_0||_{\max} \leq \lambda_0$. Then we show $||\hat{\mathbf{\Theta}}_0||_{\ell_1} \leq 3||\mathbf{\Theta}^*||_{\ell_1}$. To prove this, it is sufficient to prove $|\hat{\boldsymbol{\theta}}^0_i|_1 \leq 3|\boldsymbol{\theta}^*_i|_1$ for $i = 1, 2, \ldots, p$ where $\hat{\mathbf{\Theta}}_0 = (\hat{\boldsymbol{\theta}}^0_1, \hat{\boldsymbol{\theta}}^0_2, \ldots, \hat{\boldsymbol{\theta}}^0_p)$ and $\mathbf{\Theta}^* = (\boldsymbol{\theta}^*_1, \boldsymbol{\theta}^*_2, \ldots, \boldsymbol{\theta}^*_p)$. Let f denote the function $f(\boldsymbol{\theta}_i) = (1/2)\boldsymbol{\theta}_i^T\tilde{\mathbf{\Sigma}}\boldsymbol{\theta}_i - \boldsymbol{e}_i^T\boldsymbol{\theta}_i + \lambda_0|\boldsymbol{\theta}_i|_1$ and $\mathbf{\Delta}_i := \hat{\boldsymbol{\theta}}^0_i - \boldsymbol{\theta}^*_i$,

$$\begin{aligned} f(\hat{\boldsymbol{\theta}}^0_i) &= \frac{1}{2}(\hat{\boldsymbol{\theta}}^0_i)^T\tilde{\mathbf{\Sigma}}\hat{\boldsymbol{\theta}}^0_i - \boldsymbol{e}_i^T\hat{\boldsymbol{\theta}}^0_i + \lambda_0|\hat{\boldsymbol{\theta}}^0_i|_1 \\ &= \frac{1}{2}(\mathbf{\Delta}_i + \boldsymbol{\theta}^*_i)^T\tilde{\mathbf{\Sigma}}(\mathbf{\Delta}_i + \boldsymbol{\theta}^*_i) - \boldsymbol{e}_i^T(\mathbf{\Delta}_i + \boldsymbol{\theta}^*_i) + \lambda_0|\mathbf{\Delta}_i + \boldsymbol{\theta}^*_i|_1 \\ &= f(\boldsymbol{\theta}^*_i) + \frac{1}{2}\mathbf{\Delta}_i^T\tilde{\mathbf{\Sigma}}\mathbf{\Delta}_i + \mathbf{\Delta}_i^T\tilde{\mathbf{\Sigma}}\boldsymbol{\theta}^*_i - \boldsymbol{e}_i^T\mathbf{\Delta}_i + \lambda_0|\mathbf{\Delta}_i + \boldsymbol{\theta}^*_i|_1 - \lambda_0|\boldsymbol{\theta}^*_i|_1. \end{aligned}$$

Since $\hat{\boldsymbol{\theta}}^0_i$ is the optimal solution, $f(\hat{\boldsymbol{\theta}}^0_i) \leq f(\boldsymbol{\theta}^*_i)$. The term $\mathbf{\Delta}_i^T\tilde{\mathbf{\Sigma}}\mathbf{\Delta}_i > 0$ because of the positive definiteness of $\tilde{\mathbf{\Sigma}}$. We have

$$\begin{aligned} \lambda_0|\hat{\boldsymbol{\theta}}^0_i|_1 - \lambda_0|\boldsymbol{\theta}^*_i|_1 &= \lambda_0|\mathbf{\Delta}_i + \boldsymbol{\theta}^*_i|_1 - \lambda_0|\boldsymbol{\theta}^*_i|_1 \\ &\leq \boldsymbol{e}_i^T\mathbf{\Delta}_i - \mathbf{\Delta}_i^T\tilde{\mathbf{\Sigma}}\boldsymbol{\theta}^*_i \\ &= \mathbf{\Delta}_i^T(\mathbf{\Sigma}^* - \tilde{\mathbf{\Sigma}})\boldsymbol{\theta}^*_i. \end{aligned}$$

