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 by 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 a 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 negative definiteness of the generalized Hessian matrices of the dual objective function 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 demonstrated to be efficient for some random problems. Numerical results, including comparisons with the ADM, are presented to demonstrate that the proposed Newton-CG based PPA is stable, efficient and, in particular, outperforms the ADM, especially when higher 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

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. In the case when the set of random variables are jointly normally distributed, the graphical model is also known as a Gaussian graphical model.

Let $\{y_i \in \Re^n : i = 1, 2, \ldots, p\}$ be a set of samples independently drawn from an n-variate Gaussian distribution $N(0, \Sigma)$. We assume that the covariance matrix Σ is nonsingular. The goal is to estimate from the given samples the covariance matrix Σ, 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 Σ^{-1} is called covariance selection, which has

*This work was done while the first author was a research fellow at the National University of Singapore.

†Department of Mathematics, Nanjing University, 22 Han-Kou Road, Nanjing, 210093 (jfyang@nju.edu.cn).

‡Department of Mathematics and Risk Management Institute, Department of Mathematics National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (matsundf@nus.edu.sg).

§Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (mattohkc@nus.edu.sg).

diverse applications in, e.g., speech recognition [3], gene network analysis [13], etc. Recently, the covariance selection problem has mainly been studied in the low sample size and high dimensional setting, see [39].

Let $S := \frac{1}{p}\sum_{k=1}^{p} y_k y_k^\top$ be the sample covariance matrix. To estimate Σ^{-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, $\log\det 0 = -\infty$ is assumed wherever it might occur). Unfortunately, the MLE alone usually is 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 [30] proposed a greedy forward-backward cardinality search algorithm to determine the sparsity pattern of Σ^{-1}; Dobra and West [14] considered Bayesian covariance selection via a stochastic algorithm and utilized prior information; Li and Gui [32] applied an intuitive thresholding gradient ascent method to the log-likelihood function to estimate Σ^{-1}; Huang, Liu and Pourahmadi [28] 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 Ω is an index set. In particular, Banerjee, El Ghaoui and d'Aspremont [2] and Yuan and Lin [59] proposed to penalize the MLE by the ℓ_1-norm of X, resulting 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 ℓ_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., [49]. Recently, ℓ_1-regularization has been extensively utilized in various applications including linear regression [52], overcomplete decomposition [8], principal component analysis [11], and compressive sensing [5, 15], etc. More importantly, the ℓ_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 [59] 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 [21, 2, 61], projected subgradient method [16], Nesterov's first-order methods [41, 42] and their variants [2, 34, 35], alternating direction method (ADM) [60], Newton-CG based proximal point algorithm (PPA) [53], and inexact IPM with effective preconditioners [33]. 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/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 and tail, etc.) 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, 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 $\Re^m$. To promote group sparsity, we penalize the MLE by group Lasso regularization, resulting 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 $b \in \Re^m$. 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., [58, 38, 1, 62] for group ℓ_2-regularized (logistic) regression and [16] for group ℓ_∞-regularized covariance selection. A specific example of (1.3) is the multi-task structure learning problem for Gaussian graphical models [27]. Given k arbitrary tasks, the following problem was considered in [27, eq. (3)] to promote a consistent sparsity pattern across different tasks:

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

where, for each t, S_t denotes a given data matrix of size $m \times m$ and $X_{t,ij}$ denotes the (i,j) 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 ω 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., [36]. 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, more general problems 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 ℓ, $\varphi_\ell : \Re^{|g_\ell|} \to \Re$ is a simple, closed proper convex function (local weights can be implicitly included), $\mathcal{A}$ is a generic linear mapping from S^n to $\Re^m$, and $b \in \Re^m$. 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 $\Re^m$ is surjective.*

1.3. Motivation and contributions. Recently, Zhao, Sun and Toh [63] 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 nonsmooth equations. The extensive numerical results presented in [63] 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, 50], where it is shown that under the constraint nondegeneracy conditions the augmented Lagrangian method 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 [53] adopted a similar idea to solve the log-determinant problem (1.2) with additional linear constraints, where the problem is transformed into a smooth one via introducing auxiliary variables. The resulting algorithm was shown to be approximately 2∼20 times faster than the adaptive Nesterov's smoothing method [34], 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 [63, 53] to solving (1.5), which clearly contains a much broader class of problems. Unlike the problem considered in [53], 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. Some theoretical results, including the characterization of the nonsingularity of the generalized Hessian matrices of the dual subproblem, global and local convergence, are also presented. Moreover, based on the dual problem of (1.5) we derive a simple ADM-like algorithm, which can be used to generate a good starting point for our proposed 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. $X \in S^n_{++}$) will be also represented by $X \succeq 0$ (resp. $X \succ 0$) occasionally. The transpose operation of a vector or matrix variable will be 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 sizes will be 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 $\|(X,y)\| = \sqrt{\|X\|^2 + \|y\|^2}$. 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 for solving (1.5). Numerical results, including comparisons with the ADM, 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 $\vartheta:\mathcal{E}\to\Re$ be a closed proper convex function, see e.g., [45]. For any $\beta>0$, the Moreau-Yosida regularization [40, 57] of ϑ is defined by

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

From the closedness and the strong convexity of the objective function, it is easy to show that, for any $x\in\mathcal{E}$, (2.1) has a unique optimal solution, which is well known as the proximal point of x associated with ϑ and will be denoted by $\pi_\vartheta^\beta(x)$, i.e.,

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

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

PROPOSITION 2.1 ([26, 31, 48]). *Let $\vartheta:\mathcal{E}\to\Re$ 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*

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

(ii) *$x^*\in\mathcal{E}$ minimizes ϑ over $\mathcal{E}$ if and only if it minimizes Φ_ϑ^β over $\mathcal{E}$.*

(iii) *π_ϑ^β is globally Lipschitz continuous with modulus* 1*, i.e.,*

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

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 $\phi_\beta^-(x):=(\sqrt{x^2+4\beta}-x)/2$, $x\in\Re$, we define their matrix counterparts by

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

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

PROPOSITION 2.2. *Let $\beta>0$. For any $X\in S^n$ with eigenvalue decomposition $X=Q\,\mathrm{diag}(d_1,d_2,\ldots,d_n)Q^\top$, ϕ_β^+ and ϕ_β^- 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) ϕ_β^+ *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 ϕ_β^+ and ϕ_β^-, while the proofs for (d) and (e) can be found in [53]. □

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 $\Re^n$, 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*

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

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

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 the expression in (2.7). □

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

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

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

The proof of Proposition 2.4 can be easily fulfilled by using the famous Moreau's theorem (see e.g., [45, Theorem 31.5]). A simple proof can also be found in [20].

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

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

We note that in principle the Newton-CG based PPA proposed in this paper is applicable provided that i) φ_ℓ's are simple in the sense that their proximal point mappings have explicit formulas, and ii) the generalized Jacobian matrices of the proximal point mappings can be derived. Clearly, the φ_ℓ's prescribed in Assumption 3 satisfy the two conditions because the projections onto the ℓ_1-, ℓ_2- and ℓ_∞-norm balls all have closed form formulas and the generalized Jacobian matrices of these projection mappings can also be analytically represented. Another example that satisfies the two conditions is the vector k-norm defined by $\|x\|_{(k)} := \sum_{i=1}^k |x|_i^\downarrow$, $x \in \Re^n$, where $|x|^\downarrow$ is a reordering of x such that $|x|_1^\downarrow \ge |x|_2^\downarrow \ge \ldots \ge |x|_n^\downarrow$, see the recent manuscript [54].

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 [37] and is extensively studied in [47, 46]. Roughly, suppose we aim to minimize an objective function $f : \Re^n \to \Re$ and x^k is the current guess of 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 [46] that PPA is closely related to the method of multipliers of Hestenes [25] and Powell [43]. In the following of this section, we first reformulate (1.5) and then apply the PPA.

