MARS: A second-order reduction algorithm for high-dimensional sparse precision matrices estimation[*]

Qian Li   QIANXA.LI@CONNECT.POLYU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

Binyan Jiang   BY.JIANG@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

Defeng Sun   DEFENG.SUN@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

Editor:

Abstract

Estimation of the precision matrix (or inverse covariance matrix) is of great importance in statistical data analysis. However, as the number of parameters scales quadratically with the dimension p, computation becomes very challenging when p is large. In this paper, we propose an adaptive sieving reduction algorithm to generate a solution path for the estimation of precision matrices under the ℓ_1 penalized D-trace loss, with each subproblem being solved by a second-order algorithm. In each iteration of our algorithm, we are able to greatly reduce the number of variables in the problem based on the Karush-Kuhn-Tucker (KKT) conditions and the sparse structure of the estimated precision matrix in the previous iteration. As a result, our algorithm is capable of handling datasets with very high dimensions that may go beyond the capacity of the existing methods. Moreover, for the sub-problem in each iteration, other than solving the primal problem directly, we develop a semismooth Newton augmented Lagrangian algorithm with global linear convergence on the dual problem to improve the efficiency. Theoretical properties of our proposed algorithm have been established. In particular, we show that the convergence rate of our algorithm is asymptotically superlinear. The high efficiency and promising performance of our algorithm are illustrated via extensive simulation studies and real data applications, with comparison to several state-of-the-art solvers.

Keywords: Adaptive sieving reduction, Precision matrix, Semismooth Newton, Sparsity, Solution path

1. Introduction

The estimation of high dimensional sparse precision matrices has been a central topic in statistical learning, with a wide range of applications such as genomics (Wille et al., 2004;

1

Li and Gui, 2006), image analysis (Li, 2009), among others. Owing to the fast development of data engineering and technology, modern datasets are oftentimes having much higher dimensions than before, and the estimation of the precision matrices becomes more challenging as the number of variables scales quadratically in the dimension p. For example, in the breast cancer data set studied in our numerical experiments, p is equal to $22,283$, and the number of variables is nearly 250 million. Many existing algorithms or solvers could easily fail to produce a meaningful estimator in this case. Highly efficient algorithms with sound theoretical guarantees are thus in great need of meeting the computation requirement of the time.

Assume that $x_1, \cdots, x_n$ are independently identically distributed (iid) variables with an observation $X = (X_1, \cdots, X_n)$ from a p-dimensional Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, where μ and Σ are population mean and covariance matrix respectively. The precision matrix Θ (also known as the concentration matrix) is defined as the inverse of Σ, namely, $\Theta = \Sigma^{-1}$. For the undirected Gaussian graphical model, it is well known that the conditional independence of variables is directly reflected in the zero components of the precision matrix (Lauritzen, 1996). Specifically, for any $1 \leq i \neq j \leq p$, $\Theta_{ij} = 0$ if and only if x_i is conditionally independent of x_j given all other variables x_k, $k \neq i, j$, $1 \leq k \leq p$.

So far, many methods have been proposed to estimate the high-dimensional sparse precision matrix. Meinshausen et al. (2006) estimated the conditional independence restrictions of each node of an undirected graph model by identifying the zero patterns of the associated precision matrix through a sequence of lasso penalized least square regression models. Yuan (2010) studied the above regression models via a Dantzig selector. Later, Cai et al. (2011) proposed a constrained ℓ_1 minimization approach and established the convergence rates under different norms. However, among the methods mentioned above, none of them are truly treating the precision matrix as a whole. There is another well-known estimator called the lasso penalized Gaussian likelihood estimator (Yuan and Lin, 2007; Banerjee et al., 2008; Friedman et al., 2008), also known as graphical lasso or glasso. Given the sample covariance matrix $\widehat{\Sigma} \in \mathbb{S}^p$ and a tuning parameter $\lambda > 0$, the glasso estimator is obtained by minimizing the l_1 penalized log-likelihood function:

$$\min_{\Omega \in \mathbb{S}^p_+} \left\{ \mathrm{tr}(\Omega \widehat{\Sigma}) - \mathrm{logdet}(\Omega) + \lambda \left\| \Omega \right\|_1 \right\}, \tag{1}$$

where $\mathbb{S}^p_+$ is the space of $p \times p$ real symmetric positive definite matrices, $\mathrm{tr}(\cdot)$ and $\left\| \cdot \right\|_1$ are the trace and ℓ_1-norm, respectively. Researchers have designed different optimization algorithms to solve this problem. Some first-order methods have been applied to solve (1), such as the interior point method (Yuan and Lin, 2007), the block coordinate descent method (Banerjee et al., 2008; Friedman et al., 2008) and the alternating linearization method (Scheinberg et al., 2010). To solve the graphical lasso problem more efficiently, some second-order methods such as the quadratic approximation method (Hsieh et al., 2014) and the Newton-like methods (Oztoprak et al., 2012) were also developed. However, these two methods may not be the best choice. For the quadratic approximation method, the computational complexity could be up to $O(p^3)$ unless a block diagonal structure of the estimated precision matrix is detected. As for the Newton-like methods, the algorithms are more or less heuristic and related convergence properties are yet to be explored. We note that the graphical lasso brings much complexity to the calculation because its objective

2

function contains a log determine term. Although Witten et al. (2011) provides a strategy to further improve the efficiency by identifying the block diagonal structure, which has been implemented in the glasso package, such a strategy can easily fail in practice, especially when the tuning parameter is small. The total computation time on the breast cancer dataset shown in Table 7 can further illustrate such a conclusion.

Recently, a ℓ_1-penalized D-trace loss estimator was proposed in Zhang and Zou (2014); Liu and Luo (2015). This new estimator is obtained by solving a convex composite optimization problem, which involves a quadratic loss function with a ℓ_1-regularized penalty:

$$\min_{\Omega \in \mathbb{S}^p} \left\{ \frac{1}{2} \text{tr}(\Omega \widehat{\Sigma} \Omega^T) - \text{tr}(\Omega) + \lambda \|\Omega\|_{1,\text{off}} \right\}, \tag{2}$$

where $\mathbb{S}^p$ is the space of $p \times p$ real symmetric matrices and $\|\cdot\|_{1,\text{off}}$ is the off-diagonal ℓ_1-norm, i.e., $\|\Omega\|_{1,\text{off}} = \sum_{i \neq j} |\Omega_{i,j}|$. Zhang and Zou (2014) derived the convergence rates of this new estimator and showed that it could be comparable with the graphical lasso. Clearly, this new loss function is simpler in form than the graphical lasso, which could bring great convenience to calculation. In particular, when dealing with big data where the dimension is very large, computational efficiency owing to the simple form of the loss function would be a favorable feature for practical applications. However, existing methods for solving (2) are all first-order methods, such as the coordinate descent method (Liu and Luo, 2015) and the alternating direction method of multipliers (ADMM) (Zhang and Zou, 2014; Wang and Jiang, 2020). Although it has been shown that these methods are able to handle data with a larger dimension than glasso (Wang and Jiang, 2020), these algorithms suffer from some well-known deficiencies of first-order methods such as relatively slow convergence rates.

In this paper, we first propose an adaptive sieving reduction strategy to generate a solution path of precision matrices. The main idea of this algorithm is to reduce the number of variables in each iteration to improve efficiency. Firstly, by collecting the indexes of the non-zero components in the previous solution, we can obtain an initial non-zero index set for the new problem with a smaller tuning parameter. Secondly, we use the KKT residual for adaptive sieving to obtain another index set of non-zero components for the new problem and then solve it until obtaining a satisfactory solution. This strategy reduces the number of estimated components from p^2 to t in each iteration, where t is the number of non-zero elements in the upper-triangular precision matrix and may not be larger than $p+n(n-1)/2$ to ensure the validity of the estimation in some cases. Thus, our algorithm is not only significantly efficient but also can solve the problem of insufficient data storage space for huge data to a certain extent. Specifically, the problem we solved in the main loop is

$$\min_{\Omega \in \mathbb{S}^p} \left\{ \frac{1}{2} \text{tr}(\Omega \widehat{\Sigma} \Omega^T) - \text{tr}(\Omega) + \lambda \|\Omega\|_{1,\text{off}} - \langle \Delta, \Omega \rangle \mid \Omega_{\bar{I}} = 0 \right\}, \tag{3}$$

where $\Delta \in \mathbb{S}^p$ is an error matrix with $\|\Delta\| \leq \epsilon$ for some small $\epsilon > 0$, $\|\cdot\|$ is the ℓ_2 norm and $\bar{I}$ is the complement set of the non-zero components index set. Since we can always use the KKT residual to find an optimal index set, the difference between (2) and (3) is only the small term $-\langle \Delta, \Omega \rangle$, and we will show the optimal solution of (3) is an approximate solution of (2) with a controllable error $O(\epsilon)$. Then, we develop an efficient second-order algorithm, or more precisely, a semismooth Newton augmented Lagrangian algorithm, to solve (3) by finding a unique optimal solution of its dual. In the design of the algorithm, we use

the high dimensional setting of the data, i.e., p is much larger than n, to skillfully rewrite the original problem to obtain a dual problem where the dimension of the dual variable is $p \times n$ instead of $p \times p$. More importantly, due to the facts that the piecewise linear-quadratic structure of the primal problem provides asymptotically superlinear convergence rate of the augmented Lagrange method (Li et al., 2018), and the strong convexity of the dual problem, our algorithm only needs a few iterations to obtain a desirable solution. Although commonly, the computation cost of second-order algorithms in each step could be much more expensive than that of first-order algorithms, by making use of the second-order sparsity of the augmented Lagrangian functions here in our design, our algorithm is computationally as cheap as first-order algorithms. Moreover, we provide a technique to determine the maximum λ. This technique limits the choice of λ to avoid unnecessary waste of time for generating a solution path. In subsequent numerical experiments, we shall see that our algorithm significantly outperforms several state-of-the-art solvers and is competent to handle huge-scale problems.

We highlight the main contributions of this paper as follows:

1. We develop a dual approach for the precision matrix estimation. By equivalently rewriting the primal problem under the high-dimensional setting where p is much large than n, we obtain a dual problem where the dimension of a dual variable is $p \times n$ instead of $p \times p$. Such an approach can fundamentally improve the efficiency when n is much smaller than p.

2. This is the first attempt to implement the adaptive sieving strategy and the semismooth Newton augmented Lagrangian algorithm for variables with matrix forms. With the development of the dual approach, this combination can avoid some time-consuming operations in the main loops, such as the multiplication of two $p \times n$ matrices. More importantly, although we are adopting a second-order method for solving the subproblems in each iteration, the computational complexity of our proposed algorithm is comparable to first-order algorithms. The promising numerical performance of our algorithm is also theoretically justified by the global linear convergence and asymptotically superlinear convergence rate we established.

3. We have developed a R package for applications to estimate the sparse precision matrix effectively. Comparing with other existing solvers/packages, our algorithm is much more efficient and is able to handle datasets with much higher dimensions. For instance, on a publicly available breast cancer data set, our algorithm can be up to more than 20 times faster than the popular glasso package (Friedman et al., 2008; Witten et al., 2011) for estimating a precision matrix with five-fold cross-validation included.

The remaining subsequent arrangements are as follows. In section 2, we will develop an adaptive sieving reduction strategy for generating solution paths. In section 3, we derive an inexact augmented Lagrangian method (ALM) to solve the dual problem of the inner problem in section 2. Then for the subproblem in the inexact ALM, we design a semismooth Newton algorithm to obtain an expected solution. In section 4, after introducing some algorithms, by comparing with the introduced algorithms and several state-of-the-art solvers,

we will demonstrate the promising performance of our algorithm through some numerical experiments and the analysis of two real datasets. We conclude our paper in section 5.

Notation and preliminaries: Throughout this paper, $\mathcal{X}$ and $\mathcal{Y}$ represent finite-dimensional real Euclidean spaces. We use $\|\cdot\|_F$ to denote the Frobenius norm, and its induced inner product is denoted by $\langle \cdot, \cdot \rangle$. Specificly, let $X = (X_{ij})_{1 \leq i \leq p, \, i \leq j \leq n}$ and $Y = (Y_{ij})_{1 \leq i \leq p, \, i \leq j \leq n}$ be two real matrices, $\|X\|_F = \left(\sum_{i,j} X_{ij}^2\right)^{1/2}$ and $\langle X, Y \rangle = \mathrm{tr}\left(X^T Y\right) = \sum_{i,j} X_{ij} Y_{ij}$, where X^T denotes the transpose of X. The ℓ_1 norm denoted by $\|\cdot\|_1$, namely, $\|X\|_1 = \sum_{ij} |X_{ij}|$, and the spectral norm is denoted by $\|\cdot\|_{(2)}$. The Jacobian of f at $X \in D_F$ is denoted as $F'(X)$, where $D_F = \{X \mid F(\cdot) \text{ is differentiable at } X\}$. We also use "$\circ$" to denote the Hardmard product, i.e., $(X \circ Y)_{ij} = X_{ij} Y_{ij}$ and $\sup\{\cdot\}$ to denote the supremum. The cardinal number of a real vector or matrix V is denoted by $|V|$, but for a one-component variable v, $|v|$ denotes its absolute value.

Let $f : \mathcal{X} \to (-\infty, +\infty]$ be an extended real-valued closed and proper convex function on the real Euclidean space $\mathcal{X}$. The unique optimal solution of the Moreau-Yosida regularization of f at $X \in \mathcal{X}$ is called the proximal point mapping of X associated with f, denoted by $\mathrm{Prox}_f(X)$, where the Moreau-Yosida regularization is defined as

$$\mathcal{H}_f(X) := \min_{Y \in \mathcal{X}} \left\{ f(Y) + \frac{1}{2} \|Y - X\|_F^2 \right\}.$$

For later use, we present some useful propositions of the Moreau-Yosida regularization first. $\mathcal{H}_f(\cdot)$ is continuously differentiable, and furthermore, the gradient of $\mathcal{H}_f(\cdot)$ at $X \in \mathcal{X}$ is known as in the form of $\nabla \mathcal{H}_f(X) = X - \mathrm{Prox}_f(X)$. Another important and useful formula is the Moreau decomposition, which is, for any $X \in \mathcal{X}$, $X = \mathrm{Prox}_f(X) + \mathrm{Prox}_{f^*}(X)$, where f^* is the Fenchel conjugate of f and is defined by $f^*(Y) = \sup\{\langle Y, X \rangle - f(X) \mid X \in \mathcal{X}\}$ for any $Y \in \mathcal{X}$. It can be shown that the pointwise supremum function of a collection of convex (closed) function is also convex (closed). Hence, as long as f is a closed convex function, f^* is always convex and closed. In addition, we should emphasize here that Prox_f is globally Lipschitz continuous with modulus 1 (Lemaréchal and Sagastizábal, 1997).

2. An adaptive sieving reduction strategy

In this section, based on the adaptive sieving strategy (Lin et al., 2020), we will develop a reduction algorithm to generate solution paths for the problem (2). Since our algorithm is designed for <u>M</u>atrix estimation via an <u>A</u>daptive sieving <u>R</u>eduction and a <u>S</u>emismooth Newton algorithm, we call our algorithm **MARS**. The main idea of this strategy is to reduce the number of variables to be consistent with the number of non-zero components of the optimal solution, which can not only greatly improve the efficiency of an algorithm, but also save plenty of storage space.

As we mentioned in the introduction, we will solve (2) by finding an optimal solution of its dual after obtaining an equivalent problem of (2). Specifically, it can be equivalently written as

$$\min_{\Omega \in \mathbb{S}^p} \left\{ \frac{1}{2} \|\Omega A\|_F^2 - \langle \Omega, I_p \rangle + \lambda \|\Omega\|_{1,\mathrm{off}} \right\}, \tag{4}$$

5

where A is a real matrix with rank n such that $AA^T = \widehat{\Sigma}$. Note that we can always use the singular value decomposition for a given sample to find such matrix A. Without loss of generality, we assume that A is a $p \times n$ matrix with rank n. For later use, we also denote $\theta(\Omega) := \|\Omega\|_{1,\text{off}}, \forall \Omega \in \mathbb{S}^p$. Moreover, we further denote the optimal solution set of (4) by Θ_λ, and the associated proximal residual function by

$$R_\lambda(\Omega) := h(\Omega) + \text{Prox}_{\delta_{B_\lambda}}(\Omega - h(\Omega)), \quad \forall \Omega \in \mathbb{S}^p,$$

where $h(\Omega) := \frac{1}{2}(\Omega\widehat{\Sigma} + \widehat{\Sigma}\Omega) - I_p$ with I_p being the p dimensional identity matrix, and δ_{B_λ} is the indicator function with $B_\lambda = \{Z \in \mathbb{S}^p \mid Z_{ii} = 0, |Z_{ij}| \leq \lambda, i, j = 1, \cdots, p, i \neq j\}$, i.e., $\delta_{B_\lambda}(Z) = 0$ for any $Z \in B_\lambda$ and $\delta_{B_\lambda}(Z) = +\infty$ otherwise. By the KKT conditions, we know that $\widetilde{\Omega} \in \Theta_\lambda$ if and only if $R_\lambda(\widetilde{\Omega}) = 0$.

Detailed steps of our adaptive sieving reduction strategy are given in Algorithm 1. For a sequence of positive tuning parameters sorted in descending order, we first solve the problem (4) inexactly with λ equal to the largest parameter to obtain the initial non-zero components index set and an approximate solution with a tolerable error ϵ. Then, for the next smaller λ, we continuously use the KKT residual to perform adaptive sieving to obtain a new non-zero components index set, while updating its solution until a desirable solution is obtained. Such a procedure is performed for all the tuning parameters until the algorithm stops (we will show the while loop can terminate in a finite number of iterations in the proof of theorem 3). Noted that the existence of Δ_0 and $\{\Delta_i^l\}$ in Steps 1 and 10 means the minimization problems are solved inexactly. Both of them are not given in prior but are automatically obtained when the original minimization problems are solved inexactly.

Next, we will establish the convergence of Algorithm 1. Before that, we present the following proposition to interpret the connection between the optimal solution in Step 1 of Algorithm 1 and an approximate solution of (4) firstly.