Noticing

$$\begin{aligned} \mathbf{\Delta}_i^T(\mathbf{\Sigma}^* - \tilde{\mathbf{\Sigma}})\boldsymbol{\theta}^*_i &\leq |\mathbf{\Delta}_i|_1||(\mathbf{\Sigma}^* - \tilde{\mathbf{\Sigma}})\boldsymbol{\theta}^*_i||_{\max} \\ &\leq |\mathbf{\Delta}_i|_1||\mathbf{\Sigma}^* - \tilde{\mathbf{\Sigma}}||_{\max}|\boldsymbol{\theta}^*_i|_1 \end{aligned}$$

$$\leq \frac{\lambda_0}{2}|\boldsymbol{\Delta}_i|_1$$
$$\leq \frac{\lambda_0}{2}|\hat{\boldsymbol{\theta}}_i^0|_1 + \frac{\lambda_0}{2}|\boldsymbol{\theta}_i^*|_1,$$

it follows that $|\hat{\boldsymbol{\theta}}_i^0|_1 \leq 3|\boldsymbol{\theta}_i^*|_1$ for any i. Therefore,

$$\begin{aligned}||\hat{\boldsymbol{\Theta}}_0 - \boldsymbol{\Theta}^*||_{\max} &= ||\boldsymbol{\Theta}^*(\boldsymbol{\Sigma}^*\hat{\boldsymbol{\Theta}}_0 - \tilde{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}}_0 + \tilde{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}}_0 - \mathbf{I})||_{\max} \\ &\leq ||\boldsymbol{\Theta}^*||_{\ell_1}||\boldsymbol{\Sigma}^* - \tilde{\boldsymbol{\Sigma}}||_{\max}||\hat{\boldsymbol{\Theta}}_0||_{\ell_1} + ||\boldsymbol{\Theta}^*||_{\ell_1}||\tilde{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}}_0 - \mathbf{I}||_{\max} \\ &\leq \frac{3}{2}\lambda_0 M + \lambda_0 M \\ &= \frac{5}{2}\lambda_0 M.\end{aligned}$$

This completes the proof.

Proof of Theorem 2. For simplicity, we use $\hat{\boldsymbol{\Theta}}$ to represent $\hat{\boldsymbol{\Theta}}^{\mathrm{SGM}}$ in this proof. We first prove that $\hat{\boldsymbol{\Theta}}$ recovers the true graph under the event $||\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^*||_{\max} \leq \min\{\lambda_0/(4M), \lambda_1/2\}$. First, we note that

$$||\tilde{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^*||_{\max} \leq ||\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^*||_{\max} + a\sqrt{\frac{\log p}{n}} \leq \min\left\{\frac{\lambda_0}{2M}, \lambda_1\right\}.$$

Define $\tilde{\boldsymbol{\Theta}}$ as the optimal solution for the following problem:

$$\tilde{\boldsymbol{\Theta}} = \underset{\boldsymbol{\Theta}_{S^c}=0, \boldsymbol{\Theta}^T=\boldsymbol{\Theta}}{\operatorname{argmin}} \frac{1}{2}\operatorname{tr}(\boldsymbol{\Theta}^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{\Theta}) - \operatorname{tr}(\boldsymbol{\Theta}) + \lambda_1||\mathbf{W} \circ \boldsymbol{\Theta}||_1.$$

It suffice to prove (i) $\tilde{\boldsymbol{\Theta}}$ recovers the true graph and (ii) $\tilde{\boldsymbol{\Theta}} = \hat{\boldsymbol{\Theta}}$. To show (i) and (ii), we define two quantities $\Delta_G = ||(\tilde{\boldsymbol{\Gamma}}_{SS})^{-1} - (\boldsymbol{\Gamma}^*_{SS})^{-1}||_{\ell_\infty}$ and $\Delta_H = ||\tilde{\boldsymbol{\Gamma}}_{S^cS}(\tilde{\boldsymbol{\Gamma}}_{SS})^{-1} - \boldsymbol{\Gamma}^*_{S^cS}(\boldsymbol{\Gamma}^*_{SS})^{-1}||_{\ell_\infty}$ where $\tilde{\boldsymbol{\Gamma}} = \boldsymbol{\Gamma}(\tilde{\boldsymbol{\Sigma}})$. We first bound Δ_G and Δ_H.

