SIAM J. OPTIM.
Vol. 23, No. 2, pp. 857–893




# A PROXIMAL POINT ALGORITHM FOR LOG-DETERMINANT OPTIMIZATION WITH GROUP LASSO REGULARIZATION*

JUNFENG YANG†, DEFENG SUN‡, AND KIM-CHUAN TOH§

**Abstract.** We consider the covariance selection problem where variables are clustered into groups and the inverse covariance matrix is expected to have a blockwise sparse structure. This problem is realized via penalizing the maximum likelihood estimation of the inverse covariance matrix by group Lasso regularization. We propose to solve the resulting log-determinant optimization problem with the classical proximal point algorithm (PPA). At each iteration, as it is difficult to update the primal variables directly, we first solve the dual subproblem by an inexact semismooth Newton-CG method and then update the primal variables by explicit formulas based on the computed dual variables. We also propose to accelerate the PPA by an inexact generalized Newton's method when the iterate is close to the solution. Theoretically, we prove that at the optimal solution, the nonsingularity of the generalized Hessian matrices of the dual subproblem is equivalent to the constraint nondegeneracy condition for the primal problem. Global and local convergence results are also presented for the proposed PPA. Moreover, based on the augmented Lagrangian function of the dual problem we derive an alternating direction method (ADM), which is easily implementable and is demonstrated to be efficient for random problems. Numerical results, including comparisons with the ADM on both synthetic and real data, are presented to demonstrate that the proposed Newton-CG based PPA is stable and efficient and, in particular, outperforms the ADM when high accuracy is required.

**Key words.** covariance selection, log-determinant optimization, group Lasso regularization, proximal point algorithm, augmented Lagrangian, alternating direction method, Newton's method, Gaussian graphical model

**AMS subject classifications.** 65K05, 65K10, 65J22, 90C25

**DOI.** 10.1137/120864192

## 1. Introduction.

In many applications, e.g., multivariate data analysis, the relationships among a set of variables are usually described by an undirected graph, where each node represents a certain variable and two nodes are unconnected if and only if the corresponding variables are conditionally independent, i.e., independent with all other variables being fixed. This graph is frequently referred to as a graphical model of the set of random variables. In many cases, we are required to select a graphical model that adequately explains the observed data and yet has a simple structure, i.e., fewer edges. When the set of random variables are jointly normally distributed, the graphical model is also known as a Gaussian graphical model.

Let $\{y_i \in \mathbb{R}^n : i = 1, 2, \ldots, p\}$ be a set of samples independently drawn from an $n$-variate Gaussian distribution $N(0, \Sigma)$. We assume that the covariance matrix $\Sigma$ is

*Received by the editors January 31, 2012; accepted for publication (in revised form) February 1, 2013; published electronically May 7, 2013.
http://www.siam.org/journals/siopt/23-2/86419.html

†Department of Mathematics, Nanjing University, Nanjing, China, 210093 (jfyang@nju.edu.cn). This author was supported by the Natural Science Foundation of China NSFC-11001123, Program for New Century Excellent Talents in University NCET-12-0252, and the Fundamental Research Funds for the Central Universities. This work was done while this author was a research fellow at the National University of Singapore.

‡Department of Mathematics and Risk Management Institute, National University of Singapore, Singapore 119076 (matsundf@nus.edu.sg). This author's research was supported in part by Academic Research Fund grant R-146-000-149-112.

§Department of Mathematics, National University of Singapore, Singapore 119076 (mattohkc@nus.edu.sg).

nonsingular. The goal is to estimate from the given samples the covariance matrix $\Sigma$, whose inverse is expected to have a sparse structure, i.e., fewer nonzero entries. This is largely because sparsity in the inverse covariance matrix (a.k.a. precision or concentration matrix) corresponds to conditional independence. In fact, Dempster [12] proved that any two components, say, $x_i$ and $x_j$, of $x \sim N(0, \Sigma)$ are conditionally independent if and only if $(\Sigma^{-1})_{ij} = 0$. Based on this theoretical result, Dempster suggested directly setting some selected entries of the inverse covariance matrix to be zero, which leads to robust and efficient estimates of the covariance matrix in the case when its inverse matrix indeed has a large number of zero elements. The estimation of the sparsity pattern (and sometimes the values of the nonzero entries) of $\Sigma^{-1}$ is called covariance selection, which has diverse applications in, e.g., speech recognition [3] and gene network analysis [13]. Recently, the covariance selection problem has mainly been studied in the low sample size and high dimensional setting; see [40].

Let $S := \frac{1}{p}\sum_{k=1}^{p} y_k y_k^\top$ be the sample covariance matrix. To estimate $\Sigma^{-1}$, it is natural to consider the maximum likelihood estimation (MLE), which is given by

$$\hat{\Sigma}^{-1} = \arg\min_{X \succeq 0} \langle S, X\rangle - \log\det X. \tag{1.1}$$

Here the notation $X \succeq 0$ represents that $X$ is symmetric and positive semidefinite. (In this paper, $\log(\cdot)$ represents the natural logarithm function, and $\log 0 = -\infty$ is assumed wherever it might occur.) Unfortunately, the MLE alone is usually not sufficient for our purpose because, first, $S$ may not be positive definite (e.g., when $p < n$), in which case the objective function in (1.1) is unbounded below; and second, even if $S$ is positive definite, the MLE, which is easily shown to be given by $\hat{\Sigma}^{-1} = S^{-1}$, may not have the desired sparsity structure determined by a prior given conditional independence. Furthermore, it is well known that the MLE is not a robust estimator for many statistical purposes.

**1.1. Some existing approaches.** To promote sparsity in the inverse covariance matrix, many heuristic, statistical, and variational approaches have been suggested in the literature, e.g., Lauritzen [31] proposed a greedy forward-backward cardinality search algorithm to determine the sparsity pattern of $\Sigma^{-1}$; Dobra and West [14] considered Bayesian covariance selection via a stochastic algorithm and utilized prior information; Li and Gui [33] applied an intuitive thresholding gradient ascent method to the log-likelihood function to estimate $\Sigma^{-1}$; Huang, Liu, and Pourahmadi [29] reparameterized the covariance matrix through its modified Cholesky factorization and considered penalized MLE; and Dahl, Roychowdhury, and Vandenberghe [10] considered MLE with predetermined sparsity constraints $\{X_{ij} = 0 : (i,j) \in \Omega\}$, where $\Omega$ is an index set. In particular, Banerjee, El Ghaoui, and d'Aspremont [2] and Yuan and Lin [62] proposed to penalize the MLE by the $\ell_1$-norm of $X$, resulting in an optimization problem of the form

$$\min_{X \succeq 0} \langle S, X\rangle - \log\det X + \omega\|X\|_1. \tag{1.2}$$

Here $\|X\|_1 := \sum_{ij} |X_{ij}|$ and $\omega > 0$ is a parameter to balance the relative importance between the log-likelihood and regularization. In fact, the use of $\ell_1$-regularization to promote solution sparsity can be traced back to the 1960s and was mainly started in geophysics for searching the so-called sparse spike trains; see, e.g., [50]. Recently, $\ell_1$-regularization has been extensively utilized in various applications including linear regression [53], overcomplete decomposition [8], principal component analysis [11], and compressive sensing [5, 15]. More importantly, the $\ell_1$-norm is a simple convex

function, which facilitates efficient computation (at least theoretically). In the covariance selection setting, (1.2) is a strictly convex problem due to the presence of the strictly convex function $\log\det X^{-1}$. Therefore, standard interior point methods (IPMs) are in principle applicable, at least to problems with small $n$, e.g., in [62] the authors utilized standard IPMs to solve (1.2). Unfortunately, it is impossible to solve (1.2) efficiently on a common PC via IPMs when $n$ is large, say, more than 200. As a result, many customized algorithms for solving (1.2) and related problems have been designed in the literature, e.g., block coordinate descent method [22, 2, 64], projected subgradient method [16], Nesterov's first-order methods [42, 43] and their variants [2, 35, 36], alternating direction method (ADM) [63], Newton-CG based proximal point algorithm (PPA) [54], and inexact IPM with effective preconditioners [34]. In general, first-order algorithms (block coordinate descent, projected subgradient, ADM, Nesterov's methods and their variants) are easily implementable and fast to obtain low or moderate accuracy solutions. The Newton-CG based PPA works stably and is more efficient in obtaining solutions of higher accuracy. The customized inexact IPM with effective preconditioners can even be faster than the Newton-CG based PPA, but it is not applicable to log-determinant problems like (1.2) with other types of regularization and/or additional generic linear constraints other than $\{X_{ij} = 0 : (i,j) \in \Omega\}$.

**1.2. Covariance selection with group Lasso regularization.** In many applications, variables are naturally clustered into groups, and those from the same group are more likely to be connected than those from different ones. For example, in machine learning when modeling a two-dimensional shape made up of articulated objects, landmarks along the contours of an animal's different parts (e.g., legs, head, tail) can naturally be grouped together, as these landmarks move collectively as the animal moves through different articulated forms; see [16] for details. Another example comes from the modeling of gene networks, where genes can be grouped into pathways and interactions happen at the level of pathways, i.e., either two pathways interact or they do not interact at all. In such applications, a blockwise sparsity structure in the inverse covariance matrix is highly desired. Let $\mathcal{A}$ be a generic linear mapping from $S^n$ (the set of $n \times n$ symmetric matrices) to $\mathbb{R}^m$. To promote group sparsity, we penalize the MLE by group Lasso regularization, resulting in an optimization problem of the form

$$\min_X \left\{ \langle S, X\rangle - \log\det X + \omega \sum\nolimits_{g\in G} \|X_g\|_{\#} : \mathcal{A}X = b, X \succeq 0 \right\}. \tag{1.3}$$

Here each $g$ is a subset of $\{(i,j) : i,j = 1,2,\ldots,n\}$, $G$ is a collection of such index sets, $X_g$ is a vector of length $|g|$ (the cardinality of $g$) formed by the components of $X$ with indices in $g$, $\|\cdot\|_{\#}$ is a certain norm, and $b \in \mathbb{R}^m$. The equation $\mathcal{A}X = b$ in (1.3) enforces a set of additional linear constraints on $X$, which could be determined via prior knowledge about the inverse covariance matrix in a specific application. We note that group Lasso regularization has been used in the literature to promote blockwise sparsity; see, e.g., [61, 39, 1, 65] for group $\ell_2$-regularized (logistic) regression and [16] for group $\ell_\infty$-regularized covariance selection. A specific example of (1.3) is the multitask structure learning problem for Gaussian graphical models [28]. Given $k$ arbitrary tasks, the following problem was considered in [28, equation (3)] to promote a consistent sparsity pattern across different tasks:

$$\min_{X_1,\ldots,X_k\succeq 0} \sum_{t=1}^{k} \left(\langle S_t, X_t\rangle - \log\det X_t\right) + \omega \sum_{i,j=1}^{m} \|(X_{1,ij},\ldots,X_{k,ij})\|_\infty, \tag{1.4}$$

where for each $t$, $S_t$ denotes a given data matrix of size $m \times m$, and $X_{t,ij}$ denotes the $(i,j)$th component of $X_t$, $t = 1, 2, \ldots, k$. Let $S = \mathrm{diag}(S_1, \ldots, S_k)$ and $X = \mathrm{diag}(X_1, \ldots, X_k)$. Since the constraint that off-diagonal blocks of $X$ are equal to zero can be represented by $\mathcal{A}X = b$ with an appropriate $\mathcal{A}$, it is clear that (1.4) is a special case of (1.3) with $\|\cdot\|_{\#} = \|\cdot\|_\infty$ and an appropriate $G$. Theoretically, the group Lasso regularization is equivalent to enforcing certain bound constraints on the magnitudes of $X_g$'s. Obviously, the choice of the group structure $G$, the regularization norm $\|\cdot\|_{\#}$ and the parameter $\omega$ are very important issues, which usually depend on specific application problems. In this paper, we assume that they are given priors, and our objective is to design an efficient algorithm for solving the optimization problem. In practical applications, the group structure $G$ can be either a known prior or learned from statistical machine learning algorithms; see, e.g., [37]. For convenience, we assume that $G = \{g_i : i = 1, 2, \ldots, r\}$ and it satisfies the following assumption.

*Assumption* 1. Different groups in $G$ are disjoint, i.e., $g_i \cap g_j = \emptyset$ for all $1 \le i < j \le r$.

In practice, problems more general than (1.3) can be considered, e.g., local weights can be enforced, and different norms can be applied to different groups. Taking into account these two factors, we obtain the model problem which we will concentrate on in this paper:

$$\min_X \left\{ \langle S, X\rangle - \log\det X + \sum\nolimits_{\ell=1}^{r} \varphi_\ell(X_{g_\ell}) : \mathcal{A}X = b, X \succeq 0 \right\}, \tag{1.5}$$

where, for each $\ell$, $\varphi_\ell : \mathbb{R}^{|g_\ell|} \to \mathbb{R}$ is a simple, closed proper convex function (local weights can be implicitly included), $\mathcal{A}$ is a generic linear mapping from $S^n$ to $\mathbb{R}^m$, and $b \in \mathbb{R}^m$. Obviously, explicit sparsity constraints of the form $\{X_{ij} = 0 : (i,j) \in \Omega\}$ can be enforced via the linear constraints $\mathcal{A}X = b$. In this paper, we make the following assumption on $\mathcal{A}$.

*Assumption* 2. The generic linear mapping $\mathcal{A}$ from $S^n$ to $\mathbb{R}^m$ is surjective.

**1.3. Motivation and contributions.** Recently, Zhao, Sun, and Toh [66] proposed to solve the dual form of a standard linear semidefinite programming (SDP) problem by a Newton-CG based augmented Lagrangian (NAL) method, which is essentially a PPA applied to the primal SDP where the inner subproblems are solved by an inexact generalized Newton's method for semismooth equations. The extensive numerical results presented in [66] demonstrated that the NAL method can be highly efficient for solving large-scale linear SDP problems whenever the constraint nondegeneracy conditions hold for both the primal and the dual problems. The efficiency of the NAL method can be partly explained by the theoretical results in [6, 52], where it is shown that under the constraint nondegeneracy conditions the augmented Lagrangian method (ALM) can be locally regarded as an approximate generalized Newton's method applied to a semismooth equation. Given the efficiency of the NAL method for SDP, Wang, Sun, and Toh [54] adopted a similar idea to solve the log-determinant problem (1.2) with additional linear constraints, where the problem is transformed into a smooth problem via introducing auxiliary variables. The resulting algorithm was shown to be approximately 2∼20 times faster than the adaptive Nesterov's smoothing method [35], one of the fastest first-order methods for solving (1.2) and some of its variants.

Motivated by the robustness and the effectiveness of the Newton-CG based PPA, in this paper we extend the idea of [66, 54] to solving (1.5), which clearly contains a much broader class of problems. Unlike the problem considered in [54], in the case of

group Lasso regularization, it is generally not feasible to transform (1.5) into a smooth problem. Therefore, at each iteration a nonsmooth PPA subproblem needs to be solved. Our approach is to first solve the dual subproblem via an inexact generalized Newton's method for the dual variables and then update the primal variables via explicit formulas based on the computed dual variables. We also propose to accelerate the PPA by an inexact generalized Newton's method when the iterate is close to the solution. The key difference between this algorithmic framework and that in [54] is the nonsmoothness of the objective function in the dual subproblem. In [54], the objective function in the dual subproblem is smooth and thus Newton's method of normal type can be applied. In our case, the objective function of the dual subproblem is only first-order continuously differentiable. We used a generalized Newton's method for solving semismooth nonlinear equations and solved the generalized Newton equation iteratively. However, the efficiency of the inexact generalized Newton's method for solving the dual subproblem depends on the nonsingularity of the generalized Hessian matrices of the dual subproblem. We prove that the nonsingularity of the generalized Hessian matrices of the dual subproblem is equivalent to the constraint nondegeneracy of the primal problem. Global and local convergence results of the proposed Newton-CG based PPA are also presented. Moreover, based on the dual problem of (1.5) we derive a first-order ADM-like algorithm, which is used for comparison with the Newton-CG based PPA.

**1.4. Notation.** In the following, we let $S^n$, $S^n_+$, and $S^n_{++}$ be the sets of all $n \times n$ symmetric, symmetric positive semidefinite, and symmetric positive definite matrices, respectively. For convenience, $X \in S^n_+$ (resp., $X \in S^n_{++}$) also is represented by $X \succeq 0$ (resp., $X \succ 0$) occasionally. The derivative of a mapping $f : S^n \to S^n$ at $X$ is denoted by $f'(X)[H]$ for any $H \in S^n$, i.e., $f'(X)[H] := \lim_{t\to 0} \frac{f(X+tH)-f(X)}{t}$. The transpose operation of a vector or matrix variable is denoted by superscript $^\top$, and the adjoint operators of $\mathcal{A}$ and $\mathcal{P}$ are represented, respectively, by $\mathcal{A}^*$ and $\mathcal{P}^*$. The identity matrix of appropriate size is denoted by $I$. The signum function is denoted by sgn, which represents componentwise operation when applied to vector variables. The notation $\|\cdot\|$ represents the Frobenius norm $\|\cdot\|_F$ (resp., the 2-norm $\|\cdot\|_2$) for matrix (resp., vector) variables. For matrices $X, Y$ and vectors $x, y$ of appropriate sizes, we define $\langle (X, x), (Y, y)\rangle = \mathrm{tr}(X^\top Y) + x^\top y$, where tr represents the trace operation, and the induced norm $\|(X, y)\| = \sqrt{\|X\|^2 + \|y\|^2}$. The Hadamard product or componentwise multiplication of two vectors or matrices of the same size is denoted by $\circ$. The set of generalized Jacobian matrices of a mapping $f$ at a certain point $x$ is denoted by $\partial f(x)$ (see the definition in (3.20)). The notation $\partial$ is also used to denote partial derivative of a mapping with respect to certain variables (see, e.g., (3.25)), and with a little overloading of notation, the boundary of a set $K$ is also represented by $\partial K$ (see, e.g., (4.6)). We note that these uses of the notation $\partial$ in this paper do not cause confusion in context. Other notation will be defined when it occurs.

**1.5. Organization.** The rest of this paper is organized as follows. In section 2, we review the concept of Moreau–Yosida regularization and its basic properties which will be used in subsequent analysis. In section 3, we present a Newton-CG based PPA for solving (1.5). Some theoretical results, including global and local convergence, are given in section 4. In section 5, based on the augmented Lagrangian function of the dual problem we derive an ADM-like algorithm for solving (1.5). Numerical results, including comparisons with the ADM on various types of data, are presented in section 6. Finally, some concluding remarks are given in section 7.

**2. The Moreau–Yosida regularization.** Let $\mathcal{E}$ be a finite dimensional real Euclidean space endowed with an inner product $\langle\cdot,\cdot\rangle$ and its induced norm $\|\cdot\|$. Let $\vartheta : \mathcal{E} \to \mathbb{R}$ be a closed proper convex function; see, e.g., [46]. For any $\beta > 0$, the Moreau–Yosida regularization [41, 60] of $\vartheta$ is defined by

$$\Phi_\vartheta^\beta(x) := \min_{y\in\mathcal{E}} \left\{ \vartheta(y) + \frac{1}{2\beta}\|y-x\|^2 \right\}, \quad x \in \mathcal{E}. \tag{2.1}$$

Since $\vartheta$ is proper and convex, it is minorized by an affine function in its effective domain. Therefore, the objective function in (2.1) is coercive (i.e., it goes to infinity as $\|y\|$ does). Further considering the closedness and strong convexity of the objective function, for any $x \in \mathcal{E}$, problem (2.1) has a unique optimal solution, which is well known as the proximal point of $x$ associated with $\vartheta$ and is denoted by $\pi_\vartheta^\beta(x)$, i.e.,

$$\pi_\vartheta^\beta(x) := \arg\min_{y\in\mathcal{E}} \left\{ \vartheta(y) + \frac{1}{2\beta}\|y-x\|^2 \right\}, \quad x \in \mathcal{E}. \tag{2.2}$$

The Moreau–Yosida regularization and the proximal point mapping have the properties given in the following proposition.

PROPOSITION 2.1 (see [27, 32, 49]). *Let $\vartheta : \mathcal{E} \to \mathbb{R}$ be a closed proper convex function. For any $\beta > 0$, the Moreau–Yosida regularization $\Phi_\vartheta^\beta(\cdot)$ and the associated proximal point mapping $\pi_\vartheta^\beta(\cdot)$ defined in* (2.1) *and* (2.2)*, respectively, have the following properties:*

(i) *$\Phi_\vartheta^\beta(\cdot)$ is continuously differentiable and convex on $\mathcal{E}$. Furthermore, it holds that*

$$\nabla\Phi_\vartheta^\beta(x) = \frac{1}{\beta}\left(x - \pi_\vartheta^\beta(x)\right), \quad x \in \mathcal{E}. \tag{2.3}$$

(ii) *$x^* \in \mathcal{E}$ minimizes $\vartheta$ over $\mathcal{E}$ if and only if it minimizes $\Phi_\vartheta^\beta$ over $\mathcal{E}$.*

(iii) *$\pi_\vartheta^\beta$ is globally Lipschitz continuous with modulus* 1*, i.e.,*

$$\|\pi_\vartheta^\beta(x) - \pi_\vartheta^\beta(y)\| \le \|x-y\| \quad \forall\, x, y \in \mathcal{E}.$$

Let $X \in S^n$ and $X = Q\,\mathrm{diag}(d_1, d_2, \ldots, d_n)Q^\top$ be its eigenvalue decomposition, where $d_1 \ge \cdots \ge d_n$. Let $\beta > 0$. For the two scalar functions $\phi_\beta^+(x) := (\sqrt{x^2+4\beta} + x)/2$ and $\phi_\beta^-(x) := (\sqrt{x^2+4\beta} - x)/2$, $x \in \mathbb{R}$, we define their matrix counterparts by

$$\begin{aligned} &\phi_\beta^+(X) := Q\,\mathrm{diag}(\phi_\beta^+(d_1), \ldots, \phi_\beta^+(d_n))Q^\top \ \text{ and} \\ &\phi_\beta^-(X) := Q\,\mathrm{diag}(\phi_\beta^-(d_1), \ldots, \phi_\beta^-(d_n))Q^\top, \ X \in S^n. \end{aligned} \tag{2.4}$$

Clearly, $\phi_\beta^+(X)$ and $\phi_\beta^-(X)$ are positive definite for any $X \in S^n$. The following properties of $\phi_\beta^+$ and $\phi_\beta^-$ will be used in our subsequent analysis.