3.1. The problem reformulation. For each ℓ, 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 ℓ an auxiliary variable $y_\ell \in \Re^{|g_\ell|}$ to take $X_{g_\ell} = \mathcal{P}_\ell X$ out of the function φ_ℓ. Let $s = \sum_{\ell=1}^r |g_\ell|$ be the total number of elements of X involved in the regularization. For convenience, we let $\mathcal{P} := [\mathcal{P}_1; \mathcal{P}_2; \ldots; \mathcal{P}_r] : \Re^{n\times n} \to \Re^s$ and

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

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

$$\begin{aligned}
\min_{X,\, y}\ & \langle S, X\rangle - \log\det X + \varphi(y) && \text{(3.3a)}\\
\text{s.t. } & \mathcal{A}X = b, && \text{(3.3b)}\\
& \mathcal{P}X - y = 0, && \text{(3.3c)}\\
& X \in S^n_+,\ y \in \Re^s. && \text{(3.3d)}
\end{aligned}$$

An advantage of introducing the auxiliary variable $y \in \Re^s$ 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 $\mathcal{L}(X, y, \lambda, \eta) : \Re^{n\times n} \times \Re^s \times \Re^m \times \Re^s \to \Re \cup \{+\infty\}$ associated with (3.3) be defined by

$$\mathcal{L}(X, y, \lambda, \eta) := \begin{cases} \langle S, X\rangle - \log\det X + \varphi(y) - \lambda^\top(\mathcal{A}X - b) - \eta^\top(\mathcal{P}X - y), & \text{if } X \in \mathcal{S}^n_{++},\\ +\infty, & \text{otherwise.}\end{cases} \tag{3.4}$$

Clearly, (3.3) is equivalent to

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

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

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

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 φ^* denotes the convex conjugate of φ (see e.g., [45]). Under Assumption 3, φ^* is actually the indicator function of

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

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 $\eta \in \mathcal{B}$. Since both $\mathcal{A}$ and $\mathcal{P}$ are surjective mappings, it can be shown that $\log\det(S - \mathcal{A}^*\lambda - \mathcal{P}^*\eta)$ is strictly concave in (λ, η). Therefore, the dual problem has a unique optimal solution if one exists. The feasible sets of the primal and the dual problems (1.5) and (3.7) are, respectively, defined by

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

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

For convenience, we define

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

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

In the following, we concentrate on (3.5), to which we apply the PPA. 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_β and z_β be defined in* (3.10) *and Φ^β_φ be the Moreau-Yosida regularization of φ defined in* (3.2). *Then, the Moreau-Yosida regularization of f defined in* (3.6) *is given by*

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

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

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

where the interchange of "min" and "max" follows from [45], and

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

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 $(X^k, y^k) \in \Re^{n\times n} \times \Re^s$, according to (3.1), the next iterate (X^{k+1}, y^{k+1}) generated by PPA satisfies

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

For simplicity, here the proximal parameter β 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

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

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

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

Unfortunately, directly solving (3.14) in practice is by no means an easy task. Our strategy is that, at each iteration, we first (approximately) solve the dual problem (3.16) to obtain the dual variables $(\lambda^{k+1}, \eta^{k+1})$ and then update the primal variables 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 (λ, η), 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, ω_ℓ'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 only solved approximately, we will use the following stopping criteria considered by Rockafellar [47, 46] for the theoretical analysis in Section 4:

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

$$\sup \theta_k(\lambda, \eta) - \theta_k(\lambda^{k+1}, \eta^{k+1}) \le \delta_k^2/2\beta\|(X^{k+1}, y^{k+1}) - (X^k, y^k)\|^2,\ \delta_k \ge 0, \sum_{k=0}^{\infty} \delta_k < \infty; \tag{3.18b}$$

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

In practical implementation, a proximal term $-\frac{1}{2\beta}(\|\lambda - \lambda^k\|^2 + \|\eta - \eta^k\|^2)$ can be added to the dual objective function $\theta_k(\lambda, \eta)$. In fact, this corresponds to the PPA of multipliers considered in [46, Section 5]. Convergence analysis of this improvement can be conducted in a parallel way as for Algorithm 1. More importantly,

adding this proximal term to $\theta_k(\lambda,\eta)$ provides us a feasible way of terminating Algorithm 1. In fact, the function $\hat{\theta}_k(\lambda,\eta) := \theta_k(\lambda,\eta) - \frac{1}{2\beta}(\|\lambda-\lambda^k\|^2 + \|\eta-\eta^k\|^2)$ is strongly concave with modulus $\frac{1}{\beta}$, and thus the following estimation holds:

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

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

$$\|\nabla\hat{\theta}_k(\lambda^{k+1},\eta^{k+1})\| \le \varepsilon_k,\ \varepsilon_k \ge 0, \sum_{k=0}^{\infty}\varepsilon_k < \infty; \tag{3.19a}$$

$$\|\nabla\hat{\theta}_k(\lambda^{k+1},\eta^{k+1})\| \le \delta_k\|(X^{k+1},y^{k+1}) - (X^k,y^k)\|,\ \delta_k \ge 0, \sum_{k=0}^{\infty}\delta_k < \infty. \tag{3.19b}$$

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 practical Newton-CG algorithm (which allows more flexible choice of the proximal parameter) for solving the dual subproblem (3.16).

3.3. Solve the 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 concave function with respect to (λ,η). Therefore, solving (3.16) is equivalent to solving the following nonlinear system

$$F_k(\lambda,\eta) := -\nabla\theta_k(\lambda,\eta) = \begin{bmatrix} \mathcal{A}\phi_\beta^+(W_\beta^k(\lambda,\eta)) - b \\ \mathcal{P}\phi_\beta^+(W_\beta^k(\lambda,\eta)) - \pi_\varphi^\beta(z_\beta^k(\eta)) \end{bmatrix} = 0,\ (\lambda,\eta)\in\Re^m\times\Re^s, \tag{3.20}$$

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 ℓ_p-norm balls (hidden in the proximal point mapping π_φ^β), $\theta_k(\lambda,\eta)$ is not twice continuously differentiable. From Proposition 2.1, π_φ^β is globally Lipschitz continuous. Therefore, according to Rademacher's Theorem, π_φ^β is almost everywhere Fréchet-differentiable in the whole space. Let $\mathcal{D}_{\pi_\varphi^\beta}$ be the set of points where π_φ^β is differentiable, and

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

The set of generalized Jacobian matrices (see e.g., [9]) of π_φ^β is defined by

$$\partial\pi_\varphi^\beta(z) := \mathrm{conv}\{\partial_B\pi_\varphi^\beta(z)\},\ z\in\Re^s, \tag{3.21}$$

where "conv" denotes the convex hull. Fortunately, the projection mappings onto the ℓ_p-norm balls ($p = 1,2,\infty$) (and thus, from Proposition 2.4, the proximal point mappings) are semismooth. That is, for any fixed $\bar{z}$, π_φ^β 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, (3.20) is a semismooth equation and the generalized Newton's method developed in [29, 44] for solving semismooth equations can be applied. In our implementation, we solved (3.20) by the inexact generalized Newton's method described in Algorithm 2.

Simple computation shows that the set of generalized Jacobian matrices $\partial F_k(\lambda^{k,j},\eta^{k,j})$ have 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 our implementation, we subtracted $\epsilon_j I$ from $V^{k,j}$ to ensure that the coefficient matrix is definite. The linear system is then solved by a preconditioned conjugate gradient (PCG) method.

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

$$(V^{k,j} + \epsilon_j I)(d_\lambda; d_\eta) = -F_k(\lambda^{k,j}, \eta^{k,j}) \tag{3.22}$$

to obtain d_λ^j and d_η^j, where $V^{k,j} \in \partial F_k(\lambda^{k,j}, \eta^{k,j})$ and $\epsilon_j > 0$.

Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m for which

$$\theta_k(\lambda^{k,j} + \delta^m d_\lambda^j, \eta^{k,j} + \delta^m d_\eta^j) \geq \theta_k(\lambda^{k,j}, \eta^{k,j}) - \mu\delta^m \langle F_k(\lambda^{k,j}, \eta^{k,j}), (d_\lambda^j; d_\eta^j)\rangle, \tag{3.23a}$$

$$\|\nabla\theta_k(\lambda^{k,j} + \delta^m d_\lambda^j, \eta^{k,j} + \delta^m d_\eta^j)\| \leq c\|\nabla\theta_k(\lambda^{k,j}, \eta^{k,j})\|. \tag{3.23b}$$

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 $j = j + 1$.