Proposition 1 *The optimal solution $\Omega^*(\lambda)$ of*

$$\min_{\Omega \in S^p} \left\{ \frac{1}{2} \|\Omega A\|_F^2 - \langle \Omega, I_p \rangle + \lambda \|\Omega\|_{1,\text{off}} - \langle \Delta, \Omega \rangle \right\}, \tag{5}$$

with $\|\Delta\|_F \leq \epsilon$ can be equivalently found by

$$\Omega^*(\lambda) = \text{Prox}_{\lambda\theta}(\widehat{\Omega}(\lambda) - h(\widehat{\Omega}(\lambda))), \tag{6}$$

where $\widehat{\Omega}(\lambda)$ is an approximate solution of

$$\min_{\Omega \in S^p} \left\{ \frac{1}{2} \|\Omega A\|_F^2 - \langle \Omega, I_p \rangle + \lambda \|\Omega\|_{1,\text{off}} \right\} \tag{7}$$

such that

$$\left\| R_\lambda(\widehat{\Omega}(\lambda)) \right\|_F \leq \frac{\epsilon}{\sqrt{2}\left(1 + \|\widehat{\Sigma}\|_F\right)}. \tag{8}$$

Proof Denote the optimal solution of (7) by $\widetilde{\Omega}(\lambda)$. For any $i = 1, 2, \cdots$, let $\Omega^i \to \widetilde{\Omega}(\lambda)$, and we further define

$$\Delta^i := R_\lambda(\Omega^i) + h\left(\text{Prox}_{\lambda\theta}(\Omega^i - h(\Omega^i))\right) - h(\Omega^i). \tag{9}$$

6

Algorithm 1 Adaptive sieving reduction strategy for generating a solution path.

Require:

    A real matrix $A \in \mathbb{R}^{p \times n}$ and a tolerance constant $\epsilon \geq 0$;

    A sequence of tuning parameters $\lambda_0 > \lambda_1 > \cdots > \lambda_k > 0$ with $\lambda_{\max} \geq \lambda_0$;

Ensure:

    A solution path: $\Omega^*(\lambda_0), \Omega^*(\lambda_1), \cdots, \Omega^*(\lambda_k)$;

1: **Initialization:**

    For $\lambda_0 > 0$, solve

$$\Omega^*(\lambda_0) \in \mathrm{argmin}_{\Omega \in S^p} \left\{ \frac{1}{2} \|\Omega A\|_F^2 - \langle \Omega, I_p \rangle + \lambda_0 \|\Omega\|_{1,\mathrm{off}} - \langle \Delta_0, \Omega \rangle \right\},$$

    where $\Delta_0 \in S^p$ is an error matrix such that $\|\Delta_0\|_F \leq \epsilon$. Then let

$$I^*(\lambda_0) := \{(i,j) \mid \Omega^*(\lambda_0)_{ij} \neq 0, \ i,j = 1, \cdots, p\};$$

2: **Main loop:**

3: **for** $i = 1; \ i < k + 1; \ i{+}{+}$ **do**

4:     Set $\Omega^0(\lambda_i) = \Omega^*(\lambda_{i-1})$ and $I^0(\lambda_i) = I^*(\lambda_{i-1})$;

5:     Calculate $R_{\lambda_i}(\Omega^0(\lambda_i))$ and set $l = 0$;

6:     **while** $\left\| R_{\lambda_i}(\Omega^l(\lambda_i)) \right\|_F > \epsilon$ **do**

7:         $l{+}{+}$;

8:         Create $J^l(\lambda_i)$ by

$$J^l(\lambda_i) = \left\{ (i,j) \in \bar{I}^{l-1}(\lambda_i) \mid -\left(h(\Omega^{l-1}(\lambda_i)) \right)_{ij} \notin \lambda_i \left(\partial\theta(\Omega^{l-1}(\lambda_i)) + \frac{\epsilon}{\lambda_i \sqrt{2|\bar{I}^{l-1}(\lambda_i)|}} \mathbb{B}\infty \right)_{ij} \right\},$$

        where $\bar{I}^{l-1}(\lambda_i)$ denotes the complement of $I^{l-1}(\lambda_i)$ and $\mathbb{B}\infty$ is the infinity norm unit ball;

9:         Update $I^l(\lambda_i) = I^{l-1}(\lambda_i) \cup J^l(\lambda_i)$;

10:        Solve

$$\Omega^l(\lambda_i) \in \mathrm{argmin}_{\Omega \in S^p} \left\{ \frac{1}{2} \|\Omega A\|_F^2 - \langle \Omega, I_p \rangle + \lambda_i \|\Omega\|_{1,\mathrm{off}} - \left\langle \Delta_i^l, \Omega \right\rangle \mid \Omega_{\bar{I}^l(\lambda_i)} = 0 \right\},$$

        where $\Delta_i^l \in S^p$ is an error vector such that $\left\| \Delta_i^l \right\|_F \leq \epsilon/\sqrt{2}$ and $(\Delta_i^l)_{\bar{I}^l(\lambda_i)} = 0$;

11:        Compute $R_{\lambda_i}(\Omega^l(\lambda_i))$;

12:     **end while**

13:     Set $\Omega^*(\lambda_i) = \Omega^l(\lambda_i)$, $I^*(\lambda_i) = I^l(\lambda_i)$ and $\Delta_i = \Delta_i^l$;

14: **end for**

15: **return** Ω^*;

From Lemma 4.5 of Du (2015), we know that $\lim_{i \to \infty} \left\| \Delta^i \right\|_F = 0$. This implies the existence of $\widehat{\Omega}(\lambda)$ satisfying the inequality (8). Next we verify that $\Omega^*(\lambda)$ and $\widehat{\Omega}(\lambda)$ are connected with a proximal point mapping as in 6. Beginning with the definition of R_λ, we have

$R_\lambda(\widehat{\Omega}(\lambda)) = \widehat{\Omega}(\lambda) - \mathrm{Prox}_{\lambda\theta}(\widehat{\Omega}(\lambda) - h(\widehat{\Omega}(\lambda))) = \widehat{\Omega}(\lambda) - \Omega^*(\lambda)$. By combining this with the equation (6), we obtain

$$R_\lambda(\widehat{\Omega}(\lambda)) - h(\widehat{\Omega}(\lambda)) \in \lambda\partial\theta(\Omega^*(\lambda)).$$

Now, let us define $\Delta := R_\lambda(\widehat{\Omega}(\lambda)) + h(\Omega^*(\lambda)) - h(\widehat{\Omega}(\lambda))$. It is easily seen that

$$\Delta \in h(\Omega^*(\lambda)) + \lambda\partial\theta(\Omega^*(\lambda)),$$

which means $\Omega^*(\lambda)$ is an optimal solution of (5) with the given Δ. Besides, we have

$$\|\Delta\|_F = \left\| R_\lambda(\widehat{\Omega}(\lambda)) + h(\Omega^*(\lambda)) - h(\widehat{\Omega}(\lambda)) \right\|_F \leq \left(1 + \left\|\widehat{\Sigma}\right\|_F\right) \left\| R_\lambda(\widehat{\Omega}(\lambda)) \right\|_F \leq \epsilon/\sqrt{2}.$$

■

Remark 2 *Proposition 1 presents the connection between $\Omega^*(\lambda)$ and $\widehat{\Omega}(\lambda)$. Following similar arguments as in inequality (12), we immediately have $\left\|\Omega^*(\lambda) - \widetilde{\Omega}(\lambda)\right\| = O(\epsilon)$, where the tolerance error ϵ can be set to be arbitrarily small.*

As for the Step 10 in Algorithm 1, a more detailed interpretation can be found from the proof of the following theorem.

Theorem 3 *The solution path $\{\Omega^*(\lambda_i) \mid i = 0, 1, \cdots, k\}$ generated by Algorithm 1 are approximate optimal solutions of a sequence problems of form*

$$\min_{\Omega \in S^p} \left\{ \frac{1}{2} \|\Omega A\|_F^2 - \langle \Omega, I_p \rangle + \lambda_i \|\Omega\|_{1,\mathrm{off}} \right\},$$

with $\|R_{\lambda_i}(\Omega^*(\lambda_i))\|_F \leq \epsilon$, $i = 0, 1, \cdots, k$.

Proof Firstly, we show the index $J^l(\lambda_i)$ is nonempty when $\left\| R_{\lambda_i}(\Omega^l(\lambda_i)) \right\|_F > \epsilon$. We use the proof by contradiction. Suppose that $J^l(\lambda_i) = \emptyset$. Then we have

$$-\left(h(\Omega^{l-1}(\lambda_i))\right)_{ij} \in \lambda_i \left(\partial\theta(\Omega^{l-1}(\lambda_i)) + \frac{\epsilon}{\lambda_i\sqrt{2|\bar{I}^{l-1}(\lambda_i)|}}\mathbb{B}\infty \right)_{ij}, \quad \forall(i,j) \in \bar{I}^{l-1}(\lambda_i).$$

Thus, there is a matrix $\widehat{\Delta}_i^l \in \mathbb{S}^p$ with $(\widehat{\Delta}_i^l)_{I^{l-1}(\lambda_i)} = 0$ and $\left\|\widehat{\Delta}_i^l\right\|_\infty \leq \frac{\epsilon}{\sqrt{2|\bar{I}^{l-1}(\lambda_i)|}}$ such that

$$-\left(h(\Omega^{l-1}(\lambda_i)) - \widehat{\Delta}_i^l\right)_{ij} \in \lambda_i \left(\partial\theta(\Omega^{l-1}(\lambda_i)) \right)_{ij}, \quad \forall(i,j) \in \bar{I}^{l-1}(\lambda_i). \tag{10}$$

Since $\Omega^{l-1}(\lambda_i)$ is an optimal solution of

$$\min_{\Omega \in S^p} \left\{ \frac{1}{2} \|\Omega A\|_F^2 - \langle \Omega, I_p \rangle + \lambda_i \|\Omega\|_{1,\mathrm{off}} - \left\langle \Delta_i^{l-1}, \Omega \right\rangle \mid \Omega_{\bar{I}^{l-1}(\lambda_i)} = 0 \right\},$$

8

where Δ_i^{l-1} is an error matrix with $\left\|\Delta_i^{l-1}\right\|_F \leq \epsilon/\sqrt{2}$ and $(\Delta_i^{l-1})_{\bar{I}^{l-1}(\lambda_i)} = 0$, by the KKT conditions, we know that there exists $\Lambda \in \mathbb{S}^p$ with $\Lambda_{I^{l-1}(\lambda_i)} = 0$ such that

$$\begin{cases} 0 \in h(\Omega^{l-1}(\lambda_i)) - \Delta_i^{l-1} + \lambda_i \partial\theta(\Omega^{l-1}(\lambda_i)) - \Lambda, \\ \left(\Omega^{l-1}(\lambda_i)\right)_{\bar{I}^{l-1}(\lambda_i)} = 0. \end{cases} \tag{11}$$

Thus, combining (10) and (11), we obtain

$$-h(\Omega^{l-1}(\lambda_i)) + \widetilde{\Delta}_i^{l-1} \in \lambda_i \partial\theta(\Omega^{l-1}(\lambda_i)),$$

where $\widetilde{\Delta}_i^{l-1} \in \mathbb{S}^p$ with $(\widetilde{\Delta}_i^{l-1})_{I^{l-1}(\lambda_i)} = (\Delta_i^{l-1})_{I^{l-1}(\lambda_i)}$ and $(\widetilde{\Delta}_i^{l-1})_{\bar{I}^{l-1}(\lambda_i)} = (\widehat{\Delta}_i^{l-1})_{\bar{I}^{l-1}(\lambda_i)}$. This means

$$\Omega^{l-1}(\lambda_i) = \text{Prox}_{\lambda_i\theta}(\Omega^{l-1}(\lambda_i) - h(\Omega^{l-1}(\lambda_i)) + \widetilde{\Delta}_i^{l-1}).$$

Therefore, we have

$$\left\|R_{\lambda_i}(\Omega^{l-1}(\lambda_i))\right\|_F$$
$$= \left\|\text{Prox}_{\lambda_i\theta}(\Omega^{l-1}(\lambda_i) - h(\Omega^{l-1}(\lambda_i)) + \widetilde{\Delta}_i^{l-1}) - \text{Prox}_{\lambda_i\theta}(\Omega^{l-1}(\lambda_i) - h(\Omega^{l-1}(\lambda_i)))\right\|_F$$
$$\leq \left\|\widetilde{\Delta}_i^{l-1}\right\|_F \leq \epsilon,$$

where the first inequality is followed by the property that the proximal mapping is globally Lipschitz continuous with modulus 1. Hence, a contradiction is found. Then we can say $J^l(\lambda_i) \neq \emptyset$ if and only if $\left\|R_{\lambda_i}(\Omega^l(\lambda_i))\right\| > \epsilon$. Since the total number of components of Ω is finite, the while loop in Algorithm 1 will terminate in a finite number of iterations. Additionally, by the KKT conditions, we have $\Omega^*(\lambda_0) = \text{Prox}_{\lambda_0\theta}(\Omega^*(\lambda_0)) - h(\Omega^*(\lambda_0))) + \Delta_0)$. Thus

$$\|R_{\lambda_0}(\Omega^*(\lambda_0))\|_F \leq \|\Delta_0\|_F \leq \epsilon.$$

∎

The above proposition and theorem indicate that Algorithm 1 is well defined. In the following corollaries we show that the approximate solution path generated by Algorithm 1 is positive definite with probability tending to 1 under some standard assumptions. Before that, we give some notations and describe some assumptions. Suppose that the true precision matrix Θ is sparse and its minimum eigenvalue $\gamma_{\min}(\Theta) > r$ with $r > 0$. For the associated graph, we denote the maximum node degree and number of edges by m and s respectively. Then, we denote $s_\theta = \min\{\sqrt{s+p}, m\}$ to describe the sparse level of Θ. For convenience, we write $a_n = \Xi(b_n)$ to denote $a_n \geq c_1 b_n$ for some constants $c_1 > 0$ and $a_n = O(b_n)$ to denote $a_n \leq c_2 b_n$ for some constants $c_2 > 0$. We also give the following two conditions for later use:

$$(\mathbf{C_1}) \quad \theta_{\min} = \Xi\left(m\sqrt{\log p/n}\right),$$

$$(\mathbf{C_2}) \quad \theta_{\min} = \Xi(mp^{\eta/2q}\sqrt{n}),$$

where we use $\theta_{\min}$ to denote the minimum non-zero absolute value of Θ.

9

Now, we assume that the following irrepresentability condition holds:

$$\max_{v \in \overline{\Psi}} \left\| \Gamma_{v,\Psi} (\Gamma_{\Psi,\Psi})^{-1} \right\|_1 = 1 - \alpha,$$

for $0 < \alpha < 1$, where Ψ is the support of Θ, $\overline{\Psi}$ is its complement, $\Gamma = \frac{1}{2} \Sigma \oplus \Sigma$, and $\oplus$ denotes the Kronecker matrix sum. Similarly to the assumption in Section 3.3 of Ravikumar et al. (2011), for simplicity, we assume in addition that $\left\| \Gamma_{\Psi,\Psi}^{-1} \right\|_{1,\infty}$, $\|\Sigma\|_{1,\infty}$ and α are not scaling with p and m, where $\|X\|_{1,\infty} = \max_i \sum_j |X_{ij}|$ for any real matrix X.

Corollary 4 *Assume that $x_1, \cdots, x_n$ are iid variables from a sub-Gaussian distribution with covariance Σ and for the i-th coordinate x_j^i of any random variable x_j, $j \in [1, n]$, the following sub-Gaussian tails with parameter σ_s condition holds:*

$$E\left[\exp\{t x_j^i \Sigma_{ii}^{-1/2}\} \right] \leq \exp\{\sigma_s^2 t^2 / 2\}, \quad t \in \mathbb{R}.$$

Let $\bar{\lambda}_n = C_e \sqrt{\frac{\eta \log p}{n}}$ with a constant C_e sufficiently large for some $\eta > 2$ and the sample size $n = \Xi\left((s_\theta m / r)^2 \eta \log p \right)$ under $(\mathbf{C_1})$. Then with probability $1 - 1/p^{\eta-2}$, we have

$$\left\| \Omega^*(\bar{\lambda}_n) - \Theta \right\|_{(2)} = O(s_\theta m \sqrt{\eta \log p / n}),$$

where $\Omega^*(\bar{\lambda}_n)$ is generated by Algorithm 1 such that $\left\| R_{\bar{\lambda}_n}(\Omega^*(\bar{\lambda}_n)) \right\|_F \leq \epsilon$ for a small enough tolerance $\epsilon \geq 0$.