$$\begin{aligned}\Delta_G &= ||(\tilde{\boldsymbol{\Gamma}}_{SS})^{-1}(\tilde{\boldsymbol{\Gamma}}_{SS} - \boldsymbol{\Gamma}^*_{SS})(\boldsymbol{\Gamma}^*_{SS})^{-1}||_{\ell_\infty} \\ &\leq ||(\tilde{\boldsymbol{\Gamma}}_{SS})^{-1}||_{\ell_\infty}||\tilde{\boldsymbol{\Gamma}}_{SS} - \boldsymbol{\Gamma}^*_{SS}||_{\ell_\infty}||(\boldsymbol{\Gamma}^*_{SS})^{-1}||_{\ell_\infty} \\ &\leq (G + \Delta_G)d\lambda_1 G,\end{aligned}$$

where we use inequalities $||\mathbf{AB}||_{\ell_\infty} \leq ||\mathbf{A}||_{\ell_\infty}||\mathbf{B}||_{\ell_\infty}$ and $||\mathbf{A} + \mathbf{B}||_{\ell_\infty} \leq ||\mathbf{A}||_{\ell_\infty} + ||\mathbf{B}||_{\ell_\infty}$ for any matrices $\mathbf{A}$ and $\mathbf{B}$. It is easy to see

$$\Delta_G \leq \frac{d\lambda_1 G^2}{1 - d\lambda_1 G}. \tag{A.1}$$

The bound is larger than 0 since $\lambda_1 < 1/(2dG)$.

$$\begin{aligned}\Delta_H &\leq ||(\tilde{\boldsymbol{\Gamma}}_{S^cS} - \boldsymbol{\Gamma}^*_{S^cS})(\tilde{\boldsymbol{\Gamma}}_{SS})^{-1}||_{\ell_\infty} + ||\boldsymbol{\Gamma}^*_{S^cS}((\tilde{\boldsymbol{\Gamma}}_{SS})^{-1} - (\boldsymbol{\Gamma}^*_{SS})^{-1})||_{\ell_\infty} \\ &\leq (||\tilde{\boldsymbol{\Gamma}}_{S^cS} - \boldsymbol{\Gamma}^*_{S^cS}||_{\ell_\infty} + ||\boldsymbol{\Gamma}^*_{S^cS}(\boldsymbol{\Gamma}^*_{SS})^{-1}(\tilde{\boldsymbol{\Gamma}}_{SS} - \boldsymbol{\Gamma}^*_{SS})||_{\ell_\infty})||(\tilde{\boldsymbol{\Gamma}}_{SS})^{-1}||_{\ell_\infty}\end{aligned}$$

$$\begin{aligned} &\leq d\lambda_1(1+H)(G+\Delta_G) \\ &\leq \tfrac{d\lambda_1 G(1+H)}{1-d\lambda_1 G}. \end{aligned} \tag{A.2}$$

Now we show (i). It is enough to show that none of the elements of $\tilde{\mathbf{\Theta}}_S$ is zero. Note that $\tilde{\mathbf{\Theta}}$ satisfies the optimality condition

$$\left(\frac{1}{2}(\tilde{\mathbf{\Sigma}}\tilde{\mathbf{\Theta}} + \tilde{\mathbf{\Theta}}\tilde{\mathbf{\Sigma}}) - \mathbf{I} + \lambda_1 \mathbf{W} \circ \tilde{\mathbf{Z}}\right)_S = \mathbf{0},$$

where $\tilde{\mathbf{Z}}$ denote the sub-gradient of $|\tilde{\mathbf{\Theta}}|$, and $||\tilde{\mathbf{Z}}||_{\max} \leq 1$. Or equivalently,

$$(\tilde{\mathbf{\Gamma}}\,\mathrm{vec}(\tilde{\mathbf{\Theta}}) - \mathrm{vec}(\mathbf{I}) + \lambda_1\,\mathrm{vec}(\mathbf{W} \circ \tilde{\mathbf{Z}}))_S = \mathbf{0}. \tag{A.3}$$