3.4. Acceleration by 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 $\nabla\Phi_f^\beta(X, y)$. 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 alterative for $\partial^2\Phi_f^\beta(X, y)$ just as in [63]:

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

where $(D, d) \in \Re^{n\times n} \times \Re^s$, $\partial\pi_\varphi^\beta(z)$ 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, page 75] that $\partial^2\Phi_f^\beta(X,y)[D,d] \subseteq \hat{\partial}^2\Phi_f^\beta(X,y)[D,d]$ for any $(D,d) \in \Re^{n\times n}\times\Re^s$. 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

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

Let W and z be defined as in (3.24). By taking derivatives to $\mathcal{A}\phi_\beta^+(W) - b = 0$ and $\mathcal{P}\phi_\beta^+(W) - \pi_\varphi^\beta(z) = 0$ with respect to (X,y), 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 λ'_X and λ'_y stand for $\partial\lambda(X,y)/\partial X[D]$ and $\partial\lambda(X,y)/\partial y[d]$, respectively, and similarly for η'_X and η'_y. Note that the coefficient matrix of the above linear system is identical to that in (3.22) ($\epsilon_j = 0$). 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 via solving the above linear system. Since $\partial\pi_\varphi^\beta$ can be explicitly computed for the φ_ℓ'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), $(\bar{\lambda}, \bar{\eta})$ be the corresponding multipliers, i.e., the unique optimal solution to the dual problem (3.7) (with Z eliminated), and

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

We will show in subsection 4.1 that the constraint nondegeneracy condition for (3.3) (with a slight reformulation) at $(\bar{X}, \bar{y})$ is equivalent to the positive definiteness of the set of generalized Jacobian matrices $\partial F(\bar{\lambda}, \bar{\eta})$. We also present in subsection 4.2 global and local convergence results of Algorithm 1 based on the classical results in [47, 46].

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

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

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

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

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

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

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

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

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

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

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

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

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

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 analysis for $p = 1$ and $p = \infty$ is similar.

For $p = 2$, direct calculation shows that

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

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

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

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

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

where

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

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

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

where $B_2^{\beta w} = B_2^{\beta w_1} \times \cdots \times B_2^{\beta w_r}$ and $B_2^{\beta w_\ell} := \{v \in \Re^{|z_\ell|} : \|v\| \le \beta w_\ell\}$ for each ℓ. Clearly, we have the following formulas for the projection onto an ℓ_2-norm ball and its generalized Jacobian matrices:

$$\Pi_{B_2^w}(v) = \begin{cases} w\frac{v}{\|v\|}, & \text{if } \|v\| > w, \\ v, & \text{otherwise,} \end{cases} \quad \text{and } \partial\Pi_{B_2^w}(v) = \begin{cases} \frac{w}{\|v\|}\left(I - \frac{vv^\top}{\|v\|^2}\right), & \text{if } \|v\| > w, \\ \left\{I - t\frac{vv^\top}{w^2} : 0 \le t \le 1\right\}, & \text{if } \|v\| = w, \\ I, & \text{if } \|v\| < w. \end{cases} \tag{4.10}$$

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 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 (3.21). Let $d = (d_1, d_2) \in \Re^m \times \Re^s$ and $Jd = 0$ for some $J \in \partial F(\bar\lambda, \bar\eta)$. Then,

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

for some $V \in \partial\pi_\varphi^\beta(\bar z)$, where the first "$\ge$" holds because all eigenvalues of V are less 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 $\alpha \in \Re$. If $\bar y = 0$, then (4.9) implies that $H = \{\mathbf{0}\}$ and thus $d_2 \in \mathcal{V}^\perp$. Otherwise, it holds that $\langle d_2, v\rangle = \sum_{\ell:\,\bar y_\ell \ne 0}\langle (d_2)_\ell, v_\ell\rangle$ (since $H_\ell = \{\mathbf{0}_{|\bar y_\ell|}\}$ for all ℓ such that $\bar y_\ell = 0$), which is equal to 0 if we can show that $(d_2)_\ell = 0$ for all ℓ such that $\bar y_\ell \ne 0$. 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 ℓ. Therefore, for those ℓ such that $\bar y_\ell \ne 0$, 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 $\bar y_\ell \ne 0$, 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 $U_\ell \in \partial\Pi_{B_2^{\beta w_\ell}}(\bar z_\ell)$, $\forall \ell$. Therefore, $Vd_2 = 0$ implies that $(d_2)_\ell = 0$ for all ℓ such that $\bar y_\ell = 0$. 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

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

which implies that $(d_2)_\ell = 0$ if $\bar{y}_\ell \neq 0$. For ℓ 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 $U_\ell \in \{I - tvv^\top/(\beta w_\ell)^2 : 0 \leq t \leq 1\}$ if $\|\beta\bar{\eta}_\ell\| = \beta w_\ell$. By taking $U_\ell \equiv I$ for all ℓ 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 to 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 ℓ_1-norm is componentwise separable and the absolute value is essentially the only norm in $\Re$. 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 ℓ_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_∞, 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 ith component of a vector v (different from the notation y_ℓ, which represents the ℓth block of y). Direct calculation shows that

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

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

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

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 "$\circ$" represents componentwise multiplication and, for each ℓ, $I_\ell := \{i : |(y_\ell)_i| = \|y_\ell\|_\infty\}$. 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

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

4.2. Convergence results. In this subsection we first establish a lemma, which together with the classical results in [47, 46] ensures the global convergence of Algorithm 1. The local convergence rate of Algorithm 1 can also be directly derived from the results in [47, 46]. For completeness, we shall merely present the convergence results below but omit their proofs.

LEMMA 4.2. *Let π_f^β and Θ_β be defined in* (3.14) *and* (3.11)*, respectively. Then, (X^{k+1}, y^{k+1}) generated by Algorithm 1 satisfies*

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

Proof. From (3.13), $\Theta_\beta(X, y, \lambda, \eta)$ is the Moreau-Yosida regularization of $\mathcal{L}(\cdot,\cdot,\lambda,\eta)$ for fixed (λ,η). Thus, $\Theta_\beta(X, y, \lambda, \eta)$ is convex in (X, y). Furthermore, it is easy to show from Proposition 2.2 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 $(X, y) \in \Re^{n\times n} \times \Re^s$, it holds that

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

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 Θ_β 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

$$\begin{aligned}
\Phi_f^\beta(X^k, y^k) - \theta_k(\lambda^{k+1}, \eta^{k+1}) &\geq \frac{1}{\beta}\left\langle (X^k - X^{k+1}, y^k - y^{k+1}), (X - X^k, y - y^k)\right\rangle \\
&+ \frac{1}{2\beta}\left(\|\pi_f^\beta(X^k, y^k) - (X^k, y^k)\|^2 - \|\pi_f^\beta(X^k, y^k) - (X, y)\|^2\right), \\
&= \frac{1}{2\beta}\left(2\left\langle \pi_f^\beta(X^k, y^k) - (X^{k+1}, y^{k+1}), (X, y) - (X^k, y^k)\right\rangle - \|(X, y) - (X^k, y^k)\|^2\right).
\end{aligned} \tag{4.17}$$

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

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

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

For $(X, y) \in \Re^{n\times n} \times \Re^s$, we define

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

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

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

The following results on global and local convergence of Algorithm 1 follow directly from (4.18), [47, Theorem 1] and [46, 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**). *Assume $\mathcal{F}_P \neq \emptyset$ and let Algorithm 1 be executed with stopping criteria* (3.18a) *and* (3.18b). *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*

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

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 $a_{\mathcal{L}}$ ($\ge a_f$), then for all k sufficiently large it holds that*

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

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 alternating direction method. In this section, we derive a simple alternating minimization algorithm, called alternating direction method or ADM, based on the dual problem (3.7). As noted before, under Assumption 3, φ^* 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 $(\lambda, \eta, Z) \in \Re^m \times \mathcal{B} \times S^n_{++}$, $X \in S^n$ 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 augmented Lagrangian method (ALM) or method of multiplier of Hestenes [25] and Powell [43] iterates as follows: given X^0, for $k = 0, 1, 2, \ldots$

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

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

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 (λ, η, Z) and each block is involved in a different structure. To decouple the difficulty caused by the joint minimization with respect to (λ, η, 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 [24] and Gabay and Mercier [22]:

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

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

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

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

Now we elaborate that all the 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 λ^{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 η-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 η^{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 the 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 σ and stopping criterion, etc., 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 [18], ℓ_1-minimizations in compressive sensing [56],

nuclear norm problems in low-rank matrix recovery [7, 51, 55], and many others [4], the ADM has recently attracted a lot of attentions in the signal, image and data processing communities. We note that the classical ADM [24, 22] 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 (λ, η, 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 [17]) 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 (short for "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 (5.3). All the experiments were performed under GNU Linux 2.6.18.2-34-default x86_64 and MATLAB v7.6 (R2008a), running on a Dell desktop with an Intel(R) Xeon(TM) CPU at 3.2 GHz and 3.8 GB of memory.

