Math. Prog. Comp. (2010) 2:291–315
DOI 10.1007/s12532-010-0020-6

FULL LENGTH PAPER

An inexact interior point method for L_1-regularized sparse covariance selection

Lu Li · Kim-Chuan Toh

Received: 14 April 2010 / Accepted: 27 October 2010 / Published online: 20 November 2010

Abstract Sparse covariance selection problems can be formulated as log-determinant (log-det) semidefinite programming (SDP) problems with large numbers of linear constraints. Standard primal–dual interior-point methods that are based on solving the Schur complement equation would encounter severe computational bottlenecks if they are applied to solve these SDPs. In this paper, we consider a customized inexact primal–dual path-following interior-point algorithm for solving large scale log-det SDP problems arising from sparse covariance selection problems. Our inexact algorithm solves the large and ill-conditioned linear system of equations in each iteration by a preconditioned iterative solver. By exploiting the structures in sparse covariance selection problems, we are able to design highly effective preconditioners to efficiently solve the large and ill-conditioned linear systems. Numerical experiments on both synthetic and real covariance selection problems show that our algorithm is highly efficient and outperforms other existing algorithms.

Keywords Log-determinant semidefinite programming · Sparse inverse covariance selection · Inexact interior point method · Inexact search direction · Iterative solver

Mathematics Subject Classification (2000) 90C06 · 90C22 · 90C25 · 65F10

L. Li · K.-C. Toh (✉)
Department of Mathematics, National University of Singapore,
2 Science Drive 2, Singapore 117543, Singapore
e-mail: mattohkc@nus.edu.sg

L. Li
e-mail: lilu@nus.edu.sg

K.-C. Toh
Singapore-MIT Alliance, 4 Engineering Drive 3, Singapore 117576, Singapore

1 Introduction

Given n independent and identically-distributed (i.i.d.) observations $x^{(1)}, \dots, x^{(n)}$ drawn from a p-dimensional Gaussian distribution $\mathcal{N}(x; \mu, \Sigma_p)$, the sample covariance matrix $\widehat{\Sigma}$ is defined as the second moment matrix about the sample mean

$$\widehat{\Sigma} := \frac{1}{n} \sum_{k=1}^{n} (x^{(k)} - \hat{\mu})(x^{(k)} - \hat{\mu})^T,$$

where we use n instead of $n-1$ for the degree of freedom and $\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x^{(i)}$ is the sample mean. If Σ_p is non-singular, then the probability density function of x is given by

$$P(x; \hat{\mu}, \Sigma_p) = \frac{1}{(2\pi)^{p/2} (\det \Sigma_p)^{1/2}} \exp\left(-\frac{1}{2} (x - \hat{\mu})^T \Sigma_p^{-1} (x - \hat{\mu}) \right). \tag{1}$$

To estimate Σ_p from the sample $\mathcal{X} := \{x^{(1)}, \dots, x^{(n)}\}$, we consider the log-likelihood function

$$\log P(\mathcal{X}; \hat{\mu}, \Sigma_p) = -\frac{n}{2} \log(\det \Sigma_p) - \frac{1}{2} \sum_{k=1}^{n} (x^{(k)} - \hat{\mu})^T \Sigma_p^{-1} (x^{(k)} - \hat{\mu}) + c, \tag{2}$$

where c is a constant. Let $\mathcal{S}^p$ be the space of $p \times p$ symmetric matrices, and $\mathcal{S}^p_+$ ($\mathcal{S}^p_{++}$) be its subset of positive semidefinite (definite) matrices. We also use $X \succ 0$ ($\succeq 0$) to denote $X \in \mathcal{S}^p_{++}$ ($X \in \mathcal{S}^p_+$). Given $X, Y \in \mathcal{S}^p$, we define the inner product to be $X \bullet Y = \text{Trace}(XY)$. Then expression (2) can be rewritten as

$$\log P(\mathcal{X}; \hat{\mu}, \Sigma_p) = \frac{n}{2} \log(\det \Sigma_p^{-1}) - \frac{n}{2} \Sigma_p^{-1} \bullet \widehat{\Sigma} + c. \tag{3}$$

From (3) we can see that if $\widehat{\Sigma}$ is nonsingular (hence $n \geq p$), then

$$\widehat{\Sigma}^{-1} = \arg\max \left\{ \log P(\mathcal{X}; \hat{\mu}, \Sigma_p) \mid \Sigma_p \in \mathcal{S}^p_{++} \right\}$$

is the maximum likelihood estimator of the inverse covariance matrix Σ_p^{-1}, a.k.a. *precision matrix or concentration matrix*. However, in practice, one may not want to use $\widehat{\Sigma}^{-1}$ as the estimator of Σ_p^{-1} for a variety of reasons. The most obvious is that when $\widehat{\Sigma}$ is singular or nearly so, it is not a robust estimator of Σ_p^{-1} for many statistical purposes. The second is that one may want to impose structural conditions on Σ_p^{-1}, such as conditional independence between different components of x, which is reflected as zero entries in Σ_p^{-1} [39, Proposition 5.2].

The covariance selection problem was first introduced by Dempster [8], who suggested that *the covariance structure of a multivariate normal population can be simplified by setting elements of the inverse covariance matrix to zero*. Since then, the covariance selection model has become a common statistical tool to distinguish direct from indirect interactions among a set of variables. The graphical interpretation of the covariance selection model is called the Gaussian graphical model (GGM) [10,19]. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the GGM assumes a multivariate Gaussian distribution for the underlying data, and any nonadjacent pair in $\mathcal{G}$ indicates the independence between the underlying variables conditional on the remaining variables.

Applications of the covariance selection model or GGM can be found in various areas. In financial portfolio management, sparse portfolios with fewer assets incur less transaction costs and are more tractable. In [6], the covariance selection model is applied to find a sparse portfolio for mean-reversion trading strategies. In the research of dependency networks of genome data, a gene may play a role in many biological pathways and be associated with many other genes, though all these effects may be transmitted through direct associations of only a few genes in the neighborhood. The sparse gene association network exhibited in GGM can help to explain the known biological pathways and to provide insights on the unknowns, see for example [1,28]. Recent advances in DNA microarray technology require modeling an association network on a large number of genes (say, 10^3–10^4) from a small sample (say, 10^2), which will lead to a singular sample covariance matrix $\widehat{\Sigma}$. In this situation, the covariance selection model provides a systematic way to recover the population covariance matrix. For more applications of the covariance selection model, see [2,4].

As an important statistical problem, the covariance selection model has been intensively studied. There are many available statistical approaches, including the well known stepwise backward selection [10] and graphical-lasso [13,22]. However, the challenge from high dimensional data requires more efficient and robust algorithms to handle covariance selection problems. It is well known that covariance selection problems can be modeled as log-det semidefinite programming (SDP) problems. Typically, covariance selection problems can be divided into two classes, depending on whether the sparsity pattern is given a priori. If no sparsity pattern is assumed, sparsity can be enforced by l_1-regularized maximum log-likelihood estimation [7,13]:

$$\max \left\{ \log\det X - \widehat{\Sigma} \bullet X - H \bullet |X| \mid X \in \mathcal{S}^p_{++} \right\}. \tag{4}$$

In (4), $|X|$ takes entry-wise absolute value of the matrix X, and $H \in \mathcal{S}^p$ is a given nonnegative weight matrix. The latter controls the trade-off between the goodness-of-fit and the sparsity of X. A typical choice for H is $H = \rho E$, where E is the matrix of ones and ρ is a positive parameter. The matrix H may also assign zero weight on certain entries, such as the diagonal entries.

If the conditional independence structure between all variables is given, then the covariance selection problem can be formulated as a log-det maximization problem with linear constraints, that is, finding the maximum log-likelihood value subject to given entry-wise constraints [5,37]

$$\max \left\{ \log\det X - \widehat{\Sigma} \bullet X \mid X_{ij} = 0,\ (i,j) \in \Omega,\ \text{ and } X \in \mathcal{S}^p_{++} \right\}, \tag{5}$$

where Ω contains the indices of the upper triangular part of X that are supposed to be zero, i.e. the sparsity pattern. We let Ω^c be the set of the remaining indices of the upper triangular part of X. It is not difficult to find some connections between (4) and (5). In [13], the constraint $X_{ij} = 0$ in (5) is approximately enforced by setting H_{ij} to be a large number (say, 10^6) in (4), and this is the approach taken by [13] to solve (5) via (4).

Combining (4) and (5), we have the following l_1 regularized log-det semidefinite programming problem with linear constraints:

$$\max\left\{\log\det X - \widehat{\Sigma} \bullet X - \sum_{(ij)\notin\Omega} H_{ij}|X_{ij}| \mid X_{ij} = 0,\ (i,j)\in\Omega,\ X\in\mathcal{S}^p_{++}\right\}, \quad (6)$$

where Ω is as defined previously.

The problems (5)–(6) can be expressed as standard log-det SDP problems, which can in principle be solved by popular interior-point-method-based solvers such as SDPT3 [34] or SeDuMi [31]. However, the resulting standard log-det SDP problems typically have a large number of linear constraints m (even for moderate p, say $p \leq 100$) which the solvers in SDPT3 or SeDuMi cannot handle since the computational cost they need in each iteration is at least $\Theta(m^3)$ and the memory required is at least $\Theta(m^2)$ bytes. Thus a variety of customized algorithms have been developed to solve the problem (4) or (5), and most of them avoid the interior-point method approach.

The graphical Lasso methods developed by Meinshausen and Bühlmann [22] and Friedman et al. [13] for solving (4) are essentially block coordinate descent methods. In [1,7], d'Aspremont et al. considered Nesterov's smooth gradient method [23] as well as the block coordinate gradient method (BCG) for solving the dual of (4). The complexity of their implementation of Nesterov's first order algorithm is $\Theta(\frac{1}{\epsilon})$. For their BCG method, a box-constrained quadratic programming subproblem is to be solved in each iteration and the total complexity is unknown. Lu [20] proposed a variant of Nesterov's smoothing method for solving (4) with complexity $\Theta(\frac{1}{\sqrt{\epsilon}})$. More recently, Lu [21] proposed an adaptive version of Nesterov's smooth method (ANS) to solve (6) by solving a sequence of penalized problems of the form (4). Yuan [45] applied alternating direction methods to (4). Scheinberg and Rish [29] proposed a coordinate descent method for the primal problem of (4) in a greedy approach. First order methods only need small memory and cpu time per iteration, but they typically take many iterations to converge even for relative low accuracy. Krishnamurthy and d'Aspremont [17] developed a pathwise algorithm consisting of a predictor step with the conjugate gradient method and a corrector step with the block coordinate descent method. Second-order information is involved in their predictor step. Note that among the methods just described, the ANS method in [21] is the only one that is designed for the problem (6).