PROPOSITION 2.2. *Let $\beta > 0$. For any $X \in S^n$ with eigenvalue decomposition $X = Q\,\mathrm{diag}(d_1, d_2, \ldots, d_n)Q^\top$, $\phi_\beta^+$ and $\phi_\beta^-$ defined in* (2.4) *satisfy the following properties:*

(a) *$\phi_\beta^+(X) - \phi_\beta^-(X) = X$ and $\phi_\beta^+(X)\phi_\beta^-(X) = \beta I$.*

(b) *$\phi_\beta^+(-X) = \phi_\beta^-(X)$ and $\phi_\beta^-(-X) = \phi_\beta^+(X)$.*

(c) *For any $\alpha > 0$, there hold $\phi_\beta^+(\alpha X) = \alpha\,\phi_{\beta/\alpha^2}^+(X)$ and $\phi_\beta^-(\alpha X) = \alpha\,\phi_{\beta/\alpha^2}^-(X)$.*

(d) $\phi_\beta^+$ *is continuously differentiable and its derivative* $(\phi_\beta^+)'(X)[H]$ *at* $X$ *for any* $H \in S^n$ *is given by*

$$(\phi_\beta^+)'(X)[H] = Q(\Gamma \circ (Q^\top H Q))Q^\top,$$

*where* $\Gamma \in S^n$ *is defined by*

$$\Gamma_{ij} = \frac{\phi_\beta^+(d_i) + \phi_\beta^+(d_j)}{\sqrt{d_i^2 + 4\beta} + \sqrt{d_j^2 + 4\beta}}, \quad i, j = 1, 2, \ldots, n. \tag{2.5}$$

(e) $(\phi_\beta^+)'(X)[X_1 + X_2] = \phi_\beta^+(X)$, *where* $X_1 = \phi_\beta^+(X)$ *and* $X_2 = \phi_\beta^-(X)$.

*Proof.* The properties (a), (b), and (c) can be verified straightforwardly from the definitions of $\phi_\beta^+$ and $\phi_\beta^-$, while the proofs for (d) and (e) can be found in [54]. ∎

In the following, we derive the Moreau–Yosida regularization and the proximal point mappings of $-\log\det X$ defined on $S_{++}^n$ and the vector $p$-norm $\|\cdot\|_p$ $(1 \le p \le \infty)$ defined on $\mathbb{R}^n$, which will be utilized subsequently in designing our Newton-CG based PPA.

PROPOSITION 2.3. *Let* $\vartheta(X) = -\log\det X$ *be defined on* $S_{++}^n$ *and* $\beta > 0$. *Then it holds that*

$$\pi_\vartheta^\beta(X) = \phi_\beta^+(X) = \arg\min_{Y \in S_{++}^n} \left\{ -\log\det Y + \frac{1}{2\beta}\|Y - X\|^2 \right\}, \quad X \in S^n, \tag{2.6}$$

$$\Phi_\vartheta^\beta(X) = -\log\det \phi_\beta^+(X) + \frac{1}{2\beta}\|\phi_\beta^-(X)\|^2, \quad X \in S^n. \tag{2.7}$$

*Proof.* It is easy to show that for any $X \in S^n$, the unique optimal solution $Y^*$ to (2.6) must satisfy the condition $X = Y^* - \beta(Y^*)^{-1}$. Property (a) in Proposition 2.2 implies that $Y^* = \phi_\beta^+(X)$ satisfies this condition. By plugging $\phi_\beta^+(X)$ into the objective function of (2.6), we can show by using Proposition 2.2 that the Moreau–Yosida regularization of $\vartheta(X) = -\log\det X$, $X \in S_{++}^n$, is given by (2.7). ∎

PROPOSITION 2.4. *Let* $1 \le p \le +\infty$. *The proximal point mapping of* $\vartheta(x) = \|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p} : \mathbb{R}^n \to \mathbb{R}$ *is given by* $\pi_\vartheta^\beta(x) = x - \Pi_{B_q^\beta}(x)$, *i.e.,*

$$\pi_\vartheta^\beta(x) = x - \Pi_{B_q^\beta}(x) = \arg\min_{y \in \mathbb{R}^n} \left\{ \|y\|_p + \frac{1}{2\beta}\|y - x\|^2 \right\}, \quad x \in \mathbb{R}^n, \tag{2.8}$$

*where* $1 \le q \le +\infty$ *satisfies* $\frac{1}{p} + \frac{1}{q} = 1$, $B_q^\beta := \{x \in \mathbb{R}^n : \|x\|_q \le \beta\}$, *and* $\Pi_{B_q^\beta}(\cdot)$ *represents the Euclidean projection onto* $B_q^\beta$.

The proof of Proposition 2.4 can be easily fulfilled by using the famous Moreau's theorem (see, e.g., [46, Theorem 31.5]). A simple proof can also be found in [21].

As regularization functions, in general the $\varphi_\ell$'s in (1.5) are nonsmooth. In the rest of this paper, we make the following assumption on the $\varphi_\ell$'s.

*Assumption* 3. The $\varphi_\ell$'s in (1.5) are given by $\varphi_\ell(\cdot) = w_\ell\|\cdot\|_p$, where $w_\ell > 0$ and $p = 1, 2$ or $\infty$.

We note that in principle the Newton-CG based PPA proposed in this paper is applicable provided that (i) $\varphi_\ell$'s are simple in the sense that their proximal point mappings either have explicit formulas or can be computed efficiently, and (ii) the generalized Jacobian matrices of the proximal point mappings can be evaluated at any given point. Clearly, both projections onto the $\ell_2$-norm and the $\ell_\infty$-norm balls have closed form formulas, and the generalized Jacobian matrices of these projection

mappings can also be analytically represented. On the other hand, it is well known that the projection onto the $\ell_1$-norm ball can be computed efficiently (usually via a sorting according to the magnitudes of the vector components; see, e.g., [17, 21]). Furthermore, the corresponding generalized Jacobian can also be evaluated at any given point. Therefore, all the $\varphi_\ell$'s prescribed in Assumption 3 satisfy the two conditions required by the Newton-CG based PPA. Another example that satisfies the two conditions is the vector $k$-norm defined by $\|x\|_{(k)} := \sum_{i=1}^{k} |x|_i^{\downarrow}$, $x \in \mathbb{R}^n$, where $|x|^{\downarrow}$ is a reordering of $|x|$ (absolute value applied to each component of $x$) such that $|x|_1^{\downarrow} \geq |x|_2^{\downarrow} \geq \ldots \geq |x|_n^{\downarrow}$; see the recent manuscript [56].

**3. A Newton-CG based PPA.** In this section, we propose a Newton-CG based PPA for solving (1.5). The PPA is a classical optimization approach, which goes back to [38] and is extensively studied in [48, 47]. Roughly, suppose we aim to minimize an objective function $f : \mathbb{R}^n \to \mathbb{R}$ and $x^k$ is the current guess of an optimal solution; the PPA generates $x^{k+1}$ via (approximately) solving a perturbed problem of the form

$$\min_x f(x) + \frac{1}{2\beta_k}\|x - x^k\|^2, \tag{3.1}$$

where $\{\beta_k > 0 : k = 1, 2, \ldots\}$ is a sequence of parameters. It is shown in [47] that PPA is closely related to the method of multipliers of Hestenes [26] and Powell [44]. In the following, we reformulate (1.5) and then apply the PPA.

**3.1. The problem reformulation.** For each $\ell$, we let the operation $X \to X_{g_\ell}$ be denoted by $\mathcal{P}_\ell$, i.e., $\mathcal{P}_\ell X = X_{g_\ell}$. To decouple the difficulty caused by the overlapping of variables in the log-likelihood function and the regularization, we introduce for each $\ell$ an auxiliary variable $y_\ell \in \mathbb{R}^{|g_\ell|}$ to take $X_{g_\ell} = \mathcal{P}_\ell X$ out of the function $\varphi_\ell$. Let $s := \sum_{\ell=1}^{r} |g_\ell|$ be the total number of elements of $X$ involved in the regularization. For convenience, we let $\mathcal{P} := [\mathcal{P}_1; \mathcal{P}_2; \ldots; \mathcal{P}_r] : \mathbb{R}^{n\times n} \to \mathbb{R}^s$ and

$$\varphi(y) := \sum_{\ell=1}^{r} \varphi_\ell(y_\ell), \text{ where } y := (y_1; y_2; \ldots; y_r) \in \mathbb{R}^{|g_1|} \times \mathbb{R}^{|g_2|} \times \cdots \times \mathbb{R}^{|g_r|}. \tag{3.2}$$

We note that under Assumption 1, the operator $\mathcal{P}$ is also a surjective mapping. With the above notation, (1.5) can be equivalently transformed to

$$\min_{X,\, y} \ \langle S, X\rangle - \log\det X + \varphi(y) \tag{3.3a}$$

$$\text{s.t. } \mathcal{A}X = b, \tag{3.3b}$$

$$\mathcal{P}X - y = 0, \tag{3.3c}$$

$$X \in S_+^n,\ y \in \mathbb{R}^s. \tag{3.3d}$$

An advantage of introducing the auxiliary variable $y \in \mathbb{R}^s$ is that the objective function in (3.3) is now separable in $X$ and $y$. To apply PPA to (3.3), we need to determine the essential objective function. For this purpose, we let the generalized Lagrange function $\mathcal{L}(X, y, \lambda, \eta) : \mathbb{R}^{n\times n} \times \mathbb{R}^s \times \mathbb{R}^m \times \mathbb{R}^s \to \mathbb{R} \cup \{+\infty\}$ associated with (3.3) be defined by

(3.4)

$$\mathcal{L}(X, y, \lambda, \eta) := \begin{cases} \langle S, X\rangle - \log\det X + \varphi(y) - \lambda^\top(\mathcal{A}X - b) - \eta^\top(\mathcal{P}X - y) & \text{if } X \in \mathcal{S}_{++}^n, \\ +\infty & \text{otherwise.} \end{cases}$$

Clearly, (3.3) is equivalent to

$$\min_{X,y}\{f(X,y) : X \in \mathbb{R}^{n\times n},\, y \in \mathbb{R}^s\}, \tag{3.5}$$

where $f : \mathbb{R}^{n\times n} \times \mathbb{R}^s \to \mathbb{R}$ is the essential objective function of (3.3) defined by

$$f(X,y) := \max\{\mathcal{L}(X,y,\lambda,\eta) : \lambda \in \mathbb{R}^m,\, \eta \in \mathbb{R}^s\}. \tag{3.6}$$

It is easy to show that the dual problem of (3.3) is given by

$$\max_{\lambda,\eta,Z}\left\{b^\top\lambda + \log\det Z - \varphi^*(-\eta) + n : \mathcal{A}^*\lambda + \mathcal{P}^*\eta + Z = S,\, Z \in S^n_+\right\}, \tag{3.7}$$

where $\varphi^*$ denotes the convex conjugate of $\varphi$ (see, e.g., [46]). Under Assumption 3, $\varphi^*$ is actually the indicator function of

$$\mathcal{B} := \{\eta \in \mathbb{R}^s : \|\eta_\ell\|_q \le \omega_\ell, \ell = 1,2,\ldots,r\}, \tag{3.8}$$

where $q$ satisfies $1/p + 1/q = 1$. Therefore, the presence of $-\varphi^*(-\eta)$ in the dual problem (3.7) essentially enforces the ball constraints $\eta \in \mathcal{B}$. The feasible sets of the primal and dual problems (1.5) and (3.7) are, respectively, defined by

$$\mathcal{F}_P = \{X \in S^n_{++} : \mathcal{A}X = b\}, \tag{3.9a}$$

$$\mathcal{F}_D = \{(\lambda,\eta,Z) \in \mathbb{R}^m \times \mathbb{R}^s \times S^n_{++} : \varphi^*(-\eta) < +\infty, \mathcal{A}^*\lambda + \mathcal{P}^*\eta + Z = S\}. \tag{3.9b}$$

Throughout this paper, we make the following assumption on (1.5) and (3.7).

*Assumption* 4. Both the primal and the dual feasible sets $\mathcal{F}_P$ and $\mathcal{F}_D$ are nonempty.

Under Assumption 4, both problems (1.5) and (3.7) have optimal solutions. Furthermore, the optimal solution to (1.5) is unique since the objective function is strictly convex.

In the following, we concentrate on (3.5), to which we apply the PPA. For convenience, we define

$$W_\beta := W_\beta(X,\lambda,\eta) = X - \beta\left(S - \mathcal{A}^*\lambda - \mathcal{P}^*\eta\right), \tag{3.10a}$$

$$z_\beta := z_\beta(y,\eta) = y - \beta\eta. \tag{3.10b}$$

First, we compute the Moreau–Yosida regularization of the essential objective function $f$, which is derived in the following lemma.

LEMMA 3.1. *Let $W_\beta$ and $z_\beta$ be defined in* (3.10) *and $\Phi^\beta_\varphi$ be the Moreau–Yosida regularization of $\varphi$ defined in* (3.2). *Then, the Moreau–Yosida regularization of $f$ defined in* (3.6) *is given by*

$$\Phi^\beta_f(X,y) = \max\{\Theta_\beta(X,y,\lambda,\eta) : \lambda \in \mathbb{R}^m,\, \eta \in \mathbb{R}^s\},$$

*where*

$$\begin{aligned}\Theta_\beta(X,y,\lambda,\eta) := {} & b^\top\lambda - \frac{1}{2\beta}\|\phi^+_\beta(W_\beta)\|^2 + \frac{1}{2\beta}\|X\|^2 - \log\det\phi^+_\beta(W_\beta) + n \\ & - \frac{1}{2\beta}\|z_\beta\|^2 + \frac{1}{2\beta}\|y\|^2 + \Phi^\beta_\varphi(z_\beta).\end{aligned} \tag{3.11}$$

*Proof.* By the definition of the Moreau–Yosida regularization, it holds that

$$\begin{aligned}(3.12)\qquad \Phi_f^\beta(X,y) &= \min_{U,\,v} f(U,v) + \frac{1}{2\beta}\left(\|U-X\|^2 + \|v-y)\|^2\right)\\ &= \min_{U,\,v}\max_{\lambda,\,\eta} \mathcal{L}(U,v,\lambda,\eta) + \frac{1}{2\beta}\left(\|U-X\|^2 + \|v-y\|^2\right)\\ &= \max_{\lambda,\,\eta}\min_{U,\,v} \mathcal{L}(U,v,\lambda,\eta) + \frac{1}{2\beta}\left(\|U-X\|^2 + \|v-y\|^2\right)\\ &= \max\{\Theta_\beta(X,y,\lambda,\eta) : \lambda\in\mathbb{R}^m,\, \eta\in\mathbb{R}^s\},\end{aligned}$$

where the interchange of min and max follows from [46], and

$$\begin{aligned}(3.13)\qquad &\Theta_\beta(X,y,\lambda,\eta)\\ &= \min_{U,\,v} \mathcal{L}(U,v,\lambda,\eta) + \frac{1}{2\beta}\left(\|U-X\|^2 + \|v-y\|^2\right)\\ &= \min_{U,\,v} \langle S,U\rangle - \log\det U + \varphi(v) - \lambda^\top(\mathcal{A}U-b) - \eta^\top(\mathcal{P}U-v)\\ &\quad + \frac{1}{2\beta}\left(\|U-X\|^2 + \|v-y\|^2\right)\\ &= b^\top\lambda - \frac{1}{2\beta}\|W_\beta\|^2 + \frac{1}{2\beta}\|X\|^2 + \min_U\left\{-\log\det U + \frac{1}{2\beta}\|U-W_\beta\|^2\right\}\\ &\quad - \frac{1}{2\beta}\|z_\beta\|^2 + \frac{1}{2\beta}\|y\|^2 + \min_v\left\{\varphi(v) + \frac{1}{2\beta}\|v-z_\beta\|^2\right\}.\end{aligned}$$

From (2.6), the minimization for $U$ in (3.13) is attained at $\phi_\beta^+(W_\beta)$, while the minimization for $v$ is attained at $\pi_\varphi^\beta(z_\beta)$. By using Proposition 2.2 and (2.7), simple computation shows that $\Theta_\beta(X,y,\lambda,\eta)$ can be simplified to the expression given in (3.11). ∎

**3.2. The proposed PPA framework.** In this subsection, we present a PPA framework for solving (3.3) or equivalently (3.5). Given $(X^k,y^k)\in\mathbb{R}^{n\times n}\times\mathbb{R}^s$, according to (3.1), the next iterate $(X^{k+1},y^{k+1})$ generated by PPA satisfies

$$(3.14)\qquad (X^{k+1},y^{k+1}) \approx \pi_f^\beta(X^k,y^k) = \arg\min_{X,y} f(X,y) + \frac{1}{2\beta}\left(\|X-X^k\|^2 + \|y-y^k\|^2\right).$$

For simplicity, here the proximal parameter $\beta$ is assumed to be constant, although it is frequently varying in practice to accelerate convergence. According to (3.12), the saddle point formulation of (3.14) is given by

$$(3.15)\qquad \max_{\lambda,\eta}\min_{X,y} \mathcal{L}(X,y,\lambda,\eta) + \frac{1}{2\beta}\left(\|X-X^k\|^2 + \|y-y^k\|^2\right).$$

From Lemma 3.1, the dual problem of (3.14) is given by

$$(3.16)\qquad \max_{\lambda,\eta}\{\theta_k(\lambda,\eta) := \Theta_\beta(X^k,y^k,\lambda,\eta) : \lambda\in\mathbb{R}^m, \eta\in\mathbb{R}^s\}.$$

Unfortunately, directly solving (3.14) in practice is by no means an easy task. We emphasize that the saddle value of (3.15) exists because the objective function is strictly convex (actually strongly convex) with respect to $(X,y)$; see [46, Theorem 37.3]. Therefore, a feasible way to solve (3.14) is to first solve the dual subproblem

(3.16) for the dual variables and then compute the primal variables $(X, y)$ based on the computed dual ones. Specifically, at each iteration, we first (approximately) solve (3.16) to obtain the dual variables $(\lambda^{k+1}, \eta^{k+1})$ and then update the primal variables $(X^k, y^k)$ via

$$X^{k+1} = \phi_\beta^+(W_\beta(X^k, \lambda^{k+1}, \eta^{k+1})), \tag{3.17a}$$

$$y^{k+1} = \pi_\varphi^\beta(z_\beta(y^k, \eta^{k+1})), \tag{3.17b}$$

because it is implied by the proof of Lemma 3.1 that for fixed $(\lambda, \eta)$, the minimization in (3.15) with respect to $X$ and $y$ is attained at $X = \phi_\beta^+(W_\beta(X^k, \lambda, \eta))$ and $y = \pi_\varphi^\beta(z_\beta(y^k, \eta))$, respectively. Now we are ready to summarize the proposed PPA framework.

ALGORITHM 1. A PPA framework for solving (3.3).

1 Input $S$, $\mathcal{A}$, $b$, $\omega_\ell$'s and $\beta > 0$. Initialize $(X, y) = (X^0, y^0)$ and $k = 0$.
2 **while** *"not converged"* **do**
3 For fixed $(X^k, y^k)$, solve (3.16) approximately to obtain the dual variables $(\lambda^{k+1}, \eta^{k+1})$.
4 Update the primal variables via (3.17) and set $k = k + 1$.

Since in practice (3.16) can be solved only approximately, we will use the following stopping criteria considered by Rockafellar [48, 47] for the theoretical analysis in section 4:

$$\sup \theta_k(\lambda, \eta) - \theta_k(\lambda^{k+1}, \eta^{k+1}) \le \frac{\varepsilon_k^2}{2\beta},\ \varepsilon_k \ge 0, \sum_{k=0}^{\infty} \varepsilon_k < \infty; \tag{3.18a}$$

$$\sup \theta_k(\lambda, \eta) - \theta_k(\lambda^{k+1}, \eta^{k+1}) \le \frac{\delta_k^2}{2\beta} \|(X^{k+1}, y^{k+1}) - (X^k, y^k)\|^2, \tag{3.18b}$$

$$\delta_k \ge 0, \sum_{k=0}^{\infty} \delta_k < \infty;$$

$$\|\nabla \theta_k(\lambda^{k+1}, \eta^{k+1})\| \le \frac{\delta_k'}{\beta} \|(X^{k+1}, y^{k+1}) - (X^k, y^k)\|,\ 0 \le \delta_k' \to 0. \tag{3.18c}$$

Clearly, the main cost per iteration of Algorithm 1 is to solve the dual subproblem (3.16), which requires its own iterations. In the next subsection, we describe a Newton-CG algorithm for solving the dual subproblem (3.16) and introduce practically implementable stopping criteria in place of (3.18a) and (3.18b) by removing the unknown quantity $\sup \theta_k(\lambda, \eta)$.

**3.3. Solve the dual subproblem by a Newton-CG method.** From Proposition 2.1, for any $(X^k, y^k)$, $\theta_k(\lambda, \eta)$ defined in (3.16) is continuously differentiable and concave with respect to $(\lambda, \eta)$. Therefore, solving (3.16) is equivalent to solving the following nonlinear system:

(3.19)

$$F_k(\lambda, \eta) := -\nabla \theta_k(\lambda, \eta) = \begin{bmatrix} \mathcal{A}\phi_\beta^+(W_\beta^k(\lambda, \eta)) - b \\ \mathcal{P}\phi_\beta^+(W_\beta^k(\lambda, \eta)) - \pi_\varphi^\beta(z_\beta^k(\eta)) \end{bmatrix} = 0,\ (\lambda, \eta) \in \mathbb{R}^m \times \mathbb{R}^s,$$

where $W_\beta^k(\lambda,\eta) = X^k + \beta(\mathcal{A}^*\lambda + \mathcal{P}^*\eta - S)$ and $z_\beta^k(\eta) = y^k - \beta\eta$. However, due to the nonsmoothness of the projection mappings onto the $\ell_p$-norm balls (hidden in the proximal point mapping $\pi_\varphi^\beta$), $\theta_k(\lambda,\eta)$ is not twice continuously differentiable. From Proposition 2.1, $\pi_\varphi^\beta$ is globally Lipschitz continuous. Therefore, according to Rademacher's theorem, $\pi_\varphi^\beta$ is almost everywhere Fréchet differentiable in the whole space. Let $\mathcal{D}_{\pi_\varphi^\beta}$ be the set of points where $\pi_\varphi^\beta$ is differentiable, and

$$\partial_B \pi_\varphi^\beta(z) := \left\{ V : V = \lim_{k\to\infty} (\pi_\varphi^\beta)'(z^k), z^k \to z, z^k \in \mathcal{D}_{\pi_\varphi^\beta} \right\},\ z \in \mathbb{R}^s.$$

The set of generalized Jacobian matrices (see, e.g., [9]) of $\pi_\varphi^\beta$ is defined by

$$\partial \pi_\varphi^\beta(z) := \mathrm{conv}\{\partial_B \pi_\varphi^\beta(z)\},\ z \in \mathbb{R}^s, \tag{3.20}$$

where conv denotes the convex hull. Fortunately, the projection mappings onto the $\ell_p$-norm balls ($p = 1, 2, \infty$) (and thus, from Proposition 2.4, the proximal point mappings) are semismooth. That is, for any fixed $\bar{z}$, $\pi_\varphi^\beta$ is directional differentiable and, for any $V \in \partial\pi_\varphi^\beta(z)$, it holds that

$$\pi_\varphi^\beta(z) - \pi_\varphi^\beta(\bar{z}) - V(z - \bar{z}) = o(\|z - \bar{z}\|) \text{ as } z \to \bar{z}.$$

As a result, (3.19) is a semismooth equation and the generalized Newton's method developed in [30, 45] for solving semismooth equations can be applied. In our implementation, we solved (3.19) by the inexact generalized Newton's method described in Algorithm 2.

ALGORITHM 2. A Newton-CG algorithm for solving (3.16).