Using partitions $\tilde{\mathbf{\Theta}} = (\tilde{\mathbf{\Theta}}_S, \tilde{\mathbf{\Theta}}_{S^c}) = (\tilde{\mathbf{\Theta}}_S, \mathbf{0}), \mathbf{I} = (\mathbf{I}_S, \mathbf{I}_{S^c}) = (\mathbf{I}_S, \mathbf{0}), \mathbf{W} = (\mathbf{W}_S, \mathbf{W}_{S^c})$ and $\tilde{\mathbf{Z}} = (\tilde{\mathbf{Z}}_S, \tilde{\mathbf{Z}}_{S^c})$, we have

$$\tilde{\mathbf{\Gamma}}_{SS}\,\mathrm{vec}(\tilde{\mathbf{\Theta}}_S) - \mathrm{vec}(\mathbf{I}_S) + \lambda_1\,\mathrm{vec}(\mathbf{W}_S \circ \tilde{\mathbf{Z}}_S) = \mathbf{0}.$$

Thus, we have

$$\mathrm{vec}(\tilde{\mathbf{\Theta}}_S) = (\tilde{\mathbf{\Gamma}}_{SS})^{-1}(\mathrm{vec}(\mathbf{I}_S) - \lambda_1\,\mathrm{vec}(\mathbf{W}_S \circ \tilde{\mathbf{Z}}_S)). \tag{A.4}$$

We rewrite (A.4) as follows:

$$\begin{aligned} \mathrm{vec}(\tilde{\mathbf{\Theta}}_S) = {} & (\mathbf{\Gamma}^*_{SS})^{-1}(\mathrm{vec}(\mathbf{I}_S) - \lambda_1\,\mathrm{vec}(\mathbf{W}_S \circ \tilde{\mathbf{Z}}_S)) \\ & +((\tilde{\mathbf{\Gamma}}_{SS})^{-1} - (\mathbf{\Gamma}^*_{SS})^{-1})(\mathrm{vec}(\mathbf{I}_S) - \lambda_1\,\mathrm{vec}(\mathbf{W}_S \circ \tilde{\mathbf{Z}}_S)). \end{aligned}$$

Because $\mathbf{\Gamma}^*_{SS}\,\mathrm{vec}(\mathbf{\Theta}^*_S) = \mathrm{vec}(\mathbf{I}_S)$ and $||\mathbf{AB}||_{\max} \leq ||\mathbf{A}||_{\ell_\infty}||\mathbf{B}||_{\max}$,

$$\begin{aligned} ||\,\mathrm{vec}(\tilde{\mathbf{\Theta}}_S)||_{\min} &\geq ||\,\mathrm{vec}(\mathbf{\Theta}^*_S)||_{\min} - \lambda_1 G||\mathbf{W}_S||_{\max} - \Delta_G(1 + \lambda_1||\mathbf{W}_S||_{\max}) \\ &\geq \Psi - 2\lambda_1 G(||\mathbf{W}_S||_{\max} + dG) \\ &\geq \Psi - 2\lambda_1 G\left(\frac{2}{\Psi} + dG\right) \\ &> 0, \end{aligned}$$

where the second inequality is due to (A.1) and the third inequality is due to the following bound of $||\mathbf{W}_S||_{\max}$:

$$||\mathbf{W}_S||_{\max} \leq \frac{1}{\min_{(i,j)\in S}|\hat{\theta}^0_{ij}|} \leq \frac{1}{\Psi - (5M\lambda_0)/2} \leq \frac{2}{\Psi}. \tag{A.5}$$

Now we show (ii). The objective function in (2.1) is strictly convex since its Hessian matrix $\tilde{\mathbf{\Gamma}}$ is positive definite. So any solution that satisfies optimality condition is the unique optimal solution. Since (A.3) is already satisfied, we only

need to show the following equation to prove $\tilde{\mathbf{\Theta}} = \hat{\mathbf{\Theta}}$.