In [36], the authors considered the problem (5) but with Ω chosen to reflect local interactions between variables defined on a grid. They proposed to eliminate the constraints $X_{ij} = 0$ by using the parametrization $X = \sum_{(i,j)\in\Omega^c} X_{ij}E_{ij}$, where E_{ij} are unit matrices in $\mathcal{S}^p$. By doing so, (5) is converted to an unconstrained smooth convex problem for which they applied a standard Newton method with back-tracking line search to solve the problem. For the problems in [36], X is extremely sparse and well structured, and the authors were able to solve problems with p up to 34,000 and $|\Omega^c|$ up to 100,000 although the computer architecture used and times taken were not mentioned. More recently, Wang et al. [38] applied a Newton-CG primal proximal-point (PPA) method to solve (6). In the algorithm, each subproblem is solved by a semismooth Newton-CG method for which they used a preconditioned conjugate gradient (PCG) solver to compute an inexact Newton direction from the semismooth Newton equation in each inner iteration for the subproblem. Their numerical results show that PPA is efficient for solving problem (6) with p up to 2000 and m (the cardinality of Ω) up to 10^6. In particular, for randomly generated test examples, it can be a factor of 2–19 times faster than the ANS method in solving the problem (5).

It is well known that interior-point methods are robust and generally can obtain high accuracy solution with relatively few iterations. They are often the ideal choice for solving small to medium size generic SDP problems. It is not difficult to see that (6) can be cast as a standard log-det SDP problem with $p(p+1)/2$ linear constraints. In [44], Yuan and Lin actually applied a standard primal–dual interior-point method to solve (6). However, as we have pointed out earlier, a standard IPM solver would encounter a severe computational bottleneck or even become impractical when p is large since its computational cost per iteration is at least $O(p^6)$. But given that IPMs have the highly desirable property of being able to compute accurate solutions in relatively few iterations, it is worthwhile to design an IPM based method for solving (6) but one that overcomes the computational bottleneck just mentioned. In the case of linear and convex quadratic SDP, it has been demonstrated that inexact IPMs for which the large linear system in each iteration is solved approximately by a preconditioned iterative solver can be quite successful in solving certain classes of large primal and dual nondegenerate problems [32,33].

This motivates us to design a customized inexact primal–dual interior-point method for solving (6). The main idea in our inexact interior-point method (IIPM) is to compute the search direction at each iteration approximately by solving the large linear system of equations defining the search direction via an iterative solver such as the preconditioned conjugate gradient method. As the linear system is generally ill-conditioned, it is crucial for us to design efficient preconditioners to speed up the convergence of the iterative solver. In general, it is difficult to construct efficient preconditioners for such an ill-conditioned linear system of equations, and hence the computational cost for solving the linear system is generally quite high. (This also explains why recent approaches for solving large scale linear SDP problems have moved away from interior-point methods to algorithms based on classical methods for convex programming, such as proximal-point and augmented Lagrangian methods [26]. For details on non-interior-point based methods for solving large scale linear SDP problems, see [3,16,18,38,47].) Fortunately, for the problem (6), by exploiting the problem structure in the SDPs arising from the covariance selection model, we are able to design highly efficient preconditioners such that the condition numbers of the preconditioned matrices in each IPM iteration are bounded independent of the barrier parameter. We compare the performance of our IIPM method with the two recently developed methods: the ANS [21] and the PPA [38] methods. The latter methods are currently the most general and competitive methods for solving problems of the forms given in (4) and (5). Numerical experiments on test problems generated from synthetic and real data show that our IIPM method can outperform the ANS and PPA methods by a significant margin when solving problems (4) and (5).

We should emphasize that the focus of this paper is on the practical efficiency of the proposed IIPM method for solving (6) rather than the theoretical efficiency. Even though we have not worked out the details, it is likely that by adapting and combining the analysis in [48] and [35], polynomial iteration complexity results can be established for a theoretical version of the IIPM method proposed in this paper.

The remainder of the paper is organized as follows: Sect. 2 describes the formulation of the log-det SDP problem and the details of an inexact primal–dual interior-point algorithm (which we call Algorithm IIPM). In Sect. 3, we discusses the efficient computation of the search direction in each iteration of IIPM for the covariance selection problem (6). We also design efficient preconditioners for the linear systems of equations associated with the problem (6). Section 4 demonstrates the computational performance of IIPM on both synthetic and real examples. The performance of IIPM is compared with the ANS and PPA methods. Section 5 gives the conclusion.

In the paper, we use the following notation. For any two p-dimensional matrices G and K, the symmetrized Kronecker product between them is the linear operator on $\mathcal{S}^p$ defined by

$$(G \circledast K)(U) = \frac{1}{2}(GUK^T + KUG^T).$$

Some properties of the symmetrized Kronecker product can be found in the Appendix of [24]. We use $\|\cdot\|$ to denote the Frobenius norm for a matrix or Euclidean norm for a vector, and $\|\cdot\|_2$ to denote the spectral norm of a matrix.

2 An inexact primal–dual interior-point method

Here we design a customized inexact primal–dual path-following interior-point method for the following l_1-regularized log-det SDP problem which includes (6) as a special case:

$$\begin{aligned} (P)\ \min\ & C \bullet X - \gamma \log\det X + \boldsymbol{h}^T \boldsymbol{x_1} + \boldsymbol{h}^T \boldsymbol{x_2} \\ \text{s.t.}\ & \mathcal{A}(X) = b \\ & \mathcal{B}(X) - \boldsymbol{x_1} + \boldsymbol{x_2} = 0 \\ & \boldsymbol{x_1}, \boldsymbol{x_2} \geq 0,\ X \succ 0, \end{aligned} \tag{7}$$

where γ is a given positive constant, $C, X \in \mathcal{S}^p$, $\boldsymbol{h} \in \mathbb{R}^l_+$ and $b \in \mathbb{R}^m$ are given data; $\mathcal{A} : \mathcal{S}^p \to \mathbb{R}^m$ is a linear map defined by $\mathcal{A}(X) = [A_1 \bullet X, \ldots, A_m \bullet X]^T$ where $\{A_i \in \mathcal{S}^p : i = 1, \ldots, m\}$ are given matrices; $\mathcal{B} : \mathcal{S}^p \to \mathbb{R}^l$ is another linear map defined by $\mathcal{B}(X) = [B_1 \bullet X, \ldots, B_l \bullet X]^T$ where $\{B_i \in \mathcal{S}^p : i = 1, \ldots, l\}$ are given matrices. Without loss of generality, we assume that the linear map defining the linear equality constraints in (7) is surjective. Thus we must have $m + l \leq p(p+1)/2 + 2l$. Note that the problem (6) can be easily transformed to the form in (P) by writing $X_{ij} = X^+_{ij} - X^-_{ij}$ with $X^+_{ij}, X^-_{ij} \geq 0$ for $(i, j) \in \Omega^c$; see Sect. 4 for details. For the problem (P), if an optimal solution X^* exists, then it must be unique since the problem is equivalent to $\min\{C \bullet X - \gamma \log\det X + \boldsymbol{h}^T |\mathcal{B}(X)| : \mathcal{A}(X) = b, X \succ 0\}$, whose objective function is strictly convex with respect to X.

The dual problem of (P) is given by:

$$\begin{aligned} (D)\ \max\ & b^T y + \gamma \log\det Z + p\gamma(1 - \log\gamma) \\ \text{s.t.}\ \ & \mathcal{A}^T(y) + \mathcal{B}^T(\boldsymbol{u}) + Z = C \\ & -\boldsymbol{h} \le \boldsymbol{u} \le \boldsymbol{h},\ Z \succ 0,\ y \in \mathbb{R}^m \end{aligned} \tag{8}$$

where $\mathcal{A}^T$ and $\mathcal{B}^T$ are the adjoint of $\mathcal{A}$ and $\mathcal{B}$ respectively. For the problem (D), if an optimal solution Z^* exists, then it must be unique since the problem is equivalent to the problem: $\max\{b^T y + \gamma \log\det(Z(y, \boldsymbol{u}) := C - [\mathcal{A}^T, \mathcal{B}^T][y; \boldsymbol{u}]) : -\boldsymbol{h} \le \boldsymbol{u} \le \boldsymbol{h},\ y \in \mathbb{R}^m,\ Z(y, \boldsymbol{u}) \succ 0\}$, whose objective function is strictly concave with respect to $[y; \boldsymbol{u}]$.

The perturbed KKT optimality conditions for (P) and (D) are as follows:

$$\begin{aligned} \mathcal{A}^T(y) + \mathcal{B}^T(\boldsymbol{u}) + Z - C &= 0 \\ \mathcal{A}(X) - b &= 0, \\ \mathcal{B}(X) - \boldsymbol{x_1} + \boldsymbol{x_2} &= 0 \\ XZ - \gamma I &= 0, \quad X, Z \succ 0, \\ \boldsymbol{x_1} \circ (\boldsymbol{h} + \boldsymbol{u}) &= \nu \boldsymbol{e}, \quad \boldsymbol{x_1} > 0,\ \boldsymbol{h} + \boldsymbol{u} > 0, \\ \boldsymbol{x_2} \circ (\boldsymbol{h} - \boldsymbol{u}) &= \nu \boldsymbol{e}, \quad \boldsymbol{x_2} > 0,\ \boldsymbol{h} - \boldsymbol{u} > 0, \end{aligned} \tag{9}$$

where $\boldsymbol{e}$ is the vector of ones. Here the notation "$\circ$" denotes element-wise multiplication between two vectors. The last two equations of (9) are the perturbed complementarity conditions, where the positive barrier parameter ν is to be driven to 0 explicitly.

Due to the fact that XZ is usually nonsymmetric, the equation $XZ = I$ in (9) is usually symmetrized to $H_P(XZ) = \gamma I$, where for a given positive definite matrix P, $H_P : \mathbb{R}^{p \times p} \to \mathcal{S}^n$ is defined by $H_P(M) := (PMP^{-1} + (PMP^{-1})^T)/2$. In this paper, we choose $P = W^{-1/2}$ where $W \succ 0$ is the Nesterov–Todd (NT) scaling matrix satisfying $WZW = X$ for given $X, Z \in \mathcal{S}^p_{++}$ [24]. It has been shown in [46] that for $X, Z \in \mathcal{S}^p_{++}$ and our choice of P, $H_P(XZ) = \gamma I$ if and only if $XZ = \gamma I$.