Proof Denote $\widetilde{\Omega}(\bar{\lambda}_n)$ as an optimal solution of (4) with tuning parameter $\bar{\lambda}_n$. Then by Proposition 1 and Theorem 3, we have

$$\Omega^*(\bar{\lambda}_n) = \text{Prox}_{\lambda\theta}(\widehat{\Omega}(\bar{\lambda}_n) - h(\widehat{\Omega}(\bar{\lambda}_n))),$$

where $\widehat{\Omega}(\bar{\lambda}_n)$ is an approximate solution of $\widetilde{\Omega}(\lambda_n)$ with $\left\| R_{\bar{\lambda}_n}(\widehat{\Omega}(\bar{\lambda}_n)) \right\|_F \leq \epsilon / \left(1 + \left\|\widehat{\Sigma}\right\|_F\right)$. It is reasonable to assume that $\left\| R_{\bar{\lambda}_n}(\widehat{\Omega}(\bar{\lambda}_n)) \right\|_F > 0$, since if it is not, $\widehat{\Omega}(\bar{\lambda}_n)$ is exactly an optimal solution, and so is $\Omega^*(\bar{\lambda}_n)$. Then, there is a constant $\iota \geq 0$ such that

$$\left\| \Omega^*(\bar{\lambda}_n) - \widetilde{\Omega}(\bar{\lambda}_n) \right\|_{(2)} \leq \left\| \Omega^*(\bar{\lambda}_n) - \widetilde{\Omega}(\bar{\lambda}_n) \right\|_F \tag{12}$$

$$= \left\| \text{Prox}_{\lambda\theta}(\widehat{\Omega}(\bar{\lambda}_n) - h(\widehat{\Omega}(\bar{\lambda}_n))) - \text{Prox}_{\lambda\theta}(\widetilde{\Omega}(\bar{\lambda}_n) - h(\widetilde{\Omega}(\bar{\lambda}_n))) \right\|_F$$

$$\leq (1 + \left\|\widehat{\Sigma}\right\|_F) \left\| \widehat{\Omega}(\bar{\lambda}_n) - \widetilde{\Omega}(\bar{\lambda}_n) \right\|_F$$

$$\leq \frac{\left\| \widehat{\Omega}(\bar{\lambda}_n) - \widetilde{\Omega}(\bar{\lambda}_n) \right\|_F}{\left\| R_{\bar{\lambda}_n}(\widehat{\Omega}(\bar{\lambda}_n)) \right\|_F} \epsilon$$

$$\leq \iota \epsilon,$$

where the second inequality is followed by the proximal point mapping is globally Lipschitz continuous with modulus 1 and the last inequality is induced by the fact that the optimal solution set is compact and lemma 12 (this lemma implies the distance between any

10

feasible solution and the optimal solution set can be bounded by the associated KKT residual). Then, by $\left\|\Omega^*(\bar{\lambda}_n) - \Theta\right\|_{(2)} \leq \left\|\Omega^*(\bar{\lambda}_n) - \widetilde{\Omega}(\bar{\lambda}_n)\right\|_{(2)} + \left\|\widetilde{\Omega}(\bar{\lambda}_n) - \Theta\right\|_{(2)}$ and Theorem 2 of Zhang and Zou (2014), we can readily obtain the stated conclusion. ∎

Remark 5 *If $s_\theta m \sqrt{\eta \log p / n} \to 0$, then the estimated solution $\Omega^*(\bar{\lambda}_n)$ is positive definite with probability tending to 1. Moreover, assuming that n is the same as the statement of Theorem 2 in Zhang and Zou (2014), then the positive definite property of the optimal solution estimated by the D trace estimator can be guaranteed. Moreover, if the estimated solution is not positive definite, a common remedy is to add a matrix πI_p with a small $\pi > |\gamma_{\min}(\Omega^*(\bar{\lambda}_n))|$ to obtain a positive definite estimation.*

For the polynomial tails case, we have the following corollary to state a similar conclusion.

Corollary 6 *Assume that $x_1, \cdots, x_n$ are iid variables from a distribution with polynomial tails and for all i-th coordinate x_j^i of any random variable x_j, $j \in [1, n]$, the following finite $4q$-th moments condition holds:*

$$E\left[\Sigma_{ii}^{-1/2} x_j^i\right]^{4q} \leq K_q, \quad K_q \in \mathbb{R}.$$

Let $\bar{\lambda}_n = C_p p^{\eta/2q} / \sqrt{n}$ with a constant C_p sufficiently large for some $\eta > 2$ and the sample size $n = \Xi\left((s_\theta m / r)^2 p^{\eta/q}\right)$. Then with probability $1 - 1/p^{\eta-2}$, we have

$$\left\|\Omega^*(\bar{\lambda}_n) - \Theta\right\|_{(2)} = O(s_\Theta m \sqrt{(p^{\eta/q})/n}),$$

where $\Omega^*(\bar{\lambda}_n)$ is generated by Algorithm 1 such that $R_{\lambda_n}(\Omega^*(\bar{\lambda}_n)) \leq \epsilon$ for a small enough tolerance $\epsilon \geq 0$.

We end this section by providing some further remarks for Algorithm 1.

Remark 7 *Determination of $\lambda_{\max}$ in the "Input" of Algorithm 1. Assume that the solution set to (4) is nonempty. We can set*

$$\lambda_{\max} = \max_{i<j}\left\{\frac{1}{2}|\widehat{\Sigma}_{ij}/\widehat{\Sigma}_{ii} + \widehat{\Sigma}_{ij}/\widehat{\Sigma}_{jj}|\right\}.$$

If $\lambda \geq \lambda_{\max}$, the optimal solution of (4) is a diagonal matrix Ω^* with $\Omega_{ii}^* = 1/\widehat{\Sigma}_{ii}, i = 1, \cdots, p$. This can be easily verified: suppose that the optimal solution is not Ω^* with such a λ. We can readily know $\Omega^* - h(\Omega^*) \in B_\lambda$. Thus we have $R_\lambda(\Omega^*) = 0$, which is contradict to the KKT conditions.

Remark 8 *Direct extension to the relaxed lasso. Since we have defined the non-zero components set $\bar{I}$ in Algorithm 1, we can easily insert the relaxed lasso (Meinshausen, 2007) into our algorithm after Step 13 to obtain a solution with a better prediction accuracy.*

11

3. A semismooth Newton augmented Lagrangian method

In this section, we will develop a semismooth Newton augmented Lagrangian method for solving the minimization problems in Steps 1 and 10 of Algorithm 1. In order to implement the adaptive sieving reduction strategy more effectively, we will first define some linear operators which allow us to reformulate the original problem into a neater form by removing the zero components. After that, we shall derive an inexact augmented Lagrangian algorithm (ALM) for solving the dual of the original problem and a semismooth Newton algorithm (SSN) for solving its inner problems. Moreover, we also analyze the global linear convergence and the asymptotic superlinear convergence rate of the proposed algorithm.

After introducing a matrix $W \in \mathbb{R}^{p \times n}$, for any $\lambda \in \{\lambda_i, i = 0, 1, \cdots, k\}$, we can rewrite the original problems in Steps 1 and 10 as:

$$\min_{\Omega, W} \left\{ \frac{1}{2} \|W\|_F^2 - \langle \Omega, I_p \rangle + \lambda \|\Omega\|_{1,\text{off}} \mid W - \Omega A = 0, \ \Omega \in S_{\bar{I}(\lambda)} \right\}, \qquad (13)$$

where $S_{\bar{I}(\lambda)} := \{\Omega \in S^p \mid \Omega_{ij} = 0, (i, j) \in \bar{I}(\lambda)\}$. Note that the number of nonzero components in the upper triangle of Ω is less than or equal to $t = (|I(\lambda)| + p)/2$. Since we are dealing with a problem that is designed for a sparse estimation, t will not be very large. In practical applications, in order to ensure the statistical validity of the estimated solution, t is generally no greater than $p + n(n - 1)/2$.

Define a linear operator $L : \mathbb{S}^p \to \mathbb{R}^t$ as follows: for any $\Omega \in \mathbb{S}^p$, let $\omega = L(\Omega)$ be the vector of the remaining components of $\text{svec}(\Omega)$ with those components $\Omega_{ij}, (i, j) \in S_{\bar{I}}$ being removed, where $\text{svec}(\Omega)$ is the vectorized components of the upper triangular (including the diagonal) of Ω. The generalized inverse of L is defined by $L^\dagger : \mathbb{R}^t \to \mathbb{S}^p$ such that for any vector $\omega \in \mathbb{R}^t$, $\Omega = L^\dagger(\omega)$ is a symmetric matrix with $\omega = L(\Omega)$ and all other components $\{\Omega_{ij} \mid (ij) \in S_{\bar{I}}\}$ equaling to 0. For later use, we further denote

$$e_1 := L(I_p); \quad e_2 := 2L(E - I_p); \quad e_3 := e_1 + e_2/4; \quad e_4 := e_1 + e_2,$$

where E is the p-dimensional all-ones matrix. Let L^* and $(L^\dagger)^*$ be the adjoints of L and $L^\dagger$, respectively. For any vector $v \in \mathbb{R}^t$, by the definition of the adjoint, we have $\langle L(\Omega), v \rangle = \langle \Omega, L^*(v) \rangle$. Then we immediately have

$$L^*(v) = L^\dagger(v \circ e_3).$$

Similarly, for any matrix $V \in \mathbb{S}^p$, we know

$$(L^\dagger)^*(V) = L(V) \circ e_4.$$

We also define another linear operator $S : \mathbb{R}^{p \times n} \to \mathbb{R}^t$ by $S(Y) := \frac{1}{2} L(YA^T + AY^T)$, $\forall Y \in \mathbb{R}^{p \times n}$, whose adjoint $S^* : \mathbb{R}^t \to \mathbb{R}^{p \times n}$ is, for any vector $v \in \mathbb{R}^t$ given by

$$S^*(v) = L^*(v)A.$$

Before preceding to introduce our algorithm, we put a negative sign in front of the objective function of problem (13) to obtain a minimization dual problem. After rewriting

12

the original problem by using the operators defined above and introducing another variable $x \in \mathbb{R}^t$ such that $x = \omega \circ e_4$ with $\omega = L(\Omega)$, we have the following equivalent problem:

$$(\mathbf{P}) \quad \max_{x \in \mathbb{R}^t} \left\{ - \left(\Gamma(x) := \frac{1}{2} \|S^*(x)\|_F^2 - \langle x, e_1 \rangle + \lambda/2 \|x \circ e_2\|_1 \right) \right\},$$

whose dual is

$$(\mathbf{D}) \quad \min_{Y \in \mathbb{R}^{p \times n}, z \in \mathbb{R}^t} \left\{ \frac{1}{2} \|Y\|_F^2 + \delta_{b_\lambda}(z) \mid S(Y) + z = e_1 \right\},$$

where δ_{b_λ} is an indicator function with $b_\lambda = \{z \in \mathbb{R}^t \mid e_1 \circ z = 0, |z_i| \le \lambda, i = 1, \cdots, t\}$.

As mentioned earlier, we solve $(\mathbf{P})$ by solving its dual, provided that the KKT system associated with $(\mathbf{D})$ is nonempty (Borwein and Lewis, 2010). Assume that the solution set to $(\mathbf{P})$ is nonempty. Then, as we shall see in the proof of proof of Theorem 13, there exists $(Y^0, z^0) \in \mathbb{R}^{p \times n} \times \mathbb{R}^t$ such that the constraints $S(Y^0) + z^0 = e_1$ and $z^0 \in b_\lambda$ are satisfied. According to the Kuhn-Tucker theorem (Rockafellar, 1970), we know that $(\overline{Y}, \overline{z})$ is the optimal solution of $(\mathbf{D})$ if and only if there is $\bar{x} \in \mathbb{R}^t$ such that $(\overline{Y}, \overline{z}, \bar{x})$ satisfies the KKT system corresponding to $(\mathbf{D})$. For any $Y \times z \times x \in \mathbb{R}^{p \times n} \times \mathbb{R}^t \times \mathbb{R}^t$, the associated Lagrangian function of $(\mathbf{D})$ is

$$\mathcal{L}(Y, z, x) = \frac{1}{2} \|Y\|_F^2 + \delta_{b_\lambda}(z) - \langle S(Y) + z - e_1, x \rangle.$$

Then the KKT system can be easily obtained as follows

$$\begin{cases} Y - S^*(x) = 0, \\ 0 \in \partial(\delta_{b_\lambda}(z)) - x, & \text{with } Y \in \mathbb{R}^{p \times n}, \ z, x \in \mathbb{R}^t. \\ S(Y) + z - e_1 = 0, \end{cases} \tag{14}$$

Given $\sigma > 0$, the augmented Lagrangian function associated with $(\mathbf{D})$ is given by

$$\mathcal{L}_\sigma(Y, z; x) = \frac{1}{2} \|Y\|_F^2 + \delta_{b_\lambda}(z) - \langle S(Y) + z - e_1, x \rangle + \frac{\sigma}{2} \|S(Y) + z - e_1\|_F^2.$$

Now, all the preparation works are done, we can introduce our semismooth Newton augmented Lagrangian method in the following two subsections.

3.1 An inexact augmented Lagrangian algorithm

In this subsection, we will develop an inexact ALM for solving $(\mathbf{P})$ and $(\mathbf{D})$, and prove the global linear convergence of the proposed algorithm. This implies that the initial point of our algorithm can be chosen arbitrarily on the associated effective domain. Moreover, we will establish the asymptotic superlinear convergence of our algorithm when applied to $(\mathbf{D})$, at the end of this subsection. We remark that some standard stopping criteria are used for analyzing the convergence rate of our algorithm here since we cannot solve the inner problems of the inexact ALM exactly. A semismooth Newton algorithm to solve the inner problems of the inexact ALM together with the implementations of the stopping criteria

13

Algorithm 2 An inexact augmented Lagrangian method for solving (**D**).

Require:
   A given parameter $\sigma_0 > 0$;
   An initial point $(Y^0, z^0, x^0) \in \mathbb{R}^{p \times n} \times \mathbb{R}^t \times \mathbb{R}^t$; An integer $k = 0$;
Ensure:
   Approximate optimal solution $(\widehat{Y}, \hat{z}, \hat{x})$;
 1: **while** Stopping criteria are not satisfied **do**
 2:    Compute
$$(Y^{k+1}, z^{k+1}) \approx \arg\min\{\Psi_k(Y, z) := \mathcal{L}_{\sigma_k}(Y, z; x^k)\}; \qquad (15)$$

 3:    Compute $x^{k+1} = x^k - \sigma_k \left(S(Y^{k+1}) + z^{k+1} - e_1\right)$ and update $\sigma_{k+1} \uparrow \sigma_\infty \leq \infty$;
 4:    Update $\widehat{Y} = Y^{k+1}$, $\hat{z} = z^{k+1}$, $\hat{x} = x^{k+1}$;
 5:    $k++$;
 6: **end while**

---

will be introduced in the next subsection. Details of the inexact ALM are provided in Algorithm 2.

Next, we will establish the global linear convergence and the asymptotic superlinear convergence rate of Algorithm 2. Firstly, we shall analyze some properties of (**P**), which is a piecewise linear-quadratic programming. These properties will be further used to establish the global linear convergence when combined with the convergence theorem introduced later. Denote $\text{dist}(Y, C) := \inf_{Y' \in C} \|Y - Y'\|$, $\forall Y \in \mathcal{Y}$ and $\forall C \subset \mathcal{Y}$ and $\mathcal{O}$ as the optimal solution set of (**P**). Note that if $C = \emptyset$, then $\text{dist}(Y, C) = +\infty$. We also define two maximal monotone operators $\mathcal{T}_\Gamma$ and $\mathcal{T}_l$ by

$$\mathcal{T}_\Gamma(x) := \partial\Gamma(x), \quad \mathcal{T}_\mathcal{L}(Y, z, x) := \{(Y', z', x') \mid (Y', z', -x') \in \partial\mathcal{L}(Y, z, x)\}.$$

Now, we begin the analysis by introducing the following definition, which is given by Robinson (1981).

Definition 9 *Let $F : \mathcal{Y} \rightrightarrows \mathcal{X}$ be a closed-valued multifunction. If there exists $\kappa \geq 0$, for some neighbourhood N of $\bar{y}$ and for all $y \in N$,*

$$F(y) \subset F(\bar{y}) + \kappa \|y - \bar{y}\| B_x,$$

with $B_x = \{x \mid \|x\| \leq 1\}$, then we say F is locally upper Lipschitzian at the point $\bar{y}$ with modulus κ.

After obtaining a conclusion that $\mathcal{T}_\Gamma^{-1}$ is locally upper Lipschitzian at the origin with modulus $\kappa \geq 0$, we state the following lemma to establish an upper bound for the distance between some points $x \in \mathbb{R}^t$ and $\mathcal{O}$ with $\mathcal{O} \neq \emptyset$.

Lemma 10 *Suppose that $\mathcal{T}_\Gamma^{-1}(0)$ is nonempty. There exist $\kappa \geq 0$ and some neighborhood N_0 such that for all $x \in \mathbb{R}^t$, we have*

$$\text{dist}(x, \mathcal{O}) \leq \kappa \, \text{dist}(0, \mathcal{T}_\Gamma(x) \cap N_0). \qquad (16)$$

Proof Since $(\boldsymbol{P})$ is a ℓ_1 regularized convex piecewise linear-quadratic programming, from Sun (1986) and Robinson (1981), we know that the operators $\mathcal{T}_\Gamma$ and its inverse $\mathcal{T}_\Gamma^{-1}$ are polyhedral multifunctions and there exists a constant κ such that $\mathcal{T}_\Gamma^{-1}$ is locally upper Lipschitzian at the origin with a neighborhood N_0. We know that, for any $x \in \mathbb{R}^t$, if $\mathcal{T}_\Gamma(x) \cap N_0 = \emptyset$, the inequality (16) holds automatically. Let $\mathcal{T}_\Gamma(x) \cap N_0 \neq \emptyset$. Since $\mathcal{T}_\Gamma$ is closed-valued, then there exists $y \in \mathcal{T}_\Gamma(x) \cap N_0$ with

$$\|y\| = \mathrm{dist}(0, \mathcal{T}_\Gamma(x) \cap N_0).$$

Therefore,
$$\mathcal{T}_\Gamma(y) \subset \mathcal{T}_\Gamma(0) + \kappa \|y\| B_x,$$
but since $x \in \mathcal{T}_\Gamma^{-1}(y)$ and $\mathcal{O} = \mathcal{T}_\Gamma^{-1}(0) \neq \emptyset$, we have

$$\mathrm{dist}(x, \mathcal{O}) \leq \kappa \, \mathrm{dist}(0, \mathcal{T}_\Gamma(x) \cap N_0).$$

This completes the proof. ∎

Remark 11 *From Lemma 10, we know that there exists $\epsilon > 0$ such that for any $x \in \mathbb{R}^t$ with $\mathrm{dist}(0, \mathcal{T}_\Gamma(x)) < \epsilon$ we have $\mathrm{dist}(x, \mathcal{O}) \leq \kappa \, \mathrm{dist}(0, \mathcal{T}_\Gamma(x))$, which is also consistent with the corollary introduced by Robinson (1981).*

With the conclusion stated in Lemma 10, we can then verify that inequality (16) still holds for any point x chosen arbitrarily on the effective domain of $\mathcal{T}_\Gamma$ in the following lemma.