Given $\mu \in (0, 0.5)$ and $c, \delta \in (0, 1)$. Choose $(\lambda^{k,0}, \eta^{k,0})$ and let $j = 0$.
**while** *"not converged"* **do**
  Apply an iterative algorithm to solve

$$V^{k,j}(d_\lambda; d_\eta) = -F_k(\lambda^{k,j}, \eta^{k,j}) \tag{3.21}$$

  to obtain $d_\lambda^j$ and $d_\eta^j$, where $V^{k,j} \in \partial F_k(\lambda^{k,j}, \eta^{k,j})$.
  Set $\alpha_j = \delta^{m_j}$, where $m_j$ is the first nonnegative integer $m$ for which

$$\theta_k(\lambda^{k,j} + \delta^m d_\lambda^j, \eta^{k,j} + \delta^m d_\eta^j) \ge \theta_k(\lambda^{k,j}, \eta^{k,j}) - \mu\delta^m \langle F_k(\lambda^{k,j}, \eta^{k,j}), (d_\lambda^j; d_\eta^j)\rangle, \tag{3.22a}$$

$$\|\nabla\theta_k(\lambda^{k,j} + \delta^m d_\lambda^j, \eta^{k,j} + \delta^m d_\eta^j)\| \le c\|\nabla\theta_k(\lambda^{k,j}, \eta^{k,j})\|. \tag{3.22b}$$

  Set $\lambda^{k,j+1} = \lambda^{k,j} + \alpha_j d_\lambda^j$, $\eta^{k,j+1} = \eta^{k,j} + \alpha_j d_\eta^j$.
  If converged, set $\lambda^{k+1} = \lambda^{k,j+1}$, $\eta^{k+1} = \eta^{k,j+1}$ and break; otherwise set $j = j + 1$.

Simple computation shows that the set of generalized Jacobian matrices $\partial F_k(\lambda^{k,j}, \eta^{k,j})$ has the form

$$\partial F_k(\lambda^{k,j}, \eta^{k,j}) = \left\{ \beta \begin{bmatrix} \mathcal{A} \\ \mathcal{P} \end{bmatrix} (\phi_\beta^+)'(W_\beta^k) \begin{bmatrix} \mathcal{A} \\ \mathcal{P} \end{bmatrix}^* + \beta \begin{pmatrix} 0 & 0 \\ 0 & J \end{pmatrix} : J \in \partial\pi_\varphi^\beta(z_\beta^k) \right\},$$

where $W_\beta^k = X^k + \beta(\mathcal{A}^*\lambda^{k,j} + \mathcal{P}^*\eta^{k,j} - S)$ and $z_\beta^k = y^k - \beta\eta^{k,j}$. Clearly, all the elements in $\partial F_k(\lambda^{k,j}, \eta^{k,j})$ are positive semidefinite.

In practical implementation, a proximal term $-\frac{\epsilon}{2}(\|\lambda-\lambda^k\|^2+\|\eta-\eta^k\|^2)$ can always be added to the dual objective function $\theta_k(\lambda,\eta)$. In fact, this corresponds to the PPA of multipliers considered in [47, section 5]. Convergence analysis of this improvement can be conducted in a parallel way as for Algorithm 1. The benefits of adding a proximal term to $\theta_k(\lambda,\eta)$ are twofold. First, it provides a feasible way of terminating Algorithm 2. In fact, the function $\hat{\theta}_k(\lambda,\eta) := \theta_k(\lambda,\eta) - \frac{\epsilon}{2}(\|\lambda-\lambda^k\|^2 + \|\eta-\eta^k\|^2)$ is strongly concave with modulus $\epsilon$, and thus the following estimation holds:

$$\sup \hat{\theta}_k(\lambda,\eta) - \hat{\theta}_k(\lambda^{k+1},\eta^{k+1}) \le \frac{1}{2\epsilon}\|\nabla\hat{\theta}_k(\lambda^{k+1},\eta^{k+1})\|^2.$$

Therefore, the stopping criteria (3.18a) and (3.18b) can be practically modified to

$$\|\nabla\hat{\theta}_k(\lambda^{k+1},\eta^{k+1})\| \le \sqrt{\epsilon/\beta}\cdot\varepsilon_k,\ \varepsilon_k \ge 0, \sum_{k=0}^{\infty}\varepsilon_k < \infty; \tag{3.23a}$$

$$\|\nabla\hat{\theta}_k(\lambda^{k+1},\eta^{k+1})\| \le \sqrt{\epsilon/\beta}\cdot\delta_k\|(X^{k+1},y^{k+1}) - (X^k,y^k)\|, \quad \delta_k \ge 0, \sum_{k=0}^{\infty}\delta_k < \infty. \tag{3.23b}$$

Note that the unknown value $\sup\theta_k(\lambda,\eta)$ has been removed, and thus the criteria (3.23a) and (3.23b) are practically implementable. Furthermore, adding this proximal term to $\theta_k(\lambda,\eta)$ changes the coefficient matrix in (3.21) to $V^{k,j} + \epsilon I$, which is clearly positive definite. Therefore, the corresponding linear system can be practically solved by a preconditioned conjugate gradient (PCG) method. In practical implementation, we always choose the parameter $\epsilon$ adaptively to accelerate convergence of inner iterations.

**3.4. Acceleration by a generalized Newton's method.** For given $X$ and $y$, we let $(\lambda(X,y),\eta(X,y)) \in \arg\max_{\lambda,\eta}\Theta_\beta(X,y,\lambda,\eta)$. According to (i) of Proposition 2.1, it holds that

$$\nabla\Phi_f^\beta(X,y) = \begin{bmatrix}\nabla_X\Phi_f^\beta(X,y)\\ \nabla_y\Phi_f^\beta(X,y)\end{bmatrix} = \frac{1}{\beta}\left\{\begin{bmatrix}X\\ y\end{bmatrix} - \begin{bmatrix}\phi_\beta^+(W)\\ \pi_\varphi^\beta(z)\end{bmatrix}\right\},$$

where

$$W := W(X,y,\lambda(X,y),\eta(X,y)) = X + \beta(\mathcal{A}^*\lambda(X,y) + \mathcal{P}^*\eta(X,y) - S), \tag{3.24a}$$

$$z := z(y,\eta(X,y)) = y - \beta\eta(X,y). \tag{3.24b}$$

Therefore, the PPA framework presented in Algorithm 1, which iterates $X$ and $y$ by (3.17), is equivalent to a gradient descent method applied to $\Phi_f^\beta(X,y)$. To accelerate convergence, we propose to use the generalized Newton's method for solving semismooth equations when the iterate is close to the solution. Let $\mathcal{O}$ be a set of operators and $u$ be an element in the domain of the operators in $\mathcal{O}$. With a slight abuse of notation, in the following we let $\mathcal{O}u := \{h(u) : h \in \mathcal{O}\}$ and $v + \mathcal{O}u := \{v + h(u) : h \in \mathcal{O}\}$. Since it is difficult to express $\partial^2\Phi_f^\beta(X,y) := \partial\nabla\Phi_f^\beta(X,y)$ exactly, we define the following alternative for $\partial^2\Phi_f^\beta(X,y)$ just as in [66]:

(3.25)
$$\hat{\partial}^2\Phi_f^\beta(X,y)\begin{bmatrix}D\\ d\end{bmatrix} := \left\{\frac{1}{\beta}\begin{bmatrix}D\\ d\end{bmatrix} - \frac{1}{\beta}\begin{bmatrix}(\phi_\beta^+)'(W)\left(\frac{\partial W}{\partial X}[D]+\frac{\partial W}{\partial y}[d]\right)\\ h\left(\frac{\partial z}{\partial X}[D]+\frac{\partial z}{\partial y}[d]\right)\end{bmatrix} : h\in\partial\pi_\varphi^\beta(z)\right\},$$

where $(D,d)\in\mathbb{R}^{n\times n}\times\mathbb{R}^s$, $\partial\pi_\varphi^\beta(z)$ denotes the set of generalized Jacobian matrices of $\pi_\varphi^\beta(z)$, and

$$\frac{\partial W}{\partial X}[D] = D+\beta\mathcal{A}^*\frac{\partial\lambda(X,y)}{\partial X}[D]+\beta\mathcal{P}^*\frac{\partial\eta(X,y)}{\partial X}[D], \tag{3.26a}$$

$$\frac{\partial W}{\partial y}[d] = \beta\mathcal{A}^*\frac{\partial\lambda(X,y)}{\partial y}[d]+\beta\mathcal{P}^*\frac{\partial\eta(X,y)}{\partial y}[d], \tag{3.26b}$$

$$\frac{\partial z}{\partial X}[D] = -\beta\frac{\partial\eta(X,y)}{\partial X}[D], \tag{3.26c}$$

$$\frac{\partial z}{\partial y}[d] = d-\beta\frac{\partial\eta(X,y)}{\partial y}[d]. \tag{3.26d}$$

It follows from [9, p. 75] that $\partial^2\Phi_f^\beta(X,y)[D,d]\subseteq\hat{\partial}^2\Phi_f^\beta(X,y)[D,d]$ for any $(D,d)\in\mathbb{R}^{n\times n}\times\mathbb{R}^s$. When the iterate is close to an optimal solution, we take a generalized Newton's step, i.e.,

$$(X^{k+1},y^{k+1}) = (X^k,y^k)+(D^k,d^k), \tag{3.27}$$

where $(D^k,d^k)$ is the solution to the Newton system

$$V^k[D,d] = -\nabla\Phi_f^\beta(X^k,y^k)\ \text{ for some } V^k\in\hat{\partial}^2\Phi_f^\beta(X^k,y^k). \tag{3.28}$$

Let $W$ and $z$ be defined as in (3.24). Then, $\mathcal{A}\phi_\beta^+(W)$ and $\mathcal{P}\phi_\beta^+(W)-\pi_\varphi^\beta(z)$ are mappings of $(X,y)$ from $S^n\times\mathbb{R}^s$ to $\mathbb{R}^m$ and $\mathbb{R}^s$, respectively, and so are their derivatives. By computing the derivatives to $\mathcal{A}\phi_\beta^+(W)-b=0$ and $\mathcal{P}\phi_\beta^+(W)-\pi_\varphi^\beta(z)=0$ on both sides (for $\pi_\varphi^\beta(z)$, compute its generalized Jacobians) with respect to $(X,y)$ and applying the resulting mappings onto $(D,d)\in S^n\times\mathbb{R}^s$, we obtain

$$\mathcal{A}(\phi_\beta^+)'(W)\left[\frac{\partial W}{\partial X}[D]+\frac{\partial W}{\partial y}[d]\right] = 0, \tag{3.29a}$$

$$\mathcal{P}(\phi_\beta^+)'(W)\left[\frac{\partial W}{\partial X}[D]+\frac{\partial W}{\partial y}[d]\right]-\partial\pi_\varphi^\beta(z)\left[\frac{\partial z}{\partial X}[D]+\frac{\partial z}{\partial y}[d]\right] = 0. \tag{3.29b}$$

By plugging (3.26) into (3.29) and with simple manipulation, we obtain

$$\beta\left\{\begin{bmatrix}\mathcal{A}\\ \mathcal{P}\end{bmatrix}(\phi_\beta^+)'(W)\begin{bmatrix}\mathcal{A}\\ \mathcal{P}\end{bmatrix}^*+\begin{bmatrix}0 & 0\\ 0 & \partial\pi_\varphi^\beta(z)\end{bmatrix}\right\}\begin{bmatrix}\lambda_X'+\lambda_y'\\ \eta_X'+\eta_y'\end{bmatrix} = \begin{bmatrix}-\mathcal{A}(\phi_\beta^+)'(W)[D]\\ -\mathcal{P}(\phi_\beta^+)'(W)[D]+\partial\pi_\varphi^\beta(z)[d]\end{bmatrix}.$$

Here $\lambda_X'$ and $\lambda_y'$ stand for $\partial\lambda(X,y)/\partial X[D]$ and $\partial\lambda(X,y)/\partial y[d]$, respectively, and similarly for $\eta_X'$ and $\eta_y'$. Note that the coefficient matrix of the above linear system is identical to that in (3.21). Therefore, $\lambda_X'+\lambda_y'$ and $\eta_X'+\eta_y'$, and thus $\frac{\partial W}{\partial X}[D]+\frac{\partial W}{\partial y}[d]$ and $\frac{\partial z}{\partial X}[D]+\frac{\partial z}{\partial y}[d]$ from (3.26), can be computed by solving the above linear system.

Since $\partial\pi_\varphi^\beta$ can be explicitly computed for the $\varphi_\ell$'s prescribed in Assumption 3, (3.25) implies that for given $V \in \hat{\partial}^2\Phi_f^\beta(X,y)$ and $[D,d]$ in its domain, their multiplication can be computed. As a result, the Newton system (3.28) can be solved by a Krylov subspace method such as the CG method, which depends merely on this "matrix-vector" multiplication.

**4. Theoretical results.** In this section, we present some theoretical results of the proposed PPA. Let $(\bar{X},\bar{y})$ be the unique optimal solution of (3.3) and $(\bar{\lambda},\bar{\eta})$ be the corresponding unique multipliers. We note that the uniqueness of the optimal solution to the dual problem (3.7) (or multipliers) depends on a primal constraint nondegeneracy, as will be stated in (4.8). For $\lambda \in \mathbb{R}^m$ and $\eta \in \mathbb{R}^s$, we define

$$F(\lambda,\eta) := \begin{bmatrix} \mathcal{A}\phi_\beta^+(\bar{X}+\beta(\mathcal{A}^*\lambda+\mathcal{P}^*\eta-S))-b \\ \mathcal{P}\phi_\beta^+(\bar{X}+\beta(\mathcal{A}^*\lambda+\mathcal{P}^*\eta-S))-\pi_\varphi^\beta(\bar{y}-\beta\eta) \end{bmatrix}. \tag{4.1}$$

The efficiency of the Newton-CG method for solving the inner subproblems depends on the positive definiteness of the generalized Hessian matrices of the dual objective function, an important property for the effectiveness of applying an iterative solver, such as CG, to the generalized Newton equation (3.21). We will show in subsection 4.1 that the primal constraint nondegeneracy condition for (3.3) (with a slight reformulation) at $(\bar{X},\bar{y})$ is equivalent to the nonsingularity (positive definiteness) of the set of generalized Jacobian matrices $\partial F(\bar{\lambda},\bar{\eta})$. We note that the Newton-CG method cannot be guaranteed to be efficient if the constraint nondegeneracy condition for (3.3) does not hold. Thus, for an effective and stable implementation, a proximal term can be added when solving the dual subproblem (3.16). We also present in subsection 4.2 global and local convergence results of Algorithm 1 based on the classical results in [48, 47].

**4.1. Constraint nondegeneracy and the positive definiteness of $\partial F(\bar{\lambda},\bar{\eta})$.** Recall that under Assumption 3 the regularization function $\varphi$ has the form $\varphi(y) = \sum_{\ell=1}^r \omega_\ell \|y_\ell\|_p$, $y \in \mathbb{R}^s$, where $p = 1, 2$, or $\infty$. By introducing an auxiliary variable $t \in \mathbb{R}$, (3.3) is clearly equivalent to

$$\min_{X,\,y,t} \ \langle S, X\rangle - \log\det X + t \tag{4.2a}$$

$$\text{s.t.} \ \ \mathcal{A}X = b, \tag{4.2b}$$

$$\mathcal{P}X - y = 0, \tag{4.2c}$$

$$(X,y,t) \in S_+^n \times K_p, \tag{4.2d}$$

where $K_p$, $p = 1, 2, \infty$, is a closed convex cone defined by

$$K_p := \left\{(y,t) \in \mathbb{R}^s \times \mathbb{R} : \varphi(y) = \sum\nolimits_{\ell=1}^r \omega_\ell \|y_\ell\|_p \le t\right\}. \tag{4.3}$$

The constraint nondegeneracy condition of (4.2) at $(\bar{X},\bar{y},\bar{t})$ (here $\bar{t} = \varphi(\bar{y})$ since the constraint $(y,t) \in K_p$ must be active at the optimal solution) is

$$\begin{pmatrix} \mathcal{A} & 0 & 0 \\ \mathcal{P} & -I & 0 \end{pmatrix} \begin{pmatrix} \mathrm{lin}(\mathcal{T}_{S_+^n}(\bar{X})) \\ \mathrm{lin}(\mathcal{T}_{K_p}(\bar{y},\bar{t})) \end{pmatrix} = \begin{pmatrix} \mathbb{R}^m \\ \mathbb{R}^s \end{pmatrix}, \tag{4.4}$$

where, for a set $C$ and $v \in C$, $\mathcal{T}_C(v)$ denotes the tangent cone of $C$ at $v$, and lin represents the linearity space of a closed convex cone (the biggest linear space contained in the cone). Since $\bar{X}$ is positive definite, it follows that $\mathcal{T}_{S_+^n}(\bar{X}) = S^n$, and thus (4.4) is equivalent to

$$\begin{pmatrix} \mathcal{A} & 0 & 0 \\ \mathcal{P} & -I & 0 \end{pmatrix} \begin{pmatrix} S^n \\ \lin\left(\mathcal{T}_{K_p}(\bar{y},\bar{t})\right) \end{pmatrix} = \begin{pmatrix} \mathbb{R}^m \\ \mathbb{R}^s \end{pmatrix}. \tag{4.5}$$

For any $(y,t) \in \mathbb{R}^s \times \mathbb{R}$, the tangent cone of $K_p$ at $(\bar{y},\bar{t})$ is given by

$$\mathcal{T}_{K_p}(y,t) = \begin{cases} \mathbb{R}^{s+1} & \text{if } (y,t) \in \operatorname{int}(K_p), \\ K_p & \text{if } (y,t) = 0, \\ \{(d,\alpha) \in \mathbb{R}^s \times \mathbb{R} : \phi'((y,t);(d,\alpha)) \le 0\} & \text{if } (y,t) \in \partial K_p \setminus \{0\}, \end{cases} \tag{4.6}$$

where $\phi(y,t) := \varphi(y) - t$; see, e.g., [9, Theorem 2.4.7]. In the following, we give a detailed analysis for the case $p = 2$. As we will explain at the end of this subsection, the analyses for $p = 1$ and $p = \infty$ are similar.

For $p = 2$, direct calculation shows that

$$\mathcal{T}_{K_2}(y,t) = \left\{ (d,\alpha) \in \mathbb{R}^s \times \mathbb{R} : \sum_{\ell:\, y_\ell \neq 0} \frac{y_\ell^\top d_\ell}{\|y_\ell\|} + \sum_{\ell:\, y_\ell = 0} \|d_\ell\| \le \alpha \right\} \quad \forall\, (y,t) \in \partial K_2 \setminus \{0\}.$$

Thus, it follows from $\lin(\mathcal{T}_{K_2}(y,t)) = \mathcal{T}_{K_2}(y,t) \cap -\mathcal{T}_{K_2}(y,t)$ that

$$\lin(\mathcal{T}_{K_2}(y,t)) = \left\{ (d,\alpha) \in \mathbb{R}^s \times \mathbb{R} : \alpha = \sum_{\ell:\, y_\ell \neq 0} \frac{y_\ell^\top d_\ell}{\|y_\ell\|},\ d_\ell = 0 \text{ if } y_\ell = 0 \right\} \quad \forall\, (y,t) \in \partial K_2 \setminus \{0\}. \tag{4.7}$$

Since at the optimal solution the constraint $\varphi(y) \le t$ must be active, i.e., $(\bar{y},\bar{t}) \notin \operatorname{int}(K_2)$, it follows from (4.7) that (4.5) can be simplified to

$$\begin{pmatrix} \mathcal{A} \\ \mathcal{P} \end{pmatrix} S^n + \begin{pmatrix} 0 \\ \mathcal{V} \end{pmatrix} = \begin{pmatrix} \mathbb{R}^m \\ \mathbb{R}^s \end{pmatrix}, \tag{4.8}$$

where

$$\begin{aligned} \mathcal{V} &:= \{d : (d,\alpha) \in \lin(\mathcal{T}_{K_2}(\bar{y},\bar{t})) \text{ for some } \alpha\} \\ &= H_1 \times \ldots \times H_r,\ H_\ell = \begin{cases} \mathbb{R}^{|\bar{y}_\ell|} & \text{if } \bar{y}_\ell \neq 0, \\ \{\mathbf{0}_{|\bar{y}_\ell|}\} & \text{otherwise.} \end{cases} \end{aligned} \tag{4.9}$$

From Proposition 2.4, it holds for $p = 2$ that

$$\pi_\varphi^\beta(z) = z - \Pi_{B_2^{\beta w}}(z),$$

where $B_2^{\beta w} = B_2^{\beta w_1} \times \cdots \times B_2^{\beta w_r}$ and $B_2^{\beta w_\ell} := \{v \in \mathbb{R}^{|z_\ell|} : \|v\| \le \beta w_\ell\}$ for each $\ell$. Clearly, we have the following formulas for projection onto an $\ell_2$-norm ball and its generalized Jacobian matrices:

$$\Pi_{B_2^w}(v) = \begin{cases} w\frac{v}{\|v\|} & \text{if } \|v\| > w, \\ v & \text{otherwise,} \end{cases} \quad \text{and} \tag{4.10}$$

$$\partial\Pi_{B_2^w}(v) = \begin{cases} \frac{w}{\|v\|}\left(I - \frac{vv^\top}{\|v\|^2}\right) & \text{if } \|v\| > w, \\ \left\{ I - t\frac{vv^\top}{w^2} : 0 \le t \le 1 \right\} & \text{if } \|v\| = w, \\ I & \text{if } \|v\| < w. \end{cases}$$

The next theorem establishes the equivalence of the positive definiteness of the set of generalized Jacobian matrices $\partial F(\bar{\lambda},\bar{\eta})$ and the primal constraint nondegeneracy condition (4.8).