Given the current iterate $(X, \boldsymbol{x_1}, \boldsymbol{x_2}, y, \boldsymbol{u}, Z)$, our IPM algorithm computes a search direction for the current iterate by applying one step of Newton method to (9) with the fourth equation $XZ = \gamma I$ replaced by $H_P(XZ) = \gamma I$. Without going through the algebraic manipulations, the search direction is the solution to the following linear system of equations:

$$\begin{aligned} \mathcal{A}^T(\Delta y) + \mathcal{B}^T(\Delta \boldsymbol{u}) + \Delta Z &= R^d := C - Z - \mathcal{A}^T(y) - \mathcal{B}^T(\boldsymbol{u}), \\ \mathcal{A}(\Delta X) &= r^p\ \ = b - \mathcal{A}(X), \\ \mathcal{B}(\Delta X) - \Delta \boldsymbol{x_1} + \Delta \boldsymbol{x_2} &= \boldsymbol{r^p} := \boldsymbol{x_1} - \boldsymbol{x_2} - \mathcal{B}(X), \\ W^{-1} \circledast W^{-1}(\Delta X) + \Delta Z &= R^c := \gamma X^{-1} - Z, \\ \boldsymbol{x_1}^{-1} \circ (\boldsymbol{h} + \boldsymbol{u}) \circ \Delta \boldsymbol{x_1} + \Delta \boldsymbol{u} &= \boldsymbol{r}_1 := \nu \boldsymbol{x_1}^{-1} - \boldsymbol{h} - \boldsymbol{u}, \\ \boldsymbol{x_2}^{-1} \circ (\boldsymbol{h} - \boldsymbol{u}) \circ \Delta \boldsymbol{x_2} - \Delta \boldsymbol{u} &= \boldsymbol{r}_2 := \nu \boldsymbol{x_2}^{-1} - \boldsymbol{h} + \boldsymbol{u}. \end{aligned} \tag{10}$$

Here, for a given vector $\boldsymbol{x} > 0$, we let $\boldsymbol{x}^{-1}$ be the componentwise reciprocal of $\boldsymbol{x}$. It is clear that the linear system (10) has dimension $m + p(p + 1) + 3l$, which could easily be very large. In practice, one would not solve (10) directly, but would first perform block eliminations so that only a smaller linear system is solved.

There are two possible ways to obtain a smaller linear system for computing the search direction from (10). The first approach is based on the fact that $\Delta \boldsymbol{x_1}$, $\Delta \boldsymbol{x_2}$ and $\Delta \boldsymbol{u}$, the variables associated with the linear constraints, can easily be eliminated. From the last two equations of (10), we have

$$\begin{aligned} \Delta \boldsymbol{x_1} &= (\boldsymbol{h}+\boldsymbol{u})^{-1} \circ \boldsymbol{x_1} \circ (\boldsymbol{r}_1 - \Delta \boldsymbol{u}), \\ \Delta \boldsymbol{x_2} &= (\boldsymbol{h}-\boldsymbol{u})^{-1} \circ \boldsymbol{x_2} \circ (\boldsymbol{r}_2 + \Delta \boldsymbol{u}). \end{aligned} \tag{11}$$

Then from (11) and the third equation of (10), we get

$$\Delta \boldsymbol{u} = \boldsymbol{q}^{-1} \circ \boldsymbol{g} - \operatorname{diag}(\boldsymbol{q}^{-1})\mathcal{B}(\Delta X) \tag{12}$$

where

$$\boldsymbol{q} = (\boldsymbol{h}+\boldsymbol{u})^{-1} \circ \boldsymbol{x_1} + (\boldsymbol{h}-\boldsymbol{u})^{-1} \circ \boldsymbol{x_2}, \tag{13}$$

$$\boldsymbol{g} = -2\nu\, \boldsymbol{u} \circ (\boldsymbol{h} \circ \boldsymbol{h} - \boldsymbol{u} \circ \boldsymbol{u})^{-1} - \mathcal{B}(X). \tag{14}$$

By using (12) and eliminating ΔZ from the first and fourth equations of (10), we get the following augmented system:

$$\begin{aligned} &\begin{bmatrix} -W^{-1} \circledast W^{-1} - \mathcal{B}^T \operatorname{diag}(\boldsymbol{q}^{-1})\mathcal{B} & \mathcal{A}^T \\ \mathcal{A} & 0 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta y \end{bmatrix} \\ &= \begin{bmatrix} R^d - R^c - \mathcal{B}^T(\boldsymbol{q}^{-1} \circ \boldsymbol{g}) \\ r^p \end{bmatrix}. \end{aligned} \tag{15}$$

To compute the search direction associated with the current iterate, one can solve the linear system (15) for ΔX, Δy. Once they are obtained, $\Delta \boldsymbol{u}$, ΔZ can be obtained from (12) and the first equation of (10), respectively. After that, $\Delta \boldsymbol{x_1}$, $\Delta \boldsymbol{x_2}$ can be computed from (11).

The linear system (15) is generally dense even if $\mathcal{A}$ is sparse, and its dimension, $m + p(p+1)/2$, can be very large even for a moderate p (say $p = 500$). Thus it is impractical to solve (15) via a direct solver since it would require huge computer memory space to store the coefficient matrix as well as excessive computing time to factorize it. The only viable alternative is to use an iterative solver to compute an approximate solution with a sufficiently small residual norm. Note that if (15) is solved inexactly such that the computed solution $(\Delta X, \Delta y)$ has residual given by (Ξ, ξ), then the residual of the computed search direction for (10) is given by $(0, \xi, 0, -\Xi, 0, 0)$. In our numerical implementation, we deem $(\Delta X, \Delta y)$ computed from (15) to be sufficiently accurate if the following relative stopping criterion is satisfied:

$$\max\{\|\Xi\|, \|\xi\|\} \le \kappa \max\left\{\|R^d\|, \|r^p\|, \|\boldsymbol{r^p}\|, \|R^c\|, \|\boldsymbol{r}_1\|, \|\boldsymbol{r}_2\|\right\},$$

where $\kappa \in (0, 1)$ is an accuracy parameter. Although we do not investigate the global polynomial convergence of our inexact IPM under such a stopping criterion, we note that a similar criterion has been used in the analysis in [48].

Observe that for the system (15), if let $H = W^{-1} \circledast W^{-1} + \mathcal{B}^T \operatorname{diag}(\boldsymbol{q}^{-1})\mathcal{B}$ and $\mathcal{G} = [-H,\ \mathcal{A}^T;\ \mathcal{A},\ \mathbf{0}]$ be its coefficient matrix, then

$$\mathcal{G}^{-1} = \begin{bmatrix} -H^{-1} + H^{-1}\mathcal{A}^T Y^{-1}\mathcal{A}H^{-1} & H^{-1}\mathcal{A}^T Y^{-1} \\ Y^{-1}\mathcal{A}H^{-1} & Y^{-1} \end{bmatrix}, \tag{16}$$

where $Y = \mathcal{A}H^{-1}\mathcal{A}^T$. Thus for any given $[x;\ y]$, $\mathcal{G}^{-1}[x;\ y]$ can be computed via the following steps:

$$\text{Compute } v = Y^{-1}(\mathcal{A}H^{-1}x + y); \tag{17}$$

$$\text{Compute } \mathcal{G}^{-1}[x;\ y] = [H^{-1}(\mathcal{A}^T v - x);\ v]. \tag{18}$$

However, note that it is impractical to compute either (17) or (18) exactly since it is extremely costly to compute H^{-1} and Y^{-1}.

In the second approach for computing the search direction in (10), we rewrite the system (10) in the following block form

$$\begin{bmatrix} -U & V^T \\ V & 0 \end{bmatrix} \begin{bmatrix} \Delta\widetilde{X} \\ \Delta\widetilde{y} \end{bmatrix} = \begin{bmatrix} \widetilde{R} \\ \widetilde{r} \end{bmatrix}, \tag{19}$$

where

$$U = \operatorname{diag}\Big(W^{-1} \circledast W^{-1},\ (\boldsymbol{h}+\boldsymbol{u}) \circ \boldsymbol{x_1}^{-1},\ (\boldsymbol{h}-\boldsymbol{u}) \circ \boldsymbol{x_2}^{-1}\Big),\ \ V = (\mathcal{A}, 0, 0;\ \mathcal{B}, -\mathcal{I}, \mathcal{I})\,,$$
$$\Delta\widetilde{X} = (\Delta X,\ \Delta\boldsymbol{x_1},\ \Delta\boldsymbol{x_2})\,,\quad \Delta\widetilde{y} = (\Delta y,\ \Delta\boldsymbol{u})\,,$$
$$\widetilde{R} = \Big(R^d - R^c;\ -\boldsymbol{r}_1;\ -\boldsymbol{r}_2\Big)\,,\quad \widetilde{r} = \big(r^p;\ \boldsymbol{r^p}\big)\,.$$

Since U^{-1} exists and is easy to compute, by eliminating $\Delta\widetilde{X}$, the system (19) can be reduced to the following smaller system:

$$VU^{-1}V^T\Delta\widetilde{y} = VU^{-1}\widetilde{R} + \widetilde{r}. \tag{20}$$

By rewriting (20) with the original variables, Δy, $\Delta\boldsymbol{u}$ can be computed from the following linear system:

$$\begin{bmatrix} \mathcal{A}W \circledast W\mathcal{A}^T & \mathcal{A}W \circledast W\mathcal{B}^T \\ \mathcal{B}W \circledast W\mathcal{A}^T & \mathcal{B}W \circledast W\mathcal{B}^T + \operatorname{diag}(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta\boldsymbol{u} \end{bmatrix}$$
$$= \begin{bmatrix} \mathcal{A}W \circledast W(R^d - R^c) + r^p \\ \mathcal{B}W \circledast W(R^d - R^c) + \boldsymbol{g} \end{bmatrix} \tag{21}$$

where $\boldsymbol{g}$ is defined as in (14). Once Δy, $\Delta\boldsymbol{u}$ are computed, ΔZ can be obtained from the first equation of (10), while $\Delta\boldsymbol{x_1}$, $\Delta\boldsymbol{x_2}$ can be computed from (11). The unknown ΔX is easy to obtain since from the fourth equation of (10), we have

$$\Delta X = W \circledast W(R^c - \Delta Z). \tag{22}$$

Just like the linear system (15), the linear system (21) is generally dense and its dimension, $m + l$, can be very large. Thus it is generally impractical to solve (21) via a direct solver, and one must resort to an iterative solver to compute an approximate solution with a sufficiently small residual norm. We note that if the solution $(\Delta y, \Delta \boldsymbol{u})$ is computed inexactly from (21) with residual given by $[\eta; \zeta]$, then the residual of the computed search direction for (10) is given by $[0; \eta; \zeta; 0; 0; 0]$.