6.1. A preconditioner. Let $\mathcal{T} : S^n \to S^n$ be defined by $\mathcal{T}(H) = Q(\Gamma \circ (Q^\top H Q))Q^\top$ for $H \in S^n$, where Γ 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 (3.22) (for simplicity, here we shall 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 (3.22) is crucial for the performance of the overall algorithm. To achieve a faster convergence for the CG method to solve (3.22), an effective preconditioner is desired. In our implementation, we designed an easy-to-compute approximate diagonal preconditioner by using an idea first developed in [23].

Let the standard basis of S^n be given by $\{E_{ij} := \alpha_{ij}(e_i e_j^\top + e_j e_i^\top) : 1 \le i \le j \le n\}$, where e_i is the ith unit vector in $\Re^n$, 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 ith 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 $J \in \partial\pi_\varphi^\beta(z_\beta)$ in the preconditioner because, for φ_ℓ's prescribed in Assumption 3, the explicit representation of the elements of $\partial\pi_\varphi^\beta(z_\beta)$ 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 the assumption that the interiors of $\mathcal{F}_P$ and $\mathcal{F}_D$ are non-empty, 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 "$\texttt{Res} < \texttt{Tol}$" is 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 β 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 kth iteration. As for the penalty parameter σ in ADM, we add the following note. It is well known that the penalty parameter σ plays an important role for the convergence rate of the ALM scheme (5.2). In general, a larger value of σ leads to a faster convergence of the outer loop. However, extremely large values of σ 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 σ 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 our implementation, the ADM was initialized at $(X^0, Z^0, \eta^0) = (I, I, 0)$. Given the simplicity of the ADM (one eigenvalue decomposition periteration), in our implementation of LGL we used ADM to obtain a rough initial point before switching to the Newton-CG based PPA. This initialization stage is terminated if the condition "$\texttt{Res} < 10^{-2}$" is met or a maximum number of 50 iterations is reached. We also take an outer Newton acceleration step if after a PPA iteration, the condition "$\texttt{Res} < 0.1$" 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 with, we describe our procedure for generating random synthetic data, including the inverse covariance matrix Σ^{-1}, the sample covariance matrix S, the group structure G, and the linear constraints $\mathcal{A}X = b$.

For the inverse covariance matrix Σ^{-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 ± 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); A = full(A);
- eig_min = min(eig(A)); ep = max(-1.2*eig_min, 1E-4); A = A + ep * eye(n);

After generating Σ^{-1} (and thus Σ), we generated $2n$ 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 Ω is a subset of $\mathcal{E}$ (the set of indices of the zero elements of Σ^{-1}). In our experiments, we randomly chose approximately 50% of the elements in $\mathcal{E}$ to form the subset Ω.

In the following, we first present an illustrative example to demonstrate the potential superiority of group Lasso regularization ($p = 2, \infty$) compared to ℓ_1-regularization ($p = 1$) when the inverse covariance matrix possesses a blockwise sparsity structure, and then 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 ∞. 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 to 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 column "iter" for LGL, four numbers are given for each of the selected iterations, where the first one (the one outside of each parenthesis) represents the number of PPA iterations, and the three numbers inside each parenthesis represent, respectively, the total number of Newton systems (3.22)

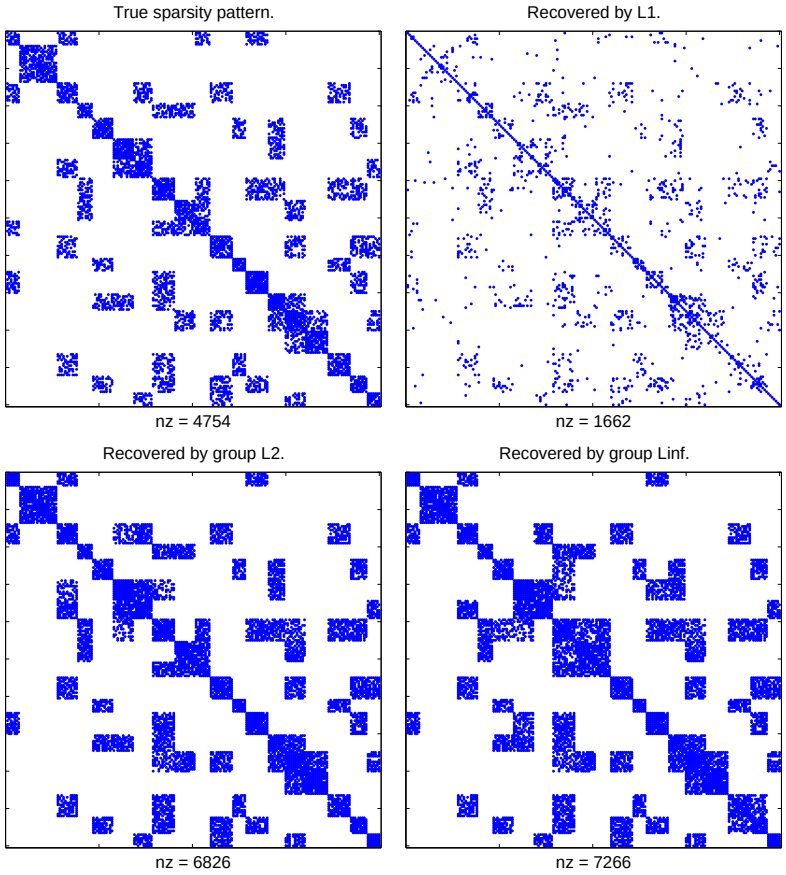

FIG. 6.1. *An illustrative example. From top left to bottom right: the sparsity pattern of the true inverse covariance matrix and those recovered from* (1.5) *with* $p = 1, 2$ *and* ∞*, respectively. Parameters:* $n = 200$*,* $n_g = 20$*,* $p_1 = 0.8$*,* $p_2 = 0.2$ *and* $p_3 = 0.5$*. The regularization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

solved, the average PCG steps taken for solving (3.22) and the total number of outer Newton acceleration steps, i.e., (3.27). We also present the final primal objective function 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 -1.47963560×10^3. 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 3.74×10^{-8} implies that the final primal objective function value obtained by ADM is $-1.47963560 \times 10^3 + 3.74 \times 10^{-8}$. To evaluate how well we have recovered the true inverse covariance matrix, we compute the quadratic loss (Loss_Q) and the normalized entropy loss (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 Σ^{-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 X in relation to that of the true matrix, we borrow some criteria from the machine