THEOREM 4.1. *Assume $p = 2$. The set of generalized Jacobian matrices $\partial F(\bar{\lambda}, \bar{\eta})$ are all positive definite if and only if the constraint nondegeneracy* (4.8) *holds.*

*Proof.* First, we show that the constraint nondegeneracy condition (4.8) implies the positive definiteness of all members in $\partial F(\bar{\lambda}, \bar{\eta})$. Let $\bar{W} := \bar{X} + \beta(\mathcal{A}^*\bar{\lambda} + \mathcal{P}^*\bar{\eta} - S)$ and $\bar{z} := \bar{y} - \beta\bar{\eta}$. Then, the set of generalized Jacobian matrices of $F$ at $(\bar{\lambda}, \bar{\eta})$ (upon a scaling of $1/\beta$) are given by

$$\partial F(\bar{\lambda}, \bar{\eta}) = \left\{ \begin{bmatrix} \mathcal{A}(\phi_\beta^+)'(\bar{W})\mathcal{A}^* & \mathcal{A}(\phi_\beta^+)'(\bar{W})\mathcal{P}^* \\ \mathcal{P}(\phi_\beta^+)'(\bar{W})\mathcal{A}^* & \mathcal{P}(\phi_\beta^+)'(\bar{W})\mathcal{P}^* + V \end{bmatrix} : V \in \partial\pi_\varphi^\beta(\bar{z}) \right\}, \tag{4.11}$$

where $\partial\pi_\varphi^\beta(\cdot)$ is defined in (3.20). Let $d = (d_1, d_2) \in \mathbb{R}^m \times \mathbb{R}^s$ and $Jd = 0$ for some $J \in \partial F(\bar{\lambda}, \bar{\eta})$. Then,

$$\begin{aligned} 0 = \langle d, Jd\rangle &= \langle \mathcal{A}^*d_1 + \mathcal{P}^*d_2, (\phi_\beta^+)'(\bar{W})(\mathcal{A}^*d_1 + \mathcal{P}^*d_2)\rangle + \langle d_2, Vd_2\rangle \\ &\geq \langle \mathcal{A}^*d_1 + \mathcal{P}^*d_2, (\phi_\beta^+)'(\bar{W})(\mathcal{A}^*d_1 + \mathcal{P}^*d_2)\rangle + \langle Vd_2, Vd_2\rangle \geq 0 \end{aligned} \tag{4.12}$$

for some $V \in \partial\pi_\varphi^\beta(\bar{z})$, where the first $\geq$ holds because all eigenvalues of $V$ are less than or equal to one (this is clear from (4.10)). Clearly (4.12) implies that $\mathcal{A}^*d_1 + \mathcal{P}^*d_2 = 0$ and $Vd_2 = 0$. Next we show that $d_2 \in \mathcal{V}^\perp$, where $\mathcal{V}$ is defined in (4.9). For any $v \in \mathcal{V}$, i.e., $(v, \alpha) \in \mathrm{lin}(\mathcal{T}_{K_2}(\bar{y}, \bar{t}))$ ($\bar{t} = \varphi(\bar{y})$) for some $\alpha \in \mathbb{R}$. If $\bar{y} = 0$, then (4.9) implies that $H_\ell \equiv \{\mathbf{0}\}$. Thus, $\mathcal{V} = H_1 \times \ldots \times H_r = \{\mathbf{0}\}$ and $d_2 \in \mathcal{V}^\perp$. Otherwise, it holds that $\langle d_2, v\rangle = \sum_{\ell:\, \bar{y}_\ell \neq 0} \langle (d_2)_\ell, v_\ell\rangle$ (since $H_\ell = \{\mathbf{0}_{|\bar{y}_\ell|}\}$ for all $\ell$ such that $\bar{y}_\ell = 0$), which is equal to 0 if we can show that $(d_2)_\ell = 0$ for all $\ell$ such that $\bar{y}_\ell \neq 0$. It follows from $\bar{y} = \pi_\varphi^\beta(\bar{z}) = \pi_\varphi^\beta(\bar{y} - \beta\bar{\eta})$ that $-\beta\bar{\eta}_\ell = \Pi_{B_2^{\beta w_\ell}}(-\beta\bar{\eta}_\ell + \bar{y}_\ell)$ for each $\ell$. Therefore, for those $\ell$ such that $\bar{y}_\ell \neq 0$, it holds that $\|\bar{\eta}_\ell\| = w_\ell$, and there exist $\delta_\ell > 0$ such that $\delta_\ell \bar{y}_\ell = -\beta\bar{\eta}_\ell$. This implies that if $\bar{y}_\ell \neq 0$, then $\|\bar{y}_\ell - \beta\bar{\eta}_\ell\| > \beta w_\ell$ and thus

$$\partial\Pi_{B_2^{\beta w_\ell}}(\bar{z}_\ell) = \left\{ \frac{\beta w_\ell}{\|\bar{z}_\ell\|} \left( I - \frac{\bar{z}_\ell \bar{z}_\ell^\top}{\|\bar{z}_\ell\|^2} \right) \right\}.$$

Here $\bar{z}_\ell = \bar{y}_\ell - \beta\bar{\eta}_\ell$. In this case, it is easy to see that

$$I - \frac{\beta w_\ell}{\|\bar{z}_\ell\|} \left( I - \frac{\bar{z}_\ell \bar{z}_\ell^\top}{\|\bar{z}_\ell\|^2} \right) \succ 0.$$

Note that $V \in \partial\pi_\varphi^\beta(\bar{z})$ implies that $V = I - U$ for some $U \in \partial\Pi_{B_2^{\beta w}}(\bar{z})$ or, equivalently, $V_\ell = I - U_\ell$ for some $U_\ell \in \partial\Pi_{B_2^{\beta w_\ell}}(\bar{z}_\ell)$ for all $\ell$. Therefore, $Vd_2 = 0$ implies that $(d_2)_\ell = 0$ for all $\ell$ such that $\bar{y}_\ell \neq 0$. In summary, we have proved $d_2 \in \mathcal{V}^\perp$. Next we show that $d = (d_1, d_2) = 0$. From (4.8), there exist $X \in S^n$ and $\hat{d} \in \mathcal{V}$ such that $\mathcal{A}X = d_1$ and $\mathcal{P}X + \hat{d} = d_2$. It thus follows from $\mathcal{A}^*d_1 + \mathcal{P}^*d_2 = 0$, $d_2 \in \mathcal{V}^\perp$, and $\hat{d} \in \mathcal{V}$ that

$$\langle d, d\rangle = \left\langle d, \begin{bmatrix} \mathcal{A} & 0 \\ \mathcal{P} & I \end{bmatrix} \begin{bmatrix} X \\ \hat{d} \end{bmatrix} \right\rangle = \left\langle \begin{bmatrix} \mathcal{A}^*d_1 + \mathcal{P}^*d_2 \\ d_2 \end{bmatrix}, \begin{bmatrix} X \\ \hat{d} \end{bmatrix} \right\rangle = \langle d_2, \hat{d}\rangle = 0, \tag{4.13}$$

which implies $d = 0$. Therefore, we have proved that the set of generalized Jacobian matrices $\partial F(\bar{\lambda}, \bar{\eta})$ defined in (4.11) are all positive definite under the constraint nondegeneracy condition (4.8).

Now we assume that the set of generalized Jacobian matrices $\partial F(\bar{\lambda}, \bar{\eta})$ are all positive definite. Let

$$\mathcal{B} = \begin{bmatrix} \mathcal{A} & 0 \\ \mathcal{P} & -I \end{bmatrix} \text{ and } \mathcal{X} = \begin{bmatrix} S^n \\ \mathcal{V} \end{bmatrix}.$$

We will show that the constraint nondegeneracy condition (4.8) holds by contradiction. Suppose otherwise; then there exists $d = (d_1, d_2) \neq 0$ such that $d \in (\mathcal{B}\mathcal{X})^\perp$. Thus, it holds that $\langle d, \mathcal{B}U\rangle = \langle \mathcal{B}^* d, U\rangle = 0$ for any $U \in \mathcal{X}$. Clearly, this is equivalent to $\langle \mathcal{A}^* d_1 + \mathcal{P}^* d_2, X\rangle = 0$ for all $X \in S^n$ and $\langle d_2, v\rangle = 0$ for all $v \in \mathcal{V}$. Thus, $\mathcal{A}^* d_1 + \mathcal{P}^* d_2 = 0$ and $d_2 \in \mathcal{V}^\perp$, where from (4.9) $\mathcal{V}^\perp$ is given by

$$\mathcal{V}^\perp = H_1^\perp \times \ldots \times H_r^\perp,\ H_\ell^\perp = \begin{cases} \{\mathbf{0}_{|\bar{y}_\ell|}\} & \text{if } \bar{y}_\ell \neq 0, \\ \mathbb{R}^{|\bar{y}_\ell|} & \text{otherwise,} \end{cases}$$

which implies that $(d_2)_\ell = 0$ if $\bar{y}_\ell \neq 0$. For $\ell$ such that $\bar{y}_\ell = 0$, it follows from $\bar{y}_\ell = \pi_\varphi^\beta(\bar{y}_\ell - \beta\bar{\eta}_\ell)$ that $\Pi_{B_2^{\beta w_\ell}}(-\beta\bar{\eta}_\ell) = -\beta\bar{\eta}_\ell$ and thus $\|\bar{y}_\ell - \beta\bar{\eta}_\ell\| \leq \beta w_\ell$. Therefore, for any $V \in \partial\pi_\varphi^\beta(\bar{z})$, it holds that

$$\langle d_2, V d_2\rangle = \sum_{\ell:\, \bar{y}_\ell = 0} \langle (d_2)_\ell, (I - U_\ell)(d_2)_\ell\rangle,$$

where $U_\ell = I$ if $\|\beta\bar{\eta}_\ell\| < \beta w_\ell$ and $U_\ell \in \{I - t\bar{z}\bar{z}^\top/(\beta w_\ell)^2 : 0 \leq t \leq 1\}$ if $\|\beta\bar{\eta}_\ell\| = \beta w_\ell$. By taking $U_\ell \equiv I$ for all $\ell$ such that $\bar{y}_\ell = 0$, we obtain $\langle d_2, V d_2\rangle = 0$. Therefore, we have found a vector $d = (d_1, d_2) \neq 0$ and a member $\mathcal{M} \in \partial F(\bar{\lambda}, \bar{\eta})$ such that $\langle d, \mathcal{M}d\rangle = 0$, which contradicts the fact that all the members in $\partial F(\bar{\lambda}, \bar{\eta})$ are positive definite. Thus, the constraint nondegeneracy condition (4.8) must hold. □

Results similar to Theorem 4.1 can be established for $p = 1$ and $p = \infty$. The analysis for the case $p = 1$ follows directly from that for $p = 2$ because the $\ell_1$-norm is componentwise separable and the absolute value is essentially the only norm in $\mathbb{R}$. The analysis for the case $p = \infty$ is also analogous to that for $p = 2$, except that the argument is more tedious in notation because the projection mapping onto the $\ell_1$-norm ball and its generalized Jacobian matrices require an ordering of the variable components according to their magnitudes. Due to this similarity, we omit the analysis for these two cases and merely present in the following the explicit representations of the linearity spaces of $K_1$ and $K_\infty$, which are the key of the proofs.

From (4.6), we only need to concentrate on the case $(y, t) \in \partial K_p \setminus \{0\}$. With a slight abuse of notation, for the case $p = 1$ we temporarily let $v_i$ be the $i$th component of a vector $v$ (different from the notation $y_\ell$, which represents the $\ell$th block of $y$). Direct calculation shows that

$$\mathcal{T}_{K_1}(y,t) = \left\{ (d,\alpha) \in \mathbb{R}^s \times \mathbb{R} : \sum_{1\leq i\leq s,\, y_i \neq 0} \operatorname{sgn}(y_i) d_i + \sum_{1\leq i\leq s,\, y_i = 0} |d_i| \leq \alpha \right\}$$
$$\forall\, (y,t) \in \partial K_1 \setminus \{0\}.$$

Thus, it follows from $\operatorname{lin}(\mathcal{T}_{K_1}(y,t)) = \mathcal{T}_{K_1}(y,t) \cap -\mathcal{T}_{K_1}(y,t)$ that

$$\operatorname{lin}(\mathcal{T}_{K_1}(y,t)) = \left\{ (d,\alpha) \in \mathbb{R}^s \times \mathbb{R} : \alpha = \sum_{1\leq i\leq s:\, y_i \neq 0} \operatorname{sgn}(y_i) d_i,\ d_i = 0 \text{ if } y_i = 0 \right\}$$
$$\forall\, (y,t) \in \partial K_1 \setminus \{0\}.$$

Analogously, for $p = \infty$ it can be shown that

$$\mathcal{T}_{K_\infty}(y,t) = \left\{ (d,\alpha) : \sum_{\ell:\, y_\ell \neq 0} \max(d_{I_\ell} \circ \operatorname{sgn}((y_\ell)_{I_\ell})) + \sum_{\ell:\, y_\ell = 0} \|d_\ell\|_\infty \leq \alpha \right\}$$
$$\forall\, (y,t) \in \partial K_\infty \setminus \{0\},$$

where, for each $\ell$, $I_\ell := \{i : |(y_\ell)_i| = \|y_\ell\|_\infty\}$. Thus, it follows from $\mathrm{lin}(\mathcal{T}_{K_\infty}(y,t)) = \mathcal{T}_{K_\infty}(y,t) \cap -\mathcal{T}_{K_\infty}(y,t)$ that

$$\mathrm{lin}(\mathcal{T}_{K_\infty}(y,t)) = \left\{ (d,\alpha) : \alpha = \sum_{\ell:\, y_\ell \neq 0} \gamma_\ell, d_{I_\ell} = \gamma_\ell \,\mathrm{sgn}((y_\ell)_{I_\ell}), d_\ell = 0 \text{ if } y_\ell = 0 \right\}$$
$$\forall\, (y,t) \in \partial K_\infty \setminus \{0\}.$$

Finally, we note that for $p = 1, 2, +\infty$ and under Assumption 4, one can prove that the primal constraint nondegeneracy condition (4.8) implies the positive definiteness of the set of generalized Hessian operators $\hat{\partial}^2 \Phi_f^\beta(\bar{X}, \bar{y})$, where $\hat{\partial}^2 \Phi_f^\beta$ is defined in (3.25) and $(\bar{X}, \bar{y})$ is the unique optimal solution to (3.3), i.e., $V[D,d] \neq 0$ for any $V \in \hat{\partial}^2 \Phi_f^\beta(\bar{X}, \bar{y})$ and $0 \neq (D,d) \in \mathbb{R}^{n\times n} \times \mathbb{R}^s$. As a consequence, the Newton acceleration step (3.27) is well defined if the current point is sufficiently close to $(\bar{X}, \bar{y})$. The proof of this statement can be done by following that of [52, Proposition 5] in the context of nonconvex SDP. Note that since we consider convex problems in this paper, we do not need the penalty parameter $\beta$ to go beyond a threshold value as in [52]. Here we omit the details of the proof for brevity.

**4.2. Convergence results.** In this subsection we first establish a lemma, which together with the classical results in [48, 47] ensures the global convergence of Algorithm 1. The local convergence rate of Algorithm 1 can also be directly derived from the results in [48, 47]. For completeness, we shall merely present the convergence results below but omit their proofs.

LEMMA 4.2. *Let $\pi_f^\beta$ and $\Theta_\beta$ be defined in* (3.14) *and* (3.11)*, respectively. Then, $(X^{k+1}, y^{k+1})$ generated by Algorithm* 1 *satisfies*

$$\|(X^{k+1}, y^{k+1}) - \pi_f^\beta(X^k, y^k)\|^2/(2\beta) \le \Phi_f^\beta(X^k, y^k) - \theta_k(\lambda^{k+1}, \eta^{k+1}). \tag{4.14}$$

*Proof.* From (3.13), $\Theta_\beta(X, y, \lambda, \eta)$ is the Moreau–Yosida regularization of $\mathcal{L}(\cdot, \cdot, \lambda, \eta)$ for fixed $(\lambda, \eta)$. Thus, $\Theta_\beta(X, y, \lambda, \eta)$ is convex in $(X, y)$. Furthermore, it is easy to show from Proposition 2.1 and (3.17) that

$$\nabla_{(X,y)} \Theta_\beta(X^k, y^k, \lambda^{k+1}, \eta^{k+1}) = \frac{1}{\beta}(X^k - X^{k+1}, y^k - y^{k+1}).$$

Thus, for any $(X, y) \in \mathbb{R}^{n\times n} \times \mathbb{R}^s$, it holds that

$$\begin{aligned}
\theta_k(\lambda^{k+1}, \eta^{k+1}) &+ \frac{1}{\beta}\left\langle (X^k - X^{k+1}, y^k - y^{k+1}), (X - X^k, y - y^k) \right\rangle \\
&\le \Theta_\beta(X, y, \lambda^{k+1}, \eta^{k+1}) \le \sup_{\lambda,\eta} \Theta_\beta(X, y, \lambda, \eta) \\
&= \sup_{\lambda,\,\eta} \inf_{U,\,v} \mathcal{L}(U, v, \lambda, \eta) + \frac{1}{2\beta}\left(\|U - X\|^2 + \|v - y\|^2\right) \\
&= \inf_{U,\,v} \sup_{\lambda,\,\eta} \mathcal{L}(U, v, \lambda, \eta) + \frac{1}{2\beta}\left(\|U - X\|^2 + \|v - y\|^2\right) \\
&= \inf_{U,\,v} f(U, v) + \frac{1}{2\beta}\left(\|U - X\|^2 + \|v - y\|^2\right) \\
&\le f\left(\pi_f^\beta(X^k, y^k)\right) + \frac{1}{2\beta}\|\pi_f^\beta(X^k, y^k) - (X, y)\|^2,
\end{aligned} \tag{4.15}$$

where the first inequality follows from the fact that $\theta_k(\lambda^{k+1},\eta^{k+1}) = \Theta_\beta(X^k,y^k,\lambda^{k+1},\eta^{k+1})$ and the convexity of $\Theta_\beta$ as a function of $(X,y)$. On the other hand, it is obvious by definition that

$$\Phi_f^\beta(X^k,y^k) = f\left(\pi_f^\beta(X^k,y^k)\right) + \frac{1}{2\beta}\|\pi_f^\beta(X^k,y^k) - (X^k,y^k)\|^2. \tag{4.16}$$

It follows from (4.15) and (4.16) that

(4.17)

$$\begin{aligned}
&\Phi_f^\beta(X^k,y^k) - \theta_k(\lambda^{k+1},\eta^{k+1})\\
&\quad\ge \frac{1}{\beta}\left\langle (X^k - X^{k+1}, y^k - y^{k+1}), (X - X^k, y - y^k)\right\rangle\\
&\qquad + \frac{1}{2\beta}\left(\|\pi_f^\beta(X^k,y^k) - (X^k,y^k)\|^2 - \|\pi_f^\beta(X^k,y^k) - (X,y)\|^2\right),\\
&\quad= \frac{1}{2\beta}\left(2\left\langle \pi_f^\beta(X^k,y^k) - (X^{k+1},y^{k+1}), (X,y) - (X^k,y^k)\right\rangle - \|(X,y) - (X^k,y^k)\|^2\right).
\end{aligned}$$

The required result in (4.14) follows directly by taking the supremum on the right-hand side of (4.17) since it holds for all $(X,y) \in \mathbb{R}^{n\times n}\times\mathbb{R}^s$. ∎

In order to present the global and local convergence results of Algorithm 1, we define two set-valued operators $T_f$ and $T_{\mathcal{L}}$. For $(X,y)\in\mathbb{R}^{n\times n}\times\mathbb{R}^s$, we define

$$T_f(X,y) = \{(U,v)\in\mathbb{R}^{n\times n}\times\mathbb{R}^s : (U,v)\in\partial f(X,y)\}.$$

For $(X,y,\lambda,\eta)\in\mathbb{R}^{n\times n}\times\mathbb{R}^s\times\mathbb{R}^m\times\mathbb{R}^s$, we define

$$T_{\mathcal{L}}(X,y,\lambda,\eta) = \{(U,v,-\mu,-\nu)\in\mathbb{R}^{n\times n}\times\mathbb{R}^s\times\mathbb{R}^m\times\mathbb{R}^s : (U,v,-\mu,-\nu)\in\partial\mathcal{L}(X,y,\lambda,\eta)\}.$$

To ensure the uniqueness of multipliers, we further make the following assumption.

*Assumption* 5. The primal constraint nondegeneracy condition (4.8) holds.

Since $\Phi_f^\beta(X^k,y^k) = \sup_{\lambda,\eta}\theta_k(\lambda,\eta)$, it follows from (4.14) and (3.18a) that

$$\|(X^{k+1},y^{k+1}) - \pi_f^\beta(X^k,y^k)\| \le \varepsilon_k,\ \varepsilon_k\ge 0, \sum_{k=0}^\infty \varepsilon_k < \infty. \tag{4.18}$$

Under the assumption that $\mathcal{F}_D \ne \emptyset$, problem (3.3) has a unique optimal solution, which we denote by $(\bar X,\bar y)$. It follows from (4.18) that

$$\begin{aligned}
\|(X^{k+1},y^{k+1}) - (\bar X,\bar y)\| - \varepsilon_k &\le \|\pi_f^\beta(X^k,y^k) - (\bar X,\bar y)\|\\
&\le \|(X^k,y^k) - (\bar X,\bar y)\|,
\end{aligned} \tag{4.19}$$

where the first $\le$ follows from the triangle inequality, and the second $\le$ follows from the nonexpansiveness of $\pi_f^\beta$ (see Proposition 2.1) and the fact that $\pi_f^\beta(\bar X,\bar y) = (\bar X,\bar y)$. Further considering $\sum_{k=0}^\infty\varepsilon_k<\infty$ and $\varepsilon_k\ge 0$, it it easy to show from (4.19) that the sequence $\{(X^k,y^k)\}$ generated by Algorithm 1 is bounded, a key property needed in the proof of the global convergence of PPA. On the other hand, the local convergence of Algorithm 1 depends on (4.19) and the Lipschitz continuity of $T_f^{-1}$ and $T_{\mathcal{L}}^{-1}$ at the origin; see [48].

Now, we are ready to present the global and local convergence results of Algorithm 1, which follow directly [48, Theorem 1] and [47, Theorems 4 and 5].

THEOREM 4.3 (global convergence). *Let Algorithm* 1 *be executed with stopping criterion* (3.18a). *If $\mathcal{F}_D \neq \emptyset$, then the sequence $\{(X^k, y^k)\}$ generated by Algorithm* 1 *converges to $(\bar{X}, \bar{y})$, the unique optimal solution to* (3.3), *and $\{(\lambda^k, \eta^k, Z^k)\}$ ($Z^k := S - \mathcal{A}^*\lambda^k - \mathcal{P}^*\eta^k$) is asymptotically maximizing the dual problem* (3.7) *with the same optimal value as the primal problem, i.e., strong duality holds.*

*If $\{(X^k, y^k)\}$ is bounded and $\mathcal{F}_P \neq \emptyset$, then the sequence $\{(\lambda^k, \eta^k, Z^k)\}$ is also bounded and thus converges to the unique optimal solution to* (3.7).

THEOREM 4.4 (local convergence). *Let Algorithm* 1 *be executed with stopping criteria* (3.18a) *and* (3.18b). *Assume that both $\mathcal{F}_P$ and $\mathcal{F}_D$ are nonempty. If $T_f^{-1}$ is Lipschitz continuous at the origin with modulus $a_f$, then $\{(X^k, y^k)\}$ converges to $(\bar{X}, \bar{y})$, the unique optimal solution to* (3.3), *and*

$$\|(X^{k+1}, y^{k+1}) - (\bar{X}, \bar{y})\| \leq \tau_k \|(X^k, y^k) - (\bar{X}, \bar{y})\|$$

*for all $k$ sufficiently large, where $\tau_k = (a_f(a_f^2 + \beta^2)^{-1/2} + \delta_k)/(1 - \delta_k) \to a_f(a_f^2 + \beta^2)^{-1/2} < 1$.*

*If in addition condition* (3.18c) *is satisfied and $T_{\mathcal{L}}^{-1}$ is Lipschitz continuous at the origin with modulus $a_{\mathcal{L}}$ ($\geq a_f$), then for all $k$ sufficiently large it holds that*

$$\|(\lambda^{k+1}, \eta^{k+1}) - (\bar{\lambda}, \bar{\eta})\| \leq \tau_k' \|(X^{k+1}, y^{k+1}) - (X^k, y^k)\|,$$

*where $\tau_k' = a_{\mathcal{L}}(1 + \delta_k')/\beta \to a_{\mathcal{L}}/\beta$ and $(\bar{\lambda}, \bar{\eta}, \bar{Z})$ ($\bar{Z} := S - \mathcal{A}^*\bar{\lambda} - \mathcal{P}^*\bar{\eta}$) is the unique optimal solution to the dual problem* (3.7).