$$(\tilde{\mathbf{\Gamma}} \operatorname{vec}(\tilde{\mathbf{\Theta}}) - \operatorname{vec}(\mathbf{I}) + \lambda_1 \operatorname{vec}(\mathbf{W} \circ \tilde{\mathbf{Z}}))_{S^c} = \mathbf{0},$$

which is equivalent to

$$|\tilde{\mathbf{\Gamma}}_{S^cS} \operatorname{vec}(\tilde{\mathbf{\Theta}}_S)| \leq \lambda_1 |\mathbf{W}_{S^c}|.$$

It is sufficient to have

$$||\tilde{\mathbf{\Gamma}}_{S^cS} \operatorname{vec}(\tilde{\mathbf{\Theta}}_S)||_{\max} \leq \lambda_1 ||\mathbf{W}_{S^c}||_{\min}. \tag{A.6}$$

Partition $\mathbf{\Gamma}^* \operatorname{vec}(\mathbf{\Theta}^*) = \operatorname{vec}(\mathbf{I})$. We have $\mathbf{\Gamma}^*_{SS} \operatorname{vec}(\mathbf{\Theta}^*_S) = \operatorname{vec}(\mathbf{I}_S)$ and $\mathbf{\Gamma}^*_{S^cS} \operatorname{vec}(\mathbf{\Theta}^*_S) = \operatorname{vec}(\mathbf{I}_{S^c}) = \mathbf{0}$. So $\operatorname{vec}(\mathbf{I}_{S^c}) = \mathbf{\Gamma}^*_{S^cS}(\mathbf{\Gamma}^*_{SS})^{-1} \operatorname{vec}(\mathbf{I}_S) = \mathbf{0}$. By (A.4), we have $\tilde{\mathbf{\Gamma}}_{S^cS} \operatorname{vec}(\tilde{\mathbf{\Theta}}_S) = \tilde{\mathbf{\Gamma}}_{S^cS}(\tilde{\mathbf{\Gamma}}_{SS})^{-1}(-\lambda_1 \operatorname{vec}(\mathbf{W}_S \circ \tilde{\mathbf{Z}}_S)) + (\tilde{\mathbf{\Gamma}}_{S^cS}(\tilde{\mathbf{\Gamma}}_{SS})^{-1} - \mathbf{\Gamma}^*_{S^cS}(\mathbf{\Gamma}^*_{SS})^{-1})$ $\operatorname{vec}(\mathbf{I}_S)$, which implies

$$||\tilde{\mathbf{\Gamma}}_{S^cS} \operatorname{vec}(\tilde{\mathbf{\Theta}}_S)||_{\max} \leq (H + \Delta_H)\lambda_1 ||\mathbf{W}_S||_{\max} + \Delta_H.$$

By (A.2), we get

$$\begin{aligned} ||\tilde{\mathbf{\Gamma}}_{S^cS} \operatorname{vec}(\tilde{\mathbf{\Theta}}_S)||_{\max} &\leq \frac{d\lambda_1 G + H}{1 - d\lambda_1 G} \lambda_1 ||\mathbf{W}_S||_{\max} + \frac{d\lambda_1 G(1+H)}{1 - d\lambda_1 G} \\ &\leq (2H+1)\lambda_1 ||\mathbf{W}_S||_{\max} + 2d\lambda_1 G(1+H). \end{aligned} \tag{A.7}$$

On the other hand,

$$||\mathbf{W}_{S^c}||_{\min} = \frac{1}{\max_{(i,j)\in S^c} |\hat{\theta}^0_{ij}| + u(n,p)} \geq \frac{1}{(5M\lambda_0)/2 + u(n,p)}. \tag{A.8}$$

Then, (A.6) is obtained by combining (A.5), (A.7), (A.8) and $\lambda_0 \leq (1/5M)$ $\{\Psi/(2H + 1 + dG\Psi(1+H)) - 2u(n,p)\}$. This completes the proof that $\hat{\mathbf{\Theta}}$ recovers the true graph.