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	1(1, 3.0, 0)	28	(1.2-3, 1.8-5, **6.5-3**)	30	17	(1.9-3, **6.7-2**, 2.4-2)
		3(3, 3.0, 2)	48	(3.5-6, 1.8-5, **1.5-4**)	36	20	(**9.7-4**, 4.2-4, 1.3-4)
		5(5, 3.0, 4)	65	(1.4-8, 4.0-7, **1.1-6**)	47	25	(**9.1-6**, 2.1-6, 3.0-7)
	pobj & Loss	-1.47963560 3, (5.48-3, 4.09-3, 1.00, 0.02)			3.74-8, (5.48-3, 4.09-3, 1.00, 0.02)		
	1000|222941	1(1, 3.0, 0)	127	(**3.3-2**, 7.9-4, 3.2-2)	32	72	(3.5-13, **9.7-2**, 3.3-2)
		5(5, 3.0, 4)	351	(1.3-5, 1.0-4, **2.6-4**)	37	85	(1.3-12, **4.9-4**, 1.9-4)
		8(8, 3.0, 7)	477	(2.8-8, **1.2-6**, 3.2-7)	41	96	(1.2-12, **8.7-6**, 2.6-6)
	pobj & Loss	-3.57578598 3, (3.63-3, 4.97-3, 1.00, 0.01)			4.91-7, (3.63-3, 4.97-3, 1.00, 0.01)		
2	500|55072	3(8, 10.7, 0)	58	(**9.4-2**, 1.2-3, 9.6-4)	48	30	(**9.8-2**, 4.1-2, 1.6-2)
		4(11, 22.3, 1)	101	(**1.6-4**, 6.3-6, 2.1-6)	150	94	(**9.7-4**, 3.4-4, 1.3-4)
		5(14, 29.4, 2)	145	(**2.5-6**, 6.1-8, 9.6-9)	259	162	(**9.9-6**, 3.8-6, 1.2-6)
	pobj & Loss	-1.54395177 3, (1.17-2, 2.98-2, 0.74, 0.63)			4.08-7, (1.17-2, 2.98-2, 0.74, 0.63)		
	1000|224590	6(12, 9.0, 0)	390	(**6.0-2**, 6.7-4, 5.4-4)	49	165	(**9.8-2**, 5.5-2, 2.1-2)
		7(16, 39.3, 1)	1035	(**1.1-5**, 3.1-6, 1.4-6)	186	623	(**9.8-4**, 2.7-4, 1.0-4)
		8(19, 50.8, 2)	1442	(**3.6-6**, 9.2-8, 2.7-8)	321	1074	(**9.7-6**, 3.1-6, 9.6-7)
	pobj & Loss	-3.60607928 3, (6.50-3, 1.94-2, 0.77, 0.62)			4.10-7, (6.50-3, 1.94-2, 0.77, 0.62)		
∞	500|54394	7(22, 35.1, 0)	355	(**1.2-2**, 9.0-5, 6.4-5)	53	76	(6.9-2, **9.9-2**, 3.8-2)
		8(27, 43.4, 1)	496	(**7.7-6**, 3.9-6, 1.7-9)	403	574	(7.4-4, **9.9-4**, 4.8-4)
		8(27, 43.4, 1)	496	(**7.7-6**, 3.9-6, 1.7-9)	876	1247	(7.4-6, **9.9-6**, 4.4-6)
	pobj & Loss	-1.62905710 3, (2.43-2, 1.09-1, 0.59, 0.83)			8.02-7, (2.43-2, 1.09-1, 0.59, 0.83)		
	1000|223472	7(15, 14.0, 0)	945	(**6.3-2**, 9.5-4, 9.4-4)	59	390	(8.8-2, **9.8-2**, 3.6-2)
		8(18, 17.4, 1)	1279	(**1.1-4**, 1.9-5, 5.2-6)	497	3279	(**9.9-4**, 8.8-4, 4.2-4)
		9(22, 19.9, 2)	1621	(**5.2-7**, 2.1-7, 3.9-8)	1071	7066	(**9.9-6**, 8.7-6, 4.0-6)
	pobj & Loss	-3.74196300 3, (1.60-2, 9.94-2, 0.58, 0.85)			1.71-6, (1.60-2, 9.94-2, 0.58, 0.85)		

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 each parenthesis behind the primal objective function values represent, respectively, Loss_Q, 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 ADM initialization and acceleration by outer Newton iterations, LGL is able to reach the required accuracy in less than 10 PPA iterations for all the tested random problems. The total number of Newton systems (3.22) solved is at most 27 (for $p = \infty$ and $n = 1000$). The average PCG steps taken for solving (3.22) is 3 for $p = 1$, and this number is increased to about about 50 and 40 for $p = 2$ and $p = \infty$, respectively. Convergence faster than linear rate can also be observed from the results of LGL in Table 6.1, e.g., for $p = 2$ and $p = \infty$, large decreases in `Res` were obtained in 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 could be faster than LGL on some of the

tested random problems for obtaining a solution that satisfies a moderate or high accuracy requirement. The differences in the final objective function values obtained by both methods are negligible. From the results of Loss_Q and 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 indicate 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 [53] (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 [59] and [19]. 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 Ω 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 same meanings as 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 are no more

than 50. The total number of Newton systems (3.22) solved is mostly less than 30, except for the two hard problems **ar1** and **circle** for which this number can be up to several hundreds. We note that problems **ar1** and **circle** are challenging since Σ^{-1} has very small eigenvalues (the minimum eigenvalue is in the order of 10^{-5} to 10^{-6} for $n = 500$ and 1000). The average PCG steps taken for solving (3.22) and the total number of outer Newton acceleration steps for these deterministic synthetic problems are also more than those for the random problems tested in Section 6.3. In contrast, ADM failed to generate solutions satisfying the final accuracy requirement for all the tests after 2000 iterations. Based on our experimental results, the residue values produced by ADM either stagnated or improved very 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 in the order of 10^{-2} to 10^{-3}, while this accuracy is increased by about 1 to 2 digits for the case of columnwise groups. For both 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 LGL consumed longer CPU time for the later case. This is reasonable because the calculations of the proximal point mapping of the ℓ_∞-norm and its generalized Jacobian require projections onto the ℓ_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 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 all but one problem. For the problem **ar2** in Table 6.3, although ADM obtained smaller objective function value for $n = 500$, the primal and the dual residue results of ADM are much worse than those of LGL. For reference purpose, the results of Loss_Q, 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 to the computed solutions, the sparsity pattern of the inverse covariance matrices are recovered very well because the specificity and the sensitivity results are close to one for all these problems except **decay**. The reason that problem **decay** gives worse specificity and sensitivity results is 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 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 [33], which will be abbreviated, respectively, as Lymph, ER, Arabidopsis, Leukemia and Hereditary. For detailed information about these gene data sets, we refer to [33] 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. 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, with 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 order of 10^{-4}. LGL also performs stably as it requires less than 10 PPA iteration