Lemma 12 *Suppose that $\mathcal{T}_\Gamma^{-1}(0)$ is nonempty. For any $r > 0$, there exists $\kappa \geq 0$ such that*

$$\mathrm{dist}(x, \mathcal{O}) \leq \kappa \, \mathrm{dist}(0, \mathcal{T}_\Gamma(x)), \quad \forall x \in R^n \text{ satisfying } \mathrm{dist}(x, \mathcal{O}) \leq r. \tag{17}$$

Proof From Lemma 10, we know that there exist $\kappa_1 \geq 0$ and some neighborhood N_0 such that for all $x \in R^n$ inequality (16) holds. Then, for any $x \in R^n$ satisfying $\mathrm{dist}(x, \mathcal{O}) \leq r$, if $\mathcal{T}_\Gamma(x) \cap N_0 \neq \emptyset$, we readily have

$$\mathrm{dist}(x, \mathcal{O}) \leq \kappa_1 \, \mathrm{dist}(0, \mathcal{T}_\Gamma(x) \cap N_0) \leq \kappa_1 \, \mathrm{dist}(0, \mathcal{T}_\Gamma(x)),$$

otherwise, there exists $\bar{\delta} > 0$ satisfying $\mathrm{dist}(0, \mathcal{T}_\Gamma(x)) \geq \bar{\delta}$, hence we can find $\kappa_2 \geq r/\bar{\delta}$ such that

$$\mathrm{dist}(x, \mathcal{O}) \leq \kappa_2 \, \mathrm{dist}(0, \mathcal{T}_\Gamma(x)).$$

We complete our proof by setting $\kappa = \max\{\kappa_1, \kappa_2\}$. ∎

Now, we are ready to proceed with the analysis of convergence properties of Algorithm 2. Since we can't obtain an exact optimal solution for the inner problem (15), we use the following standard stopping criterion introduced in Rockafellar (1976) to obtain an approximated solution:

$$\Psi_k(Y^{k+1}, z^{k+1}) - \inf \Psi_k \leq \epsilon_k^2/2\sigma_k, \quad \sum_{k=0}^{\infty} \epsilon_k \leq \alpha_\epsilon < \infty. \tag{18}$$

Besides, for analyzing the convergence rate, we need to introduce the following two stopping criteria (Li et al., 2018):

$$(S1) \quad \Psi_k(Y^{k+1}, z^{k+1}) - \inf \Psi_k \leq (\theta_k^2/2\sigma_k) \left\| x^{k+1} - x^k \right\|^2, \ \sum_{k=0}^{\infty} \theta_k < +\infty,$$

$$(S2) \quad \mathrm{dist}(0, \partial \Psi_k(Y^{k+1}, z^{k+1})) \leq (\theta_k'/\sigma_k) \left\| x^{k+1} - x^k \right\|, \ 0 \leq \theta_k' \to 0.$$

Based on Rockafellar (1976); Li et al. (2018) and Lemma 12, the following theorem establishes convergence results for the primal iteration sequence $\{x^k\}$ and the dual iteration sequence $\{(y^k, z^k)\}$ generated by the inexact ALM.

Theorem 13 *Suppose that the solution set to $(\boldsymbol{P})$ is nonempty and the initial point $x^0 \in \mathbb{R}^t$ satisfies $\mathrm{dist}(x^0, \mathcal{O}) \leq r - \alpha_\epsilon$, where α_ϵ is a constant as given in (18). $\{(Y^k, z^k, x^k), k = 1, 2, \cdots\}$ is a sequence generated by Algorithm 2 with stopping criteria (18) and $(S1)$. There is a constant $\gamma_f \geq 0$ such that the sequence $\{x^k\}$ converges to the optimal solution $x^* \in \mathcal{O}$ and for all $k \geq 0$, we have*

$$\mathrm{dist}(x^{k+1}, \mathcal{O}) \leq \zeta_k \mathrm{dist}(x^k, \mathcal{O}), \tag{19}$$

where $\zeta_k = (\gamma_f(\gamma_f^2 + \sigma_k^2)^{-1/2} + 2\theta_k)(1 - \theta_k)^{-1}$ and $\zeta_k \to \zeta_\infty = \gamma_f(\gamma_f^2 + \sigma_\infty^2)^{-1/2} < 1$ when $k \to \infty$. In addition, the sequence $\{Y^k, z^k\}$ converges to the unique optimal solution (Y^*, z^*) of $(\boldsymbol{D})$. Furthermore, if the stopping criteria $(S2)$ is satisfied, for all $k \geq 0$, there is a constant $\gamma_l \geq 0$ such that

$$\left\| (Y^{k+1}, z^{k+1}) - (Y^k, z^k) \right\| \leq \zeta_k' \left\| x^{k+1} - x^k \right\|, \tag{20}$$

where $\zeta_k' = \gamma_l(1 + \theta_k')/\sigma_k$ and $\zeta_k' \to \zeta_\infty' = \gamma_l/\sigma_\infty$ as $k \to \infty$.

Proof Since the solution set to $(\boldsymbol{P})$ is nonempty and the objective function of $(\boldsymbol{D})$ is strongly convex on a convex set, according to Fenchel's duality theorem (Rockafellar, 1970), the solution set to $(\boldsymbol{D})$ is nonempty and the optimal values of these two problems are equal to each other and also finite. This implies the solutions of the associated KKT system is nonempty. The uniqueness of the optimal solution of $(\boldsymbol{D})$ is obtained directly by the strong convexity of its objective function. Then, The first part of the theorem can be obtained directly from Lemma 12, Theorem 2.1 in Luque (1984) and Theorem 5 in Rockafellar (1976). For the second part, from Theorem 2.7 in Li et al. (2018), we know that the operator $\mathcal{T}_l$ is satisfied with the condition in theorem 3.3 in Li et al. (2018). Hence, it can be concluded trivially. ∎

Remark 14 *From Theorem 13, we know that Algorithm 2 enjoys a global linear convergence.*

Remark 15 *If $\sigma_\infty = +\infty$, from (19), the sequence $\{x^k\}$ generated by Algorithm 2 will converge Q-superlinearly. Combing this with (20), we know that the sequence $\{(y^k, z^k)\}$ converges R-superlinearly. Thus, according to Theorem 13, we can say that our algorithm here converges asymptotically superlinearly.*

3.2 A semismooth Newton algorithm for solving the subproblem in Algorithm 2

In this subsection, we will develop a semismooth Newton algorithm (SSN) for solving (15), and introduce the implementations of the stopping criteria used in the previous subsection. Given $\sigma > 0$ and $x \in \mathbb{R}^t$, the problem is to find an optimal solution of $\min_{Y,z} \Psi(Y, z)$, $\forall (Y, z) \in \mathbb{R}^{p \times n} \times \mathbb{R}^t$. Since $\Psi(\cdot)$ is strongly convex, there is a unique optimal solution $(\overline{Y}, \overline{z}) \in \mathbb{R}^{p \times n} \times \mathbb{R}^t$ and it can be obtained by solving $\min_Y \{\inf_Z \Psi(Z, Y)\}$. For any $Y \in \mathbb{R}^{p \times n}$, we first denote

$$\psi(Y) := \inf_z \Psi(Y, z)$$
$$= \frac{1}{2} \|Y\|_F^2 - \frac{1}{2\sigma} \|x\|_F^2 + \sigma \inf_z \left\{ \sigma^{-1} \delta_{b_\lambda}(z) + \frac{1}{2} \|z - (x/\sigma - S(Y) + e_1)\|_F^2 \right\}.$$

Thus, we can obtain $(\overline{Y}, \overline{z})$ simultaneously by

$$\overline{Y} = \arg\min \psi(Y), \qquad \overline{z} = \text{Prox}_{\delta_{b_\lambda}}(x/\sigma - S(\overline{Y}) + e_1). \tag{21}$$

To solve (21), for any $Y \in \mathbb{R}^{p \times n}$, we need to find $\nabla\psi(Y)$ first. Let us start by defining $f(Y) := x/\sigma - S(Y) + e_1, \forall Y \in \mathbb{R}^{p \times n}$ and $\mathcal{G}(v) := \inf_z \sigma^{-1} \delta_{B_\lambda}(z) + \frac{1}{2} \|z - v\|_F^2, \forall v \in \mathbb{R}^t$. Notice that $\nabla\mathcal{G}(\cdot)$ is continuously differentiable. We thus have

$$\nabla\psi(Y) = Y - \sigma S^* (\nabla\mathcal{G}(f(Y))), \tag{22}$$

where $\nabla\mathcal{G}(f(Y)) = \text{Prox}_\phi(f(Y))$ with $\phi(v) = \lambda/2 \|v \circ e_4\|_1, \forall v \in \mathbb{R}^t$. Therefore, $\overline{Y}$ can be found by solving

$$\nabla\psi(Y) = 0. \tag{23}$$

For deriving a semismooth Newton algorithm, we need to find the second-order information of the objective function, i.e., the generalized Hessian matrix of $\psi(\cdot)$. For any $Y \in \mathbb{R}^{p \times n}$, we denote $\partial^2 \psi(Y)$ as the generalized Hessian matrix of $\psi(\cdot)$ at Y. According to Hiriart-Urruty et al. (1984), for any vector $v, \bar{v} \in \mathbb{R}^t$, since $(S^*)'(v)(\bar{v}) = L^\dagger(\bar{v} \circ e_3)A$, we have

$$\partial^2 \psi(Y)(D) = \{D + \sigma L^\dagger(S(D) \circ u \circ e_3)A \mid u \in \mathcal{U}\}, \quad \forall D \in \mathbb{R}^{p \times n},$$

where $\mathcal{U} := \partial\text{Prox}_\phi(f(Y))$. For later use, we define $\hat{\partial}^2\psi(Y)$ as follows:

$$V \in \hat{\partial}^2\psi(Y) \Leftrightarrow \exists u \in \mathcal{U} \text{ such that } V(D) \in \partial^2\psi(Y)(D), \quad \forall D \in \mathbb{R}^{p \times n}. \tag{24}$$

We know from Clarke (1990) that there is a useful fact: $\partial^2\psi(Y)(\Delta Y) \subseteq \hat{\partial}^2\psi(Y)(\Delta Y)$, $\forall \Delta Y \in \mathbb{R}^{p \times n}$. Now, we can introduce our semismooth Newton algorithm for solving (23) as in Algorithm 3.

In order to establish the convergence of our semismooth Newton algorithm, two conditions must be satisfied. The first condition is that any element in $\hat{\partial}^2\psi(Y)$, $Y \in \mathbb{R}^{p \times n}$ are non-singular. This can be checked by simple calculation. Another condition is that $\nabla\psi(\cdot)$ is Lipschitz continuous and semismooth. $\nabla\psi(\cdot)$ is globally Lipschitz continuous provided

Algorithm 3 A semismooth Newton algorithm for solving (23).

Require:

   Given parameters $\mu \in (0, 1/2)$, $\bar{\eta} \in (0, 1)$, $\tau \in (0, 1]$, and $\delta \in (0, 1)$;
   An initial point $Y^0 \in \mathbb{R}^{p \times n}$ and a given $x \in \mathbb{R}^t$;
   An integer $j = 0$;

Ensure:

   Approximate optimal solution $\widehat{Y}$;

1: **while** Stopping criteria are not satisfied **do**
2:    Choose $u_j \in \partial \mathrm{Prox}_\phi(x/\sigma - S(Y^j) + e_1)$. For $D \in \mathbb{R}^{p \times n}$, let $V_j D := D + \sigma L^\dagger (S(D) \circ u_j \circ e_3) A$. Solve the equation

$$V_j D + \nabla \psi(Y^j) = 0 \tag{25}$$

   by the conjugate gradient algorithm to find D^j such that

$$\left\| V_j D^j + \nabla \psi(Y^j) \right\| \leq \min(\bar{\eta}, \left\| \nabla \psi(Y^j) \right\|^{1+\tau}); \tag{26}$$

3:    (Line search) Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m such that

$$\psi(Y^j + \delta^{m_j} D^j) \leq \psi(Y^j) + \mu \delta^{m_j} \left\langle \nabla \psi(Y^j), D^j \right\rangle; \tag{27}$$

4:    Set $Y^{j+1} = Y^j + \alpha_j D^j$ and update $\hat{Y} = Y^{k+1}$;
5:    $j + +$;
6: **end while**

that the proximal point mapping is Lipschitz continuous. As for the semismooth property, we first present a useful conclusion that twice continuously differentiable functions and continuous piecewise affine functions are strongly semismooth. Since $\mathrm{Prox}_{\lambda \|\cdot\|_1}(\cdot)$ is a Lipschitz continuous piecewise affine function, we know that $\nabla \psi(\cdot)$ is strongly semismooth. Finally, we can state the convergence results of SSN in the following theorem.

Theorem 16 *The sequence $\{Y^k\}$ generated by the Algorithm 3 is converge to the unique optimal solution $\overline{Y} \in \mathbb{R}^{p \times n}$ of the problem in (21) and the convergence is of the order $1 + \tau$, that is*

$$\left\| Y^{j+1} - \overline{Y} \right\| = O\left(\left\| Y^j - \overline{Y} \right\|^{1+\tau} \right).$$

Proof Since any $V \in \hat{\partial}^2 \psi(Y)$ is self-adjoint positive definite, Algorithm 3 always can find a descent direction as long as $\nabla \psi(Y^j) \neq 0$. Due to the fact that $\nabla \psi(\cdot)$ is strongly semismooth, the stated conclusion can be easily derived from Theorem 3.5 of Zhao et al. (2010). ∎

Theorem 16 shows that the convergence rate of SSN is of order $(1 + \tau)$. This implies that SSN can converge quadratically if $\tau = 1$. However, this setting will results in more

iterations in the conjugate gradient method for solving (25). In practice, we suggest to set τ to be smaller, such as 0.1 or 0.2, to obtain a superlinear convergence.

Now, we can discuss how to insert the stopping criteria (18), (S1), and (S2) into Algorithm 3 to guarantee the convergence results discussed in Section 3.1. Since ψ is strongly convex with a parameter $\tau > 1$, we can obtain

$$\Psi_k(Y^{k+1}, z^{k+1}) - \inf \Psi_k = \psi_k(Y^{k+1}) - \inf \psi_k \leq 1/2\tau \left\| \nabla \psi_k(Y^{k+1}) \right\|^2,$$

and $(\nabla \psi_k(Y^{k+1}), 0) \in \partial \Psi_k(Y^{k+1}, Z^{k+1})$. As a result, in practical implementation, we can replace the stopping criteria (18), (S1) and (S2) with $\nabla \psi_k(Y^{k+1})$ to

$$
\begin{cases}
\left\| \nabla \psi_k(Y^{k+1}) \right\| \leq \sqrt{\tau/\sigma_k}\epsilon_k, \; \sum_{k=0}^{\infty} \epsilon_k < \infty, \\[2mm]
\left\| \nabla \psi_k(Y^{k+1}) \right\| \leq \sqrt{\tau\sigma_k}\theta_k \left\| \frac{1}{2}Y^{k+1}A^T + \frac{1}{2}A(Y^{k+1})^T + z^{k+1} - c \right\|, \quad \sum_{k=0}^{\infty} \theta_k < +\infty, \\[2mm]
\left\| \nabla \psi_k(Y^{k+1}) \right\| \leq \theta'_k \left\| \frac{1}{2}Y^{k+1}A^T + \frac{1}{2}A(Y^{k+1})^T + z^{k+1} - c \right\|, \quad 0 \leq \theta'_k \to 0.
\end{cases}
$$

In other words, the stopping criteria (18), (S1), and $S2$ will be satisfied when $\left\| \nabla \psi_k(Y^{k+1}) \right\|$ is small enough.

4. Numerical experiments

In this section, we will conduct several tests to illustrate the performance of our MARS. For comparison, we consider several popular solvers including SCIO (Liu and Luo, 2015), EQUAL (Wang and Jiang, 2020), CLIME (Pang et al., 2014), glasso (Friedman et al., 2008) and BigQuic (Hsieh et al., 2013). Since the existing methods are all first-order and the stopping criteria of those algorithms are different from each other and also ours, for better comparison, we will also introduce some other algorithms for solving (4) in Section 4.1. Specifically, we will introduce a second-order algorithm, namely "a semismooth Newton augmented Lagrangian method (SSNAL)", and two kinds of alternating direction methods of multipliers, where one is derived by solving the sub-problem inexactly (iADMM) and the other derived by solving it exactly (eADMM). The tests here are divided into two parts by the source of the data. The first part is conducted for some random data generated by five given models, and for the second part, the data is derived from real applications.

Before proceeding to the experiments, we provide some explanations about our MARS first. For any vector $\nu \in \mathbb{R}^t$, to find a $u \in \partial \text{Prox}_\phi(\nu)$, we can choose the i-th component of u as

$$u_i = \begin{cases} 0, & \text{if } d_i \neq 0 \,\&\, |\nu_i| \leq \lambda, \\ 1, & \text{otherwise}, \end{cases} \quad i = 1, 2, \ldots, t.$$

Apparently we have $u \in \partial \text{Prox}_\phi(\nu)$, which is because the components of $\nu^p := \text{Prox}_\phi(\nu)$ can be found by

$$\nu_i^p = \begin{cases} \text{sign}(\nu_i) \cdot \max\{|\nu_i| - \lambda, 0\}, & \text{if } d_i \neq 0, \\ \nu_i, & \text{if } d_i = 0, \end{cases} \quad i = 1, 2, \ldots, t.$$

19

For the stopping criteria of our MARS, we use the relative KKT residual

$$\eta = \frac{\|R(\Omega)\|_F}{1 + \|h\left(\Omega\right)\|_F + \|\Omega\|_F}$$

to estimate the distance between the generated solution Ω and the optimal solution set of (4), and we use η to decide whether our MARS should be stopped or not. The stopping tolerance we set in the following experiments is 10^{-4} for all the solvers/algorithms except EQUAL. Based on several tests, the stopping tolerance of EQUAL is set to be 10^{-6}. The reason for such an adjustment is that their stopping criterion is determined by the distance between two subsequent generating solutions, and a slightly larger stopping tolerance may cause the generated solution to be too far from the optimal solution set, in terms of the associated objective value and relative KKT residual. Moreover, from the test results in section 4.3, we found that even if the stopping tolerance is set to be 10^{-6}, the relative KKT residuals of the solutions obtained by EQUAL are still not less than 10^{-3} (we also try to set stopping tolerance to be 10^{-5} here, but none of the associated relative KKT residuals is less than 5×10^{-2}. More details can be found from appendix 5). All the numerical results are obtained by running Microsoft R Open 4.0.2 on a Windows workstation (Intel(R) Core(TM) i7-10700 CPU @2.90GHz 2.00GHz RAM 32GB). For simplicity, we will use R to represent Microsoft R Open 4.0.2.