We should emphasize that when we apply an iterative solver to (15) or (21), it is crucial for us to construct efficient preconditioners for the systems since they are generally ill-conditioned, as we will explain next. Assume that strict complementarity conditions hold for the last two equations in (9) at an optimal solution $(X^*, \boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \boldsymbol{u}^*, y^*, Z^*)$, i.e., there exists a positive constant κ independent of ν such that for $\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{u}$ that are sufficiently close to the optimal solution, we have $\boldsymbol{x_1} + \boldsymbol{h} + \boldsymbol{u} \geq \kappa$, $\boldsymbol{x_2} + \boldsymbol{h} - \boldsymbol{u} \geq \kappa$. Let $L := \{1, \ldots, l\}$. We define

$$L_1 = \{k \in L \, : \, (\mathbf{x}_1)_k = \Theta(\nu) \text{ and } (\mathbf{x}_2)_k = \Theta(\nu)\}, \tag{23}$$

and $L_2 = L \backslash L_1 = \{k \in L \, : \, (\boldsymbol{h} + \boldsymbol{u})_k = \Theta(\nu) \text{ or } (\boldsymbol{h} - \boldsymbol{u})_k = \Theta(\nu)\}$. It is clear that for the vector $\boldsymbol{q}$ in (13), its components would have the following order of magnitudes:

$$\boldsymbol{q}_{(k)} = \begin{cases} \Theta(\nu) & \text{for } k \in L_1 \\ \Theta(1/\nu) & \text{for } k \in L_2. \end{cases}$$

Thus when $L_1 \neq \emptyset$, the coefficient matrix in (15) would have its norm increase like $O(1/\nu)$ as $\nu \downarrow 0$ since its (1,1) block involves $\boldsymbol{q}^{-1}$. Thus the condition number of the coefficient matrix in (15) is at least of the order $\Theta(1/\nu)$ when the iterate is close to optimality. Similarly, when $L_2 \neq \emptyset$, the condition number of the coefficient matrix in (21) is at least of the order $\Theta(1/\nu)$ when the iterate is close to optimality.

As the focus of this paper is on the special problem (6), the design of efficient preconditioners for (15) and (21) arising from the general problem (7) is an interesting topic that we will not pursue here but leave it for future research.

Now we describe the details of our inexact primal–dual path-following algorithm.

Algorithm (IIPM) *Choose starting points* $X^0 = Z^0 \in \mathcal{S}^p_{++}$, $\boldsymbol{x_1}^0 = \boldsymbol{x_2}^0 > 0$, $y^0 = \mathbf{0}$, $\boldsymbol{u}^0 = 0$. *Let* $\sigma, \tau \in (0, 1)$ *be given parameters.*
For $k = 0, 1, 2, \ldots$
Let the current and next iterate be $(X, \boldsymbol{x_1}, \boldsymbol{x_2}, y, \boldsymbol{u}, Z)$ *and* $(X^+, \boldsymbol{x_1}^+, \boldsymbol{x_2}^+, y^+, \boldsymbol{u}^+, Z^+)$*, respectively.*

1. *(Convergence test) Terminate the iteration if*

$$\phi := \max\left\{\frac{gap}{1 + |\,pobj| + |\,dobj|},\ pinf,\ dinf\right\} \leq \texttt{Tol}, \tag{24}$$

where pobj, dobj are the primal and dual objective values, and

$$gap = (\boldsymbol{h} + \boldsymbol{u})^T \boldsymbol{x_1} + (\boldsymbol{h} - \boldsymbol{u})^T \boldsymbol{x_2} + X \bullet Z - \gamma \log\det(XZ) - p\gamma(1 - \log\gamma)$$

$$pinf = \max\left\{\frac{\|r^p\|}{1 + \|b\|}, \frac{\|\boldsymbol{r^p}\|}{1 + \|X\|}\right\}, \quad dinf = \frac{\|R^d\|}{1 + \|C\|}.$$

2. *Compute the search direction* $(\Delta X, \Delta \boldsymbol{x_1}, \Delta \boldsymbol{x_2}, \Delta \boldsymbol{u}, \Delta y, \Delta Z)$ *from* (10) *by solving the augmented system* (15) *for* $(\Delta X, \Delta y)$ *or* (21) *for* $(\Delta y, \Delta \boldsymbol{u})$ *with* $\nu = \sigma\mu$*, where*

$$\mu \;=\; \frac{(\boldsymbol{h}+\boldsymbol{u})^T\boldsymbol{x_1} + (\boldsymbol{h}-\boldsymbol{u})^T\boldsymbol{x_2}}{2l}.$$

3. *Determine the maximum step lengths* $\bar{\alpha}, \bar{\beta} \in (0, \infty)$ *such that* $X+\bar{\alpha}\Delta X$*,* $Z+\bar{\beta}\Delta Z$ *remain positive semidefinite;* $\boldsymbol{x_1}+\bar{\alpha}\Delta\boldsymbol{x_1}$*,* $\boldsymbol{x_2}+\bar{\alpha}\Delta\boldsymbol{x_2}$*,* $\boldsymbol{h}+\boldsymbol{u}+\bar{\beta}\Delta\boldsymbol{u}$*,* $\boldsymbol{h}-\boldsymbol{u}-\bar{\beta}\Delta\boldsymbol{u}$ *remain nonnegative.*
4. (*Update*) *Compute the next iterate as:*

$$\begin{aligned} &X^+ = X+\alpha\Delta X, \quad \boldsymbol{x_1}^+ = \boldsymbol{x_1}+\alpha\Delta\boldsymbol{x_1}, \quad \boldsymbol{x_2}^+ = \boldsymbol{x_2}+\alpha\Delta\boldsymbol{x_2},\\ &Z^+ = Z+\beta\Delta Z, \quad y^+ = y+\beta\Delta y, \quad \boldsymbol{u}^+ = \boldsymbol{u}+\beta\Delta\boldsymbol{u}, \end{aligned}$$

where $\alpha = \min\{1, \tau\bar{\alpha}\}$ *and* $\beta = \min\{1, \tau\bar{\beta}\}$.

3 Computation of search direction for the special case (6)

For the special case (6), the computation of the search direction via the systems (15) and (21) can further be simplified. More importantly, we can also design efficient preconditioners for the simplified systems.

Recall that for the problem (6), we have $m+l = p(p+1)/2$, and $[\mathcal{A}^T, \mathcal{B}^T]$ is just a permutation of the identity operator. As a result, we have the following properties:

$$\mathcal{A}^T\mathcal{A}+\mathcal{B}^T\mathcal{B} = I_{\bar{p}}, \;\; \mathcal{A}\mathcal{A}^T = I_m, \;\; \mathcal{B}\mathcal{B}^T = I_l, \;\; \mathcal{A}\mathcal{B}^T = \mathbf{0}_{m\times l}, \tag{25}$$

where $\bar{p} := p(p+1)$.

3.1 Computing $(\Delta X, \Delta y)$ first

First we consider the linear system (15) corresponding to the problem (6). By using (25), we have $\Delta X = \mathcal{A}^T(\mathcal{A}\Delta X)+\mathcal{B}^T(\mathcal{B}\Delta X) = \mathcal{A}^T r^p + \mathcal{B}^T(\mathcal{B}\Delta X)$, thus the system (15) can be rewritten as follows:

$$\left(\mathcal{B}W^{-1} \circledast W^{-1}\mathcal{B}^T + \operatorname{diag}(\boldsymbol{q}^{-1})\right)\Delta\xi = f, \tag{26}$$

where $\Delta\xi = \mathcal{B}\Delta X$ and $f = \boldsymbol{q}^{-1}\circ\boldsymbol{g} - \mathcal{B}(R^d - R^c + W^{-1}(\mathcal{A}^T r^p)W^{-1})$. Once $\Delta\xi$ is computed, $(\Delta X, \Delta y)$ can be recovered from the following equation:

$$\Delta X = \mathcal{A}^T r^p + \mathcal{B}^T\Delta\xi, \quad \Delta y = \mathcal{A}(W^{-1}\Delta X W^{-1} + R^d - R^c). \tag{27}$$

Suppose that the computed solution $\Delta\xi$ from (26) has residual δ. Then for the direction $(\Delta X, \Delta y)$ computed based on (26) and (27), the residual vector associated with the system (15) is given by $[-\mathcal{B}^T\delta; 0]$.

It is easy to see that when X, Z are sufficiently close to the optimal solutions X^*, Z^*, there exists a positive constant τ (independent of the barrier parameter ν) such that

$$W^{-1} \circledast W^{-1} \succeq \tau \Lambda^{-1} \circledast \Lambda^{-1},$$

where Λ is a given positive definite diagonal matrix, for example, $\Lambda = I$. In our numerical implementation of the IIPM algorithm, we take $\Lambda^{-1} = \operatorname{diag}(W^{-1})$.

We can rewrite (26) as

$$\left(\mathcal{B}(W^{-1} \circledast W^{-1} - \tau \Lambda^{-1} \circledast \Lambda^{-1})\mathcal{B}^T + \mathcal{M}_3\right) \Delta\xi \;=\; f, \qquad (28)$$

where

$$\mathcal{M}_3 = \tau \mathcal{B} \Lambda^{-1} \circledast \Lambda^{-1} \mathcal{B}^T + \operatorname{diag}(\boldsymbol{q}^{-1}). \qquad (29)$$

If we precondition (28) by $\mathcal{M}_3$, then we get

$$\left(I + \mathcal{M}_3^{-1/2} \mathcal{B}(W^{-1} \circledast W^{-1} - \tau \Lambda^{-1} \circledast \Lambda^{-1})\mathcal{B}^T \mathcal{M}_3^{-1/2}\right)\left(\mathcal{M}_3^{1/2} \Delta\xi\right) = \mathcal{M}_3^{-1/2} f. \qquad (30)$$

For the above preconditioned system (30), which has a symmetric positive definite coefficient matrix, the iterative solver of choice is the minimum residual (MINRES) method [27, p. 194]. We can expect the MINRES method (also known as the conjugate residual method) to be efficient in solving (30), as the result in Theorem 1 indicates.