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	16(28, 6.8, 13)	188	(2.9-3, 1.3-2, **9.5-2**)	15	8	(1.4-3, **9.4-2**, 7.6-2)
		29(78, 16.2, 26)	452	(1.0-6, 6.2-7, **9.3-4**)	2000	1069	(3.3-6, 3.0-5, **1.3-2**)
		37(113, 51.2, 34)	1032	(1.7-6, 5.1-9, **7.6-6**)	———		
	pobj & Loss	8.46292260 2, (3.00-1, 1.12-1, 1.00, 1.00)			1.01 0, (2.79-1, 1.07-1, 1.00, 1.00)		
	1000|249251	20(41, 7.3, 19)	1453	(1.6-4, 7.7-4, **9.2-2**)	15	44	(9.0-4, **8.6-2**, 6.9-2)
		35(107, 24.2, 34)	3885	(5.5-6, 1.7-7, **6.8-4**)	2000	5941	(3.2-6, 3.8-5, **5.0-2**)
		43(151, 104.6, 42)	12047	(**9.3-6**, 2.3-9, 9.2-6)	———		
	pobj & Loss	1.70058114 3, (3.21-1, 1.10-1, 1.00, 1.00)			3.49 1, (2.36-1, 1.19-1, 1.00, 1.00)		
ar2	500|61877	1(1, 3.0, 0)	30	(**1.4-1**, 3.2-3, 1.3-2)	34	18	(1.7-2, **9.6-2**, 5.8-2)
		3(5, 12.3, 1)	52	(3.6-5, **2.8-4**, 5.4-5)	2000	1058	(5.0-13, **1.2-2**, 8.8-4)
		4(8, 16.5, 2)	74	(1.7-6, **3.0-6**, 1.9-6)	———		
	pobj & Loss	6.69868993 2, (2.55-2, 1.07-1, 1.00, 0.95)			1.15-1, (2.57-2, 1.08-1, 1.00, 0.95)		
	1000|248752	1(1, 3.0, 0)	166	(**3.1-1**, 4.7-3, 1.9-2)	40	120	(1.2-2, **9.9-2**, 5.9-2)
		6(9, 9.3, 1)	389	(**2.3-5**, 2.3-5, 1.5-5)	2000	5905	(6.0-13, **1.5-2**, 5.6-4)
		7(12, 13.1, 2)	532	(5.0-7, **7.5-7**, 5.5-7)	———		
	pobj & Loss	1.34264941 3, (1.79-2, 1.06-1, 1.00, 0.99)			1.56-1, (1.80-2, 1.06-1, 1.00, 0.99)		
ar3	500|61628	1(1, 3.0, 0)	30	(**1.1-1**, 2.8-3, 9.6-3)	34	18	(1.3-2, **9.3-2**, 5.5-2)
		3(5, 11.3, 1)	50	(2.5-5, **1.9-4**, 4.7-6)	2000	1057	(1.6-11, **8.2-3**, 2.0-3)
		5(8, 11.6, 3)	79	(**4.7-6**, 8.1-8, 3.9-8)	———		
	pobj & Loss	6.17191297 2, (2.37-2, 9.93-2, 1.00, 0.91)			2.15-1, (2.42-2, 1.02-1, 1.00, 0.92)		
	1000|248253	1(1, 3.0, 0)	163	(**2.4-1**, 4.1-3, 1.4-2)	40	118	(9.8-3, **9.8-2**, 5.6-2)
		5(7, 9.6, 1)	346	(4.9-5, **1.9-4**, 1.1-4)	2000	5831	(1.2-12, **1.1-2**, 1.6-3)
		6(10, 13.5, 2)	484	(**5.9-6**, 1.8-6, 5.2-7)	———		
	pobj & Loss	1.23482105 3, (1.65-2, 9.69-2, 1.00, 0.94)			3.70-1, (1.67-2, 9.88-2, 1.00, 0.94)		
ar4	500|61380	1(1, 3.0, 0)	30	(**2.9-2**, 9.4-4, 2.7-3)	28	15	(5.9-2, **9.2-2**, 4.9-2)
		3(6, 15.3, 2)	69	(1.3-5, **7.9-5**, 2.8-5)	2000	1059	(1.3-11, **8.3-3**, 2.2-3)
		4(9, 18.8, 3)	89	(**8.1-6**, 2.1-6, 4.1-7)	———		
	pobj & Loss	6.04705333 2, (2.25-2, 9.78-2, 0.99, 0.92)			2.50-1, (2.31-2, 1.01-1, 0.99, 0.92)		
	1000|247755	1(1, 3.0, 0)	164	(**8.3-2**, 1.7-3, 5.0-3)	32	96	(3.7-2, **9.2-2**, 4.8-2)
		3(5, 11.7, 2)	370	(1.2-4, 2.8-4, **3.3-4**)	2000	5908	(1.3-12, **1.1-2**, 1.8-3)
		5(9, 15.8, 4)	585	(**1.4-6**, 5.0-7, 1.1-7)	———		
	pobj & Loss	1.21368337 3, (1.57-2, 9.61-2, 1.00, 0.93)			4.09-1, (1.60-2, 9.83-2, 1.00, 0.93)		
decay	500|57961	1(1, 3.0, 0)	30	(**1.4-2**, 2.8-4, 4.7-4)	28	23	(4.5-2, **8.8-2**, 4.2-2)
		2(3, 9.5, 1)	49	(7.1-5, **1.1-4**, 3.0-6)	2000	1086	(9.6-7, 2.6-3, **4.7-3**)
		5(8, 15.0, 4)	113	(**2.0-6**, 4.1-7, 3.8-9)	———		
	pobj & Loss	5.11076383 2, (1.89-2, 8.11-2, 0.68, 0.37)			5.14-1, (2.00-2, 8.76-2, 0.69, 0.37)		
	1000|240836	1(1, 3.0, 0)	164	(**2.8-2**, 5.6-4, 1.3-3)	31	139	(3.4-2, **9.9-2**, 4.6-2)
		2(3, 10.0, 1)	277	(1.4-4, **1.8-4**, 8.6-5)	2000	6024	(3.2-11, **7.2-3**, 3.9-3)
		3(6, 16.7, 2)	454	(4.1-6, **9.5-6**, 4.4-7)	———		
	pobj & Loss	1.02992099 3, (1.30-2, 7.79-2, 0.74, 0.30)			8.04-1, (1.35-2, 8.27-2, 0.73, 0.31)		
circle	500|62125	18(34, 7.4, 15)	210	(8.7-5, 8.5-5, **9.7-2**)	15	8	(1.4-3, **9.1-2**, 7.6-2)
		32(88, 19.4, 29)	521	(1.3-6, 4.7-7, **7.3-4**)	2000	1071	(3.4-6, 3.0-5, **1.5-2**)
		39(123, 58.1, 36)	1166	(**9.9-6**, 4.9-9, 7.6-6)	———		
	pobj & Loss	8.48353750 2, (3.00-1, 1.11-1, 1.00, 1.00)			1.29 0, (2.76-1, 1.06-1, 1.00, 1.00)		
	1000|249250	1(1, 3.0, 0)	164	(7.1-5, 7.8-5, **1.0-1**)	14	41	(3.4-4, **9.6-2**, 6.8-2)
		33(100, 27.6, 32)	3874	(2.9-6, 2.2-7, **9.5-4**)	2000	5900	(2.9-6, 3.7-5, **5.2-2**)
		42(165, 158.8, 41)	16415	(**9.9-6**, 2.0-9, 8.7-6)	———		
	pobj & Loss	1.70124497 3, (3.05-1, 1.11-1, 1.00, 1.00)			3.98 1, (2.18-1, 1.23-1, 1.00, 1.00)		