4.1 Some other algorithms

In this subsection, we will introduce some other algorithms to compare the performance with our MARS. The first algorithm is an SSNAL, which is similar to our algorithm, but do not use the adaptive sieving reduction strategy. For later use, we define a linear operator $\mathcal{S} : \mathbb{R}^{p \times n} \to \mathbb{S}^p$ by $\mathcal{S}(Y) = \frac{1}{2}(YA^T + AY^T), \forall Y \in \mathbb{R}^{p \times n}$, whose conjugate $\mathcal{S}^*$ can be expressed in the form $\mathcal{S}^*(\Omega) = \Omega A, \forall \Omega \in \mathbb{S}^p$. By putting a negative sign in front of the objective function of the original problem (4), we obtain an equivalent problem

$$\max_{\Omega \in \mathbb{S}^p} \left\{ - \left(\frac{1}{2} \|\Omega A\|_F^2 - \langle \Omega, I_p \rangle + \lambda \|\Omega\|_{1, \text{off}} \right) \right\}, \tag{28}$$

whose dual is

$$\min_{Y \in \mathbb{R}^{p \times n}, Z \in \mathbb{S}^p} \left\{ \frac{1}{2} \|Y\|_F^2 + \delta_{B_\lambda}(Z) \mid \frac{1}{2}(YA^T + AY^T) + Z = I_p \right\}. \tag{29}$$

Given $\sigma > 0$, the augmented Lagrangian function associated with (29) is given by

$$\mathcal{L}_\sigma^m(Y, Z; \Omega)$$
$$= \frac{1}{2} \|Y\|_F^2 + \delta_{B_\lambda}(Z) - \left\langle \frac{1}{2}(YA^T + AY^T) + Z - I_p, \Omega \right\rangle + \frac{\sigma}{2} \left\| \frac{1}{2}(YA^T + AY^T) + Z - I_p \right\|_F^2.$$

With a fixed $\Omega \in \mathbb{S}^p$, the inner problem of SSNAL in each iteration is to find an optimal solution of the problem $\min_{Y,Z} \mathcal{L}_\sigma^m(Y, Z; \Omega)$, $(Y, Z) \in \mathbb{R}^{p \times n} \times \mathbb{S}^p$. Since $\mathcal{L}_\sigma^m(\cdot)$ is strongly convex, there is a unique optimal solution $(\overline{Y}, \overline{Z}) \in \mathbb{R}^{p \times n} \times \mathbb{S}^p$. Besides, this unique solution

20

Algorithm 4 A semismooth Newton augmented Largangian method for solving (29).

Require:
    Given parameters $\sigma_0 > 0$, $\mu \in (0, 1/2)$, $\bar{\eta} \in (0, 1)$, $\tau \in (0, 1]$, and $\delta \in (0, 1)$;
    An initial point $(Y^0, Z^0, \Omega^0) \in \mathbb{R}^{p \times n} \times \mathbb{S}^p \times \mathbb{S}^p$; An integer $k = 0$;
Ensure:
    Approximate optimal solution $(\widehat{Y}, \widehat{Z}, \widehat{\Omega})$;
1: **while** Stopping criteria are not satisfied **do**
2:     An integer j = 0; Set an initial value $Y_0 = Y^k$ for the inner loop;
3:     **while** Stopping criteria of the inner problem are not satisfied **do**
4:         Choose $U_j \in \partial \mathrm{Prox}_\theta (\Omega^k / \sigma - \mathcal{S}(Y_j) + I_p)$. For $D \in \mathbb{R}^{p \times n}$, let $V_j D := D + \sigma (\mathcal{S}(D) \circ U_j) A$. Solve the equation

$$V_j D + \nabla \psi^m (Y_j) = 0 \tag{30}$$

        by the conjugate gradient algorithm to find D_j such that

$$\| V_j D_j + \nabla \psi^m (Y_j) \| \leq \min(\bar{\eta}, \| \nabla \psi^m (Y_j) \|^{1+\tau}); \tag{31}$$

5:         (Line search) Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m such that

$$\psi^m (Y_j + \delta^{m_j} D_j) \leq \psi^m (Y_j) + \mu \delta^{m_j} \langle \nabla \psi^m (Y_j), D_j \rangle; \tag{32}$$

6:         Set $Y_{j+1} = Y_j + \alpha_j D_j$ and update $Y^{k+1} = Y_{j+1}$;
7:         $j + +$;
8:     **end while**
9:     Compute $Z^{k+1} = \mathrm{Prox}_{\delta_{B_\lambda}} (\Omega^k / \sigma_k - \mathcal{S}(Y^{k+1}) + I_p)$;
10:     Compute $\Omega^{k+1} = \Omega^k - \sigma_k (\mathcal{S}(Y^{k+1}) + Z^{k+1} - I_p)$ and update $\sigma_{k+1} \uparrow \sigma_\infty \leq \infty$;
11:     Update $\widehat{Y} = Y^{k+1}$, $\widehat{Z} = Z^{k+1}$, $\widehat{\Omega} = \Omega^{k+1}$;
12:     $k + +$;
13: **end while**
---

can be obtained by solving $\min_Y \{ \inf_Z \mathcal{L}_\sigma^m (Y, Z; \Omega) \}$. For any $Y \in \mathbb{R}^{p \times n}$, we first define

$$
\begin{aligned}
\psi^m (Y) &:= \inf_Z \mathcal{L}_\sigma^m (Y, Z; \Omega) \\
&= \frac{1}{2} \| Y \|_F^2 - \frac{1}{2\sigma} \| \Omega \|_F^2 + \sigma \inf_Z \left\{ \sigma^{-1} \delta_{B_\lambda}(Z) + \frac{1}{2} \| \Omega/\sigma - \mathcal{S}(Y) + I_p - Z \|_F^2 \right\}.
\end{aligned}
$$

Then, we can fleetly obtain $(\overline{Y}, \overline{Z}) \in \mathbb{R}^{p \times n} \times \mathbb{S}^p$ by

$$\overline{Y} = \arg\min \psi^m (Y), \quad \overline{Z} = \mathrm{Prox}_{\delta_{B_\lambda}} (\Omega/\sigma - \mathcal{S}(\overline{Y}) + I_p). \tag{33}$$

Similar to the arguments in Section 3.2, we obtain that the first order derivative of ψ^m at $Y \in \mathbb{R}^{p \times n}$ is

$$\nabla \psi^m (Y) = Y - \sigma \mathrm{Prox}_\theta (\Omega/\sigma - \mathcal{S}(Y) + I_p) A, \tag{34}$$

and the generalized Hessian of ψ^m at $Y \in \mathbb{R}^{p \times n}$ is

$$\partial^2 \psi^m(Y)(D) = \{D + \sigma \mathcal{S}(D) \circ U)A \mid U \in \mathcal{U}\}, \quad \forall D \in \mathbb{R}^{p \times n}.$$

Now, assume that the solution set of the KKT system of (29) is nonempty, the detailed steps of SSNAL are then introduced in Algorithm 4.

Next, we will introduce an iADMM for solving (29). Given $Z, \Omega \in \mathbb{S}^p$, we first define

$$\psi^a(Y) := \mathcal{L}_\sigma^m(Y, Z; \Omega) = \frac{1}{2} \|Y\|_F^2 + \frac{\sigma}{2} \|\mathcal{S}(Y) - C\|_F^2 + \delta_{B_\lambda}(Z) - \frac{1}{2\sigma} \|\Omega\|_F^2 ,$$

where $C = \Omega/\sigma + I_p - Z$. Similarly, we can obtain that the gradient of $\psi^a(\cdot)$ at $Y \in \mathbb{R}^{p \times n}$ is

$$\nabla \psi^a(Y) = Y + \mathcal{S}(Y)A - \sigma CA. \tag{35}$$

Then the optimal solution $\overline{Y}_a = \operatorname{argmin}_Y \psi^a(Y)$ can be found by solving the equality system $\nabla \psi^a(Y) = 0$, which can be rewritten as

$$(I_p + \sigma \mathcal{S}^* \mathcal{S})(Y) = \sigma CA, \quad \forall Y \in \mathbb{R}^{p \times n}. \tag{36}$$

Since the operator $\mathcal{SS}^*$ is positive semidefinite, we can use the conjugate gradient method to find a solution of (36). Detailed steps of iADMM are provided in Algorithm 5.

Algorithm 5 An inexact ADMM for solving (29).

Require:
  Given parameters $\sigma > 0$ and $\pi \in (0, \infty)$; An initial point $(Y^0, Z^0, \Omega^0) \in \mathbb{R}^{p \times n} \times \mathbb{S}^p \times \mathbb{S}^p$; An integer $k = 0$;
Ensure:
  Approximate optimal solution $(\widehat{Y}, \widehat{Z}, \widehat{\Omega})$;
1: **while** Stopping criteria are not satisfied **do**
2:   Use conjugate gradient method to find an optimal solution Y^{k+1} such that

$$Y^{k+1} \approx \operatorname{argmin}_{Y \in \mathbb{R}^{p \times n}} \psi^a(Y).$$

3:   Compute $Z^{k+1} = \operatorname{Prox}_{\delta_{B_\lambda}}(\Omega^k/\sigma - \mathcal{S}(Y^{k+1}) + I_p)$;
4:   Compute $\Omega^{k+1} = \Omega^k - \pi \sigma \left(\mathcal{S}(Y^{k+1}) + Z^{k+1} - I_p \right)$;
5:   Update $\widehat{Y} = Y^{k+1}$, $\widehat{Z} = Z^{k+1}$, $\widehat{\Omega} = \Omega^{k+1}$;
6:   $k + +$;
7: **end while**

For introducing an eADMM, we should first equivalently rewrite (28) to be

$$\max_{\Omega, M \in \mathbb{S}^p} \left\{ -\left(\frac{1}{2} \|MA\|_F^2 - \langle \Omega, I_p \rangle + \lambda \|\Omega\|_{1,\text{off}} \right) \mid M - \Omega = 0 \right\}, \tag{37}$$

whose dual is

$$\min_{V, Z \in \mathbb{S}^p} \left\{ \frac{1}{2} \|VA\|_F^2 + \delta_{B_\lambda}(Z) \mid \frac{1}{2}(V\widehat{\Sigma} + \widehat{\Sigma}V) + Z = I_p \right\}. \tag{38}$$

22

This reformulation is designed for implementing the conclusion in Wang and Jiang (2020) to let eADMM be able to solve the inner problems exactly. We point out that, the main difference between (29) and (38) is the dimensions of variables Y and V are $p \times n$ and $p \times p$ respectively. Under the high-dimensional setting, it is easily seen that solving (29) can be much more efficient than solving (38). But, when p is relative small, eADMM could be slightly efficient than other algorithms. Now, we can start introducing eADMM. Given $\sigma > 0$, for any $V, Z \in \mathbb{S}^p$, the augmented Lagrangian function associated with (38) is given by

$$\mathcal{L}_\sigma^e(V, Z; \Omega) = \frac{1}{2} \|VA\|_F^2 + \delta_{B_\lambda}(Z) - \langle T(V) + Z - I_p, \Omega \rangle + \frac{\sigma}{2} \|T(V) + Z - I_p\|_F^2,$$

where $T(V) := \frac{1}{2}(VS + SV)$. Likewise, given fixed $Z, \Omega \in \mathbb{S}^p$, we first define

$$\psi^e(V) := \mathcal{L}_\sigma^e(V, Z; \Omega) = \frac{1}{2} \|VA\|_F^2 + \frac{\sigma}{2} \|T(V) - C\|_F^2 + \delta_{B_\lambda}(Z) - \frac{1}{2\sigma} \|\Omega\|_F^2,$$

where $C = \Omega/\sigma + I_p - Z$. Then then optimal solution $\overline{V}_e$ of problem $\arg\min \psi^e(V)$ can be obtained by solving

$$V/\sigma + T(V) - C = 0. \tag{39}$$

Algorithm 6 An exact ADMM for solving (38).

Require:

    Given parameters $\sigma > 0$ and $\pi \in (0, \infty)$; An initial point $(V^0, Z^0, \Omega^0) \in \mathbb{S}^p \times \mathbb{S}^p \times \mathbb{S}^p$;

    An integer $k = 0$;

Ensure:

    Approximate optimal solution $(\widehat{V}, \widehat{Z}, \widehat{\Omega})$;

  1: Calculate Λ_1 and Λ_2;

  2: **while** Stopping criteria are not satisfied **do**

  3:     Update $C^k = \Omega^k/\sigma + I_p - Z^k$;

  4:     Compute V^{k+1} using formula (40);

  5:     Compute $Z^{k+1} = \text{Prox}_{\delta_{B_\lambda}}(\Omega^k/\sigma - T(V^{k+1}) + I_p)$;

  6:     Compute $\Omega^{k+1} = \Omega^k - \pi\sigma\left(T(V^{k+1}) + Z^{k+1} - I_p\right)$;

  7:     Update $\widehat{Y} = Y^{k+1}$, $\widehat{Z} = Z^{k+1}$, $\widehat{\Omega} = \Omega^{k+1}$;

  8:     $k++$;

  9: **end while**

By applying the thin singular value decomposition to the sample covariance matrix, we can obtain $\mathcal{V} \in \mathbb{R}^{p \times n}$ and $\Lambda = \text{diag}(\tau_1, \cdots, \tau_m)$ with $\tau_1, \cdots, \tau_m \geq 0$ such that $\widehat{\Sigma} = \mathcal{V}\Lambda\mathcal{V}^T$ and $\mathcal{V}^T\mathcal{V} = I_n$. After calculating Λ_1 and Λ_2 by

$$\Lambda_1 = \text{diag}\left\{\frac{\tau_1}{\tau_1 + 2/\sigma}, \cdots, \frac{\tau_m}{\tau_m + 2/\sigma}\right\},$$

$$\Lambda_2 = \left\{\frac{\tau_i\tau_j(\tau_i + \tau_j + 4/\sigma)}{(\tau_i + 2/\sigma)(\tau_j + 2/\sigma)(\tau_i + \tau_j + 2/\sigma)}\right\}_{m \times m},$$

we then have

$$\overline{V}_e = \sigma\left(C - C\mathcal{V}\Lambda_1\mathcal{V}^T - \mathcal{V}\Lambda_1\mathcal{V}^T C + \mathcal{V}(\Lambda_2 \circ (\mathcal{V}^T C\mathcal{V}))\mathcal{V}^T\right). \tag{40}$$

Now, we give the detailed steps of eADMM in Algorithm 6.

Table 1: Average performance among different methods for precision matrix estimation with 100 replications, p = 500 in models 1 to 4, p = 484 in model 5 and n = 400 for all the models

	Frobenius	Spectral	Infity	TP	TN	s_{off}	$\bar{s}_{\text{off}}$
	mean | sd	mean | sd	mean | sd	mean | sd	mean | sd	mean	mean
Model 1							
MARS	7.0335 | 0.1717	0.7498 | 0.0227	1.1045 | 0.0433	0.9040 | 0.0214	0.9838 | 0.0029	5755.06	5753.90
SSNAL	7.0304 | 0.1691	0.7495 | 0.0225	1.1047 | 0.0426	0.9045 | 0.0211	0.9836 | 0.0028	5814.36	5774.74
iADMM	7.0317 | 0.1719	0.7497 | 0.0227	1.1048 | 0.0434	0.9046 | 0.0213	0.9834 | 0.0030	5873.46	5795.42
eADMM	7.0318 | 0.1719	0.7497 | 0.0227	1.1048 | 0.0434	0.9046 | 0.0213	0.9834 | 0.0030	5873.68	5796.04
SCIO	7.2442 | 0.1734	0.7639 | 0.0221	1.0730 | 0.0371	0.8720 | 0.0250	0.9887 | 0.0021	4461.72	4459.66
glasso	7.4821 | 0.0943	0.7834 | 0.0138	1.2662 | 0.0640	0.8851 | 0.0161	0.9762 | 0.0028	7591.48	7588.22
Model 2							
MARS	11.4759 | 0.0392	1.5585 | 0.0122	1.9502 | 0.0459	0.6748 | 0.0071	0.9818 | 0.0004	6982.66	6981.14
SSNAL	11.4761 | 0.0392	1.5585 | 0.0122	1.9501 | 0.0459	0.6751 | 0.0071	0.9817 | 0.0004	7024.72	6985.08
iADMM	11.4744 | 0.0392	1.5583 | 0.0122	1.9504 | 0.0460	0.6772 | 0.0072	0.9812 | 0.0004	7160.74	7019.38
eADMM	11.4745 | 0.0392	1.5583 | 0.0122	1.9504 | 0.0460	0.6773 | 0.0072	0.9812 | 0.0004	7161.74	7020.46
SCIO	11.6522 | 0.0380	1.5777 | 0.0117	1.9261 | 0.0375	0.6270 | 0.0076	0.9872 | 0.0003	5461.84	5459.66
glasso	11.8553 | 0.0308	1.6073 | 0.0087	2.1313 | 0.0661	0.6549 | 0.0078	0.9724 | 0.0007	9206.80	9202.38
Model 3							
MARS	8.5468 | 0.1284	0.9120 | 0.0193	1.1033 | 0.0318	0.5969 | 0.0457	0.9922 | 0.0022	2931.00	2930.48
SSNAL	8.5418 | 0.1259	0.9114 | 0.0189	1.1044 | 0.0330	0.5993 | 0.0452	0.9919 | 0.0022	2990.91	2966.44
iADMM	8.5453 | 0.1284	0.9120 | 0.0192	1.1036 | 0.0319	0.5978 | 0.0459	0.9920 | 0.0023	2974.42	2948.34
eADMM	8.5454 | 0.1284	0.9120 | 0.0192	1.1036 | 0.0319	0.5978 | 0.0459	0.9920 | 0.0023	2974.72	2948.28
SCIO	8.7575 | 0.0893	0.9284 | 0.0168	1.0693 | 0.0214	0.5129 | 0.0325	0.9959 | 0.0012	1785.08	1784.12
glasso	8.7660 | 0.0558	0.9197 | 0.0135	1.1593 | 0.0361	0.5450 | 0.0227	0.9923 | 0.0017	2774.44	2772.96
Model 4							
MARS	5.1517 | 0.1276	0.5186 | 0.0162	0.7655 | 0.0446	0.0130 | 0.0024	0.9925 | 0.0039	2744.56	2743.52
SSNAL	5.1527 | 0.1274	0.5188 | 0.0162	0.7652 | 0.0445	0.0132 | 0.0024	0.9924 | 0.0040	2785.84	2757.18
iADMM	5.1503 | 0.1271	0.5186 | 0.0162	0.7660 | 0.0448	0.0132 | 0.0024	0.9922 | 0.0041	2798.62	2775.30
eADMM	5.1503 | 0.1271	0.5186 | 0.0162	0.7660 | 0.0448	0.0132 | 0.0024	0.9922 | 0.0041	2798.82	2775.22
SCIO	5.2148 | 0.0935	0.5209 | 0.0148	0.7515 | 0.0404	0.0118 | 0.0013	0.9938 | 0.0031	2432.92	2431.60
glasso	5.2565 | 0.0564	0.5263 | 0.0119	0.8307 | 0.0429	0.0155 | 0.0007	0.9901 | 0.0036	3367.54	3365.96
Model 5							
MARS	4.9748 | 0.1298	0.5449 | 0.0209	0.9325 | 0.0452	0.9949 | 0.0022	0.9820 | 0.0035	6007.48	6006.36
SSNAL	4.9701 | 0.1281	0.5444 | 0.0214	0.9332 | 0.0461	0.9949 | 0.0021	0.9818 | 0.0034	6047.16	6046.34
iADMM	4.9795 | 0.1342	0.5455 | 0.0209	0.9321 | 0.0455	0.9953 | 0.0021	0.9815 | 0.0038	6122.64	5974.92
eADMM	4.9822 | 0.1320	0.5458 | 0.0208	0.9313 | 0.0464	0.9953 | 0.0021	0.9816 | 0.0038	6104.16	5956.28
SCIO	5.0845 | 0.0595	0.5481 | 0.0163	0.8936 | 0.0424	0.9922 | 0.0023	0.9877 | 0.0003	4680.18	4678.16
glasso	5.2337 | 0.0559	0.5754 | 0.0146	1.0700 | 0.0591	0.9974 | 0.0014	0.9734 | 0.0007	8004.50	8000.88

4.2 Simulation studies

In this subsection, we will compare performance of the algorithms by using randomly generated data from the following five different models:

1. $\Theta_{ij} = 0.2$, if $i \neq j$ and $1 \leq |i - j| \leq 2$; $\Theta_{ii} = 1$; $\Theta_{ij} = 0$ otherwise.