**5. A dual based ADM.** In this section, we derive a simple alternating minimization algorithm, the ADM, based on the dual problem (3.7). As noted, under Assumption 3, $\varphi^*$ is the indicator function of the set $\mathcal{B}$ defined in (3.8). Thus, it is easy to see that (3.7) is equivalent to

$$\min_{\lambda,\eta,Z} \left\{ -b^\top\lambda - \log\det Z : \mathcal{A}^*\lambda + \mathcal{P}^*\eta + Z = S,\ Z \in S^n_{++},\ \eta \in \mathcal{B} \right\}. \tag{5.1}$$

The augmented Lagrangian function associated with (5.1) is given by

$$\mathcal{L}_A(\lambda, \eta, Z, X) = -b^\top\lambda - \log\det Z + \frac{\sigma}{2}\|\mathcal{A}^*\lambda + \mathcal{P}^*\eta + Z - S + X/\sigma\|^2 - \|X\|^2/2\sigma,$$

where $(\lambda, \eta, Z) \in \mathbb{R}^m \times \mathcal{B} \times S^n_{++}$, $X \in S^n$ is the Lagrangian multiplier associated with the equality constraints, and $\sigma > 0$ is a penalty parameter. To solve the dual problem (5.1), the classical ALM or method of multiplier of Hestenes [26] and Powell [44] iterates as follows: given $X^0$, for $k = 0, 1, 2, \ldots$,

$$(\lambda^{k+1}, \eta^{k+1}, Z^{k+1}) = \arg\min \left\{ \mathcal{L}_A(\lambda, \eta, Z, X^k) : (\lambda, \eta, Z) \in \mathbb{R}^m \times \mathcal{B} \times S^n_+ \right\}, \tag{5.2a}$$

$$X^{k+1} = X^k + \sigma\left(\mathcal{A}^*\lambda^{k+1} + \mathcal{P}^*\eta^{k+1} + Z^{k+1} - S\right). \tag{5.2b}$$

It is easy to verify that the multiplier $X$ is actually the primal variable in (1.5). Therefore, whenever the generated sequence $\{X^k\}$ converges, it converges to the optimal solution of the primal problem (1.5). Obviously, the practical efficiency of the ALM framework (5.2) depends on our ability to solve the subproblem (5.2a). Unfortunately, solving (5.2a) is not an easy task since it has three blocks of variables ($\lambda$, $\eta$, $Z$) and each block is involved in a different structure. To decouple the difficulty caused by the joint minimization with respect to $(\lambda, \eta, Z)$, we minimize with respect to each of them separately while keeping the others fixed. Meanwhile, we adopt the

idea of Gauss–Seidel iteration to utilize the latest information. After each sweep of alternating minimization, we update $X$ according to (5.2b). This leads to an iterative algorithm that is known as the ADM pioneered by Glowinski and Marrocco [25] and Gabay and Mercier [23]:

$$\lambda^{k+1} = \arg\min \left\{ \mathcal{L}_A(\lambda, \eta^k, Z^k, X^k) : \lambda \in \mathbb{R}^m \right\}, \tag{5.3a}$$

$$\eta^{k+1} = \arg\min \left\{ \mathcal{L}_A(\lambda^{k+1}, \eta, Z^k, X^k) : \eta \in \mathcal{B} \right\}, \tag{5.3b}$$

$$Z^{k+1} = \arg\min \left\{ \mathcal{L}_A(\lambda^{k+1}, \eta^{k+1}, Z, X^k) : Z \in S^n_+ \right\}, \tag{5.3c}$$

$$X^{k+1} = X^k + \sigma \left( \mathcal{A}^*\lambda^{k+1} + \mathcal{P}^*\eta^{k+1} + Z^{k+1} - S \right). \tag{5.3d}$$

Now we elaborate that all three subproblems in (5.3) can be solved easily. First, it is easy to see that (5.3a) is a least squares problem with normal equations given by

$$\mathcal{A}\mathcal{A}^*\lambda = b/\sigma - \mathcal{A}\left( \mathcal{P}^*\eta^k + Z^k - S + X^k/\sigma \right). \tag{5.4}$$

For the linear map $\mathcal{A}$ which enforces explicit sparsity constraints of the form $\{X_{ij} = 0 : (i,j) \in \Omega\}$, i.e., $\mathcal{A}X = X_\Omega$, we have $\mathcal{A}\mathcal{A}^* = \mathcal{I}$ (the identity operator). In this case, the solution $\lambda^{k+1}$ to (5.4) is trivial to obtain. Second, it follows from Assumption 1 and the definition of $\mathcal{P}$ that $\mathcal{P}\mathcal{P}^* = \mathcal{I}$, and thus the $\eta$-subproblem (5.3b) always has an analytical solution given by

$$\eta^{k+1} = \Pi_{\mathcal{B}}\left( \mathcal{P}(S - \mathcal{A}^*\lambda^{k+1} - Z^k - X^k/\sigma) \right), \tag{5.5}$$

where $\Pi_{\mathcal{B}}$ denotes the Euclidean projection onto $\mathcal{B}$. For $p = 1, 2, \infty$, the projection onto $\mathcal{B}$, and thus $\eta^{k+1}$ in (5.5), can be computed easily. Third, from Proposition 2.2, the solution to the $Z$-subproblem (5.3c) is analytically given by

$$Z^{k+1} = \phi^+_{1/\sigma}\left( S - \mathcal{A}^*\lambda^{k+1} - \mathcal{P}^*\eta^{k+1} - X^k/\sigma \right). \tag{5.6}$$

Obviously, the computational cost of $Z^{k+1}$ in (5.6) is one eigenvalue decomposition. In summary, all three subproblems are easily solvable and thus the ADM framework (5.3) is easily implementable. The implementation details of the ADM framework (5.3), including adaptive choice of the parameter $\sigma$ and stopping criterion, will be discussed in section 6.

Due to the simplicity and surprising effectiveness of ADM for a wide range of optimization problems including total variation problems in image processing [19, 59], $\ell_1$-minimizations in compressive sensing [58], nuclear norm problems in low-rank matrix reconstruction and factorization [7, 57, 51], semidefinite programming [55], and many others [4], the ADM has recently attracted a lot of attention in the signal, image, and data processing communities. We note that the classical ADM [25, 23] is designed for linear equality constrained convex optimization problems where the objective function contains only two blocks of variables. However, here we separated the objective function in (5.1) into three blocks ($\lambda$, $\eta$, $Z$) because for fixed $X = X^k$, the joint minimization of $\mathcal{L}_A$ in its effective domain with respect to any two of them is not easily solvable. Although the classical ADM is a special case of the PPA (see [18]) and thus its convergence can be guaranteed even under certain inexactness of solutions to the subproblems, the convergence of the ADM-like algorithm (5.3), which is a natural generalization of the classical ADM when the objective function has three blocks of variables, is still ambiguous. In section 6, we will verify the convergence of (5.3) numerically.

**6. Numerical results.** In this section, we present numerical results to demonstrate the performance of the proposed Newton-CG based PPA on (1.5) with both synthetic and real data. We implemented the algorithm in MATLAB and referred to it as LGL (log-determinant optimization with group Lasso regularization). Since we are not aware of any publicly available codes customized for solving (1.5) with group Lasso regularization ($p = 2, \infty$), we only compared LGL with the ADM-like algorithm (5.3). All the experiments were performed with Microsoft Windows XP and MATLAB v7.9 (R2009b), running on a Lenovo desktop with an Intel Core i7-2600 CPU at 3.40 GHz and 3 GB of memory.

**6.1. A preconditioner.** Let $\mathcal{T} : S^n \to S^n$ be defined by $\mathcal{T}(H) = Q(\Gamma \circ (Q^\top HQ))Q^\top$ for $H \in S^n$, where $\Gamma$ and $Q$ are given in Proposition 2.2. Let $\mathbf{A}$, $\mathbf{P}$, and $\mathbf{T}$ be the matrix representations of the linear mappings $\mathcal{A}$, $\mathcal{P}$, and $\mathcal{T}$, respectively. Then the coefficient matrix in (3.21) (for simplicity, here we omit the dependence on the iteration counters $k$ and $j$) has the form

$$M := \beta \begin{bmatrix} \mathbf{A} \\ \mathbf{P} \end{bmatrix} \mathbf{T} \begin{bmatrix} \mathbf{A} \\ \mathbf{P} \end{bmatrix}^\top + \beta \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} + \epsilon I, \tag{6.1}$$

where $J \in \partial\pi^\beta_\varphi(z_\beta)$. Clearly, the efficiency for solving the linear system (3.21) is crucial for the performance of the overall algorithm. To achieve a faster convergence for the CG method to solve (3.21), an effective preconditioner is desired. In our implementation, we designed an easy-to-compute approximate diagonal preconditioner by using an idea first developed in [24].

Let the standard basis of $S^n$ be given by $\{E_{ij} := \alpha_{ij}(e_ie_j^\top + e_je_i^\top) : 1 \le i \le j \le n\}$, where $e_i$ is the $i$th unit vector in $\mathbb{R}^n$, and $\alpha_{ij} = 1/\sqrt{2}$ if $i \ne j$ and $\alpha_{ij} = 1/2$ if otherwise. Then the diagonal element of $\mathbf{T}$ with respect to the basis element $E_{ij}$ is given by

$$\mathbf{T}_{(ij),(ij)} = \langle E_{ij}, \mathcal{T}(E_{ij})\rangle = \begin{cases} ((Q \circ Q)\Gamma(Q \circ Q)^\top)_{ij} + \langle v^{(ij)}, \Gamma v^{(ij)}\rangle & \text{if } i \ne j, \\ ((Q \circ Q)\Gamma(Q \circ Q)^\top)_{ij} & \text{otherwise,} \end{cases} \tag{6.2}$$

where letting $Q_i$ be the $i$th column of $Q$, $v^{(ij)} = Q_i \circ Q_j$. It is easy to see from (6.2) that the computational cost for all the diagonal elements of $\mathbf{T}$ is $O(n^4)$. In our implementation, we merely computed the first term on the right-hand side of (6.2), which is typically a good approximation of $\mathbf{T}_{(ij)(ij)}$, and the computational cost is reduced to $O(n^3)$. Let $\mathbf{d}_{(ij)} := ((Q \circ Q)\Gamma(Q \circ Q)^\top)_{ij}$ for $i, j = 1, 2, \ldots, n$. We used the following preconditioner for $M$:

$$M_D := \beta \begin{bmatrix} \mathbf{A} \\ \mathbf{P} \end{bmatrix} \mathrm{diag}(\mathbf{d}) \begin{bmatrix} \mathbf{A} \\ \mathbf{P} \end{bmatrix}^\top + \epsilon I.$$

We note that it is also possible to take into account $J \in \partial\pi^\beta_\varphi(z_\beta)$ in the preconditioner because for $\varphi_\ell$'s prescribed in Assumption 3, the explicit representation of the elements of $\partial\pi^\beta_\varphi(z_\beta)$ is not complicated. Based on our numerical experience, taking $M_D$ as the preconditioner works reasonably well in practice, and thus we adopted it for simplicity.

**6.2. Implementation details.** We measured the primal and the dual infeasibility of (1.5) and (3.7), respectively, by

$$R_P := \frac{\|\mathcal{A}X - b\|}{1 + \|b\|} \quad \text{and} \quad R_D := \max\left\{\frac{\|\mathcal{A}^*\lambda + \mathcal{P}^*\eta + Z - S\|}{1 + \|S\|}, \frac{\|\eta_\ell\|_q - \omega_\ell}{\omega_\ell},\ \ell = 1, \ldots, r\right\}.$$

Let $\texttt{pobj} := \langle S, X\rangle - \log\det X + \sum_{\ell=1}^{r}\omega_\ell\|X_{g_\ell}\|_p$ and $\texttt{dobj} := b^\top\lambda + \log\det Z + n$ be the primal and the dual objective function values. Under Assumption 4, strong duality holds. Therefore, in our experiments we terminated both LGL and ADM by $\texttt{Res} := \max\{R_P, R_D, R_G\} < \texttt{Tol}$, where $\texttt{Tol} > 0$ is a tolerance and $R_G$ is the relative duality gap defined by

$$R_G := \frac{|\texttt{pobj} - \texttt{dobj}|}{1 + |\texttt{pobj}| + |\texttt{dobj}|}. \tag{6.3}$$

We also terminated ADM if the requirement $\texttt{Res} < \texttt{Tol}$ was not satisfied after a maximum number of 2000 iterations. In all experiments, we set $\texttt{Tol} = 10^{-5}$. For LGL, we initialized $\beta_0 = 1$ and updated $\beta$ by

$$\beta_{k+1} = \begin{cases} \min(2\beta_k, 10^8) & \text{if } R_D^{k+1}/R_D^k > 0.5, \\ \beta_k & \text{otherwise,} \end{cases}$$

where $R_D^k$ represents the dual infeasibility at the $k$th iteration. As for the penalty parameter $\sigma$ in ADM, we add the following note. It is well known that the penalty parameter $\sigma$ plays an important role for the convergence rate of the ALM scheme (5.2). In general, a larger value of $\sigma$ leads to a faster convergence of the outer loop. However, extremely large values of $\sigma$ may cause numerical difficulty and thus should be avoided in practice. The same comments apply to the ADM since it is a practical variant of the ALM for structured problems. In our experiments we initialized $\sigma_0 = 1$ for constrained problems and updated it in a way such that the primal and the dual infeasibilities are well balanced. Specifically, we updated $\sigma$ as follows:

$$\sigma_{k+1} = \begin{cases} \min(2\sigma_k, 10^8) & \text{if } R_P/R_D < 0.1, \\ \max(0.5\sigma_k, 10^{-2}) & \text{if } R_P/R_D > 10, \\ \sigma_k & \text{otherwise.} \end{cases}$$

For unconstrained problems, we first rescale the problem data and then set $\sigma = 1$ without dynamic adjustment. In all the experiments, we initialized LGL and ADM, respectively, by $(X^0, y^0) = (I, \mathcal{P}I)$ and $(X^0, Z^0, \eta^0) = (I, I, 0)$. A Newton acceleration step (3.28) was also taken if after a PPA iteration the condition $\max(R_P, R_D) < 10^{-2}$ is satisfied at the current point.

**6.3. Results on random synthetic data.** In this section, we present experimental results to demonstrate the performance of LGL and ADM on (1.5) with random synthetic data. To begin, we describe our procedure for generating random synthetic data, including the inverse covariance matrix $\Sigma^{-1}$, the sample covariance matrix $S$, the group structure $G$, and the linear constraints $\mathcal{A}X = b$.

For the inverse covariance matrix $\Sigma^{-1}$, we first generate its sparsity pattern and then the values of its nonzero entries. By reordering the components of $x \sim N(0, \Sigma)$ if necessary, without loss of generality, we assume that $x$ can be partitioned into $n_g$ groups where the indices of components in each group are adjacent. That is, $x = (x_{I_1}, x_{I_2}, \ldots, x_{I_{n_g}})$, where $I_j = \{i_{j-1} + 1, i_{j-1} + 2, \ldots, i_j\}$ for $j = 1, 2, \ldots, n_g$. Here, $i_0 := 0$ and $i_{n_g} := n$. The group sizes $\{|I_j| : j = 1, 2, \ldots, n_g\}$ are determined randomly such that each $|I_j|$ is around the mean value $n/n_g$. In the graphical model of $x$, we let two nodes $x_i$ and $x_j$ from the same group be connected with probability $p_1$. On the other hand, for any two different groups $I_{j_1}$ and $I_{j_2}$, we let the probability of "there exist connections between $I_{j_1}$ and $I_{j_2}$" be $p_2$. In the case that indeed there exist connections between $I_{j_1}$ and $I_{j_2}$, we let two nodes, one from each group,

be connected with probability $p_3$. Based on the principle that connections within a group are more likely than connections between different groups, we assume that $0 < p_2 < p_3 < p_1 < 1$. Let `Mask` be the generated sparsity pattern and $U$ be an $n \times n$ matrix having the sparsity pattern `Mask` and entry values $\pm 1$ generated with equal probability. The inverse covariance matrix (denoted by $A$) is generated via the following MATLAB scripts:

- d = diag(U'*U); A = sign(sprandn(`Mask`)) + diag(d+1);
- eig_min = min(eig(A)); ep = max(-1.2*eig_min, 1E-4); A = A + ep * eye(n);

After generating $\Sigma^{-1}$ (and thus $\Sigma$), we generated $2n$ independent and identically distributed (i.i.d.) random vectors from $N(0, \Sigma)$ and calculate the sample covariance matrix $S$. For two index sets $I_i$ and $I_j$, we let $I_i \times I_j := \{(k,l) : k \in I_i, l \in I_j\}$. The group structure is then set to be $G = \{I_i \times I_j : i, j = 1, 2, \ldots, n_g\}$. The linear constraints $\mathcal{A}X = b$ are determined by $\{X_{ij} = 0 : (i,j) \in \Omega\}$, where $\Omega$ is a subset of $\mathcal{E}$ (the set of indices of the zero elements of $\Sigma^{-1}$). In our experiments, we randomly chose approximately 50% of the elements in $\mathcal{E}$ to form the subset $\Omega$.

In the following, we first present an illustrative example to demonstrate the potential superiority of group Lasso regularization ($p = 2, \infty$) compared to $\ell_1$-regularization ($p = 1$) when the inverse covariance matrix possesses a blockwise sparsity structure, and then we present comparison results of LGL and ADM on random synthetic data with different problem sizes.

Figure 6.1 shows the results recovered from (1.5) with $p = 1, 2$, and $\infty$. In this experiment, the regularization parameters were chosen by trial and error so that the recovered sparsity pattern approximates that of the true inverse covariance matrix sufficiently well. It can be seen from Figure 6.1 that with appropriate choices of groups and regularization parameters, group Lasso regularization with $p = 2$ and $p = \infty$ can give better results than $p = 1$. Specifically, the blockwise sparsity structure of the true inverse covariance matrix is approximately recovered by (1.5) with $p = 2$ and $p = \infty$, while the result for $p = 1$ is much worse no matter how we tune the regularization parameters.

Table 6.1 presents detailed comparison results of LGL and ADM on solving these random problems with different problem sizes, where the number of iterations (iter), the consumed CPU time (measured in seconds), and the resulting residues ($R_P$, $R_D$ and $R_G$) are given. To better understand the convergence speed (in terms of iterations), for each test we present the results of two or three intermediate iterations for both methods, where the iteration numbers are those where LGL and ADM attain the required accuracy in the solution (measured by `Res`). In particular, in our experiments all the final solutions obtained by LGL satisfy the condition `Res` $< 10^{-5}$. Therefore, the three iteration numbers will be those where the iterates first meet the conditions `Res` $< 10^{-1}, 10^{-3}$, and $10^{-5}$. On the other hand, most of the final solutions obtained by ADM failed to meet the condition `Res` $< 10^{-5}$. In this case, we present the final results as well as one or two intermediate iterations to make a consistent comparison with LGL. The reason for us to present detailed results of several intermediate iterations is that one can compare the two methods to obtain a solution of modest accuracy. In the Iter column for LGL, four numbers are given for each of the selected iterations, where the first (the one outside of the parentheses) represents the number of PPA iterations, and the three numbers inside the parentheses represent, respectively, the total number of Newton systems (3.21) solved, the average PCG steps taken for solving (3.21), and the total number of outer Newton acceleration steps, i.e., (3.27). We also present the final primal objective function

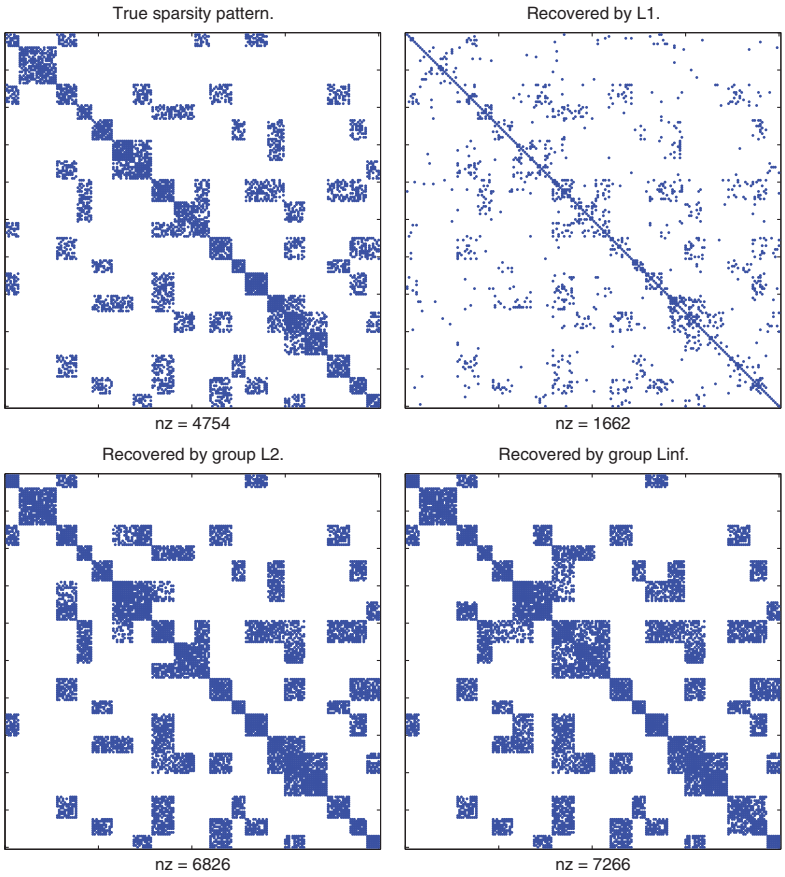


FIG. 6.1. *An illustrative example. From top left to bottom right: the sparsity pattern of the true inverse covariance matrix and those recovered from* (1.5) *with* $p = 1, 2$, *and* $\infty$, *respectively. Parameters:* $n = 200$, $n_g = 20$, $p_1 = 0.8$, $p_2 = 0.2$, *and* $p_3 = 0.5$. *The regularization parameters are set to be* $\omega_\ell \equiv \omega = 0.03, 0.13$, *and* $0.8$ *for* $p = 1, 2$, *and* $\infty$, *respectively.*

values (`pobj`) below the detailed results of the selected iterations. For example, for $p = 1$, $n = 500$, and $m = 55194$, LGL obtained the final primal objective function value of $-1.47963560 \times 10^3$. The value at the corresponding position for ADM represents the difference between the final primal objective function value obtained by ADM and that obtained by LGL, e.g., for $p = 1$, $n = 500$, and $m = 55194$, the value $-5.48 \times 10^{-7}$ implies that the final primal objective function value obtained by ADM is $-1.47963560 \times 10^3 - 5.48 \times 10^{-7}$. To evaluate how well we have recovered the true inverse covariance matrix, we compute the quadratic loss ($\text{Loss}_Q$) and the normalized entropy loss ($\text{Loss}_E$) defined, respectively, by

$$\text{Loss}_Q := \frac{1}{n}\|\Sigma X - I\| \text{ and } \text{Loss}_E := \frac{1}{n}(\langle \Sigma, X\rangle - \log\det \Sigma X - n).$$

We note that in general it is impossible to recover $\Sigma^{-1}$ accurately based on $S$ via solving (1.5). The purpose of solving (1.5) is not to recover the true inverse covariance matrix accurately but to detect its sparsity pattern while maintaining a reasonable approximation to the true matrix. To measure the quality of the sparsity pattern in