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	16(45, 13.6, 15)	380	(5.8-4, 2.6-3, **9.3-2**)	15	13	(1.4-3, **9.4-2**, 7.6-2)
		32(130, 34.6, 31)	1145	(4.7-6, 4.6-7, **7.1-4**)	2000	1717	(3.8-6, 3.0-5, **1.7-2**)
		39(175, 52.4, 38)	1775	(1.6-6, 6.3-9, **9.7-6**)	———		
	pobj & Loss	8.30496132 2, (3.22-1, 1.28-1, 1.00, 1.00)			1.79 0, (2.91-1, 1.16-1, 1.00, 1.00)		
	1000|249251	1(1, 3.0, 0)	198	(1.4-4, 6.8-5, **9.9-2**)	14	50	(3.0-4, **9.4-2**, 6.7-2)
		37(207, 69.1, 36)	9452	(3.0-6, 1.9-7, **9.3-4**)	2000	7221	(3.0-6, 3.9-5, **5.8-2**)
		47(490, 169.7, 46)	27900	(2.2-6, 1.5-9, **7.3-6**)	———		
	pobj & Loss	1.66820339 3, (3.28-1, 1.29-1, 1.00, 1.00)			5.27 1, (2.18-1, 1.27-1, 1.00, 1.00)		
ar2	500|61877	1(2, 8.0, 0)	50	(**5.3-2**, 2.2-3, 1.4-2)	22	19	(2.3-2, **9.6-2**, 7.5-2)
		5(11, 26.8, 4)	224	(2.2-4, 4.4-5, **3.0-4**)	81	69	(3.0-3, 6.7-3, **9.9-3**)
		8(17, 29.8, 7)	362	(**1.9-6**, 2.4-7, 2.5-7)	2000	1693	(**2.5-4**, 2.5-4, 1.7-8)
	pobj & Loss	5.95021764 2, (3.66-2, 1.68-1, 1.00, 1.00)			-1.43-5, (3.66-2, 1.68-1, 1.00, 1.00)		
	1000|248752	1(2, 8.0, 0)	215	(**8.3-2**, 2.6-3, 1.5-2)	20	72	(6.8-2, **9.5-2**, 7.4-2)
		5(11, 26.8, 4)	1026	(**7.4-4**, 7.6-5, 1.7-4)	99	356	(2.1-3, 6.6-3, **9.8-3**)
		7(15, 29.1, 6)	1419	(**7.3-6**, 7.5-8, 2.5-8)	2000	7100	(**3.5-4**, 2.5-4, 3.2-8)
	pobj & Loss	1.19135635 3, (2.57-2, 1.67-1, 1.00, 1.00)			1.74-5, (2.57-2, 1.67-1, 1.00, 1.00)		
ar3	500|61628	1(1, 3.0, 0)	46	(**1.7-1**, 1.3-3, 9.1-3)	21	18	(2.1-2, **9.9-2**, 7.4-2)
		5(7, 9.8, 1)	96	(**2.8-4**, 6.9-5, 2.5-5)	2000	1674	(**6.3-3**, 1.2-3, 2.2-6)
		7(12, 19.7, 3)	186	(9.9-7, **7.0-6**, 1.4-7)	———		
	pobj & Loss	5.45504085 2, (3.49-2, 1.65-1, 1.00, 1.00)			6.83-4, (3.49-2, 1.65-1, 1.00, 1.00)		
	1000|248253	1(2, 7.0, 0)	210	(**8.5-2**, 2.2-3, 1.1-2)	20	71	(5.4-2, **9.2-2**, 7.1-2)
		5(12, 27.2, 4)	1052	(1.9-5, 3.3-5, **2.1-4**)	2000	7046	(2.7-4, **1.3-3**, 1.0-6)
		8(18, 28.6, 7)	1650	(**7.6-7**, 5.4-8, 6.7-9)	———		
	pobj & Loss	1.09233427 3, (2.43-2, 1.61-1, 1.00, 1.00)			6.41-4, (2.43-2, 1.61-1, 1.00, 1.00)		
ar4	500|61380	1(1, 3.0, 0)	47	(**2.7-1**, 2.8-3, 1.2-2)	35	30	(1.3-2, **9.4-2**, 5.3-2)
		6(14, 41.8, 1)	217	(**9.6-4**, 4.2-5, 1.5-6)	2000	1688	(5.3-12, **5.6-3**, 1.2-3)
		8(23, 68.9, 3)	425	(**7.9-7**, 6.4-7, 1.6-8)	———		
	pobj & Loss	5.89329097 2, (2.57-2, 1.17-1, 0.98, 0.97)			5.59-1, (2.60-2, 1.18-1, 0.98, 0.97)		
	1000|247755	1(1, 3.0, 0)	205	(**2.8-1**, 2.2-3, 8.9-3)	32	119	(4.3-2, **9.8-2**, 5.5-2)
		5(14, 69.0, 1)	1301	(**8.5-4**, 1.0-4, 2.4-5)	2000	7389	(6.4-13, **8.1-3**, 5.7-4)
		7(26, 112.6, 3)	2696	(**6.0-6**, 6.2-7, 3.3-8)	———		
	pobj & Loss	1.18230793 3, (1.80-2, 1.16-1, 1.00, 0.99)			7.24-1, (1.81-2, 1.16-1, 1.00, 0.99)		
decay	500|57961	1(1, 3.0, 0)	47	(**4.8-2**, 6.1-4, 1.1-3)	28	32	(5.5-2, **9.3-2**, 5.0-2)
		3(12, 75.3, 2)	217	(**5.1-5**, 1.6-5, 5.4-7)	2000	1707	(2.1-9, **4.2-3**, 2.1-3)
		5(19, 79.8, 4)	365	(**6.0-6**, 8.5-7, 1.3-8)	———		
	pobj & Loss	4.97737320 2, (2.35-2, 1.13-1, 0.62, 0.40)			8.59-1, (2.39-2, 1.16-1, 0.62, 0.40)		
	1000|240836	1(1, 3.0, 0)	206	(**1.1-1**, 9.0-4, 2.6-3)	32	163	(3.6-2, **9.2-2**, 4.9-2)
		3(15, 155.3, 1)	1662	(1.4-4, **1.6-4**, 3.7-5)	2000	7367	(1.3-12, **6.8-3**, 1.2-3)
		5(22, 125.2, 3)	2381	(**6.9-6**, 2.3-7, 6.3-9)	———		
	pobj & Loss	1.00403411 3, (1.61-2, 1.09-1, 0.68, 0.34)			1.11 0, (1.62-2, 1.09-1, 0.68, 0.34)		
circle	500|62125	16(39, 11.5, 15)	352	(1.2-3, 4.0-3, **7.5-2**)	15	13	(1.4-3, **9.1-2**, 7.6-2)
		31(124, 36.3, 30)	1146	(9.9-7, 4.0-7, **6.6-4**)	2000	1709	(3.9-6, 3.0-5, **1.9-2**)
		38(188, 73.4, 37)	2237	(2.3-6, 4.4-9, **7.3-6**)	———		
	pobj & Loss	8.32534972 2, (3.22-1, 1.27-1, 1.00, 1.00)			2.22 0, (2.88-1, 1.14-1, 1.00, 1.00)		
	1000|249250	1(1, 3.0, 0)	209	(1.5-4, 7.0-5, **1.0-1**)	14	50	(2.7-4, **9.5-2**, 6.7-2)
		35(176, 51.5, 34)	7375	(2.9-6, 1.7-7, **8.0-4**)	2000	7205	(3.0-6, 3.9-5, **5.7-2**)
		44(404, 128.0, 43)	20837	(8.6-6, 2.0-9, **9.4-6**)	———		
	pobj & Loss	1.66808468 3, (3.23-1, 1.29-1, 1.00, 1.00)			5.09 1, (2.18-1, 1.25-1, 1.00, 1.00)		