Theorem 1 *Let* $\beta = \|\mathcal{M}_3^{-1/2} \mathcal{B}(W^{-1} \circledast W^{-1} - \tau \Lambda^{-1} \circledast \Lambda^{-1})\mathcal{B}^T \mathcal{M}_3^{-1/2}\|_2$. *The MINRES method applied to* (30) *would converge at a rate given by*

$$\frac{\sqrt{1+\beta} - 1}{\sqrt{1+\beta} + 1}.$$

Proof Let $\mathcal{H}$ be the preconditioned matrix in (30). It is clear that the eigenvalues of $\mathcal{H}$ are contained in the interval $[1, 1 + \beta]$. Thus the condition number of $\mathcal{H}$ is no more than $1 + \beta$, and the required convergence rate follows by adapting the standard convergence result for the conjugate gradient method [27, p. 203] to the MINRES method. □

Note that the quantity β in Theorem 1 can be bounded independent of the barrier parameter ν.

In our implementation of Algorithm IIPM, we use the symmetric quasi-minimal residual (SQMR) iterative method [12] to solve the linear system (30) instead of the MINRES method. We note that the SQMR method is mathematically equivalent to the MINRES method when the coefficient matrix is symmetric positive definite. But we prefer the former in our implementation as it has slightly better numerical performance in finite-precision arithmetic.

3.2 Computing $(\Delta y,\ \Delta \boldsymbol{u})$ first

Next, we consider the linear system (21) corresponding to the problem (6). In this case, by using (25), the linear system (21) can be rewritten as follows:

$$\left(W \circledast W + \mathcal{B}^T \operatorname{diag}(\boldsymbol{q})\mathcal{B}\right) \Delta V \;=\; F \tag{31}$$

where $\Delta V = \mathcal{A}^T(\Delta y) + \mathcal{B}^T(\Delta \boldsymbol{u})$ and $F = \mathcal{A}^T r^p + W(R^d - R^c)W + \mathcal{B}^T \boldsymbol{g}$. After solving for ΔV, the search direction can be found as follows:

$$\Delta y = \mathcal{A}(\Delta V), \quad \Delta \boldsymbol{u} = \mathcal{B}(\Delta V), \quad \Delta Z = R^d - \Delta V,$$

and $\Delta \boldsymbol{x_1}$, $\Delta \boldsymbol{x_2}$ can be computed from (11). The unknown ΔX can be computed either from (22) or from the following equation: $\Delta X = \mathcal{A}^T r^p + \mathcal{B}^T(\boldsymbol{g} - \boldsymbol{q} \circ \Delta \boldsymbol{u})$. We found that the former has better numerical stability and we adopt it in the implementation of Algorithm IIPM.

Suppose ΔV is computed from (31) with residual Φ. Then the residual corresponding to the system (21) for the computed $(\Delta y, \Delta \boldsymbol{u})$ above is given by $[\mathcal{A}\Phi; \mathcal{B}\Phi]$. Note that if ΔX is computed from (22), then the residual vector associated with the computed search direction for (10) is given by $(0, \mathcal{A}\Phi, \mathcal{B}\Phi, 0, 0, 0)$.

It is easy to see that when X, Z are sufficiently close to the optimal solutions X^*, Z^*, there exists a positive constant τ (independent of the barrier parameter ν) such that

$$W \circledast W \succeq \tau \Lambda \circledast \Lambda,$$

where Λ is a given positive definite diagonal matrix. In our implementation of the IIPM algorithm, we take $\Lambda = \operatorname{diag}(W)$.

We can rewrite (31) as

$$(W \circledast W - \tau \Lambda \circledast \Lambda + \mathcal{M}_4)\, \Delta V \;=\; F, \tag{32}$$

where

$$\mathcal{M}_4 = \tau \Lambda \circledast \Lambda + \mathcal{B}^T \operatorname{diag}(\boldsymbol{q})\mathcal{B}. \tag{33}$$

If we precondition (32) by $\mathcal{M}_4$, then we get

$$\left(I + \mathcal{M}_4^{-1/2}\,(W \circledast W - \tau \Lambda \circledast \Lambda)\,\mathcal{M}_4^{-1/2}\right)\left(\mathcal{M}_4^{1/2} \Delta V\right) = \mathcal{M}_4^{-1/2} F. \tag{34}$$

For the above preconditioned system (34), which has a symmetric positive definite coefficient matrix, again the iterative solver of choice is the MINRES method. As in (30), we can expect the MINRES method to be efficient in solving (34), as the result in the next theorem indicates.

Theorem 2 *Let* $\beta = \|\mathcal{M}_4^{-1/2}(W \circledast W - \tau\Lambda \circledast \Lambda)\mathcal{M}_4^{-1/2}\|_2$. *The MINRES method applied to* (34) *would converge at a rate given by*

$$\frac{\sqrt{1+\beta}-1}{\sqrt{1+\beta}+1}.$$

Proof Similar to that of Theorem 1. □

Note that as before, the quantity β in Theorem 2 can be bounded independent of the barrier parameter ν. Again, for the same reason mentioned in the last subsection, we use the SQMR method to solve (34) instead of the MINRES method in our implementation of Algorithm IIPM.

Remark Given the systems (26) and (31), we have the flexibility to choose a better conditioned system among the two to compute the search direction. By noting that $W \approx X/\sqrt{\gamma}$ when (X, Z) is close to optimality, the system (31) is preferred if $\|X\|$ is moderate. On the other hand, the system (26) is preferred if $\|X^{-1}\|$ is moderate. In our implementation of Algorithm IIPM for solving (5) and (4), we replace Step 2 in the algorithm as follows:

2′. If $\|X^{-1}\| < 10^{-2}\|X\|$, compute the search direction $(\Delta X, \Delta \boldsymbol{x_1}, \Delta \boldsymbol{x_2}, \Delta \boldsymbol{u}, \Delta y, \Delta Z)$ for (10) via solving the system (30) for $(\Delta X, \Delta y)$; otherwise, compute the search direction via solving the system (34) for $(\Delta y, \Delta \boldsymbol{u})$.

4 Numerical experiment

In this section, we conduct numerical experiments to evaluate the performance of **Algorithm IIPM** for solving the problem (6) arising from covariance selection. Specifically, we solve the following problem:

$$\begin{aligned} \min\ & \widehat{\Sigma} \bullet X - \log\det X + \boldsymbol{h}^T(\mathbf{x}^+ + \mathbf{x}^-) \\ \text{s.t.}\ & X_{ij} = 0 \quad \forall (i,j) \in \Omega, \\ & \mathcal{B}(X) - \mathbf{x}^+ + \mathbf{x}^- = 0, \\ & \mathbf{x}^+, \mathbf{x}^- \geq 0,\ X \succ 0, \end{aligned} \tag{35}$$

where $\widehat{\Sigma}$ is a given $p \times p$ sample covariance matrix, $\boldsymbol{h}$ corresponds to $H = \rho E$, and E is the matrix of ones. The linear map $\mathcal{B} : \mathcal{S}^p \to \mathbb{R}^l$ is defined by $\mathcal{B}(X) = X_{\Omega^c}$, where X_{Ω^c} is the vector in $\mathbb{R}^l$ that is obtained by stacking the elements X_{ij} with $(i,j) \in \Omega^c$ in lexicographical order into a column vector.

The input sample covariance matrices $\widehat{\Sigma}$ are chosen from both synthetic data and real data. For synthetic data, the sparsity pattern of the true inverse covariance matrix Σ^{-1} is assumed to be known. In this case, we create linear constraints $X_{ij} = 0$ by letting Ω to be a subset of Ξ, where Ξ is the set of indices of the zero elements of Σ^{-1}. In our experiments, we randomly choose 50% of the elements in Ξ to form the subset Ω and expect to recover the rest by solving (35). For the real data considered in this section, we have no priori knowledge on the sparsity pattern. Hence, we set $\Omega = \emptyset$ in the problem (35).

In Algorithm IIPM, we use the following starting iterate:

$$\begin{gathered} Z^0 = \widehat{\Sigma} + I, \quad y^0 = 0, \quad \boldsymbol{u}^0 = 0, \\ X^0 = (Z^0)^{-1}, \quad \mathbf{x}_+^0 = \gamma \boldsymbol{e}, \quad \mathbf{x}_-^0 = \gamma \boldsymbol{e}, \end{gathered}$$

where $\gamma = 0.1\boldsymbol{e} + \|X^0\|/p$, with p being the dimension of $\widehat{\Sigma}$, and $\boldsymbol{e}$ is the vector of ones.

All the numerical experiments are carried out in MATLAB 7.6 on a 3.0 GHz Intel Xeon PC with 4.0GB RAM running Linux 9.10. We compare the performance of our inexact interior point algorithm (Algorithm IIPM) with the Adaptive Nesterov Smoothing (ANS) method proposed by Lu [20,21] and the Newton-CG primal proximal-point (PPA) method proposed in [38]. For the IIPM method, we use the stopping condition (24) with `Tol` $= 10^{-6}$. For the ANS method, its stopping conditions depend on two tolerance parameters, ε_o and ε_c, which control the duality gap and constraint violation, respectively. When solving examples with linear constraints, $X_{ij} = 0$, $(i, j) \in \Omega$ in (5), we use ANS method with its default updating parameter $r_\rho = 2$. For the PPA method, the stopping condition used is similar to that in (24), and the tolerance is set to be `Tol` $= 10^{-6}$.

4.1 Synthetic examples

Example 1 We adopt the idea from d'Aspremont [7] to construct a random sparse inverse covariance matrix. In particular, let U be a $p \times p$ sparse matrix with a few randomly chosen nonzero entries that are equal to ± 1, then we generate a sparse inverse covariance matrix as follows:

$$\begin{aligned} &A = U^T U; \quad d = \operatorname{diag}(A); \quad A = \max(\min(A - \operatorname{diag}(d), 1), -1); \\ &A = A + \operatorname{diag}(d + 1); \quad \Sigma^{-1} = A + \max(-1.2\lambda_{\min}(A), \varepsilon) I \end{aligned} \tag{36}$$

where $\varepsilon = 10^{-4}$ is a small perturbation to ensure that Σ^{-1} is positive definite.

The above choice has been frequently considered when constructing a synthetic testing example for covariance selection problems, see for example [20,21,38]. It is worth pointing out that (36) is a slight modification of d'Aspremont's original example. The reason for doing so is to generate a true covariance matrix Σ such that the problem (6) can recover Σ^{-1} reasonably well.