Table 2: Average performance among different methods for precision matrix estimation with 100 replications, p = 1000 in models 1 to 4, p = 1024 in model 5 and n = 400 for all the models

	Frobenius	Spectral	Infity	TP	TN	s_{off}	$\bar{s}_{\text{off}}$
	mean | sd	mean | sd	mean | sd	mean | sd	mean | sd	mean	mean
Model 1							
MARS	10.9437 | 0.0433	0.8193 | 0.0119	1.0949 | 0.0280	0.8134 | 0.0069	0.9947 | 0.0001	8353.36	8351.84
SSNAL	10.9469 | 0.0438	0.8195 | 0.0119	1.0947 | 0.0280	0.8134 | 0.0069	0.9947 | 0.0001	8375.55	8350.06
iADMM	10.9460 | 0.0433	0.8195 | 0.0119	1.0949 | 0.0280	0.8148 | 0.0070	0.9945 | 0.0001	8528.64	8348.64
eADMM	10.9460 | 0.0433	0.8195 | 0.0119	1.0949 | 0.0280	0.8148 | 0.0070	0.9945 | 0.0001	8529.00	8349.04
SCIO	11.2131 | 0.0424	0.8315 | 0.0117	1.0782 | 0.0237	0.7697 | 0.0065	0.9964 | 0.0001	6399.24	6396.24
glasso	11.2632 | 0.0346	0.8287 | 0.0081	1.2692 | 0.0374	0.8111 | 0.0065	0.9903 | 0.0002	12724.18	12717.21
Model 2							
MARS	17.2284 | 0.0372	1.6560 | 0.0094	1.9535 | 0.0282	0.5399 | 0.0052	0.9944 | 0.0001	9390.04	9388.16
SSNAL	17.2320 | 0.0372	1.6564 | 0.0094	1.9534 | 0.0282	0.5399 | 0.0051	0.9944 | 0.0001	9407.36	9382.32
iADMM	17.2295 | 0.0372	1.6561 | 0.0094	1.9536 | 0.0282	0.5427 | 0.0050	0.9942 | 0.0001	9624.22	9377.70
eADMM	17.2295 | 0.0372	1.6561 | 0.0094	1.9536 | 0.0282	0.5427 | 0.0050	0.9942 | 0.0001	9625.02	9377.74
SCIO	17.4139 | 0.0363	1.6704 | 0.0092	1.9402 | 0.0236	0.4965 | 0.0052	0.9962 | 0.0001	7208.28	7204.52
glasso	17.3500 | 0.0300	1.6623 | 0.0064	2.1461 | 0.0480	0.5721 | 0.0052	0.9883 | 0.0003	15757.24	15748.44
Model 3							
MARS	12.6053 | 0.0964	0.9441 | 0.0117	1.0930 | 0.0247	0.4480 | 0.0351	0.9978 | 0.0007	3461.28	3459.78
SSNAL	12.6059 | 0.0961	0.9441 | 0.0117	1.0928 | 0.0247	0.4498 | 0.0329	0.9978 | 0.0007	3486.69	3473.66
iADMM	12.6056 | 0.0957	0.9442 | 0.0117	1.0931 | 0.0248	0.4508 | 0.0332	0.9977 | 0.0007	3528.22	3473.72
eADMM	12.6057 | 0.0959	0.9442 | 0.0117	1.0931 | 0.0248	0.4508 | 0.0332	0.9977 | 0.0007	3528.30	3473.78
SCIO	12.8151 | 0.0246	0.9555 | 0.0103	1.0715 | 0.0142	0.3708 | 0.0089	0.9991 | 0.0001	1710.32	1709.18
glasso	12.7205 | 0.0376	0.9425 | 0.0099	1.1469 | 0.0319	0.4338 | 0.0151	0.9976 | 0.0005	3583.68	3581.24
Model 4							
MARS	7.7640 | 0.0493	0.5496 | 0.0115	0.7630 | 0.0332	0.0064 | 0.0001	0.9973 | 0.0001	4289.52	4288.96
SSNAL	7.7651 | 0.0493	0.5496 | 0.0115	0.7629 | 0.0331	0.0064 | 0.0001	0.9973 | 0.0001	4367.77	4347.22
iADMM	7.7642 | 0.0493	0.5497 | 0.0115	0.7631 | 0.0331	0.0065 | 0.0001	0.9972 | 0.0001	4414.12	4347.84
eADMM	7.7642 | 0.0493	0.5497 | 0.0115	0.7631 | 0.0331	0.0065 | 0.0001	0.9972 | 0.0001	4414.08	4347.86
SCIO	7.9572 | 0.0492	0.5570 | 0.0120	0.7479 | 0.0267	0.0054 | 0.0001	0.9982 | 0.0001	3341.48	3339.84
glasso	7.7360 | 0.0516	0.5459 | 0.0094	0.8877 | 0.0399	0.0096 | 0.0003	0.9943 | 0.0003	7458.78	7454.63
Model 5							
MARS	8.0015 | 0.0603	0.6061 | 0.0145	0.9493 | 0.0358	0.9893 | 0.0021	0.9938 | 0.0001	10389.18	10386.06
SSNAL	8.0058 | 0.0603	0.6064 | 0.0145	0.9493 | 0.0358	0.9893 | 0.0021	0.9932 | 0.0001	11059.55	10378.32
iADMM	8.0000 | 0.0604	0.6060 | 0.0145	0.9494 | 0.0358	0.9904 | 0.0019	0.9935 | 0.0001	10662.36	10385.10
eADMM	8.0002 | 0.0604	0.6060 | 0.0145	0.9494 | 0.0358	0.9904 | 0.0019	0.9935 | 0.0001	10663.20	10384.46
SCIO	8.3156 | 0.0613	0.6213 | 0.0147	0.9294 | 0.0325	0.9825 | 0.0026	0.9963 | 0.0001	7747.98	7745.16
glasso	8.2708 | 0.0559	0.6212 | 0.0111	1.1149 | 0.0560	0.9951 | 0.0015	0.9881 | 0.0002	16319.38	16310.56

2. $\Theta_{ij} = 0.2$, if $i \neq j$ and $1 \leq |i-j| \leq 4$; $\Theta_{ii} = 1$; $\Theta_{ij} = 0$ otherwise.

3. $\Theta = \text{diag}\{\Theta_0, \cdots, \Theta_0\}$ with $\Theta_0 \in \mathbb{S}^5$, the off-diagonal components are equal to 0.2 and the diagonal is all 1; $\Theta_{ij} = 0$ otherwise.

4. $\Theta_{ij} = 0.2^{|i-j|}$.

5. $\Theta_{ij} = 0.2$, if the remainder after division of i by $\sqrt{p}$ is not equal to 0 and $j = i + 1$; $\Theta_{ij} = 0.2$, if $j = i + \sqrt{p}$; $\Theta_{ii} = 1$; $\Theta_{ij} = 0$ otherwise.

For the last model, the sample dimension p must satisfy with that $\sqrt{p}$ is an integer. Thus, we set $p = 500$ in models 1 to 4 and $p = 484$ in model 5 in test 1, $p = 1000$ in models 1 to 4 and $p = 1024$ in model 5 in test 2 and $n = 400$ for all the models in both two tests. We point out that models 1, 2, and 5 are derived from Zhang and Zou (2014). In the tests, the estimated precision matrix for each random sample is chosen by five-fold cross-validation. The test results conducted by 100 replications are shown in Tables 1 and 2. The performance among different algorithms compared in terms of seven quantities: the Frobenius norm (Frobenius), the spectral norm (Spectral) and the infinity norm (Infinity) between the estimated precision matrix and the true precision matrix, the ratio (TP) of correctly estimated non-zero components, the ratio (TN) of correctly estimated zero components, the number (s_{off}) of the off-diagonal non-zero components and the number ($\bar{s}_{\text{off}}$) of the off-diagonal components whose absolute values are greater than 10^{-5} in the estimated solution.

By comparing the first five quantities of the results in Tables 1 and 2, we can see that the performance among the first four algorithms is similar to each other. The reason is that they are using the same estimator and stopping criteria. The performance of these four algorithms is slightly better than that of both SCIO and glasso in most cases. We point out that the main difference between our MARS and SCIO is that SCIO estimates the precision matrix column by column, and the stopping criteria set by SCIO is based on the values of the updates instead of the relative KKT residual we used. As for the sparsity of the estimated solutions, the results are quite different. Specifically, SCIO always generates more sparse solutions compared with others and the solutions obtained by glasso are less sparse than others for all the cases except model 3 in test 1. MARS can generate more sparse solutions when comparing with SSNAL and two kinds of ADMM, and more importantly, it can generate solutions with fewer small value components. In other words, we do not need to artificially remove some components with insignificant values. In addition, all the estimated precision matrices in Tables 1 and 2 are positive definite.

Next, we compare the total computation time with different algorithms for generating a solution path. For simplicity, here we still use the five models given earlier to generate random data. Firstly, we set p to be 1000 and 2000 in models 1 to 4, set p to be 1024 and 2025 in model 5 and n to be 50 and 100 to obtain four combinations for the usage of the following numerical experiments. After observing the test results in Table 3, we found that MARS, SSNAL, glasso and EQUAL are much more efficient than other algorithms, and so when p is set to be 3000 and 5000 for models 1 to 4 and 3025 and 5041 in model 5, we will only focus on the comparison among these four algorithms. Due to the obvious gap of the computation time among different algorithms, we have carried out 10 repeated experiments here only. The vector of tuning parameter λ is fixed from 0.5 to 0.99 by 0.01

Table 3: Average computation time (seconds) with different algorithms of 10 replications to generating a solution path for some low-dimensional generated data

		Model 1 mean | sd	Model 2 mean | sd	Model 3 mean | sd	Model 4 mean | sd	Model 5 mean | sd
Models 1 to 4: p = 1000; Modle 5: p = 1024						
n = 50	MARS	1.34 | 0.08	1.31 | 0.07	1.31 | 0.11	1.28 | 0.08	1.45 | 0.08
	SSNAL	5.59 | 0.63	5.44 | 0.48	5.42 | 0.78	5.16 | 0.54	6.72 | 0.58
	iADMM	21.63 | 1.54	21.19 | 1.07	20.69 | 2.02	20.44 | 1.33	24.78 | 1.71
	eADMM	30.43 | 1.94	30.38 | 1.84	29.34 | 2.51	29.79 | 1.65	34.76 | 2.59
	SCIO	19.53 | 0.09	19.53 | 0.11	19.58 | 0.24	19.65 | 0.17	19.61 | 0.18
	EQUAL	7.08 | 0.30	7.21 | 0.21	7.56 | 0.18	7.53 | 0.32	8.11 | 0.30
	CLIME	57.71 | 0.31	58.43 | 0.50	58.20 | 0.40	58.53 | 0.73	57.82 | 0.56
	glasso	1.67 | 0.01	1.67 | 0.02	1.71 | 0.02	1.69 | 0.03	1.70 | 0.06
	BigQuic	10.88 | 0.40	10.58 | 0.05	10.62 | 0.11	10.60 | 0.05	11.06 | 0.83
n = 100	MARS	0.97 | 0.07	1.00 | 0.07	0.96 | 0.03	0.99 | 0.06	1.18 | 0.17
	SSNAL	1.70 | 0.31	1.77 | 0.23	1.65 | 0.03	1.71 | 0.38	2.86 | 0.91
	iADMM	5.43 | 0.78	5.86 | 0.79	5.12 | 0.04	5.46 | 0.88	8.28 | 1.75
	eADMM	7.09 | 0.65	7.45 | 0.81	6.80 | 0.09	7.06 | 0.70	9.83 | 1.81
	SCIO	19.39 | 0.04	19.55 | 0.18	19.53 | 0.11	19.82 | 0.29	19.58 | 0.12
	EQUAL	4.26 | 0.08	4.26 | 0.07	4.40 | 0.08	4.45 | 0.04	4.36 | 0.09
	CLIME	57.57 | 0.37	58.47 | 0.60	57.99 | 0.49	58.99 | 0.82	57.86 | 0.52
	glasso	1.65 | 0.01	1.66 | 0.01	1.68 | 0.03	1.68 | 0.04	1.68 | 0.02
	BigQuic	14.65 | 0.03	14.60 | 0.02	14.60 | 0.03	14.65 | 0.05	14.71 | 0.08
Models 1 to 4: p = 2000; Modle 5: p = 2025						
n = 50	MARS	5.47 | 0.24	5.61 | 0.25	5.45 | 0.24	5.66 | 0.32	5.54 | 0.18
	SSNAL	26.08 | 1.62	27.31 | 2.38	25.79 | 1.28	27.70 | 2.63	28.70 | 2.33
	iADMM	171.31 | 7.47	167.03 | 8.69	168.18 | 7.73	178.36 | 13.31	183.69 | 10.00
	eADMM	393.00 | 15.69	402.37 | 16.71	384.16 | 15.20	411.98 | 17.17	429.69 | 27.03
	SCIO	224.02 | 0.67	225.31 | 2.49	224.05 | 0.59	225.00 | 1.23	224.85 | 2.74
	EQUAL	56.53 | 2.47	58.66 | 2.60	56.35 | 2.63	58.92 | 3.37	58.19 | 1.10
	CLIME	494.14 | 1.70	494.70 | 2.62	493.80 | 1.15	495.18 | 3.03	493.86 | 3.10
	glasso	8.09 | 0.18	8.00 | 0.12	7.99 | 0.06	8.05 | 0.09	8.07 | 0.13
	BigQuic	46.49 | 0.22	46.16 | 0.28	46.09 | 0.23	46.21 | 0.48	46.59 | 0.38
n = 100	MARS	4.40 | 0.06	4.35 | 0.12	4.24 | 0.18	4.49 | 0.21	4.79 | 0.21
	SSNAL	7.96 | 0.39	8.01 | 0.45	7.73 | 0.24	8.23 | 0.82	11.13 | 1.65
	iADMM	39.17 | 4.03	34.83 | 2.20	35.70 | 2.53	41.19 | 6.06	58.23 | 6.08
	eADMM	62.98 | 11.00	61.06 | 9.99	56.52 | 5.55	67.83 | 14.71	100.20 | 8.77
	SCIO	224.31 | 0.53	224.14 | 0.80	224.12 | 0.37	224.14 | 1.00	225.11 | 2.88
	EQUAL	30.81 | 0.14	30.79 | 0.22	29.60 | 1.51	30.77 | 0.34	28.01 | 0.33
	CLIME	495.46 | 1.06	496.80 | 3.90	496.02 | 5.03	495.19 | 1.87	493.40 | 3.59
	glasso	7.93 | 0.02	7.91 | 0.01	7.93 | 0.02	7.94 | 0.05	7.95 | 0.02
	BigQuic	56.35 | 0.27	56.08 | 0.25	55.96 | 0.22	55.97 | 0.28	56.15 | 0.26

Table 4: Average computation time (seconds) with different algorithms of 10 replications to generating a solution path for some high-dimensional generated data

		Model 1	Model 2	Model 3	Model 4	Model 5
		mean | sd	mean | sd	mean | sd	mean | sd	mean | sd
Models 1 to 4: p = 3000; Modle 5: p = 3024						
n = 50	MARS	12.73 | 0.48	12.94 | 0.22	12.63 | 0.36	12.73 | 0.29	13.93 | 0.29
	SSNAL	64.71 | 5.17	66.19 | 2.63	64.09 | 2.80	65.44 | 2.31	76.58 | 3.83
	EQUAL	177.31 | 3.46	174.95 | 1.08	175.21 | 0.67	175.41 | 0.56	188.42 | 3.08
	glasso	21.13 | 0.40	22.76 | 1.08	22.83 | 0.88	23.06 | 0.97	22.69 | 0.66
n = 100	MARS	9.59 | 0.17	9.71 | 0.40	9.54 | 0.25	9.61 | 0.34	11.27 | 0.47
	SSNAL	19.15 | 0.79	21.13 | 2.11	20.85 | 0.94	20.70 | 2.44	31.12 | 3.67
	EQUAL	85.32 | 0.42	86.35 | 0.61	85.87 | 0.54	85.61 | 0.37	87.16 | 0.92
	glasso	22.29 | 0.82	22.71 | 1.46	22.85 | 0.94	22.31 | 0.52	22.80 | 1.53
Models 1 to 4: p = 5000; Modle 5: p = 5041						
n = 50	MARS	33.14 | 0.48	32.83 | 0.56	32.73 | 0.94	32.80 | 0.55	34.43 | 0.40
	SSNAL	184.99 | 6.77	182.30 | 5.16	176.63 | 5.56	179.72 | 4.85	203.69 | 3.53
	EQUAL	628.45 | 9.60	636.69 | 30.54	638.59 | 26.00	642.46 | 25.60	652.96 | 16.36
	glasso	−	−	−	−	−
n = 100	MARS	25.22 | 0.54	25.13 | 0.35	25.10 | 0.26	25.20 | 0.36	28.51 | 1.21
	SSNAL	52.06 | 4.05	50.39 | 1.31	50.46 | 1.86	51.90 | 2.54	73.57 | 8.13
	EQUAL	306.33 | 9.36	308.95 | 10.73	303.11 | 1.16	308.40 | 9.15	322.45 | 14.73
	glasso	−	−	−	−	−

"−" indicates out of memory.

with 50 parameters in total. This setting is because when λ is too small, there may not exist a feasible solution for the estimator. Especially when the gap between p and n is greater, the smallest λ that guarantees the existence of optimal solutions of the estimator will be greater. For fairness of comparison, we did not use Remark 7 to narrow the range of λ path for MARS, SSNAL, iADMM, and eADMM, and the generated data are all standardized in the first beginning.