Finally, we show results of various matrix norms. Because $\tilde{\mathbf{\Theta}} = \hat{\mathbf{\Theta}}$, it is easy to use (A.1), (A.4) and (A.5) to show that

$$\begin{aligned} ||\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*||_{\max} &= ||\operatorname{vec}(\tilde{\mathbf{\Theta}}_S) - \operatorname{vec}(\mathbf{\Theta}^*_S)||_{\max} \\ &= ||((\tilde{\mathbf{\Gamma}}_{SS})^{-1} - (\mathbf{\Gamma}^*_{SS})^{-1}) \operatorname{vec}(\mathbf{I}_S) - \lambda_1 (\tilde{\mathbf{\Gamma}}_{SS})^{-1} \operatorname{vec}(\mathbf{W}_S \circ \tilde{\mathbf{Z}}_S)||_{\max} \\ &\leq \Delta_G + \lambda_1 ||\mathbf{W}_S||_{\max} ||(\tilde{\mathbf{\Gamma}}_{SS})^{-1}||_{\ell_\infty} \\ &\leq \Delta_G + \lambda_1 \frac{2}{\Psi}(\Delta_G + G) \\ &< 2\left(\frac{2}{\Psi} + dG\right) G\lambda_1. \end{aligned}$$

Then

$$||\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*||_F \leq \sqrt{s}||\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*||_{\max} < 2\sqrt{s}\left(\frac{2}{\Psi} + dG\right)G\lambda_1,$$

$$||\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*||_2 \leq \min\{\sqrt{s}, d\}||\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*||_{\max} < 2\min\{\sqrt{s}, d\}\left(\frac{2}{\Psi} + dG\right)G\lambda_1.$$

Because $\lambda_1 < \lambda_{\min}(\mathbf{\Theta}^*)/\{2\min\{\sqrt{s}, d\}(2/\Psi + dG)G\}$, $||\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*||_2 < \lambda_{\min}(\mathbf{\Theta}^*)$, so $\lambda_{\min}(\hat{\mathbf{\Theta}}) > 0$. This completes the proof.

A.2. Code

The code for implementing the SGM estimator is available at the following Github link: https://github.com/songyng/A-Simple-Method-for-Estimating-Gaussian-Graphical-Models .

References

Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* **3**, 1–122.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for High-Dimensional Data*. Springer, New York.

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

Fan, J. and Fan, Y. (2008). High-dimensional classification using features annealed independence rules. *The Annals of Statistics* **36**, 2605–2637.

Fan, J., Feng, Y. and Wu, Y. (2009). Network exploration via the adaptive LASSO and SCAD penalties. *The Annals of Applied Statistics* **3**, 521–541.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96**, 1348–1360.

Fan, J., Li, R., Zhang, C.-H. and Zou, H. (2020). *Statistical Foundations of Data Sciences*. Chapman & Hall, CRC Press, Boca Raton.

Fan, J., Xue, L. and Zou, H. (2014). Strong oracle optimality of folded concave penalized estimation. *The Annals of Statistics* **42**, 819–849.

Friedman, J., Hastie, T. and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics* **9**, 432–441.

Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1–22.

Hess, K. R., Anderson, K., Symmans, W. F., Valero, V., Ibrahim, N., Mejia, J. A. et al. (2006). Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of Clinical Oncology* **24**, 4236–4244.

Lauritzen, S. L. (1996). *Graphical models*. Clarendon Press.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics* **34**, 1436–1462.

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

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

Singh, D., Febbo, P. J., Ross, K., Jackson, D. G., Manola, J., Ladd, C. et al. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* **1**, 203–209.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika* **94**, 19–35.

Zhang, T. and Zou, H. (2014). Sparse precision matrix estimation via Lasso penalized D-trace loss. *Biometrika* **101**, 103–120.

Zhao, P. and Yu, B. (2006). On model selection consistency of Lasso. *Journal of Machine Learning Research* **7**, 2541–2567.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association* **101**, 1418–1429.

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics* **36**, 1509–1533.

Yiyi Yin

School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA.

E-mail: yinxx307@umn.edu

Yang Song

School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA.

E-mail: song0214@umn.edu

Hui Zou

School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA.

E-mail: zouxx019@umn.edu

(Received August 2021; accepted February 2023)