Using the true sparse inverse covariance matrix Σ^{-1} generated in (36), we first generate n i.i.d. random vectors from the p-dimensional Gaussian distribution $\mathcal{N}(0, \Sigma)$. Then we calculate the sample covariance matrix $\widehat{\Sigma}$. Note that in [7,21], the sample covariance matrix is obtained by adding an i.i.d. uniform random noise term to Σ. Here we prefer the simulation approach to the noise term approach since it is more commonly employed in statistics [41,42].

Table 1 presents the results obtained by Algorithm IIPM and the ANS and PPA methods for various instances of Example 1 on the problems (5) and (4). Note that a number of the form "9.01-4" under the column "primal objective value" means the

Table 1 Comparison of the IIPM and ANS methods in solving the problems (4) and (5) with the data matrix $\widehat{\Sigma}$ generated from Example 1

<table>
<tr><th rowspan="2">Problem</th><th rowspan="2">$p \mid m$</th><th colspan="2">Iteration count</th><th colspan="2">Primal objective value</th><th colspan="2">Time (s)</th></tr>
<tr><th>IIPM</th><th>ANS|PPA</th><th>IIPM</th><th>ANS|PPA</th><th>IIPM</th><th>ANS|PPA</th></tr>
<tr><td>Random</td><td>500 | 0</td><td>14 (11.2| 2.6-2| 1.6-1| 0.90| 0.87)</td><td>239 |68</td><td>−1.751007242</td><td>9.01-4|−9.34-5</td><td>19.2</td><td>51.5|79.5</td></tr>
<tr><td>Random</td><td>1000 | 0</td><td>15 (12.2| 2.2-2| 1.8-1| 0.88| 0.90)</td><td>310 |82</td><td>−6.488578832</td><td>7.64-4|−2.16-4</td><td>122.1</td><td>365.9|547.6</td></tr>
<tr><td>Random</td><td>1500 | 0</td><td>15 (11.9| 2.3-2| 2.3-1| 0.84| 0.83)</td><td>295 |73</td><td>−1.441082843</td><td>5.72-4|−3.71-4</td><td>359.7</td><td>1089.6|1410.4</td></tr>
<tr><td>Random</td><td>2000 | 0</td><td>15 (10.3| 2.3-2| 2.7-1| 0.82| 0.75)</td><td>307 |76</td><td>−2.413956933</td><td>3.53-4|−5.64-4</td><td>735.6</td><td>2602.5|3188.6</td></tr>
<tr><td>Random</td><td>500 | 56774</td><td>13 (14.2| 2.4-2| 1.5-1| 0.94| 0.89)</td><td>3087 |69</td><td>−1.682948952</td><td>9.28-4|−6.75-5</td><td>19.3</td><td>654.4|80.5</td></tr>
<tr><td>Random</td><td>1000 | 221990</td><td>16 (18.1| 2.1-2| 1.7-1| 0.92| 0.93)</td><td>5462 |82</td><td>−6.312553392</td><td>9.78-4|−1.02-5</td><td>157.4</td><td>6325.8|628.5</td></tr>
<tr><td>Random</td><td>1500 | 491764</td><td>16 (17.9| 2.1-2| 2.2-1| 0.90| 0.87)</td><td>5714 |80</td><td>−1.404170163</td><td>9.81-4| 5.44-6</td><td>473.6</td><td>19959.0|1762.6</td></tr>
<tr><td>Random</td><td>2000 | 862392</td><td>15 (16.5| 2.1-2| 2.5-1| 0.89| 0.81)</td><td>5958 |83</td><td>−2.351873733</td><td>9.07-4|−5.13-5</td><td>945.1</td><td>47690.0|4100.8</td></tr>
</table>

The regularization parameter ρ is set to $\rho = 5/p$ for all the problems. The numbers in each parenthesis are the average number of SQMR steps taken in each iteration, Loss_Q, Loss_E, Specificity and Sensitivity, respectively number “9.01 × 10^{-4}”. For all the problems, we set $\rho = 5/p$. We use the primal objective value and computing time to compare the performance of the two algorithms. Observe that the CPU time taken by the IIPM method to solve (5) is only slightly more than that taken to solve (4) for the same sample covariance matrix. The same observation also applies to the PPA method. But for the ANS method, the time it takes to solve (5) is about 20 times of that needed for solving (4). Thus the IIPM and PPA methods are equally efficient for solving (5) and (4), but the ANS method is typically much slower in solving (5) compared to (4). Overall, we see that the IIPM method outperforms the ANS and PPA methods by quite a big margin. The IIPM method is faster than the ANS method by a factor of 2.7–3.5 and 33.9–50.5 in solving the problems (4) and (5), respectively. Comparing the IIPM and PPA methods, the former is faster by a factor of 3.9–4.5 and 3.7–4.3 in solving (4) and (5), respectively. One may expect the IIPM method to outperform the ANS method by an even larger margin when the matrix dimension p increases. As one may observe from Table 1, the number of iterations and the average number of SQMR steps needed to solve each linear system do not increase visibly when p increases for the IIPM method. But for the ANS method, the number of iterations increases moderately when p increases.

Due to the difference in stopping criteria for different algorithms, we set different accuracy tolerances for the IIPM and ANS methods. For the ANS method, we set the tolerances to $\varepsilon_o = 10^{-3}$, and $\varepsilon_c = 10^{-5}$. For the IIPM method, we set $\texttt{Tol} = 10^{-6}$ in (24). They are chosen in such a way the both algorithms would obtain roughly the same primal objective values while the primal infeasibilities are below 10^{-6}. As we can see from Table 1, the columns of “primal objective value ANS (PPA)” show the differences between the primal objective values obtained by ANS (PPA) and those obtained by IIPM. A positive difference means IIPM achieved a better (smaller) primal objective value while a negative difference indicates a worse (larger) result by IIPM. As we can observe from the table, the differences are usually insignificant.

To evaluate how well we have recovered the true inverse covariance matrix Σ^{-1}, we compute the normalized entropy loss (Loss_E) and quadratic loss (Loss_Q)

$$\text{Loss}_E := \frac{1}{p}(\,\text{Tr}(\Sigma X) - \log\det \Sigma X - p), \quad \text{Loss}_Q := \frac{1}{p}\|\Sigma X - I\|. \tag{37}$$

In general, it is impossible to recover Σ^{-1} accurately based on $\widehat{\Sigma}$ by solving (6). Thus the purpose of solving (6) is not to recover the true matrix Σ^{-1} accurately but to detect the sparsity pattern while maintaining a reasonable approximation to the true matrix. To measure the quality of the sparsity pattern in X in relation to that of the true matrix, we borrow some criteria from the machine learning literature:

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

where TP, TN, FP, and FN denotes the number of true positives, true negatives, false positives, and false negatives, respectively. In our situation, specificity measures the quality of zero entries while sensitivity measures the quality of nonzero entries. As we may observe from the results in Table 1, by solving (4) or (5) with an appropriately chosen regularization parameter ρ, one can obtain a reasonably good estimation X

of the true inverse covariance matrix Σ^{-1} from the sample covariance matrix $\widehat{\Sigma}$. In particular we see that the Specificity and Sensitivity of the sparsity pattern of the estimated matrix X are both quite close to 1.

Next we consider a collection of problems considered in [11,44].

Example 2 Let A denotes a $p \times p$ sparse inverse covariance matrix. We consider the following problems.

AR(1) An autoregressive process of order one, defined as $A_{ii} = 1$, $A_{i,i-1} = A_{i-1,i} = 0.5$;
AR(2) $A_{ii} = 1,\ A_{i,i-1} = A_{i-1,i} = 0.5,\ A_{i,i-2} = A_{i-2,i} = 0.25$;
AR(3) $A_{ii} = 1,\ A_{i,i-1} = A_{i-1,i} = 0.4,\ A_{i,i-2} = A_{i-2,i} = A_{i,i-3} = A_{i-3,i} = 0.2$;
AR(4) $A_{ii} = 1,\ A_{i,i-1} = A_{i-1,i} = 0.4,\ A_{i,i-2} = A_{i-2,i} = A_{i,i-3} = A_{i-3,i} = 0.2,\ A_{i,i-4} = A_{i-4,i} = 0.1$;
Full $A_{ii} = 2,\ A_{ij} = 1,\ \forall i \neq j$;
Decay Exponential decay matrix $A_{ij} = \exp(-2|i-j|)$, far-end off-diagonal entries could be close to zero for A with large dimensions;
Star Every node connects to the first node $A_{ii} = 1,\ A_{i1} = A_{1i} = 1/p$;
Circle $A_{ii} = 1,\ A_{i,i-1} = A_{i-1,i} = 0.5,\ A_{1p} = A_{p1} = 0.4$.

For each of the problem, we generate $2p$ i.i.d. random samples from the p-dimensional Gaussian distribution $\mathcal{N}(0, A^{-1})$ and use the sample covariance matrix $\widehat{\Sigma}$ as the input of the covariance selection problems (4) and (5). The numerical results for the problems (4) and (5) are presented in Tables 2 and 3, respectively. As in Table 1, we see that the IIPM method can solve both the problems (4) and (5) equally efficiently but the ANS method is much slower in solving (5) compared to (4). The PPA method is slightly slower in solving (5) compared to (4) for the same data matrix. Overall, the IIPM method outperforms both the ANS and PPA methods by a rather big margin. It is faster than the ANS method by a factor of 1.0–5.1 and 1.8–26.9 in solving the problems (4) and (5), respectively. The IIPM method is faster than the PPA method by a factor of 2.7–5.7 and 3.2–5.2 in solving the problems (4) and (5), respectively.

4.2 Real world examples

Gaussian graphical model (GGM) has become a popular statistical tool in the reverse engineering of genetic regulatory networks, where individual genes are represented by the nodes in a graph and the conditional dependencies between their expression profiles are indicated by edges. The GGM constructed from the sample data is usually dense, which covers underlying interactions among the genes. Moreover, the number of genes can reach thousands while the number of samples is limited. Note that the rank of a sample covariance matrix cannot exceed n, where n is the sample size. Thus for such a “large p small n” data set, the sample covariance matrix is not positive definite and it is not suitable for many statistical purposes. The sparse covariance selection model (35) can help to reduce spurious edges in the graph and also to estimate a positive definite covariance matrix. In this section, we consider several gene expression data sets that have been widely used in the model selection and classification literature.