The computation time (in terms of the mean and standard deviation (sd)) of the tests are shown in Tables 3 and 4. We can see that MARS is significantly faster than other algorithms. Especially, when p is larger, the computation time of MARS is roughly half of that of glasso, which performs better compared with the remaining algorithms. In the next subsection, we will illustrate the promising performance of MARS on two real data sets with very high dimensions.

4.3 Real data analysis

In this subsection, we will use some real data sets to demonstrate the promising performance of our MARS for generating a solution path. The publicly available data sets we are going to use include a prostate data set (`https://web.stanford.edu/~hastie/CASI_files/DATA/prostate.html`) and a breast cancer data set (Hess et al., 2006), which can be found on (`https://bioinformatics.mdanderson.org/public-datasets/`). The prostate data set contains two groups, the first one is 6033 genetic activity measurements of 50 control subjects and the other is that of 52 prostate cancer subjects. Thus, the number of variables contained in the precision matrix that needs to be estimated is more than 18 million. As for

the breast cancer data set, it contains the measurements of 22283 genes with 133 subjects, where 99 of them are labeled as residual disease (RD) and the remaining 34 subjects are labeled as pathological complete response (pCR). For this data set, the estimated precision matrix contains about 250 million parameters.

Firstly, after standardizing the two groups of the prostate data set, we use MARS, SSNAL, EQUAL, and SCIO to generate solution paths for the two groups separately. We should note that, when λ is too small, there may not exist optimal solutions of the precision matrix estimator. Therefore, before going further to the main comparison tests, we should conduct some pre-tests to find a suitable smallest λ. The performance of the estimations generated by different algorithms is concluded in Table 5. Since η is to measure the distance between an estimated solution and the optimal solution set, we notice that when λ is larger, the estimated solutions generated by SCIO perform very well, but when λ gradually becomes smaller, its estimated solutions are far away from the optimal solution set, which can also be observed from the associated objective value. The performance of EQUAL is the opposite, that is, it performs better when λ is smaller. We should point out that even if the stopping tolerance of EQUAL has been set to be 10^{-6}, none of the generated solutions make η less than 10^{-3}. Thus, by comparing the objective value and η, we conclude that MARS and SSNAL can outperform both EQUAL and SCIO since all the η are smaller than the set tolerance 10^{-4}. Although both MARS and SSNAL can generate satisfactory solutions, from Table 6, we find that MARS is much more efficient. In particular, the computation time of SSNAL to generate the solution path is more than 17 times of that of MARS in the Control group, and is more than 20 times of that of MARS in the Cancer group. This can also be seen from Figure 1, which illustrates that MARS has high efficiency in generating solutions for each λ. Besides, we obtain the final precision matrix estimations of the two different groups through 5-fold cross-validation, and the corresponding graphs are shown in Figure 2. From this figure, we can clearly see that the genes of the control group and the cancer group have different connections.

Next, we will test the performance of MARS and glasso on the breast cancer data set. We follow the same assumption stated in Cai et al. (2011), that is, this gene measurements data are normally distributed with $N(\mu_k, \Sigma)$, $k = 1, 2$, where Σ is the same for RD and pCR group, but the means are different. Firstly, two-sample t-tests are performed with some given p-value tolerances, which are set to be 0.005, 0.01, 0.05, 0.1, and 1, to obtain the most significant genes (with a smaller p-value). Under those set p-values, the number of chosen genes are 1228, 1646, 3640, 5418, and 22283 respectively. Note that, the last one is just all the genes with nearly 250 million parameters. We point out that, the λ paths for all the tests, except the test with p-value tolerance 0.05, are set from $\lambda_{\min}$ to 1 by 0.01, where $\lambda_{\min}$ is decided by some pre-tests with the D-trace estimator. When the p-value tolerance is 0.05, if the gap between two subsequent tuning parameters in the path is 0.01, glasso will fail due to insufficient memory, so we set the λ gap for this test to be 0.02. The tuning parameters for each test are chosen by five-fold cross-validation, and the total computation times are concluded in Table 7. The estimated graphs obtained by MARS and glasso with p-value tolerance 0.005, 0.01, and 0.01 can be found in Figure 3. From this figure, we notice that the graphs obtained by MARS and glasso are similar to each other, but the times taken by MARS are obviously less than those taken by glasso. Especially when the p-value tolerance is 0.05, the total computation time of glasso is more than 20 times that of

Table 5: The objective values and the relative KKT residuals (η) of paths of estimated precision matrices generated by different algorithms for the prostate data set

| | control group | | | | | | | | cancer group | | | | | | | |
| | Objective value | | | | η | | | | Objective value | | | | η | | | |
λ	MARS	SSNAL	EQUAL	SCIO	MARS	SSNAL	EQUAL	SCIO	MARS	SSNAL	EQUAL	SCIO	MARS	SSNAL	EQUAL	SCIO
0.99	-3016.50	-3016.50	924.68	-3016.50	6.47e-06	4.80e-06	5.05e-02	2.05e-07	-3016.50	-3016.50	-1964.80	-3016.50	9.36e-06	6.95e-06	3.40e-02	2.33e-07
0.98	-3016.50	-3016.50	-861.70	-3016.53	3.31e-05	3.15e-05	3.89e-01	3.44e-07	-3016.50	-3016.50	-1966.85	-3016.56	4.33e-05	4.11e-05	3.41e-02	4.69e-07
0.97	-3016.50	-3016.50	-1798.68	-3016.63	7.56e-05	7.43e-05	3.54e-02	5.44e-07	-3016.50	-3016.50	-2448.14	-3016.72	9.84e-05	9.67e-05	3.9e-02	6.86e-07
0.96	-3016.83	-3016.52	-1802.12	-3016.83	4.86e-06	9.40e-05	3.56e-02	7.04e-07	-3016.68	-3017.04	-2447.90	-3017.05	7.30e-06	9.43e-05	6.40e-02	7.99e-07
0.95	-3017.10	-3016.94	-2351.89	-3017.19	9.54e-05	7.80e-05	7.22e-02	8.51e-07	-3017.24	-3017.58	-2456.80	-3017.59	1.04e-05	8.73e-05	6.31e-02	1.01e-06
0.94	-3017.75	-3017.53	-2356.68	-3017.75	3.39e-06	7.18e-06	7.18e-02	1.03e-06	-3018.06	-3018.40	-2700.02	-3018.41	1.15e-05	8.54e-05	2.20e-02	1.21e-06
0.93	-3018.55	-3018.22	-2371.45	-3018.55	7.20e-06	9.92e-06	7.04e-02	5.63e-05	-3019.23	-3019.57	-2698.98	-3019.58	1.16e-05	8.50e-05	2.22e-02	1.35e-06
0.92	-3019.66	-3019.39	-2645.23	-3019.66	8.62e-06	8.12e-05	2.33e-02	1.21e-04	-3020.79	-3021.14	-2704.31	-3021.15	1.17e-05	8.53e-05	2.22e-02	1.49e-06
0.91	-3021.13	-3020.90	-2643.28	-3021.13	7.98e-06	7.48e-06	2.35e-02	1.82e-04	-3022.80	-3023.16	-2858.82	-3023.17	1.24e-05	8.55e-05	2.50e-02	1.58e-06
0.9	-3023.02	-3022.80	-2648.98	-3023.01	8.74e-06	7.22e-06	2.35e-02	2.47e-04	-3025.32	-3025.69	-2858.34	-3025.70	1.17e-05	8.59e-05	2.54e-02	1.70e-06
0.89	-3025.39	-3025.18	-2832.42	-3025.37	8.11e-06	7.04e-05	2.70e-02	3.28e-04	-3028.40	-3028.77	-2860.17	-3028.78	1.16e-05	8.60e-05	2.55e-02	1.04e-05
0.88	-3028.30	-3028.09	-2833.94	-3028.27	8.69e-06	6.95e-06	2.73e-02	4.00e-04	-3032.08	-3032.46	-2863.98	-3032.47	1.22e-05	8.63e-05	2.56e-02	1.13e-04
0.87	-3031.79	-3031.60	-2837.75	-3031.75	1.51e-05	6.88e-05	2.74e-02	4.81e-04	-3036.42	-3036.81	-2931.86	-3036.77	1.17e-05	8.69e-05	1.40e-02	4.20e-04
0.86	-3035.94	-3035.74	-2912.43	-3035.86	1.12e-05	6.84e-05	1.49e-02	6.08e-04	-3041.49	-3041.88	-2939.44	-3041.71	1.17e-05	8.67e-05	1.39e-02	7.96e-04
0.85	-3040.77	-3040.58	-2918.38	-3040.61	1.57e-05	6.79e-05	1.49e-02	8.34e-04	-3047.34	-3047.73	-2995.49	-3046.91	1.30e-05	8.65e-05	1.05e-02	1.68e-03
0.84	-3046.36	-3046.16	-2989.64	-3046.13	9.47e-06	6.66e-05	9.71e-04	9.71e-04	-3054.01	-3054.41	-3002.93	-3050.94	1.19e-05	8.64e-05	1.08e-02	3.73e-03
0.83	-3052.73	-3052.54	-2993.46	-3051.80	1.60e-05	6.73e-05	1.20e-02	1.81e-03	-3061.57	-3061.97	-3008.80	-3057.54	1.15e-05	8.62e-05	1.14e-02	3.70e-03
0.82	-3059.95	-3059.77	-2997.82	-2992.17	1.74e-05	6.70e-05	1.30e-02	5.80e-02	-3070.04	-3070.43	-3020.96	-3015.15	1.97e-05	8.61e-05	9.59e-03	3.61e-02
0.81	-3068.06	-3067.88	-3014.42	-806.76	1.82e-05	6.70e-05	1.00e-02	4.25e-01	-3079.48	-3079.89	-3040.97	-2534.57	1.35e-05	8.59e-05	7.75e-03	1.47e-01
0.8	-3077.11	-3076.93	-3057.88	1170.63	1.84e-05	6.66e-05	5.86e-03	4.60e-01	-3089.94	-3090.34	-3067.80	-464.84	2.07e-05	8.56e-05	5.64e-03	2.49e-01
0.79	-3087.13	-3086.95	-3069.02	4.49e+03	1.90e-05	6.62e-05	6.06e-03	4.36e-01	-3101.46	-3101.87	-3081.18	9861.79	1.38e-05	8.55e-05	5.79e-03	3.60e-01
0.78	-3098.15	-3097.98	-3066.00	1.24e+04	1.86e-05	6.59e-05	1.06e-02	4.22e-01	-3114.08	-3114.48	-3084.28	6.00e+04	2.51e-05	8.52e-05	8.97e-03	4.75e-01
0.77	-3110.23	-3110.05	-3085.58	3.13e+04	1.85e-05	6.57e-05	7.64e-03	4.21e-01	-3127.84	-3128.26	-3102.08	1.73e+05	3.06e-05	8.50e-05	7.10e-03	5.18e-01
0.76	-3123.39	-3123.21	-3106.15	7.23e+04	1.82e-05	6.54e-05	5.13e-03	4.29e-01	-3143.20	-3143.20	-3124.53	3.72e+05	2.91e-05	8.46e-05	4.94e-03	5.32e-01
0.75	-3137.67	-3137.49	-3121.64	1.54e+05	3.02e-05	6.55e-05	4.35e-03	4.38e-01	-3158.91	-3159.40	-3144.88	6.84e+05	2.53e-05	9.04e-05	3.75e-03	5.34e-01
0.74	-3153.12	-3152.93	-3139.48	3.12e+05	1.95e-05	6.48e-05	4.26e-03	4.52e-01	-3176.32	-3176.87	-3164.75	1.15e+06	2.60e-05	8.64e-05	3.75e-03	5.32e-01
0.73	-3169.78	-3169.57	-3144.13	5.54e+05	3.63e-05	6.50e-05	8.05e-03	4.58e-01	-3195.04	-3195.68	-3172.27	1.82e+06	4.81e-05	8.83e-05	7.52e-03	5.29e-01
0.72	-3187.73	-3187.46	-3162.36	9.22e+05	4.16e-05	6.75e-05	6.12e-03	4.62e-01	-3215.12	-3215.87	-3190.12	2.77e+06	5.54e-05	9.05e-05	6.33e-03	5.25e-01
0.71	-3206.96	-3206.65	-3188.84	1.55e+06	2.69e-05	6.93e-05	4.08e-03	4.72e-01	-3236.59	-3237.51	-3220.13	4.06e+06	4.12e-05	9.38e-05	3.79e-03	5.21e-01
0.7	-3227.61	-3227.17	-3204.06	2.58e+06	6.86e-05	7.28e-05	4.86e-03	4.85e-01	-3259.52	-3260.82	-3244.18	5.76e+06	2.41e-05	9.77e-05	3.82e-03	5.15e-01
0.69	-3249.70	-3249.08	-3229.79	4.01e+06	6.15e-05	7.72e-05	3.71e-03	4.93e-01	-3284.16	-3285.61	-3271.68	8.07e+06	3.79e-05	8.62e-05	3.70e-03	5.11e-01
0.68	-3273.48	-3272.43	-3257.15	6.07e+06	2.37e-05	8.30e-05	3.59e-03	4.99e-01	-3310.34	-3312.60	-3285.98	1.13e+07	3.12e-05	8.70e-05	7.49e-03	5.08e-01
0.67	-3298.90	-3297.28	-3289.17	9.13e+06	2.45e-05	8.99e-05	3.31e-03	5.07e-01	-3338.18	-3342.17	-3310.62	1.56e+07	3.54e-05	9.14e-05	6.34e-03	5.07e-01
0.66	-3326.19	-3323.70	-3310.93	1.38e+07	2.91e-05	9.78e-05	5.40e-03	5.18e-01	-3367.78	-3374.62	-3349.71	2.16e+07	3.83e-05	9.92e-05	3.58e-03	5.06e-01
0.65	-3355.73	-3352.01	-3330.89	2.05e+07	3.15e-05	9.37e-05	6.44e-03	5.28e-01	-3400.77	-3410.80	-3370.41	2.97e+07	4.81e-05	9.16e-05	7.03e-03	5.07e-01
0.64	-3388.70	-3382.07	-3370.90	3.08e+07	3.97e-05	9.88e-05	3.53e-03	5.40e-01	-3436.03	-3436.03	-3424.31	4.05e+07	5.28e-05	9.07e-05	2.13e-03	5.08e-01
0.63	-3426.33	-3415.49	-3379.63	—	4.39e-05	8.55e-05	5.77e-03	—	-3474.07	-3474.07	-3445.53	—	6.00e-05	9.92e-05	7.45e-03	—
0.62	-3469.25	-3449.88	-3429.19	—	5.38e-05	9.86e-05	5.79e-03	—	-3519.00	-3556.03	-3475.93	—	6.72e-05	9.98e-05	6.86e-03	—

"—" indicates out of memory.

Table 6: The computation time (seconds) of different algorithms for generating a solution path with the prostate data set

	MARS	SSNAL	EQUAL	SCIO
control group	97.29	1726.92	1283.25*	5372.70+
cancer group	103.86	2329.45	1235.68*	5614.00+

"*" indicates that none of the relative KKT residuals of EQUAL is less than 10^{-3}.