TABLE 6.1

*Results on random problems. Parameters:* $n_g = n/10$, $p_1 = 0.8$, $p_2 = 0.2$, *and* $p_3 = 0.5$; $\omega_\ell \equiv \omega = 1/n$.

| | | LGL | | | ADM | | |
|---|---|---|---|---|---|---|---|
| $p$ | $n\|m$ | Iter | Time | ($R_P$, $R_D$, $R_G$) | Iter | Time | ($R_P$, $R_D$, $R_G$) |
| 1 | 500\|55194 | 18(24, 4.0, 0) | 13 | (**7.6-2**, 5.9-2, 3.8-2) | 30 | 4 | (1.9-3, **6.7-2**, 2.4-2) |
| | | 22(32, 4.4, 1) | 18 | (**8.4-4**, 1.2-4, 6.7-5) | 36 | 5 | (**9.7-4**, 4.2-4, 1.3-4) |
| | | 24(36, 4.5, 3) | 23 | (1.1-8, **7.7-6**, 4.6-7) | 47 | 6 | (**9.1-6**, 2.1-6, 3.0-7) |
| | pobj & Loss | -1.47963560 3, (5.48-3, 4.09-3, 1.00, 0.02) | | | -5.48-7, (5.48-3, 4.09-3, 1.00, 0.02) | | |
| | 1000\|222941 | 22(32, 4.4, 0) | 93 | (1.4-2, **2.1-2**, 1.6-2) | 32 | 20 | (3.9-13, **9.7-2**, 3.3-2) |
| | | 25(39, 4.7, 1) | 119 | (5.3-9, **7.9-5**, 4.6-5) | 37 | 24 | (1.3-12, **4.9-4**, 1.9-4) |
| | | 28(45, 4.8, 4) | 170 | (5.7-9, **8.9-6**, 4.2-6) | 41 | 26 | (1.4-12, **8.7-6**, 2.6-6) |
| | pobj & Loss | -3.57578598 3, (3.63-3, 4.97-3, 1.00, 0.01) | | | -1.73-6, (3.63-3, 4.97-3, 1.00, 0.01) | | |
| 2 | 500\|55072 | 24(39, 13.8, 0) | 47 | (**5.6-2**, 7.5-4, 6.1-4) | 48 | 8 | (**9.8-2**, 4.1-2, 1.6-2) |
| | | 28(60, 18.8, 1) | 75 | (3.0-6, **1.3-5**, 9.5-6) | 150 | 24 | (**9.7-4**, 3.4-4, 1.3-4) |
| | | 29(64, 20.2, 2) | 85 | (6.3-6, **9.5-6**, 6.9-6) | 259 | 41 | (**9.9-6**, 3.8-6, 1.2-6) |
| | pobj & Loss | -1.54395177 3, (1.17-2, 2.98-2, 0.74, 0.63) | | | -7.17-7, (1.17-2, 2.98-2, 0.74, 0.63) | | |
| | 1000\|224590 | 25(42, 23.6, 0) | 379 | (**8.4-2**, 1.3-3, 1.1-3) | 49 | 40 | (**9.8-2**, 5.5-2, 2.1-2) |
| | | 29(70, 51.4, 0) | 897 | (1.8-5, **2.8-5**, 2.0-5) | 186 | 150 | (**9.8-4**, 2.7-4, 1.0-4) |
| | | 32(83, 75.2, 3) | 1418 | (**6.9-6**, 1.6-6, 1.1-6) | 321 | 258 | (**9.7-6**, 3.1-6, 9.6-7) |
| | pobj & Loss | -3.60607928 3, (6.50-3, 1.94-2, 0.77, 0.62) | | | 3.35-7, (6.50-3, 1.94-2, 0.77, 0.62) | | |
| $\infty$ | 500\|54394 | 26(79, 47.2, 0) | 299 | (**4.5-2**, 1.6-4, 1.6-4) | 53 | 17 | (6.9-2, **9.9-2**, 3.8-2) |
| | | 29(95, 47.0, 1) | 343 | (**4.4-5**, 1.6-5, 1.4-5) | 403 | 132 | (7.4-4, **9.9-4**, 4.8-4) |
| | | 30(100, 48.2, 2) | 369 | (3.5-6, **7.5-6**, 7.0-6) | 876 | 287 | (7.4-6, **9.9-6**, 4.4-6) |
| | pobj & Loss | -1.62905709 3, (2.43-2, 1.09-1, 0.59, 0.83) | | | -5.78-7, (2.43-2, 1.09-1, 0.59, 0.83) | | |
| | 1000\|223472 | 27(75, 45.4, 0) | 1323 | (**6.8-2**, 4.1-4, 4.0-4) | 59 | 84 | (8.8-2, **9.8-2**, 3.6-2) |
| | | 30(90, 45.5, 0) | 1506 | (2.9-5, **3.7-5**, 3.4-5) | 497 | 715 | (**9.9-4**, 8.8-4, 4.2-4) |
| | | 33(104, 47.9, 3) | 1779 | (1.6-6, **2.3-6**, 2.2-6) | 1071 | 1549 | (**9.9-6**, 8.7-6, 4.0-6) |
| | pobj & Loss | -3.74196300 3, (1.60-2, 9.94-2, 0.58, 0.85) | | | 1.35-6, (1.60-2, 9.94-2, 0.58, 0.85) | | |

$X$ 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 denote the number of true positives, true negatives, false positives, and false negatives, respectively. For each computed solution $X$, we first determine an appropriate thresholding value according to the clustered distribution of the magnitudes of the elements of $X$ and then classify $X_{ij}$ as 0 if its magnitude is less than this value. In our situation, specificity measures the quality of zero entries, while sensitivity measures the quality of nonzero entries. In Table 6.1, the four values in parentheses behind the primal objective function values represent, respectively, $\text{Loss}_Q$, $\text{Loss}_E$, specificity, and sensitivity.

It can be seen from Table 6.1 that both ADM and LGL perform very well on these random problems because both methods are able to reduce `Res` to less than $10^{-5}$. With the help of acceleration by outer Newton iterations, LGL is able to reach the required accuracy in less than around 30 PPA iterations for all the tested random problems. The total number of Newton systems (3.21) solved is at most 104 (for $p = \infty$ and $n = 1000$). The average PCG steps taken for solving (3.21) is less than 5 for $p = 1$, and this number is increased to about 70 and 50 for $p = 2$ and $p = \infty$, respectively. Convergence faster than the linear rate can also be observed from the results of LGL in Table 6.1, i.e., large decreases in `Res` were obtained in very few or even two consecutive iterations, which is due to the help of the outer Newton acceleration steps. The performance of ADM on these random problems is

also favorable because `Res` decreases continuously at a relatively stable and fast speed. It can also be seen that ADM is much faster than LGL on these random problems. The differences in the final objective function values obtained by both methods are negligible. From the results of $\text{Loss}_Q$ and $\text{Loss}_E$, we see that the recovered solutions approximate the true inverse covariance matrices very well. The sensitivity results obtained by $p = 2$ and $p = \infty$ are much better than those obtained by $p = 1$, which indicates the superior quality of nonzero elements of the recovered solutions from group Lasso regularization. The specificity results for $p = 1$ are approximately 1 because usually the recovered solutions are overly sparse, while those for $p = 2$ and $p = \infty$ are smaller because some false nonzeros are detected, as illustrated by the results in Figure 6.1.

In section 6.4, we mainly examine the performance of LGL for $p = 2$ and $p = \infty$ because first, as illustrated by the results in Figure 6.1, $p = 1$ is not suitable for problems with blockwise sparse inverse covariance matrices, and second, it is intuitively true and also justified by our experimental results that the performance of LGL for $p = 1$ is comparable with that of the NAL method [54] (which has been well illustrated therein) since both methods adopt the idea of applying Newton's method to solving PPA subproblems. In section 6.5, we present comparison results of LGL and ADM on solving (1.5) with $p = 1$ and gene expression data.

**6.4. Results on deterministic synthetic data.** In this section, we present extensive comparison results between LGL and ADM on the deterministic synthetic examples considered in [62] and [20]. Specifically, we tested the following sparse inverse covariance matrices (denoted by $A$):

**ar1** $A_{ii} = 1, A_{i,i-1} = A_{i-1,i} = 0.5 \; \forall\, i$;
**ar2** $A_{ii} = 1, A_{i,i-1} = A_{i-1,i} = 0.5, A_{i,i-2} = A_{i-2,i} = 0.25 \; \forall\, i$;
**ar3** $A_{ii} = 1, A_{i,i-1} = A_{i-1,i} = 0.4, A_{i,i-2} = A_{i-2,i} = A_{i,i-3} = A_{i-3,i} = 0.2 \; \forall\, i$;
**ar4** $A_{ii} = 1, A_{i,i-1} = A_{i-1,i} = 0.4, A_{i,i-2} = A_{i-2,i} = A_{i,i-3} = A_{i-3,i} = 0.2, A_{i,i-4} = A_{i-4,i} = 0.1 \; \forall\, i$;
**decay** $A_{ij} = \exp(-2|i-j|) \; \forall\, i, j$;
**circle** $A_{ii} = 1, A_{i,i-1} = A_{i-1,i} = 0.5 \; \forall\, i, A_{1n} = A_{n1} = 0.4$.

Again, for each test we first generated $2n$ i.i.d. random samples from the $n$-dimensional Gaussian distribution $N(0, \Sigma)$ and then computed the sample covariance matrix $S$. For all these tests, we set $\omega_\ell \equiv \omega = 0.1$. The constraints $\mathcal{A}X = b$ are generated in the same manner as in section 6.3. Here, the nonzero entries of these inverse covariance matrices exhibit a "diagonal" structure (for **decay**, $A_{ij}$ decays very fast as $|i-j|$ increases and thus can be reset to zero for $|i-j|$ large). Therefore, in our experiments we tested the "diagonal" group structure, i.e., the elements in the same diagonal are grouped together. Specifically, the group structure is given by $G = \{g_\ell : \ell = 1, 2, \ldots, 2n-1\}$, where

$$g_\ell = \begin{cases} \{(1, n-\ell+1), (2, n-\ell+2), \ldots, (\ell, n)\} \setminus \Omega, & \ell = 1, \ldots, n; \\ \{(\ell-n+1, 1), (\ell-n+2, 2), \ldots, (n, 2n-\ell)\} \setminus \Omega, & \ell = n+1, \ldots, 2n-1. \end{cases}$$

Here $\Omega$ denotes the set of indices which determine $\mathcal{A}X = b$. To illustrate the performance of LGL and ADM on different groups, we also tested the "columnwise" group structure, i.e.,

$$G = \{g_\ell : \ell = 1, 2, \ldots, n\}, \text{ where } g_\ell = \{(1, \ell), (2, \ell), \ldots, (n, \ell)\} \setminus \Omega, \; \ell = 1, 2, \ldots, n.$$

Detailed experimental results of LGL and ADM are given in Tables 6.2 (diagonal groups, $p = 2$), 6.3 (diagonal groups, $p = \infty$), 6.4 (columnwise groups, $p = 2$), and 6.5

(columnwise groups, $p = \infty$), where all the quantities have the meanings explained in section 6.3.

It can be seen from Tables 6.2–6.5 that for all the tests, LGL is able to obtain solutions satisfying the condition $\texttt{Res} < 10^{-5}$. The total number of PPA iterations taken by LGL for these problems is no more than 40. The total number of Newton systems (3.21) solved is mostly less than 100, except for the two hard problems **ar1** and **circle**, for which this number was increased to 198 at the most (**ar1** and $n = 1000$ in Table 6.3). We note that problems **ar1** and **circle** are challenging since $\Sigma^{-1}$ has very small eigenvalues. (The minimum eigenvalue is on the order of $10^{-5}$ to $10^{-6}$ for $n = 500$ and 1000.) For these two harder problems, the average PCG steps taken for solving (3.21) and the total number of outer Newton acceleration steps are also more than those for the random problems tested in section 6.3, while for the other problems, the average PCG steps and the total number of outer Newton acceleration steps are comparable with those in Table 6.1 for random problems. Specifically, the total number of outer Newton acceleration steps are no more than 6 for the two harder problems and 3 for the others. It is worth noting that the outer Newton acceleration steps are usually able to decrease `Res` substantially in very few iterations. For example, it can be seen from Tables 6.2–6.5 that for the four easier problems the solution accuracy can be increased by 1∼3 digits in the final one or two iterations, while for the two harder problems, the acceleration is less. Based on our experimental results, without the outer Newton acceleration, the CPU time consumed by the Newton-CG based PPA to obtain solutions of the same accuracy can increase by about 10–20% on average.

In contrast, the ADM obtained low-accuracy solutions (`Res` fell into the range $10^{-2} \sim 10^{-3}$) for most of the tests after 2000 iterations, except for the four easier problems in Table 6.5. Based on our experimental results, the residue values produced by ADM either stagnated or improved extremely slowly after it was decreased to a certain level. As a result, it is generally very difficult and even impossible for ADM to produce a solution satisfying the final accuracy requirement $\texttt{Res} < 10^{-5}$ for most of these tests. For diagonal groups the final accuracy reached by ADM is mostly on the order of $10^{-2}$ to $10^{-3}$, while this accuracy is increased by about $1 \sim 2$ digits for the case of columnwise groups. For both type groups, ADM failed to obtain solutions with accuracy $\texttt{Res} < 10^{-2}$ for the two hard problems **ar1** and **circle**.

By comparing the results for $p = 2$ with those for $p = \infty$, we see that both LGL and ADM consumed longer CPU time for the later case. This is reasonable because the calculations of the proximal point mapping of the $\ell_\infty$-norm and its generalized Jacobian require projections onto the $\ell_1$-norm ball which is practically more expensive than for the case of $p = 2$. On the other hand, by comparing the results for the two types of group structures, we see that LGL performs stably across the two types of problems in the sense that it can attain the desired accuracy in a comparable number of iterations. But for ADM, the case of columnwise groups appears to be easier than the case of diagonal groups, since it can attain higher accuracy for the former case as compared to the latter case. It can also be seen from Tables 6.2–6.5 that LGL obtained smaller primal objective function values for about two-thirds of the tested problems. In all cases, LGL obtained solutions of much higher accuracy measured by `Res`.

For reference purpose, the results of $\text{Loss}_Q$, $\text{Loss}_E$, specificity, and sensitivity at the final iterations are also presented for all the tests. It can be seen from these results that with appropriate postprocessing of the computed solutions, the sparsity pattern of the inverse covariance matrices are recovered very well because the specificity and

TABLE 6.2
*Results on synthetic problems. Diagonal groups, $p = 2$.*

| | | LGL | | | ADM | | |
|---|---|---|---|---|---|---|---|
| Prob. | $n\|m$ | Iter | Time | $(R_P,\ R_D,\ R_G)$ | Iter | Time | $(R_P,\ R_D,\ R_G)$ |
| ar1 | 500\|62126 | 4(4, 3.0, 0) | 2 | (4.1-2, 4.3-2, **7.8-2**) | 15 | 2 | (1.5-3, **9.6-2**, 7.6-2) |
| | | 30(85, 29.2, 1) | 99 | (6.7-6, 2.7-7, **4.0-4**) | 2000 | 278 | (3.1-6, 3.1-5, **1.2-2**) |
| | | 34(100, 36.4, 5) | 133 | (**4.8-6**, 2.0-9, 2.9-6) | ——— | | |
| | pobj & Loss | 8.46651163 2, (2.91-1, 1.13-1, 1.00, 1.00) | | | 9.15-1, (2.71-1, 1.08-1, 1.00, 1.00) | | |
| | 1000\|249251 | 4(4, 3.0, 0) | 13 | (3.2-2, 3.7-2, **7.0-2**) | 14 | 10 | (2.6-4, **9.6-2**, 6.7-2) |
| | | 33(100, 40.2, 1) | 740 | (1.2-6, 9.5-8, **4.1-4**) | 2000 | 1467 | (7.5-6, 2.9-5, **4.3-2**) |
| | | 38(117, 52.1, 6) | 1071 | (**1.3-6**, 2.4-10, 9.9-7) | ——— | | |
| | pobj & Loss | 1.69925377 3, (3.00-1, 1.11-1, 1.00, 1.00) | | | 6.28 0, (2.63-1, 1.06-1, 1.00, 1.00) | | |
| ar2 | 500\|61877 | 18(27, 7.5, 0) | 19 | (**8.5-2**, 1.3-3, 2.5-3) | 34 | 5 | (1.7-2, **9.6-2**, 5.8-2) |
| | | 21(40, 9.0, 0) | 26 | (**8.8-3**, 7.3-5, 1.5-4) | 2000 | 276 | (4.4-13, **1.2-2**, 9.0-4) |
| | | 22(44, 11.5, 1) | 34 | (**8.8-6**, 1.8-6, 4.8-6) | ——— | | |
| | pobj & Loss | 6.71682925 2, (2.55-2, 1.08-1, 1.00, 0.95) | | | -3.88-1, (2.56-2, 1.08-1, 1.00, 0.97) | | |
| | 1000\|248752 | 19(30, 10.1, 0) | 128 | (**4.8-2**, 1.2-3, 2.4-3) | 40 | 30 | (1.2-2, **9.9-2**, 5.9-2) |
| | | 22(44, 11.6, 0) | 173 | (**7.6-3**, 8.7-5, 1.8-4) | 2000 | 1465 | (2.9-12, **1.5-2**, 5.6-4) |
| | | 23(48, 13.4, 1) | 209 | (**9.5-6**, 2.7-6, 3.2-6) | ——— | | |
| | pobj & Loss | 1.34126164 3, (1.79-2, 1.06-1, 1.00, 0.98) | | | 5.93-1, (1.79-2, 1.06-1, 1.00, 0.99) | | |
| ar3 | 500\|61628 | 18(27, 7.4, 0) | 19 | (**7.1-2**, 9.0-4, 1.5-3) | 34 | 5 | (1.3-2, **9.4-2**, 5.5-2) |
| | | 21(41, 8.4, 0) | 25 | (**6.0-3**, 9.3-5, 1.6-4) | 260 | 36 | (**9.9-3**, 8.3-3, 2.1-3) |
| | | 22(45, 9.9, 1) | 31 | (**6.4-6**, 2.6-6, 2.4-6) | 2000 | 273 | (1.5-11, **8.3-3**, 2.1-3) |
| | pobj & Loss | 6.16605249 2, (2.36-2, 9.82-2, 1.00, 0.92) | | | -1.08 0, (2.42-2, 1.02-1, 1.00, 0.92) | | |
| | 1000\|248253 | 20(34, 9.8, 0) | 136 | (**3.8-2**, 4.4-4, 7.7-4) | 40 | 29 | (9.7-3, **9.8-2**, 5.7-2) |
| | | 22(45, 10.8, 0) | 167 | (**7.3-3**, 6.5-5, 1.1-4) | 2000 | 1443 | (1.1-12, **1.1-2**, 1.6-3) |
| | | 23(49, 12.9, 1) | 204 | (**7.0-6**, 2.0-6, 5.8-8) | ——— | | |
| | pobj & Loss | 1.23551533 3, (1.65-2, 9.68-2, 1.00, 0.94) | | | -5.41-1, (1.67-2, 9.88-2, 1.00, 0.95) | | |
| ar4 | 500\|61380 | 18(29, 8.2, 0) | 20 | (**7.5-2**, 7.2-4, 1.1-3) | 28 | 4 | (5.8-2, **9.1-2**, 4.8-2) |
| | | 21(42, 9.0, 0) | 27 | (**8.1-3**, 6.4-5, 9.2-5) | 245 | 34 | (**9.8-3**, 8.7-3, 2.4-3) |
| | | 22(46, 10.4, 1) | 32 | (**7.6-6**, 1.5-6, 1.1-6) | 2000 | 276 | (1.1-11, **8.7-3**, 2.4-3) |
| | pobj & Loss | 6.05884352 2, (2.25-2, 9.79-2, 0.99, 0.92) | | | 3.19-1, (2.29-2, 1.00-1, 0.99, 0.92) | | |
| | 1000\|247755 | 20(33, 9.3, 0) | 129 | (**8.5-2**, 6.4-4, 9.4-4) | 32 | 23 | (3.7-2, **9.2-2**, 4.9-2) |
| | | 22(43, 10.6, 0) | 162 | (**8.3-3**, 8.4-5, 1.2-4) | 2000 | 1459 | (1.1-12, **1.1-2**, 1.8-3) |
| | | 23(47, 12.6, 1) | 199 | (**9.2-6**, 1.8-6, 1.6-6) | ——— | | |
| | pobj & Loss | 1.21405855 3, (1.57-2, 9.61-2, 1.00, 0.93) | | | -5.15-1, (1.60-2, 9.85-2, 1.00, 0.93) | | |
| decay | 500\|57961 | 18(29, 7.4, 0) | 20 | (**5.5-2**, 6.1-4, 7.5-4) | 28 | 4 | (4.5-2, **8.8-2**, 4.2-2) |
| | | 20(35, 7.6, 0) | 23 | (**8.9-3**, 1.2-4, 1.4-4) | 207 | 29 | (9.7-3, **9.7-3**, 3.5-3) |
| | | 21(39, 8.8, 1) | 28 | (**5.1-6**, 2.4-6, 1.8-6) | 2000 | 276 | (9.8-7, 2.6-3, **4.7-3**) |
| | pobj & Loss | 5.13281600 2, (1.89-2, 8.12-2, 0.69, 0.37) | | | -1.72 0, (2.00-2, 8.79-2, 0.69, 0.37) | | |
| | 1000\|240836 | 19(29, 7.2, 0) | 105 | (**9.2-2**, 1.2-3, 1.5-3) | 31 | 23 | (3.4-2, **9.8-2**, 4.6-2) |
| | | 22(43, 8.3, 0) | 141 | (**5.3-3**, 5.3-5, 6.4-5) | 311 | 227 | (**9.8-3**, 7.2-3, 3.9-3) |
| | | 23(47, 9.7, 1) | 171 | (**6.6-6**, 1.7-6, 5.7-7) | 2000 | 1459 | (2.1-6, 3.7-3, **4.5-3**) |
| | pobj & Loss | 1.02865383 3, (1.30-2, 7.75-2, 0.74, 0.30) | | | 2.32 0, (1.36-2, 8.32-2, 0.73, 0.31) | | |
| circle | 500\|62125 | 4(4, 3.0, 0) | 3 | (3.7-2, 4.2-2, **7.8-2**) | 15 | 2 | (1.3-3, **8.8-2**, 7.6-2) |
| | | 30(87, 32.6, 1) | 106 | (7.7-6, 2.0-7, **3.1-4**) | 2000 | 278 | (3.4-6, 3.0-5, **1.5-2**) |
| | | 34(102, 38.4, 5) | 138 | (**4.7-6**, 1.6-9, 2.4-6) | ——— | | |
| | pobj & Loss | 8.48828879 2, (3.20-1, 1.13-1, 1.00, 1.00) | | | 7.59-1, (2.76-1, 1.06-1, 1.00, 1.00) | | |
| | 1000\|249250 | 4(4, 3.0, 0) | 13 | (2.9-2, 3.7-2, **7.0-2**) | 14 | 10 | (2.9-4, **9.6-2**, 6.7-2) |
| | | 34(102, 36.9, 3) | 719 | (1.1-6, 7.3-8, **3.3-4**) | 2000 | 1470 | (3.0-6, 3.7-5, **5.2-2**) |
| | | 37(112, 43.6, 6) | 896 | (5.5-6, 2.2-9, **9.6-6**) | ——— | | |
| | pobj & Loss | 1.70184613 3, (3.09-1, 1.11-1, 1.00, 1.00) | | | 3.93 1, (2.10-1, 1.23-1, 1.00, 1.00) | | |

the sensitivity results are close to one for all these problems except **decay**. Problem **decay** gives worse specificity and sensitivity results because its components are less well separated than those of the other tested problems, which causes difficulty in determining appropriate values for thresholding.