Table 2 Comparison of the IIPM and ANS methods in solving the problem (4) with the data matrix $\widehat{\Sigma}$ generated from Example 2

Problem	$p \mid m$	Iteration count		Primal objective value		Time (s)	
		IIPM	ANS|PPA	IIPM	ANS|PPA	IIPM	ANS|PPA
ar1	500 | 0	22 (24.6| 2.2-1| 4.7-2| 0.99| 1.00)	957 |111	9.13018630 2	9.76-4| 3.49-3	42.5	189.4|204.5
ar1	1000 | 0	28 (37.5| 2.3-1| 3.9-2| 1.00| 1.00)	2109 |130	1.84038733 3	−4.00-1|−3.50-1	446.8	2363.2|1632.7
ar2	500 | 0	14 (11.1| 1.3-2| 5.0-2| 0.98| 1.00)	248 |51	7.51955161 2	3.74-1| 3.73-1	17.3	56.6|61.2
ar2	1000 | 0	15 (12.1| 8.2-3| 4.5-2| 1.00| 1.00)	291 |48	1.51119408 3	9.24-1| 9.23-1	115.8	374.1|313.0
ar3	500 | 0	13 (10.3| 1.3-2| 5.4-2| 0.99| 0.76)	208 |45	6.92658631 2	1.42 0| 1.42 0	15.5	44.8|53.1
ar3	1000 | 0	12 (9.0| 8.5-3| 5.2-2| 1.00| 0.74)	268 |48	1.39055266 3	−6.84-1|−6.85-1	80.5	325.3|293.1
ar4	500 | 0	12 (6.7| 1.2-2| 5.7-2| 0.99| 0.52)	110 |37	6.77784642 2	2.24 0| 2.24 0	12.2	26.1|44.2
ar4	1000 | 0	12 (7.5| 8.5-3| 5.5-2| 1.00| 0.52)	118 |37	1.35939369 3	1.46-1| 1.45-1	74.7	159.0|225.0
Full	500 | 0	10 (3.1| 4.7-3| 1.6-2| NaN| 0.00)	34 |24	5.45765356 2	1.75 0| 1.74 0	8.3	9.3|32.3
Full	1000 | 0	10 (3.1| 3.1-3| 1.1-2| NaN| 0.00)	36 |29	1.09305256 3	2.12 0| 2.12 0	47.7	58.0|187.2
Decay	500 | 0	10 (3.6| 7.4-3| 1.6-2| 1.00| 0.01)	32 |25	5.62153506 2	1.73 0| 1.73 0	8.6	8.7|32.7
Decay	1000 | 0	10 (3.5| 5.2-3| 1.6-2| 1.00| 0.00)	31 |29	1.12579339 3	5.14-1| 5.13-1	49.3	51.6|185.6
Star	500 | 0	11 (3.2| 5.0-1| 6.2-3| 1.00| 0.33)	51 |44	5.58744528 2	8.90-1| 9.01-1	9.1	12.5|51.4
Star	1000 | 0	11 (3.1| 4.5-1| 5.1-3| 1.00| 0.33)	65 |47	1.10745413 3	9.78-1| 1.17 0	52.4	90.8|296.5
Circle	500 | 0	22 (25.7| 2.2-1| 4.8-2| 0.99| 1.00)	1115 |108	9.14061736 2	1.73-1| 1.78-1	43.1	220.2|198.3
Circle	1000 | 0	29 (40.2| 2.3-1| 3.9-2| 1.00| 1.00)	2232 |133	1.84145113 3	2.16 0| 2.20 0	485.5	2499.2|1703.0

The regularization parameter ρ is set to $\rho = 0.1$ for all the problems. The numbers in each parenthesis are the average number of SQMR steps taken in each iteration, Loss_Q, Loss_E, Specificity and Sensitivity, respectively

Table 3 Comparison of the IIPM and ANS methods in solving the problem (5) with the data matrix $\widehat{\Sigma}$ generated from Example 2

Problem	$p \mid m$	Iteration count		Primal objective value		Time (s)	
		IIPM	ANS|PPA	IIPM	ANS|PPA	IIPM	ANS|PPA
ar1	500 | 62126	22 (34.3| 2.2-1| 4.1-2| 1.00| 1.00)	8524 |122	9.16396506 2	2.90-1| 2.93-1	52.3	1679.4|244.4
ar1	1000 | 249251	30 (53.1| 2.3-1| 3.5-2| 1.00| 1.00)	12211 |146	1.84492497 3	1.03 0| 1.07 0	633.7	13633.0|2233.6
ar2	500 | 61877	13 (11.3| 1.2-2| 4.8-2| 0.99| 1.00)	1250 |44	7.53991596 2	3.28-1| 3.27-1	15.5	285.1|50.6
ar2	1000 | 248752	13 (10.9| 8.3-3| 4.4-2| 1.00| 1.00)	1299 |53	1.51324471 3	2.94-1| 2.93-1	93.4	1675.3|333.6
ar3	500 | 61628	11 (9.7| 1.3-2| 5.4-2| 0.99| 0.77)	693 |41	6.93360782 2	7.55-5|−9.17-4	12.4	150.3|45.3
ar3	1000 | 248253	13 (11.2| 8.7-3| 5.2-2| 1.00| 0.74)	614 |50	1.39111340 3	5.76-1| 5.75-1	95.3	762.0|298.7
ar4	500 | 61380	11 (7.4| 1.2-2| 5.6-2| 1.00| 0.53)	284 |34	6.78121646 2	1.41 0| 1.41 0	11.2	72.1|38.0
ar4	1000 | 247755	11 (7.9| 8.6-3| 5.6-2| 1.00| 0.52)	252 |38	1.35948285 3	1.33 0| 1.33 0	68.8	381.5|220.1
Full	500 | 62375	10 (3.1| 4.7-3| 1.6-2| NaN| 0.00)	52 |25	5.45773261 2	1.64-1| 1.63-1	7.9	22.9|30.0
Full	1000 | 249750	10 (3.1| 3.1-3| 1.1-2| NaN| 0.00)	56 |28	1.09305145 3	1.16 0| 1.15 0	46.6	210.5|173.6
Decay	500 | 57961	10 (3.7| 7.4-3| 1.6-2| 1.00| 0.01)	42 |25	5.62165596 2	1.69 0| 1.69 0	8.2	14.8|30.6
Decay	1000 | 240836	10 (3.6| 5.2-3| 1.6-2| 1.00| 0.00)	47 |28	1.12579253 3	1.59 0| 1.59 0	48.4	171.0|175.2
Star	500 | 62126	11 (3.1| 5.0-1| 6.1-3| 1.00| 0.33)	66 |43	5.58755927 2	1.08 0| 1.10 0	8.6	25.6|49.0
Star	1000 | 249251	11 (3.0| 4.5-1| 5.1-3| 1.00| 0.33)	82 |46	1.10745324 3	2.17 0| 2.50 0	50.6	225.8|263.1
Circle	500 | 62125	23 (36.1| 2.2-1| 4.1-2| 1.00| 1.00)	9876 |126	9.17446117 2	3.50-1| 3.53-1	56.0	1955.7|259.9
Circle	1000 | 249250	28 (42.6| 2.3-1| 3.5-2| 1.00| 1.00)	12161 |149	1.84600264 3	8.96-1| 9.52-1	498.8	13513.2|2262.3

The regularization parameter ρ is set to $\rho = 0.1$ for all the problems. The numbers in each parenthesis are the average number of SQMR steps taken in each iteration, Loss_Q, Loss_E, Specificity and Sensitivity, respectively

Example 3 (Lymph node status data) Lymph node status is an important clinical risk factor affecting the long-term outlook of breast cancer treatment outcome. Pittman et al. [25] analyzed the prediction of Lymph mode positivity status at gene expression level. Here, we use the data after the pre-processing of Dobra [9], which consists of 4514 genes from 148 samples. The samples can be divided into two classes, 100 low-risk (node-negative) and 48 high-risk (high-node-positive).

Example 4 (Estrogen receptor data) Increasingly, patterns of gene expressions are combined with traditional clinical risk factors in the prediction of disease outcome at the individual patient level. As mentioned before, Pittman et al. [25] demonstrated substantially improved accuracy in the combined prediction of primary breast cancer recurrence. Their study involves 158 breast cancer patients. The data we use is after an initial pre-processing [9] and contains 7027 probe sets in 158 samples that are potentially related to estrogen receptor pathway. The log-scaled and normalized data can be downloaded from Dobra's BMSS package [9].

Example 5 (Arabidopsis thaliana data) *Arabidopsis thaliana*, a small flowering plant, is important for understanding the genetic pathways of many plant traits, partially due to its small genome. Wille et al. [40] studied a gene network for isoprenoid biosynthesis in *Arabidopsis thaliana*, which links to many other biochemical products in plants such as sterols (membranes), gibberellins(hormones), carotenoids and chlorophylls (photosynthetic pigments). Their data set contains the gene expression data from 40 isoprenoid genes in mevalonate (MVA) and non-mevalonate (MEP) pathways as well as 795 additional genes from 56 downstream pathways. All gene expression values were monitored under various experimental conditions using 118 GeneChip (Affymetrix) microarrays.

Example 6 (Leukemia data) Golub et al. [14] developed a generic approach to cancer classification based on gene expression data. Their data contains 7129 human genes monitored by DNA microarrays from 72 samples. The samples are divided into two cancer classes, 25 in the class acute myeloid leukemia (AML) and 47 in the class acute lymphoblastic leukemia (ALL). Yeung et al. [43] further reduced the data set to 3501 genes with significant variance across the two classes. We use their data for analysis.

Example 7 (Hereditary breast cancer data) In order to discover the connections between a mutant BRCA1 or BRCA2 gene and the risk of inherited breast cancer, Hedenfalk et al. [15] studied 3226 genes of primary breast tumors from both hereditary and sporadic cases, including 7 BRCA1-mutation-positive, 8 BRCA2-mutation-positive and 7 sporadic cases.

In Examples 4 and 7, we only select a sub-matrix of the sample covariance matrix for testing. The selection is based on [1, Theorem 4], where we remove columns and rows whose off-diagonal entries are all smaller than the regularization parameter ρ in absolute value. The rank of the matrix after the selection is expected to be the same as the original matrix. The dimension of the sub-matrix can be found in Table 4. In Examples 3 and 6, to reduce the dimension of the initial data, we apply false discovery rate (FDR) multiple testing and select q-values at 5% significance level; see [30] and [9].