"+" indicates that, due to out of memory, the time here does not include the time for generating estimations by SCIO with the two smallest λ.
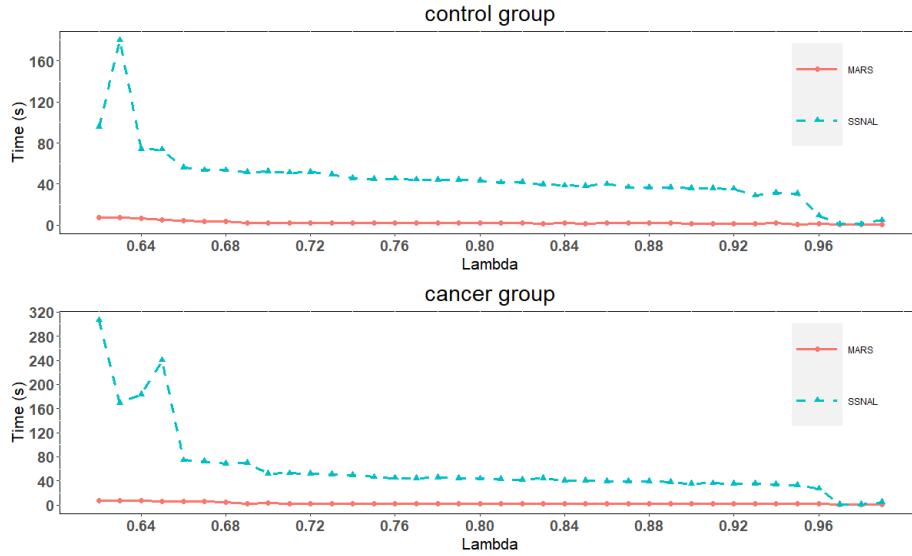

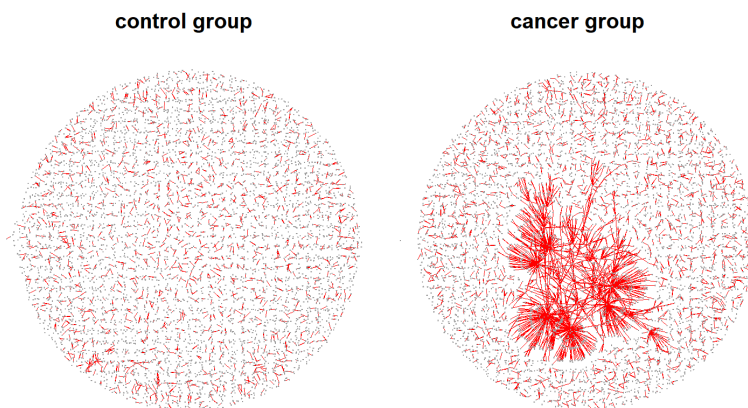Figure 1: The computation time of MARS and SSNAL for each λ with the prostate data set

Figure 2: The estimated graphs chosen by five-fold cross-validation generated by MARS with the prostrate data set

MARS. Besides, Figure 4 shows the estimated graphs obtained by MARS when the p-value tolerances are 0.1 and 1, but the figure on the right only plots the connections among the most significant 5418 genes.

Table 7: Test results of MARS and glasso on the breast cancer data sets with different p-value tolerances

p-value tolerance	No. of genes	time (mins) including cross-validation		No. of λ
		MARS	glasso	
0.005	1228	20.26	71.51	63
0.01	1646	23.06	159.79	60
0.05	3640	58.32	1257.81	28
0.1	5418	150.54	−	54
1	22283	553.35	−	29

"−" indicates out of memory.

5. Conclusions

In this paper, we have derived an efficient second-order algorithm for high-dimensional sparse precision matrices estimation under the ℓ_1-penalized D-trace loss. By using a dual approach and adopting an adaptive sieving reduction strategy, our algorithm is capable of handling large-scale datasets. Theoretical properties of our algorithm have been well established. In particular, we have shown that our algorithm enjoys global linear convergence and converges asymptotically superlinearly. Numerical results have further convincingly demonstrated the promising performance and high efficiency of our algorithm when compared with other state-of-the-art solvers. For instance, our algorithm can be up to $3 - 20$

32

MARS (0.005) glasso (0.005)

MARS (0.01) glasso (0.01)
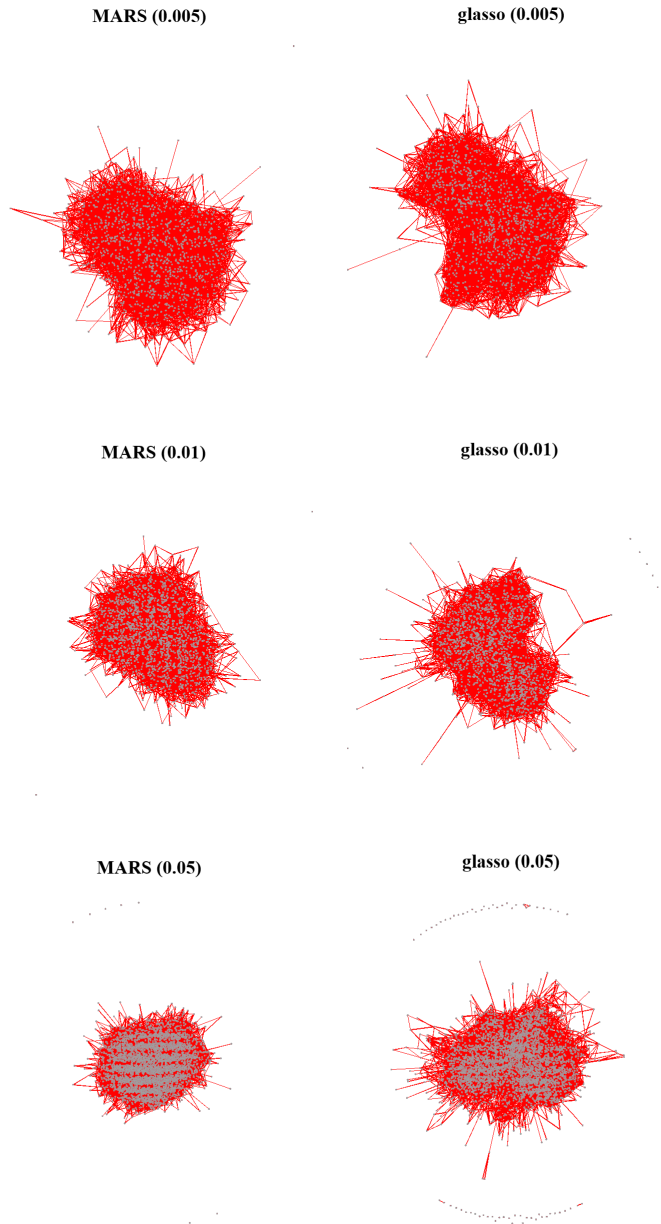
MARS (0.05) glasso (0.05)

Figure 3: The estimated graphs for the breast cancer data set chosen by five-fold cross-validation with using MARS and glasso under different p-value tolerances
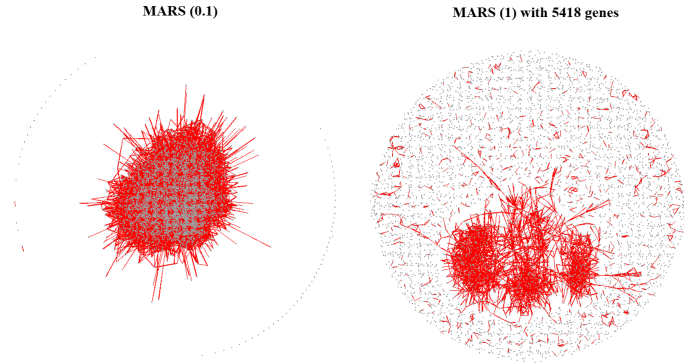
MARS (0.1)   **MARS (1) with 5418 genes**

Figure 4: The estimated graphs for the breast cancer data set chosen by five-fold cross-validation with using MARS under different p-value tolerances

times faster than glasso for some subsets of a breast cancer dataset with much less storage requirement.

We conclude by pointing out that our algorithm is not only designed for sparse precision matrix estimation but also can be extended to solve other matrix-form problems under a penalized quadratic loss function. More specifically, our algorithm can be extended to solve problems of the following form:

$$\min_{\Omega \in \mathbb{S}^p} \left\{ \frac{1}{2} \mathrm{tr}(\Omega \widehat{\Sigma} \Omega^T) - \mathrm{tr}(\widehat{Q}\Omega) + \mathrm{pen}_\lambda(\Omega) \right\}, \tag{41}$$

where $\mathrm{pen}_\lambda(\Omega)$ is a penalty term to encourage different structures. The algorithm we derived in this paper can be viewed as a special case with $\widehat{Q} = I_p$, and $\mathrm{pen}_\lambda(\Omega) = \lambda \|\Omega\|_{1,\mathrm{off}}$. With different choices of $\widehat{\Sigma}$ and $\widehat{Q}$, the quadratic loss (41) outputs sparse solutions for different statistical analysis such as canonical vectors for Fisher's LDA (Gaynanova et al., 2016), sparse canonical correlation analysis, and sparse sliced inverse regression (Tan et al., 2018). This is left for future work.

Appendix A.

This is the test performance of EQUAL, in terms of the objective value and relative KKT residual (η), for generating solution paths under different stopping tolerances, which are set to be 10^{-5} and 10^{-6}.

Table 8: Test performance of the estimated solution paths with different stopping tolerances by using EQUAL on the prostate data set

	control group				cancer group			
	EQUAL (1e-6)		EQAUL (1e-5)		EQUAL (1e-6)		EQAUL (1e-5)	
λ	objective value	η	objective value	η	objective value	η	objective value	η
0.99	924.68	5.05e-02	276912.70	5.16e-01	-1964.80	3.40e-02	264907.50	5.03e-01
0.98	-861.70	3.89e-01	21190.85	1.30e-01	-1966.85	3.41e-02	263387.60	5.06e-01
0.97	-1798.68	3.54e-02	20554.88	1.31e-01	-2448.14	6.39e-02	18529.18	1.36e-01
0.96	-1802.12	3.56e-02	9137.18	9.49e-02	-2447.90	6.40e-02	7894.90	8.70e-02
0.95	-2351.89	7.22e-02	9518.00	9.85e-02	-2456.80	6.31e-02	8194.18	8.98e-02
0.94	-2356.68	7.18e-02	9696.31	1.01e-01	-2700.02	2.20e-02	8304.20	9.18e-02
0.93	-2371.45	7.04e-02	9670.20	1.03e-01	-2698.98	2.22e-02	8225.00	9.29e-02
0.92	-2645.23	2.33e-02	9444.32	1.04e-01	-2704.31	2.22e-02	7962.94	9.30e-02
0.91	-2643.28	2.35e-02	9029.55	1.03e-01	-2858.82	2.50e-02	7530.24	9.22e-02
0.90	-2648.98	2.35e-02	8442.32	1.01e-01	-2858.34	2.54e-02	6943.96	9.04e-02
0.89	-2832.42	2.70e-02	2736.37	2.65e-01	-2860.17	2.55e-02	2018.53	3.05e-01
0.88	-2833.94	2.73e-02	2927.30	2.55e-01	-2863.98	2.56e-02	2186.41	2.92e-01
0.87	-2837.75	2.74e-02	3034.11	2.49e-01	-2931.86	1.40e-02	2280.82	2.84e-01
0.86	-2912.43	1.49e-02	3051.66	2.47e-01	-2939.44	1.39e-02	2295.27	2.81e-01
0.85	-2918.38	1.49e-02	2977.49	2.49e-01	-2995.49	1.05e-02	2226.93	2.82e-01
0.84	-2989.64	1.12e-02	2814.58	2.53e-01	-3002.93	1.08e-02	2076.93	2.88e-01
0.83	-2993.46	1.20e-02	2566.14	2.62e-01	-3008.80	1.14e-02	1850.30	2.98e-01
0.82	-2997.82	1.30e-02	2240.63	2.76e-01	-3020.96	9.59e-03	1554.29	3.14e-01
0.81	-3014.42	1.00e-02	1848.97	2.96e-01	-3040.97	7.75e-03	1198.97	3.37e-01
0.80	-3057.88	5.86e-03	1404.33	3.24e-01	-3067.80	5.64e-03	796.57	3.67e-01
0.79	-3069.02	6.06e-03	-480.09	5.09e-02	-3081.18	5.79e-03	-942.61	4.70e-02
0.78	-3066.00	1.06e-02	-236.76	5.41e-02	-3084.28	8.97e-03	-714.84	4.97e-02
0.77	-3085.58	7.64e-03	-41.10	5.71e-02	-3102.08	7.10e-03	-528.46	5.22e-02
0.76	-3106.15	5.13e-03	89.13	5.97e-02	-3124.53	4.94e-03	-401.82	5.43e-02
0.75	-3121.64	4.35e-03	141.05	6.16e-02	-3144.88	3.75e-03	-345.45	5.59e-02
0.74	-3139.48	4.26e-03	110.25	6.29e-02	-3164.75	3.75e-03	-363.44	5.68e-02
0.73	-3144.13	8.05e-03	-2.31	6.32e-02	-3172.27	7.52e-03	-455.55	5.70e-02
0.72	-3162.36	6.12e-03	-191.14	6.27e-02	-3190.12	6.33e-03	-616.71	5.65e-02
0.71	-3188.84	4.08e-03	-2316.43	5.47e-02	-3220.13	3.79e-03	-837.45	5.51e-02
0.70	-3204.06	6.15e-03	-2231.19	6.00e-02	-3244.18	3.82e-03	-2394.92	5.10e-02
0.69	-3229.79	3.71e-03	-2140.55	6.55e-02	-3271.68	3.70e-03	-2338.51	5.46e-02
0.68	-3257.15	3.59e-03	-2048.69	7.11e-02	-3285.98	7.49e-03	-2279.14	5.85e-02
0.67	-3289.17	3.31e-03	-1958.72	7.68e-02	-3310.62	6.34e-03	-2220.84	6.23e-02
0.66	-3310.93	5.40e-03	-1873.05	8.23e-02	-3349.71	3.58e-03	-2166.39	6.62e-02
0.65	-3330.89	6.44e-03	-1793.58	8.77e-02	-3370.41	7.03e-03	-2117.65	7.00e-02
0.64	-3370.90	3.53e-03	-1721.91	9.27e-02	-3424.12	2.13e-03	-2075.66	7.36e-02
0.63	-3399.63	5.77e-03	-1659.54	9.72e-02	-3445.53	7.45e-03	-2040.94	7.71e-02
0.62	-3429.19	5.79e-03	-1607.92	1.01e-01	-3475.93	6.86e-03	-2013.80	8.04e-02

References

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

Jonathan Borwein and Adrian S Lewis. *Convex analysis and nonlinear optimization: theory and examples.* Springer Science & Business Media, 2010.

Tony Cai, Weidong Liu, and Xi Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494): 594–607, 2011.

Frank H Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.

Mengyu Du. *An inexact alternating direction method of multipliers for convex composite conic programming with nonlinear constraints*. PhD thesis, Department of Mathematics, National University of Singapore, Singapore, 2015.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Irina Gaynanova, James G Booth, and Martin T Wells. Simultaneous sparse estimation of canonical vectors in the $p >> N$ setting. *Journal of the American Statistical Association*, 111(514):696–706, 2016.

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

Jean-Baptiste Hiriart-Urruty, Jean-Jacques Strodiot, and V Hien Nguyen. Generalized hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data. *Applied Mathematics and Optimization*, 11(1):43–56, 1984.

Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S Dhillon, Pradeep Ravikumar, and Russell A Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. In *NIPS*, volume 26, pages 3165–3173, 2013.

Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S Dhillon, and Pradeep Ravikumar. Quic: quadratic approximation for sparse inverse covariance estimation. *Journal of Machine Learning Research*, 15(1):2911–2947, 2014.

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

Claude Lemaréchal and Claudia Sagastizábal. Practical aspects of the moreau–yosida regularization: Theoretical preliminaries. *SIAM Journal on Optimization*, 7(2):367–385, 1997.

Hongzhe Li and Jiang Gui. Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2): 302–317, 2006.

Stan Z Li. *Markov random field modeling in image analysis*. Springer Science & Business Media, 2009.

Xudong Li, Defeng Sun, and Kim-Chuan Toh. A highly efficient semismooth newton augmented lagrangian method for solving lasso problems. *SIAM Journal on Optimization*, 28(1):433–458, 2018.

Meixia Lin, Yancheng Yuan, Defeng Sun, and Kim-Chuan Toh. Adaptive sieving with ppdna: Generating solution paths of exclusive lasso models. *arXiv preprint arXiv:2009.08719*, 2020.

Weidong Liu and Xi Luo. Fast and adaptive sparse precision matrix estimation in high dimensions. *Journal of Multivariate Analysis*, 135:153–162, 2015.

Fernando Javier Luque. Asymptotic convergence analysis of the proximal point algorithm. *SIAM Journal on Control and Optimization*, 22(2):277–293, 1984.

Nicolai Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1): 374–393, 2007.

Nicolai Meinshausen, Peter Bühlmann, et al. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

Figen Oztoprak, Jorge Nocedal, Steven Rennie, and Peder A Olsen. Newton-like methods for sparse inverse covariance estimation. *Advances in Neural Information Processing Systems*, 25:755–763, 2012.

Haotian Pang, Han Liu, and Robert J Vanderbei. The fastclime package for linear programming and large-scale precision matrix estimation in R. *Journal of Machine Learning Research*, 2014.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, Bin Yu, et al. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Stephen M Robinson. Some continuity properties of polyhedral multifunctions. In *Mathematical Programming at Oberwolfach*, pages 206–214. Springer, 1981.

R Tyrrell Rockafellar. Augmented lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976.

Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 1970.

Katya Scheinberg, Shiqian Ma, and Donald Goldfarb. Sparse inverse covariance selection via alternating linearization methods. *arXiv preprint arXiv:1011.0097*, 2010.

Jie Sun. *On monotropic piecewise quadratic programming*. PhD thesis, Department of Mathematics, University of Washington, 1986.

Kean Ming Tan, Zhaoran Wang, Han Liu, and Tong Zhang. Sparse generalized eigenvalue problem: Optimal statistical rates via truncated rayleigh flow. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5):1057–1086, 2018.

Cheng Wang and Binyan Jiang. An efficient admm algorithm for high dimensional precision matrix estimation via penalized quadratic loss. *Computational Statistics & Data Analysis*, 142:106812, 2020.

Anja Wille, Philip Zimmermann, Eva Vranová, Andreas Fürholz, Oliver Laule, Stefan Bleuler, Lars Hennig, Amela Prelić, Peter von Rohr, Lothar Thiele, et al. Sparse graphical gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. *Genome biology*, 5(11):1–13, 2004.

Daniela M Witten, Jerome H Friedman, and Noah Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4): 892–900, 2011.

Ming Yuan. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11:2261–2286, 2010.

Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Teng Zhang and Hui Zou. Sparse precision matrix estimation via lasso penalized d-trace loss. *Biometrika*, 101(1):103–120, 2014.

Xinyuan Zhao, Defeng Sun, and Kim-Chuan Toh. A newton-cg augmented lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4):1737–1765, 2010.