**6.5. Results on gene expression data.** In this section, we present comparison results of LGL and ADM on gene expression data sets that have been widely used in the model selection and classification literature. Specifically, we will test the lymph

TABLE 6.3
*Results on synthetic problems. Diagonal groups, $p = \infty$.*

| | | LGL | | | ADM | | |
|---|---|---|---|---|---|---|---|
| Prob. | $n\|m$ | Iter | Time | ($R_P$, $R_D$, $R_G$) | Iter | Time | ($R_P$, $R_D$, $R_G$) |
| ar1 | 500\|62126 | 4(4, 3.0, 0) | 4 | (**7.8-2**, 5.0-2, 7.7-2) | 15 | 3 | (1.5-3, **9.6-2**, 7.6-2) |
| | | 32(126, 60.7, 0) | 299 | (6.5-6, 3.0-7, **4.8-4**) | 2000 | 410 | (3.7-6, 3.0-5, **1.6-2**) |
| | | 37(151, 70.1, 5) | 390 | (1.0-6, 1.3-9, **2.0-6**) | ——— | | |
| | pobj & Loss | 8.30947890 2, (3.14-1, 1.29-1, 1.00, 1.00) | | | 1.67 0, (2.84-1, 1.17-1, 1.00, 1.00) | | |
| | 1000\|249251 | 4(4, 3.0, 0) | 15 | (**8.9-2**, 3.6-2, 7.0-2) | 14 | 12 | (2.6-4, **9.6-2**, 6.7-2) |
| | | 33(178, 143.4, 1) | 2906 | (6.2-6, 1.1-7, **5.0-4**) | 2000 | 1748 | (3.0-6, 3.9-5, **5.7-2**) |
| | | 37(198, 144.8, 5) | 3284 | (1.2-6, 1.4-9, **6.5-6**) | ——— | | |
| | pobj & Loss | 1.66685871 3, (3.23-1, 1.28-1, 1.00, 1.00) | | | 4.99 1, (2.18-1, 1.25-1, 1.00, 1.00) | | |
| ar2 | 500\|61877 | 19(69, 58.1, 0) | 169 | (**5.0-2**, 3.2-4, 7.4-4) | 35 | 8 | (1.7-2, **9.6-2**, 6.2-2) |
| | | 22(90, 55.5, 0) | 191 | (**5.0-3**, 2.2-5, 5.0-5) | 175 | 38 | (9.1-3, **9.5-3**, 3.9-4) |
| | | 23(95, 59.1, 1) | 213 | (2.4-6, 2.5-6, **2.5-6**) | 2000 | 432 | (4.7-13, **7.2-3**, 6.1-4) |
| | pobj & Loss | 6.54974002 2, (2.86-2, 1.23-1, 1.00, 1.00) | | | -2.96-1, (2.86-2, 1.23-1, 1.00, 1.00) | | |
| | 1000\|248752 | 21(93, 99.5, 0) | 1300 | (**8.4-2**, 4.6-4, 1.1-3) | 42 | 37 | (1.2-2, **9.8-2**, 6.3-2) |
| | | 25(125, 94.4, 1) | 1501 | (**7.5-5**, 5.5-6, 3.9-6) | 194 | 171 | (**9.1-3**, 8.5-3, 2.7-4) |
| | | 26(129, 94.8, 2) | 1578 | (**8.6-6**, 2.0-6, 1.7-6) | 2000 | 1747 | (5.6-13, **6.6-3**, 4.1-4) |
| | pobj & Loss | 1.30625650 3, (2.02-2, 1.22-1, 1.00, 1.00) | | | 1.03 0, (2.02-2, 1.23-1, 1.00, 1.00) | | |
| ar3 | 500\|61628 | 19(66, 56.1, 0) | 160 | (**7.6-2**, 3.5-4, 6.7-4) | 35 | 7 | (1.3-2, **9.4-2**, 5.9-2) |
| | | 23(92, 55.8, 1) | 201 | (**1.3-5**, 2.2-6, 2.3-6) | 201 | 42 | (**9.7-3**, 8.1-3, 7.9-4) |
| | | 24(95, 56.0, 2) | 213 | (**5.8-6**, 1.5-6, 1.8-6) | 2000 | 411 | (4.4-13, **8.0-3**, 7.9-4) |
| | pobj & Loss | 6.01294444 2, (2.68-2, 1.17-1, 1.00, 0.99) | | | -9.63-1, (2.71-2, 1.19-1, 1.00, 0.99) | | |
| | 1000\|248253 | 21(86, 94.0, 0) | 1227 | (**6.9-2**, 3.3-4, 6.4-4) | 41 | 36 | (1.0-2, **9.9-2**, 6.1-2) |
| | | 24(108, 92.5, 1) | 1401 | (**4.0-5**, 5.1-6, 5.3-6) | 205 | 178 | (**9.5-3**, 7.7-3, 5.3-4) |
| | | 25(112, 92.6, 2) | 1471 | (**8.4-6**, 1.6-6, 2.0-6) | 2000 | 1734 | (5.3-13, **7.7-3**, 5.2-4) |
| | pobj & Loss | 1.20377195 3, (1.89-2, 1.16-1, 1.00, 1.00) | | | 1.93 0, (1.89-2, 1.17-1, 1.00, 1.00) | | |
| ar4 | 500\|61380 | 19(43, 22.0, 0) | 70 | (**8.3-2**, 3.9-4, 6.4-4) | 35 | 7 | (1.3-2, **9.4-2**, 5.3-2) |
| | | 23(72, 29.4, 1) | 118 | (**1.5-5**, 2.2-6, 1.9-6) | 198 | 41 | (**9.8-3**, 9.0-3, 9.2-4) |
| | | 24(75, 30.9, 2) | 130 | (**5.3-6**, 1.5-6, 1.5-6) | 2000 | 414 | (9.0-12, **6.1-3**, 1.3-3) |
| | pobj & Loss | 5.90682889 2, (2.57-2, 1.17-1, 0.98, 0.97) | | | 5.37-1, (2.58-2, 1.17-1, 0.98, 0.97) | | |
| | 1000\|247755 | 20(67, 75.5, 0) | 950 | (**5.8-2**, 6.2-4, 1.0-3) | 32 | 28 | (4.3-2, **9.8-2**, 5.5-2) |
| | | 23(86, 76.9, 1) | 1126 | (**9.5-5**, 8.7-6, 1.2-5) | 190 | 166 | (**9.7-3**, 8.3-3, 5.9-4) |
| | | 25(94, 82.8, 3) | 1330 | (6.7-7, 2.1-6, **2.7-6**) | 2000 | 1744 | (5.6-13, **8.3-3**, 5.9-4) |
| | pobj & Loss | 1.18272795 3, (1.80-2, 1.16-1, 1.00, 0.99) | | | -3.42-1, (1.81-2, 1.17-1, 1.00, 0.99) | | |
| decay | 500\|57961 | 20(66, 46.4, 0) | 147 | (**5.1-2**, 1.5-4, 1.9-4) | 28 | 6 | (5.5-2, **9.3-2**, 5.0-2) |
| | | 22(77, 45.8, 0) | 161 | (**4.5-3**, 1.8-5, 2.3-5) | 145 | 31 | (9.6-3, **9.6-3**, 1.4-3) |
| | | 23(81, 49.7, 1) | 182 | (**6.3-6**, 1.9-6, 6.4-7) | 2000 | 417 | (2.0-9, **4.3-3**, 2.1-3) |
| | pobj & Loss | 5.00048518 2, (2.33-2, 1.12-1, 0.63, 0.40) | | | -1.37 0, (2.39-2, 1.15-1, 0.63, 0.40) | | |
| | 1000\|240836 | 21(80, 78.0, 0) | 1047 | (**1.9-2**, 1.6-4, 2.0-4) | 32 | 29 | (3.6-2, **9.2-2**, 4.9-2) |
| | | 23(91, 73.6, 0) | 1092 | (**5.0-3**, 2.8-5, 3.6-5) | 140 | 125 | (9.2-3, **9.6-3**, 9.0-4) |
| | | 24(95, 75.0, 1) | 1166 | (**6.3-6**, 2.7-6, 1.3-6) | 2000 | 1754 | (1.4-9, **4.6-3**, 1.4-3) |
| | pobj & Loss | 1.00280524 3, (1.61-2, 1.08-1, 0.68, 0.34) | | | 1.80 0, (1.63-2, 1.10-1, 0.68, 0.34) | | |
| circle | 500\|62125 | 21(66, 15.1, 0) | 64 | (1.9-2, 1.4-4, **9.0-2**) | 15 | 3 | (1.3-3, **8.8-2**, 7.6-2) |
| | | 31(136, 64.1, 1) | 306 | (3.8-6, 2.4-7, **3.8-4**) | 2000 | 412 | (3.9-6, 3.0-5, **1.9-2**) |
| | | 35(155, 71.8, 5) | 381 | (1.2-6, 5.4-9, **8.4-6**) | ——— | | |
| | pobj & Loss | 8.33272097 2, (3.45-1, 1.29-1, 1.00, 1.00) | | | 1.53 0, (2.88-1, 1.14-1, 1.00, 1.00) | | |
| | 1000\|249250 | 4(4, 3.0, 0) | 15 | (**9.0-2**, 3.5-2, 7.0-2) | 14 | 12 | (3.7-4, **9.4-2**, 6.7-2) |
| | | 33(175, 147.8, 1) | 2961 | (8.5-6, 1.2-7, **5.7-4**) | 2000 | 1748 | (3.0-6, 3.9-5, **5.8-2**) |
| | | 37(194, 151.9, 5) | 3397 | (6.8-6, 1.8-9, **8.2-6**) | ——— | | |
| | pobj & Loss | 1.66945708 3, (3.34-1, 1.28-1, 1.00, 1.00) | | | 5.33 1, (2.17-1, 1.27-1, 1.00, 1.00) | | |

node status data ($n = 587$), the estrogen receptor data ($n = 692$), the arabidopsis thaliana data ($n = 834$), the leukemia data ($n = 1255$), and the hereditary breast cancer data ($n = 1869$) tested in [34], which will be abbreviated, respectively, as lymph, ER, arabidopsis, leukemia, and hereditary. For detailed information about these gene data sets, we refer to [34] and the references therein. Since the sparsity structure of the inverse covariance matrices is unknown for these gene expression data sets, we tested (1.5) with $p = 1$ and without explicit sparsity linear constraints.

TABLE 6.4
*Results on synthetic problems. Columnwise groups, $p = 2$.*

| | | LGL | | | ADM | | |
|---|---|---|---|---|---|---|---|
| Prob. | $n\|m$ | Iter | Time | $(R_P, R_D, R_G)$ | Iter | Time | $(R_P, R_D, R_G)$ |
| ar1 | 500\|62126 | 3(3, 3.0, 0) | 2 | (6.3-2, 5.7-2, **7.6-2**) | 15 | 2 | (1.5-3, **9.6-2**, 7.6-2) |
| | | 30(86, 26.2, 1) | 88 | (5.5-6, 2.4-7, **3.7-4**) | 2000 | 272 | (3.3-6, 1.9-5, **1.4-2**) |
| | | 34(101, 28.9, 5) | 108 | (2.4-6, 2.2-9, **3.3-6**) | ——— | | |
| | pobj & Loss | 8.40510928 2, (3.02-1, 1.26-1, 1.00, 1.00) | | | 1.18 0, (2.78-1, 1.18-1, 1.00, 1.00) | | |
| | 1000\|249251 | 2(2, 3.0, 0) | 7 | (8.8-2, **8.8-2**, 6.6-2) | 14 | 10 | (2.6-4, **9.6-2**, 6.7-2) |
| | | 33(95, 31.0, 1) | 594 | (5.4-6, 8.3-8, **3.7-4**) | 2000 | 1450 | (2.9-6, 3.7-5, **5.4-2**) |
| | | 38(111, 33.3, 6) | 736 | (**4.3-6**, 2.5-10, 1.1-6) | ——— | | |
| | pobj & Loss | 1.68754056 3, (3.09-1, 1.24-1, 1.00, 1.00) | | | 4.32 1, (2.17-1, 1.28-1, 1.00, 1.00) | | |
| ar2 | 500\|61877 | 18(25, 6.3, 0) | 16 | (**6.9-2**, 7.5-4, 1.6-3) | 34 | 5 | (1.7-2, **9.5-2**, 6.0-2) |
| | | 20(34, 7.3, 0) | 21 | (**5.3-3**, 9.4-5, 2.0-4) | 163 | 23 | (3.1-4, **9.8-4**, 8.4-4) |
| | | 21(38, 8.5, 1) | 25 | (3.4-6, 2.4-6, **4.2-6**) | 2000 | 271 | (2.2-7, **1.7-4**, 3.1-6) |
| | pobj & Loss | 6.64536766 2, (2.80-2, 1.23-1, 1.00, 0.97) | | | -3.05-1, (2.80-2, 1.22-1, 1.00, 0.98) | | |
| | 1000\|248752 | 20(30, 7.6, 0) | 111 | (**7.2-2**, 5.7-4, 1.2-3) | 40 | 29 | (1.3-2, **9.9-2**, 6.2-2) |
| | | 22(40, 8.5, 0) | 138 | (**6.9-3**, 1.1-4, 2.3-4) | 168 | 124 | (2.4-4, **9.7-4**, 9.5-4) |
| | | 23(44, 9.6, 1) | 163 | (2.7-6, **3.7-6**, 3.0-6) | 2000 | 1450 | (2.4-6, **8.6-5**, 1.7-6) |
| | pobj & Loss | 1.32725189 3, (1.97-2, 1.21-1, 1.00, 0.99) | | | 3.64-1, (1.97-2, 1.21-1, 1.00, 0.99) | | |
| ar3 | 500\|61628 | 18(25, 5.6, 0) | 15 | (**6.8-2**, 1.1-3, 1.9-3) | 34 | 5 | (1.4-2, **9.4-2**, 5.8-2) |
| | | 22(37, 6.3, 0) | 21 | (**6.8-3**, 5.8-5, 1.1-4) | 134 | 18 | (2.8-4, **9.8-4**, 6.8-4) |
| | | 23(41, 7.0, 1) | 25 | (**5.1-6**, 1.7-6, 3.6-7) | 2000 | 270 | (3.8-6, **8.9-5**, 4.2-6) |
| | pobj & Loss | 6.08967487 2, (2.61-2, 1.14-1, 1.00, 0.93) | | | -1.36 0, (2.62-2, 1.15-1, 1.00, 0.92) | | |
| | 1000\|248253 | 19(27, 6.7, 0) | 95 | (**4.2-2**, 9.5-4, 1.7-3) | 40 | 29 | (1.0-2, **9.8-2**, 6.0-2) |
| | | 21(35, 7.4, 0) | 118 | (**9.2-3**, 1.5-4, 2.7-4) | 145 | 105 | (2.3-4, **9.9-4**, 7.9-4) |
| | | 22(39, 8.5, 1) | 141 | (**3.3-6**, 3.1-6, 2.7-6) | 2000 | 1438 | (8.2-7, **8.6-5**, 1.9-6) |
| | pobj & Loss | 1.22030395 3, (1.83-2, 1.13-1, 1.00, 0.95) | | | 8.52-1, (1.82-2, 1.12-1, 1.00, 0.95) | | |
| ar4 | 500\|61380 | 18(25, 5.4, 0) | 15 | (**6.7-2**, 8.6-4, 1.4-3) | 28 | 4 | (6.2-2, **9.0-2**, 5.2-2) |
| | | 21(34, 5.9, 0) | 19 | (**8.9-3**, 8.4-5, 1.3-4) | 99 | 14 | (2.8-4, **9.9-4**, 5.4-4) |
| | | 22(38, 6.6, 1) | 23 | (**5.8-6**, 2.0-6, 1.1-6) | 2000 | 273 | (3.3-6, **8.7-5**, 4.1-6) |
| | pobj & Loss | 5.98152690 2, (2.49-2, 1.14-1, 0.98, 0.92) | | | 1.93-1, (2.48-2, 1.13-1, 0.98, 0.93) | | |
| | 1000\|247755 | 20(29, 6.3, 0) | 99 | (**4.8-2**, 5.3-4, 8.3-4) | 32 | 23 | (3.9-2, **9.1-2**, 5.2-2) |
| | | 22(37, 6.9, 0) | 120 | (**8.2-3**, 7.5-5, 1.2-4) | 127 | 93 | (2.3-4, **9.7-4**, 6.4-4) |
| | | 23(41, 7.7, 1) | 142 | (**3.1-6**, 2.4-6, 6.9-7) | 2000 | 1447 | (7.0-7, **8.5-5**, 1.9-6) |
| | pobj & Loss | 1.19862076 3, (1.74-2, 1.12-1, 1.00, 0.94) | | | -9.90-1, (1.75-2, 1.12-1, 1.00, 0.94) | | |
| decay | 500\|57961 | 15(17, 3.8, 0) | 10 | (**7.0-2**, 8.9-3, 1.2-2) | 28 | 4 | (4.9-2, **8.4-2**, 4.7-2) |
| | | 20(27, 4.4, 0) | 15 | (**7.6-3**, 1.9-4, 2.4-4) | 74 | 11 | (2.0-4, **9.9-4**, 2.5-4) |
| | | 21(31, 4.9, 1) | 18 | (**4.6-6**, 2.3-6, 8.0-7) | 2000 | 276 | (5.6-7, **7.5-5**, 4.0-6) |
| | pobj & Loss | 5.05587081 2, (2.14-2, 9.89-2, 0.64, 0.39) | | | -2.20 0, (2.15-2, 9.90-2, 0.64, 0.39) | | |
| | 1000\|240836 | 14(16, 3.7, 0) | 50 | (**7.2-2**, 5.9-2, 6.2-2) | 31 | 23 | (3.7-2, **9.5-2**, 5.1-2) |
| | | 20(29, 5.1, 0) | 89 | (**5.6-3**, 3.3-4, 4.3-4) | 87 | 65 | (2.5-4, **9.6-4**, 4.6-4) |
| | | 21(33, 5.7, 1) | 109 | (**6.8-6**, 3.6-6, 1.7-6) | 2000 | 1453 | (3.0-6, **3.8-5**, 2.0-6) |
| | pobj & Loss | 1.01223229 3, (1.48-2, 9.48-2, 0.70, 0.32) | | | 3.32-1, (1.48-2, 9.47-2, 0.70, 0.32) | | |
| circle | 500\|62125 | 4(4, 3.0, 0) | 2 | (3.3-2, 4.4-2, **7.7-2**) | 15 | 2 | (1.3-3, **8.8-2**, 7.6-2) |
| | | 30(85, 26.7, 1) | 88 | (8.3-6, 3.5-7, **5.5-4**) | 2000 | 274 | (3.6-6, 2.1-5, **1.7-2**) |
| | | 34(101, 29.9, 5) | 110 | (5.1-6, 3.4-9, **5.2-6**) | ——— | | |
| | pobj & Loss | 8.42717878 2, (3.33-1, 1.26-1, 1.00, 1.00) | | | 1.26 0, (2.82-1, 1.15-1, 1.00, 1.00) | | |
| | 1000\|249250 | 2(2, 3.0, 0) | 7 | (**8.9-2**, 8.8-2, 6.6-2) | 14 | 10 | (3.7-4, **9.4-2**, 6.7-2) |
| | | 33(96, 31.0, 1) | 596 | (4.9-6, 8.6-8, **3.9-4**) | 2000 | 1450 | (2.9-6, 3.7-5, **5.5-2** |
| | | 38(112, 33.1, 6) | 736 | (**3.3-6**, 2.7-10, 1.2-6) | ——— | | |
| | pobj & Loss | 1.68857852 3, (3.09-1, 1.24-1, 1.00, 1.00) | | | 4.77 1, (2.15-1, 1.29-1, 1.00, 1.00) | | |

We set $\omega_\ell \equiv \omega = 0.5$ for all the gene data sets. Detailed comparison results are given in Table 6.6, where all presented quantities have the same meanings as those in section 6.3.