Table 4 Comparison of the IIPM and ANS methods on the problem (4) using gene data sets

Problem	$p \mid r \mid \rho$	Iteration count		Primal objective value		Time (s)	
		IIPM	ANS|PPA	IIPM	ANS|PPA	IIPM	ANS|PPA
Lymph	587 | 147 | 0.50	20 (8.2)	443 | 43	8.13260834 2	5.96-4 |−3.87-4	34.6	131.6 | 80.5
ER	692 | 157 | 0.50	20 (13.6)	931 | 49	9.23106034 2	−9.26-4 |−1.76-3	62.1	415.3 | 146.1
Arabidopsis	834 | 117 | 0.50	24 (14.2)	1074 | 56	1.10930058 3	4.84-4 |−2.11-4	126.4	752.6 | 321.7
Leukemia	1255 | 71 | 0.50	30 (16.7)	1718 | 60	1.69788920 3	−1.32-3 |−2.21-3	533.6	3829.0 | 1258.6
Hereditary bc	1869 | 21 | 0.50	29 (17.2)	3567 | 70	2.37258798 3	−1.11-3 |−1.98-3	1563.7	24619.2 | 4787.4

The number in parenthesis is the average number of SQMR steps taken in each iteration. In the table, r is the rank of $\widehat{\Sigma}$

We have the data log-scaled and normalized so that the sample mean is zero and the sample variance for each gene is one. The parameter ρ is set to be 0.5 for all the examples. The performance of the IIPM and ANS methods on the problem (4) for the five real data sets is summarized in Table 4. As before, the stopping tolerance for the IIPM and PPA methods is set to $\texttt{Tol} = 10^{-6}$ while for the ANS method, the tolerance is set to $\varepsilon_o = 10^{-3}$. As we can see from the table, the IIPM method consistently outperforms the ANS and PPA methods in terms of the CPU time taken to achieve almost the same objective values. For the largest problem with $p = 1869$, the IIPM method is about 15.8 times faster than the ANS method, and it is about 3.1 times faster than the PPA method.

5 Conclusion

We have designed an inexact primal–dual interior-point algorithm (IIPM) for solving large scale log-det SDP problems. We also customized it to solve sparse covariance selection problems. To ensure that our IIPM is practically viable, we designed efficient preconditioners for the ill-conditioned linear systems of equations arising in each iteration of the IIPM. Our IIPM enjoys the robustness and efficiency (in terms of iteration count) of classic interior-point methods and is capable of solving large scale log-det SDP problems arising from sparse covariance selection. Numerous numerical experiments conducted on sparse covariance selection problems with both synthetic and real data have shown that IIPM outperforms other current major algorithms in terms of computing time and accuracy. Observing that IIPM can achieve satisfactory practical efficiency (in terms of computing time and memory requirement) on log-det SDP and convex quadratic SDP problems [33], we hope to extend the IIPM approach for more general types of convex optimization problems in the future, for example, linearly constrained convex SDP where the objective function is a smooth convex function of the matrix variable X.

Acknowledgments We thank the referees for their helpful comments and suggestions to improve the paper.

References

1. Banerjee, O., El Ghaoui, L., d'Aspremont, A.: Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. J. Mach. Learn. Res. **9**, 485–516 (2008)
2. Bilmes, J.A.: Natural statistical models for automatic speech recognition. PhD thesis, University of California, Berkeley (1999)
3. Burer, S., Monteiro, R.D.C., Zhang, Y.: A computational study of a gradient-based log-barrier algorithm for a class of large-scale SDPs. Math. Program. **95**, 359–379 (2003)
4. Chen, S.S., Gopinath, R.A.: Model selection in acoustic modeling. In: Proc. EUROSPEECH'99, pp. 1087–1090, Budapest, Hungary (1999)
5. Dahl, J., Vandenberghe, L., Roychowdhury, V.: Covariance selection for nonchordal graphs via chordal embedding. Optim. Methods Softw. **23**, 501–520 (2008)
6. d'Aspremont, A.: Identifying small mean reverting portfolios. Quant. Finance (2010, to appear)
7. d'Aspremont, A., Banerjee, O., El Ghaoui, L.: First-order methods for sparse covariance selection. SIAM J. Matrix Anal. Appl. **30**, 56–66 (2008)
8. Dempster, A.P.: Covariance selection. Biometrics **28**, 157–175 (1972)

9. Dobra, A.: Variable selection and dependency networks for genomewide data. Biostatistics **10**, 621–639 (2009)
10. Edwards, D.: Introduction to graphical modelling, 2nd edn. Springer, New York (2000)
11. Fan, J., Feng, Y., Wu, Y.: Network exploration via the adaptive LASSO and SCAD penalties. Ann. Appl. Stat. **3**, 521–541 (2009)
12. Freund, R., Nachtigal, N.: A new Krylov-subspace method for symmetric indefinite linear system. In: Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics, Atlanta, USA, pp. 1253–1256 (1994)
13. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics **9**, 432–441 (2008)
14. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science **286**, 531–537 (1999)
15. Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Raffeld, M., Yakhini, Z., Ben-Dor, A., Dougherty, E., Kononen, J., Bubendorf, L., Fehrle, W., Pittaluga, S., Gruvberger, S., Loman, N., Johannsson, O., Olsson, H., Wilfond, B., Sauter, G., Kallioniemi, O.-P., Borg, A., Trent, J.: Gene-expression profiles in hereditary breast cancer. N. Engl. J. Med. **344**, 539–548 (2001)
16. Jarre, F., Rendl, F.: An augmented primal-dual method for linear conic programs. SIAM J. Optim. **19**, 808–823 (2008)
17. Krishnamurthy, V., d'Aspremont, A.: A pathwise algorithm for covariance selection. Preprint (2009)
18. Lan, G., Lu, Z., Monterio, R.D.: Primal-dual first-order methods with $\mathcal{O}(1/\epsilon)$ iteration-complexity for cone programming. Math. Program. (2010, to appear)
19. Lauritzen, S.L.: Graphical models. In: Oxford Statistical Science Series, vol. 17. The Clarendon Press/Oxford University Press/Oxford Science Publications, New York (1996)
20. Lu, Z.: Smooth optimization approach for sparse covariance selection. SIAM J. Optim. **19**, 1807–1827 (2008)
21. Lu, Z.: Adaptive first-order methods for general sparse inverse covariance selection. SIAM J. Matrix Anal. Appl. **31**, 2000–2016 (2010)
22. Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the lasso. Ann. Stat. **34**, 1436–1462 (2006)
23. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**, 127–152 (2005)
24. Nesterov, Y., Todd, M.J.: Primal-dual interior-point methods for self-scaled cones. SIAM J. Optim. **8**, 324–364 (1998)
25. Pittman, J., Huang, E., Dressman, H., Horng, C.-F., Cheng, S.H., Tsou, M.-H., Chen, C.-M., Bild, A., Iversen, E.S., Huang, A.T., Nevins, J.R., West, M.: Integrated modeling of clinical and gene expression information for personalized prediction of disease outcomes. Proc. Natl. Acad. Sci. USA **101**(22), 8431–8436 (2004)
26. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. Oper. Res. **1**, 97–116 (1976)
27. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2003)
28. Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D.A., Nolan, G.P.: Causal protein-signaling networks derived from multiparameter single-cell data. Science **308**, 523–529 (2005)
29. Scheinberg, K., Rish, I.: Learning sparse Gaussian Markov networks using a greedy coordinate ascent approach. In: Balcázar, J., Bonchi, F., Gionis, A., Sebag, M. (eds.) Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science 6323. pp. 196–212 (2010)
30. Storey, J.D., Tibshirani, R.: Statistical significance for genome-wide studies. Proc. Natl. Acad. Sci. USA **100**(16), 9440–9445 (2003)
31. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11/12**, 625–653 (1999)
32. Toh, K.-C.: Solving large scale semidefinite programs via an iterative solver on the augmented systems. SIAM J. Optim. **14**, 670–698 (2003)
33. Toh, K.-C.: An inexact primal-dual path following algorithm for convex quadratic SDP. Math. Program. **112**, 221–254 (2008)
34. Toh, K.-C., Todd, M.J., Tütüncü, R.H.: SDPT3—a MATLAB software package for semidefinite programming, version 1.3. Optim. Methods Softw. **11/12**, 545–581 (1999)

35. Tsuchiya, T., Xia, Y.: An extension of the standard polynomial-time primal-dual path-following algorithm to the weighted determinant maximization problem with semidefinite constraints. Pac. J. Optim. **3**, 165–182 (2007)
36. Ueno, U., Tsuchiya, T.: Covariance regularization in inverse space. Q. J. R. Meteorol. Soc. **135**, 1133–1156 (2009)
37. Vandenberghe, L., Boyd, S., Wu, S.-P.: Determinant maximization with linear matrix inequality constraints. SIAM J. Matrix Anal. Appl. **19**, 499–533 (1998)
38. Wang, C., Sun, D., Toh, K.-C.: Solving log-determinant optimization problems by a newton-cg proximal point algorithm. SIAM J. Optim. **20**, 2994–3013 (2010)
39. Whittaker, J.: Graphical models in applied multivariate statistics. In: Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. Wiley, Chichester (1990)
40. Wille, A., Zimmermann, P., Vranová, E., Fürholz, A., Laule, O., Bleuler, S., Hennig, L., Prelić, A., von Rohr, P., Thiele, L., Zitzler, E., Gruissem, W., Bühlmann, P.: Sparse graphical gaussian modeling of the isoprenoid gene network in *Arabidopsis thaliana*. Genome Biol. **5**, R92 (2004)
41. Wong, F., Carter, C.K., Kohn, R.: Efficient estimation of covariance selection models. Biometrika **90**, 809–830 (2003)
42. Wu, W.B., Pourahmadi, M.: Nonparameteric estimation of large covariance matrices of longitudinal data. Biometrika **90**, 831–844 (2003)
43. Yeung, K.Y., Bumgarner, R.E., Raftery, A.E.: Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. Bioinformatics **21**, 2394–2402 (2005)
44. Yuan, M., Lin, Y.: Model selection and estimation in the Gaussian graphical model. Biometrika **94**, 19–35 (2007)
45. Yuan, X.: Alternating direction methods for sparse covariance selection. Preprint (2009)
46. Zhang, Y.: On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming. SIAM J. Optim. **8**, 365–386 (1998)
47. Zhao, X.Y., Sun, D., Toh, K.-C.: A Newton-CG augmented Lagrangian method for semidefinite programming. SIAM J. Optim. **20**, 1737–1765 (2010)
48. Zhou, G., Toh, K.-C.: Polynomiality of an inexact infeasible interior point algorithm for semidefinite programming. Math. Program. **99**, 261–282 (2004)