It can be seen from the results in Table 6.6 that for these gene data sets and $p = 1$, (1.5) is somehow easier than the problems tested in section 6.4 because ADM reached the final accuracy requirement $\texttt{Res} < 10^{-5}$ in less than 2000 iterations for the first four data sets, while for the Hereditarybc data set the accuracy obtained by ADM is in the

TABLE 6.5
*Results on synthetic problems. Columnwise groups, $p = \infty$.*

| | | LGL | | | ADM | | |
|---|---|---|---|---|---|---|---|
| Prob. | $n\|m$ | Iter | Time | $(R_P, R_D, R_G)$ | Iter | Time | $(R_P, R_D, R_G)$ |
| ar1 | 500\|62126 | 3(3, 3.0, 0) | 2 | (5.7-2, 5.1-2, **7.6-2**) | 15 | 3 | (1.5-3, **9.6-2**, 7.6-2) |
| | | 31(92, 48.7, 1) | 194 | (1.6-6, 2.5-7, **4.0-4**) | 2000 | 349 | (3.7-6, 2.3-5, **1.7-2**) |
| | | 35(110, 58.6, 5) | 258 | (1.3-6, 5.9-9, **9.2-6**) | ——— | | |
| | pobj & Loss | 8.28721853 2, (3.19-1, 1.34-1, 1.00, 1.00) | | | 1.84 0, (2.87-1, 1.20-1, 1.00, 1.00) | | |
| | 1000\|249251 | 3(3, 3.0, 0) | 11 | (**9.6-2**, 3.9-2, 6.9-2) | 14 | 11 | (2.6-4, **9.6-2**, 6.7-2) |
| | | 33(99, 54.8, 1) | 1094 | (2.1-6, 1.4-7, **6.4-4**) | 2000 | 1603 | (3.0-6, 3.8-5, **5.7-2**) |
| | | 37(117, 67.0, 5) | 1468 | (1.2-6, 2.0-9, **9.0-6**) | ——— | | |
| | pobj & Loss | 1.66394918 3, (3.25-1, 1.32-1, 1.00, 1.00) | | | 5.11 1, (2.19-1, 1.26-1, 1.00, 1.00) | | |
| ar2 | 500\|61877 | 17(24, 10.2, 0) | 27 | (**6.5-2**, 2.1-3, 4.9-3) | 35 | 6 | (1.7-2, **9.5-2**, 6.2-2) |
| | | 21(44, 13.8, 0) | 45 | (**8.6-3**, 6.1-5, 1.4-4) | 175 | 31 | (3.0-4, 8.4-4, **9.7-4**) |
| | | 22(48, 16.0, 1) | 55 | (**5.6-6**, 2.6-6, 4.3-6) | 308 | 55 | (3.1-6, 9.0-6, **9.7-6**) |
| | pobj & Loss | 6.52491442 2, (2.95-2, 1.28-1, 1.00, 1.00) | | | -3.85-1, (2.94-2, 1.28-1, 1.00, 1.00) | | |
| | 1000\|248752 | 21(33, 12.9, 0) | 189 | (**3.1-2**, 4.8-4, 1.1-3) | 42 | 34 | (1.2-2, **9.7-2**, 6.3-2) |
| | | 24(50, 15.5, 0) | 262 | (**1.9-3**, 4.0-5, 9.4-5) | 199 | 160 | (2.1-4, 8.5-4, **9.9-4**) |
| | | 25(54, 17.7, 1) | 322 | (**7.2-6**, 4.2-6, 4.4-6) | 386 | 311 | (2.1-6, 8.8-6, **9.9-6**) |
| | pobj & Loss | 1.30284846 3, (2.07-2, 1.27-1, 1.00, 1.00) | | | 3.96-1, (2.07-2, 1.27-1, 1.00, 1.00) | | |
| ar3 | 500\|61628 | 18(26, 9.7, 0) | 28 | (**8.0-2**, 9.7-4, 1.9-3) | 34 | 6 | (1.5-2, **9.9-2**, 6.3-2) |
| | | 21(42, 12.1, 0) | 41 | (**3.7-3**, 5.6-5, 1.1-4) | 151 | 26 | (3.5-4, **9.9-4**, 9.7-4) |
| | | 22(46, 14.1, 1) | 51 | (**3.8-6**, 2.0-6, 2.1-6) | 273 | 47 | (3.4-6, **9.9-6**, 9.3-6) |
| | pobj & Loss | 5.98991552 2, (2.77-2, 1.23-1, 1.00, 0.99) | | | -1.38 0, (2.79-2, 1.24-1, 1.00, 0.99) | | |
| | 1000\|248253 | 20(31, 11.8, 0) | 169 | (**3.8-2**, 7.4-4, 1.5-3) | 41 | 33 | (1.0-2, **9.9-2**, 6.2-2) |
| | | 24(50, 14.0, 0) | 245 | (**3.6-3**, 3.6-5, 7.1-5) | 172 | 139 | (2.5-4, **9.8-4**, 9.7-4) |
| | | 25(54, 15.5, 1) | 287 | (**6.0-6**, 4.2-6, 2.0-6) | 339 | 273 | (2.4-6, **9.9-6**, 9.4-6) |
| | pobj & Loss | 1.20049312 3, (1.94-2, 1.21-1, 1.00, 1.00) | | | 8.52-1, (1.93-2, 1.21-1, 1.00, 1.00) | | |
| ar4 | 500\|61380 | 19(30, 10.0, 0) | 31 | (**4.6-2**, 4.2-4, 7.1-4) | 34 | 6 | (1.5-2, **9.9-2**, 5.7-2) |
| | | 21(42, 11.8, 0) | 40 | (**5.8-3**, 6.0-5, 1.0-4) | 154 | 27 | (3.7-4, **9.9-4**, 8.3-4) |
| | | 22(46, 13.9, 1) | 50 | (**3.9-6**, 2.2-6, 2.6-6) | 270 | 47 | (3.6-6, **9.9-6**, 7.9-6) |
| | pobj & Loss | 5.88382863 2, (2.65-2, 1.22-1, 0.97, 0.97) | | | 2.08-1, (2.64-2, 1.21-1, 0.97, 0.97) | | |
| | 1000\|247755 | 20(31, 11.1, 0) | 164 | (**3.8-2**, 6.6-4, 1.1-3) | 32 | 25 | (4.3-2, **9.7-2**, 5.6-2) |
| | | 22(39, 13.4, 1) | 221 | (**1.4-4**, 2.4-5, 2.9-5) | 155 | 123 | (2.6-4, **9.9-4**, 8.3-4) |
| | | 23(43, 15.0, 2) | 262 | (4.3-6, **8.8-6**, 6.7-6) | 315 | 250 | (2.6-6, **9.9-6**, 8.0-6) |
| | pobj & Loss | 1.17921219 3, (1.85-2, 1.20-1, 1.00, 0.99) | | | -9.44-1, (1.85-2, 1.20-1, 1.00, 0.99) | | |
| decay | 500\|57961 | 19(28, 6.9, 0) | 25 | (**6.8-2**, 3.2-4, 4.2-4) | 28 | 5 | (5.7-2, **9.0-2**, 5.0-2) |
| | | 21(36, 8.3, 1) | 34 | (**1.5-4**, 4.9-6, 7.0-6) | 73 | 13 | (5.2-4, **9.9-4**, 6.4-4) |
| | | 22(40, 9.3, 2) | 41 | (**5.6-6**, 1.4-6, 9.0-7) | 127 | 23 | (5.0-6, **9.5-6**, 5.9-6) |
| | pobj & Loss | 5.00206096 2, (2.40-2, 1.18-1, 0.63, 0.40) | | | -2.20 0, (2.42-2, 1.18-1, 0.63, 0.40) | | |
| | 1000\|240836 | 20(30, 8.0, 0) | 131 | (**5.7-2**, 4.0-4, 5.4-4) | 32 | 26 | (3.7-2, **9.0-2**, 5.0-2) |
| | | 23(41, 9.3, 1) | 185 | (**3.1-5**, 3.1-6, 2.4-6) | 93 | 76 | (3.6-4, **9.9-4**, 6.4-4) |
| | | 24(45, 10.0, 2) | 216 | (**1.6-6**, 1.2-6, 9.5-7) | 165 | 135 | (3.6-6, **9.7-6**, 6.1-6) |
| | pobj & Loss | 1.00188885 3, (1.65-2, 1.13-1, 0.68, 0.34) | | | 3.50-1, (1.65-2, 1.13-1, 0.68, 0.34) | | |
| circle | 500\|62125 | 3(3, 3.0, 0) | 2 | (**9.0-2**, 5.1-2, 7.5-2) | 15 | 3 | (1.3-3, **8.8-2**, 7.6-2) |
| | | 30(92, 52.0, 1) | 200 | (1.4-6, 5.2-7, **8.7-4**) | 2000 | 346 | (3.9-6, 2.4-5, **2.0-2**) |
| | | 34(110, 67.3, 5) | 283 | (1.5-6, 2.6-9, **4.2-6**) | ——— | | |
| | pobj & Loss | 8.30955347 2, (3.50-1, 1.34-1, 1.00, 1.00) | | | 2.00 0, (2.90-1, 1.17-1, 1.00, 1.00) | | |
| | 1000\|249250 | 3(3, 3.0, 0) | 11 | (**9.7-2**, 3.8-2, 6.9-2) | 14 | 11 | (3.7-4, **9.4-2**, 6.7-2) |
| | | 34(103, 49.1, 0) | 1034 | (1.9-6, 1.5-7, **7.6-4**) | 2000 | 1590 | (3.0-6, 3.8-5, **5.8-2**) |
| | | 40(127, 61.4, 6) | 1491 | (**1.5-6**, 3.0-10, 1.4-6) | ——— | | |
| | pobj & Loss | 1.66643596 3, (3.38-1, 1.32-1, 1.00, 1.00) | | | 5.43 1, (2.17-1, 1.28-1, 1.00, 1.00) | | |

order of $10^{-4}$. LGL also performs stably as it requires no more than 16 PPA iterations for all the tested gene data sets. The total number of Newton equations solved, the average PCG steps for solving each Newton equation, and the total number of outer Newton acceleration steps taken by LGL are also reasonable. In particular, only one or two outer Newton acceleration steps were taken at the final iterations. From Table 6.6, these Newton acceleration steps usually decrease the residue `Res` substantially, e.g., two digits of accuracy was obtained in the final iteration for the last three gene

TABLE 6.6
*Results on gene data sets.*

| $p=1$ | | LGL | | | | ADM | | | |
|---|---|---|---|---|---|---|---|---|---|
| Gene name | $n$ | Iter | Time | $(R_D, R_G)$ | pobj | Iter | Time | $(R_D, R_G)$ | pobj |
| Lymph | 587 | 6(6, 3.0, 0) | 5 | (1.7-2, **7.3-2**) | 8.667588 2 | 45 | 7 | (6.5-2, **9.9-2**) | 8.309654 2 |
| | | 11(13, 3.5, 0) | 10 | (3.6-4, **9.2-4**) | 8.133484 2 | 325 | 54 | (8.0-4, **9.9-4**) | 8.132632 2 |
| | | 14(19, 4.4, 1) | 16 | (**2.4-6**, 2.2-6) | 8.132611 2 | 672 | 112 | (**9.9-6**, 8.0-6) | 8.132604 2 |
| ER | 692 | 6(6, 3.0, 0) | 7 | (3.7-2, **6.5-2**) | 9.545812 2 | 40 | 10 | (7.3-2, **9.8-2**) | 9.493477 2 |
| | | 13(17, 4.3, 0) | 20 | (1.3-4, **7.2-4**) | 9.236315 2 | 422 | 110 | (6.5-4, **9.9-4**) | 9.231078 2 |
| | | 16(24, 7.6, 2) | 39 | (**8.7-7**, 4.0-7) | 9.231049 2 | 947 | 248 | (7.0-6, **9.9-6**) | 9.231042 2 |
| Arabidopsis | 834 | 7(7, 3.1, 0) | 15 | (7.9-3, **5.1-2**) | 1.182199 3 | 57 | 24 | (5.0-2, **9.9-2**) | 1.139305 3 |
| | | 12(18, 5.4, 0) | 35 | (6.0-5, **3.3-4**) | 1.109685 3 | 602 | 245 | (5.0-4, **9.9-4**) | 1.109305 3 |
| | | 13(23, 9.2, 1) | 56 | (**6.2-6**, 8.8-7) | 1.109301 3 | 1292 | 526 | (6.3-6, **9.9-6**) | 1.109300 3 |
| Leukemia | 1255 | 9(10, 3.3, 0) | 57 | (9.5-3, **6.2-2**) | 1.759269 3 | 100 | 123 | (3.8-2, **9.9-2**) | 1.738725 3 |
| | | 15(23, 6.3, 0) | 134 | (8.1-5, **5.2-4**) | 1.698618 3 | 925 | 1144 | (4.0-4, **9.9-4**) | 1.697893 3 |
| | | 16(27, 9.5, 1) | 196 | (**7.9-6**, 1.7-6) | 1.697891 3 | 1935 | 2394 | (5.1-6, **9.9-6**) | 1.697887 3 |
| Hereditarybc | 1869 | 8(9, 3.4, 0) | 139 | (5.4-3, **9.3-2**) | 2.694112 3 | 92 | 332 | (4.1-2, **9.9-2**) | 2.463273 3 |
| | | 14(29, 9.4, 0) | 494 | (7.6-5, **4.5-4**) | 2.373668 3 | 1554 | 5722 | (2.7-4, **9.9-4**) | 2.372595 3 |
| | | 15(34, 15.1, 1) | 759 | (**6.1-6**, 6.1-6) | 2.372587 3 | 2000 | 7388 | (8.8-5, **3.3-4**) | 2.372587 3 |

data sets. The differences in final objective function values obtained by both methods are negligible. From the CPU time results, it is easy to see that LGL is much faster than ADM on these gene data sets.

**6.6. Summary.** From the extensive experimental results presented in sections 6.3–6.5 on both synthetic and real data, we see that the proposed Newton-CG based PPA, together with the outer acceleration by Newton's method, performs very stably and efficiently to obtain solutions of relatively higher accuracy. Specifically, for all the tested problems LGL successfully generated solutions satisfying the accuracy requirement $\texttt{Res} < 10^{-5}$. By appropriately tuning the algorithmic parameters for inner subproblems, the total number of Newton systems solved and the average PCG steps taken for solving each of the Newton system are also reasonable. Aided by the outer Newton acceleration, LGL demonstrated superlinear convergence. Therefore, the outer Newton acceleration was only taken for very few iterations at the final stage of algorithm. In contrast, though easily implementable and having lower cost per iteration, the ADM scheme (5.3) performs very differently for different problem data. In our experiments, the ADM seems to be efficient only for solving random problems where the inverse covariance matrices are well-conditioned. For the deterministic synthetic problems tested in section 6.4, ADM performs poorly in most cases. For the gene expression data, ADM is also much slower than LGL. Even in cases where ADM obtained solutions of relatively higher accuracy, it takes many iterations and thus its overall efficiency can be much inferior to LGL. Based on our extensive experiments, we observed that the performance of ADM is highly sensitive to the penalty parameter $\sigma$, and in many cases ADM performs poorly no matter how we tune the parameter $\sigma$, either manually or adaptively. In contrast, LGL with a unified parameter setting performs efficiently and robustly for all the tested problems. Thus, LGL is a promising algorithm for applications in a much wider class of problem scenarios, especially when solutions of relatively higher accuracy are desired.

**7. Concluding remarks.** We designed a practical implementation of the classical PPA for solving the log-determinant optimization problem with group Lasso regularization. At each iteration, we first solve the dual subproblem with a CG based Newton's method to obtain the dual variables and then update the primal variables via explicit formulas based on the computed dual variables. An outer Newton acceleration strategy is also developed when the iterate is close to the optimal solution,

which is helpful for fast local convergence. Some theoretical results, including convergence of the Newton-CG based PPA and the nonsingularity of the Newton systems, are also presented. Based on the classical augmented Lagrangian function, we also derived an ADM for solving (1.5) via solving its dual problem. Extensive experimental results on both synthetic and real data sets are presented to illustrate the performance of the proposed Newton-CG based PPA and the ADM. These results clearly demonstrated that the Newton-CG based PPA is stable and efficient and, in particular, outperforms the ADM in obtaining solutions of relatively higher accuracy. For some easy problems where the inverse covariance matrices are well-conditioned, or when a low-accuracy solution is sufficient for a certain application, the ADM can be faster than the Newton-CG based PPA.

Finally, we note that given the simplicity and the potential superiority of the ADM in certain situations, in practical implementation it is advantageous to incorporate the ADM into the Newton-CG based PPA to provide an initial point. The ADM initialization stage can be terminated either by a maximum number of iterations or when `Res` is decreased to a certain level. Clearly, a more flexible switching criterion can be used for this initialization stage, e.g., whenever a satisfactory speed of convergence is detected (which can be realized by checking the values of `Res`), one should allow ADM to iterate more steps before switching to the more stable and robust Newton-CG based PPA. This way, the advantages of both methods can be fully adopted into a unified practical implementation for solving the log-determinant optimization problem (1.5). In our implementation of the Newton-CG based PPA, we have incorporated the ADM into initialization. For a fair comparison, however, we did not activate the ADM initialization when computing all the results reported in section 6.

**Acknowledgments.** We are grateful to two anonymous referees for their valuable comments and suggestions which have helped improve the paper. The first author would like to thank Dr. Caihua Chen of Nanjing University for many helpful discussions.

## REFERENCES

[1] F. R. Bach, *Consistency of the group lasso and multiple kernel learning*, J. Mach. Learn. Res., 9 (2008), pp. 1179–1225.

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

[3] J. A. Bilmes, *Factored sparse inverse covariance matrices*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE Computer Society, Washington, D.C., 2000, pp. 1009–1012.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Machine Learing, 3 (2010), pp. 1–122.

[5] E. J. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52 (2006), pp. 489–509.

[6] Z. Chan and D. F. Sun, *Constraint nondegeneracy, strong regularity, and nonsingularity in semidefinite programming*, SIAM J. Optim., 19 (2008), pp. 370–396.

[7] C. H. Chen, B. S. He, and X. M. Yuan, *Matrix completion via an alternating direction method*, IMA J. Numer. Anal., 32 (2012), pp. 227–245.

[8] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput., 20 (1998), pp. 33–61.

[9] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Canadian Mathematical Society Series of Monographs and Advanced Texts, John Wiley & Sons, New York, 1983.

[10] Dahl, Joachim, Vandenberghe, Lieven and Roychowdhury, Vwani, *Covariance selection for nonchordal graphs via chordal embedding*, Optim. Methods Softw., 23(4), (2008), pp. 501–520.
[11] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and Gert R. G. Lanckriet, *A direct formulation for sparse PCA using semidefinite programming*, SIAM Rev., 49 (2007), pp. 434–448.
[12] A. Dempster, *Covariance selection*, Biometrics, 28 (1972), pp. 157–175.
[13] A. Dobra, C. Hans, B. Jones, J. R. Nevins, G. A. Yao, and M. West, *Sparse graphical models for exploring gene expression data*, J. Multivariate Anal., 90 (2004), pp. 196–212.
[14] A. Dobra and M. West, *Bayesian Covariance Selection*, 2004, http://ftp.stat.duke.edu/WorkingPapers/04-23.pdf.
[15] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.
[16] J. Duchi, S. Gould, and D. Koller, *Projected subgradient methods for learning sparse gaussians*, in Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI), 2008.
[17] J. Duchi, S. Shalew-Shwartz, Y. Singer, and T. Chandra, *Efficient projections onto the $\ell_1$-ball for learning in high dimensions*, in Proceedings of the 25th International Conference on Machine Learning, Andrew McCallum and Sam Roweis, eds., Helsinki, Finland, 2008, pp. 272–279.
[18] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Program., 55 (1992), pp. 293–318.
[19] E. Esser, *Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman*, UCLA CAM Report 09-31, 2009.
[20] J. Fan, Y. Feng, and Y. Wu, *Network exploration via the adaptive lasso and SCAD penalties*, Ann. Appl. Stat., 3 (2009), pp. 521–541.
[21] M. Fornasier and H. Rauhut, *Recovery algorithms for vector-valued data with joint sparsity constraints*, SIAM J. Numer. Anal., 46 (2008), pp. 577–613.
[22] J. Friedman, T. Hastie, and R. Tibshirani, *Sparse inverse covariance estimation with the graphical lasso*, Biostatistics, 9 (2008), pp. 432–441.
[23] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2 (1976), pp. 17–40.
[24] Y. Gao and D. F. Sun, *Calibrating least squares semidefinite programming with equality and inequality constraints*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 1432–1457.
[25] R. Glowinski and A. Marrocco, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires*, Rev. Française Automat. Informat. Recherche Opérationnelle, 9 (1975), pp. 41–76.
[26] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320.
[27] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.
[28] J. Honorio and D. Samaras, *Multi-task learning of gaussian graphical models*, in Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 2010, pp. 447–454.
[29] J. Huang, N. Liu, M. Pourahmadi, and L. Liu, *Covariance matrix selection and estimation via penalised normal likelihood*, Biometrika, 93 (2006), pp. 85–98.
[30] B. Kummer, *Newton's method for nondifferentiable functions*, in Advances in Mathematical Optimization, Math. Res. 15, Akademie-Verlag, Berlin, 1988, pp. 114–125.
[31] S. L. Lauritzen, *Graphical Models*, Oxford Statist. Sci. Ser. 17, Clarendon Press, New York, 1996.
[32] C. Lemaréchal and C. Sagastizábal, *Practical aspects of the Moreau-Yosida regularization: Theoretical preliminaries*, SIAM J. Optim., 7 (1997), pp. 367–385.
[33] H. Li and J. Gui, *Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks*, Biostatistics, 7 (2006), pp. 302–317.
[34] L. Li and K.-C. Toh, *An inexact interior point method for $L_1$-regularized sparse covariance selection*, Math. Program. Comput., 2 (2010), pp. 291–315.
[35] Z. Lu, *Smooth optimization approach for sparse covariance selection*, SIAM J. Optim., 19 (2009), pp. 1807–1827.
[36] Z. Lu, *Adaptive first-order methods for general sparse inverse covariance selection*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2000–2016.
[37] B. M. Marlin and K. P. Murphy, *Sparse Gaussian graphical models with unknown block structures*, in Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada, 2009, pp. 705–712.
[38] B. Martinet, *Régularisation d'inéquations variationnelles par approximations successives*, Rev. Française Informat. Recherche Opérationnelle, 4 (1970), pp. 154–158.

[39] L. Meier, S. van de Geer, and P. Bühlmann, *The group Lasso for logistic regression*, J. Roy. Statist. Soc. Ser. B Stat. Methodol., 70 (2008), pp. 53–71.
[40] N. Meinshausen and P. Buehlmann, *High-dimensional graphs and variable selection with the Lasso*, Ann. Statist., 34 (2006), pp. 1436–1462.
[41] J.-J. Moreau, *Proximitéet dualité dans un espace hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.
[42] Yu. Nesterov, *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547.
[43] Yu. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152.
[44] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, Academic Press, London, 1969, pp. 283–298.
[45] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Program., 58 (1993), pp. 353–367.
[46] R. T. Rockafellar, *Convex analysis*. Princeton Math. Ser. 28, Princeton University Press, Princeton, NJ, 1996.
[47] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.
[48] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.
[49] N. Sagara and M. Fukushima, *A trust region method for nonsmooth convex optimization*, J. Ind. Manag. Optim., 1 (2005), pp. 171–180.
[50] F. Santosa and W. W. Symes, *Linear inversion of band-limited reflection seismograms*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1307–1330.
[51] Y. Shen, Z. W. Wen, and Y. Zhang, *Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization*, Optim. Methods Softw., to appear.
[52] D. F. Sun, J. Sun, and L. Zhang, *The rate of convergence of the augmented Lagrangian method for nonlinear semidefinite programming*, Math. Program., 114 (2008), pp. 349–391.
[53] R. Tibshirani, *Regression shrinkage and selection via the Lasso*, J. Roy. Statist. Soc. Ser. B, 58 (1996), pp. 267–288.
[54] C. Wang, D. F. Sun, and K.-C. Toh, *Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm*, SIAM J. Optim., 20 (2010), pp. 2994–3013.
[55] Z. W. Wen, D. Goldfarb, and W. Yin, *Alternating direciotn augmented Lagrangina methods for semidefinite porgramming*, Math. Program. Comput., 2 (2010), pp. 203–230.
[56] B. Wu, C. Ding, D. F. Sun, and K.-C. Toh, *On the Moreau-Yosida regularization of the vector k-norm related functions*, Optimization, 2011, http://www.optimization-online.org/DB_FILE/2011/03/2978.pdf.
[57] J. F. Yang and X. M. Yuan, *Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization*, Math. Comput., 82 (2013), pp. 301–329.
[58] J. F. Yang and Y. Zhang, *Alternating direction algorithms for* $\ell_1$*-problems in compressive sensing*, SIAM J. Sci. Comput., 33 (2011), pp. 250–278.
[59] J. F. Yang, Y. Zhang, and W. Yin, *A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data*, IEEE J. Selected Topics in Signal Processing, 4 (2010), pp. 288–297.
[60] K. Yosida, *Functional Analysis*, Grundlehren Math. Wiss. 123, Springer-Verlag, Berlin, 1980.
[61] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, J. Roy. Statist. Soc. Ser. B, 68 (2006), pp. 49–67.
[62] M. Yuan and Y. Lin, *Model selection and estimation in the Gaussian graphical model*, Biometrika, 94 (2007), pp. 19–35.
[63] X. M. Yuan, *Alternating direction method of multipliers for covariance selection models*, J. Sci. Comput., 51 (2012), pp. 261–273.
[64] S. Yun, P. Tseng, and K.-C. Toh, *A block coordinate gradient descent method for regularized convex separable optimization and covariance selection*, Math. Program., 129 (2011), pp. 331–355.
[65] P. Zhao, G. Rocha, and B. Yu, *The composite absolute penalties family for grouped and hierarchical variable selection*, Ann. Statist., 37 (2009), pp. 3468–3497.
[66] X.-Y. Zhao, D. F. Sun, and K.-C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim., 20 (2010), pp. 1737–1765.