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	16(26, 5.9, 13)	170	(1.8-2, 2.0-2, **9.0-2**)	15	8	(1.4-3, **9.4-2**, 7.6-2)
		26(62, 14.5, 23)	346	(6.7-6, 1.2-6, **9.0-4**)	2000	1021	(3.5-6, 2.0-5, **1.5-2**)
		35(97, 18.9, 32)	535	(8.4-6, 7.2-9, **8.5-6**)	———		
	pobj & Loss	8.40387649 2, (3.10-1, 1.25-1, 1.00, 1.00)			1.29 0, (2.84-1, 1.17-1, 1.00, 1.00)		
	1000|249251	18(33, 6.2, 17)	1247	(1.6-3, 3.0-3, **8.1-2**)	15	43	(9.0-4, **8.6-2**, 6.9-2)
		38(115, 19.6, 37)	3726	(8.6-6, 2.3-7, **9.6-4**)	2000	5760	(3.2-6, 3.7-5, **5.1-2**)
		47(161, 36.8, 46)	6269	(3.4-6, 2.1-9, **8.7-6**)	———		
	pobj & Loss	1.68866255 3, (3.31-1, 1.23-1, 1.00, 1.00)			3.85 1, (2.38-1, 1.24-1, 1.00, 1.00)		
ar2	500|61877	1(1, 3.0, 0)	29	(**1.6-1**, 3.4-3, 1.5-2)	34	18	(1.7-2, **9.6-2**, 6.0-2)
		3(5, 9.3, 1)	47	(2.7-5, **3.1-4**, 1.0-4)	160	83	(3.2-4, **9.9-4**, 8.8-4)
		4(8, 12.5, 2)	63	(2.0-6, **3.7-6**, 2.0-6)	2000	1017	(8.2-6, **8.8-5**, 3.5-6)
	pobj & Loss	6.62841394 2, (2.80-2, 1.22-1, 1.00, 0.96)			8.96-5, (2.80-2, 1.23-1, 1.00, 0.96)		
	1000|248752	1(1, 3.0, 0)	163	(**3.3-1**, 5.0-3, 2.1-2)	40	118	(1.3-2, **9.9-2**, 6.2-2)
		5(7, 6.6, 1)	300	(1.6-5, **1.5-4**, 4.9-5)	168	495	(2.4-4, **9.8-4**, 9.6-4)
		6(10, 9.3, 2)	407	(7.7-7, **1.6-6**, 1.4-6)	2000	5820	(2.4-6, **8.5-5**, 1.6-6)
	pobj & Loss	1.32854297 3, (1.97-2, 1.22-1, 1.00, 0.99)			7.68-5, (1.97-2, 1.22-1, 1.00, 0.99)		
ar3	500|61628	1(1, 3.0, 0)	29	(**1.2-1**, 3.0-3, 1.1-2)	34	18	(1.4-2, **9.3-2**, 5.8-2)
		3(5, 8.3, 1)	45	(1.8-5, **1.9-4**, 5.3-5)	136	76	(3.0-4, **9.9-4**, 7.1-4)
		4(8, 11.3, 2)	62	(7.0-7, **1.6-6**, 8.4-7)	2000	1059	(4.0-6, **8.7-5**, 4.0-6)
	pobj & Loss	6.09508729 2, (2.62-2, 1.16-1, 1.00, 0.92)			8.21-5, (2.62-2, 1.16-1, 1.00, 0.92)		
	1000|248253	1(1, 3.0, 0)	161	(**2.7-1**, 4.4-3, 1.6-2)	40	116	(1.0-2, **9.8-2**, 6.0-2)
		4(6, 6.8, 1)	274	(2.2-5, **1.6-4**, 4.8-5)	144	417	(2.4-4, **9.9-4**, 8.1-4)
		5(9, 9.4, 2)	371	(8.8-7, **1.5-6**, 1.4-6)	2000	5729	(7.8-7, **8.2-5**, 1.8-6)
	pobj & Loss	1.21955557 3, (1.83-2, 1.13-1, 1.00, 0.95)			7.05-5, (1.83-2, 1.13-1, 1.00, 0.95)		
ar4	500|61380	1(1, 3.0, 0)	29	(**3.4-2**, 1.1-3, 3.3-3)	28	14	(6.2-2, **9.1-2**, 5.2-2)
		2(3, 7.5, 1)	44	(1.4-4, **8.8-4**, 8.2-4)	99	51	(2.8-4, **9.9-4**, 5.5-4)
		4(8, 11.5, 3)	75	(5.4-7, **3.8-6**, 3.4-6)	2000	1016	(3.1-6, **8.4-5**, 3.9-6)
	pobj & Loss	5.97035845 2, (2.50-2, 1.14-1, 0.98, 0.92)			7.55-5, (2.50-2, 1.14-1, 0.98, 0.92)		
	1000|247755	1(1, 3.0, 0)	162	(**9.4-2**, 1.9-3, 5.9-3)	32	93	(3.9-2, **9.1-2**, 5.2-2)
		3(5, 8.7, 2)	333	(1.1-4, **9.4-4**, 3.9-4)	126	367	(2.4-4, **9.9-4**, 6.5-4)
		5(10, 11.0, 4)	498	(**4.2-6**, 2.1-7, 8.6-8)	2000	5750	(6.7-7, **8.2-5**, 1.8-6)
	pobj & Loss	1.19823056 3, (1.74-2, 1.12-1, 1.00, 0.94)			7.13-5, (1.74-2, 1.12-1, 1.00, 0.94)		
decay	500|57961	1(1, 3.0, 0)	26	(**8.9-3**, 5.0-4, 1.2-3)	28	22	(4.9-2, **8.4-2**, 4.7-2)
		2(3, 5.5, 1)	39	(7.2-5, **1.6-4**, 1.2-4)	73	53	(2.2-4, **9.9-4**, 2.8-4)
		4(7, 6.3, 3)	70	(**2.0-6**, 1.3-6, 1.0-6)	2000	1059	(4.8-7, **7.2-5**, 3.8-6)
	pobj & Loss	5.03351603 2, (2.16-2, 9.92-2, 0.63, 0.39)			3.91-5, (2.16-2, 9.92-2, 0.63, 0.39)		
	1000|240836	1(1, 3.0, 0)	162	(**2.0-2**, 7.6-4, 1.8-3)	31	138	(3.7-2, **9.6-2**, 5.1-2)
		2(3, 5.5, 1)	241	(1.6-4, **2.4-4**, 1.7-4)	87	348	(2.5-4, **9.6-4**, 4.7-4)
		4(7, 6.5, 3)	445	(**1.8-6**, 1.7-6, 1.4-6)	2000	6027	(3.0-6, **3.7-5**, 1.9-6)
	pobj & Loss	1.01337677 3, (1.48-2, 9.53-2, 0.70, 0.32)			3.87-5, (1.48-2, 9.53-2, 0.70, 0.32)		
circle	500|62125	20(39, 8.0, 17)	225	(8.1-5, 1.2-4, **9.3-2**)	15	8	(1.4-3, **9.1-2**, 7.6-2)
		33(88, 16.7, 30)	488	(2.8-6, 3.7-7, **6.3-4**)	2000	1020	(3.6-6, 2.1-5, **1.6-2**)
		40(118, 22.7, 37)	677	(1.3-6, 3.9-9, **6.7-6**)	———		
	pobj & Loss	8.42363441 2, (3.10-1, 1.24-1, 1.00, 1.00)			1.63 0, (2.82-1, 1.15-1, 1.00, 1.00)		
	1000|249250	1(1, 3.0, 0)	159	(7.1-5, 7.8-5, **1.0-1**)	14	41	(3.4-4, **9.6-2**, 6.8-2)
		32(96, 21.6, 31)	3236	(4.1-6, 1.5-7, **7.0-4**)	2000	5797	(2.9-6, 3.7-5, **5.4-2**)
		41(141, 44.0, 40)	6038	(3.8-6, 1.8-9, **8.2-6**)	———		
	pobj & Loss	1.68946432 3, (3.15-1, 1.24-1, 1.00, 1.00)			4.36 1, (2.20-1, 1.28-1, 1.00, 1.00)		

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	20(41, 10.4, 19)	347	(2.3-4, 1.3-4, **9.4-2**)	15	10	(1.4-3, **9.4-2**, 7.6-2)
		34(98, 27.6, 33)	833	(7.7-6, 5.9-7, **9.8-4**)	2000	1361	(3.9-6, 2.3-5, **1.8-2**)
		42(130, 42.3, 41)	1284	(**5.7-6**, 3.2-9, 5.3-6)	———		
	pobj & Loss	8.28565384 2, (3.26-1, 1.33-1, 1.00, 1.00)			1.96 0, (2.93-1, 1.19-1, 1.00, 1.00)		
	1000|249251	21(49, 11.6, 20)	1933	(2.3-4, 5.2-4, **7.9-2**)	15	49	(9.0-4, **8.6-2**, 6.9-2)
		36(116, 36.7, 35)	5431	(3.5-6, 1.5-7, **6.8-4**)	2000	6588	(3.3-6, 3.9-5, **5.5-2**)
		45(151, 52.0, 44)	8324	(6.7-6, 1.8-9, **7.9-6**)	———		
	pobj & Loss	1.66495354 3, (3.48-1, 1.32-1, 1.00, 1.00)			4.59 1, (2.41-1, 1.22-1, 1.00, 1.00)		
ar2	500|61877	1(1, 3.0, 0)	38	(**3.4-1**, 3.4-3, 1.9-2)	35	24	(1.7-2, **9.5-2**, 6.3-2)
		6(9, 10.3, 1)	90	(**1.2-4**, 4.1-5, 5.7-6)	173	119	(3.0-4, 8.6-4, **9.9-4**)
		8(12, 13.4, 3)	147	(**2.9-6**, 3.3-8, 1.3-8)	307	211	(3.0-6, 9.1-6, **9.7-6**)
	pobj & Loss	6.50721373 2, (2.95-2, 1.28-1, 1.00, 1.00)			5.25-6, (2.95-2, 1.28-1, 1.00, 1.00)		
	1000|248752	1(1, 3.0, 0)	182	(**6.1-1**, 5.0-3, 2.6-2)	42	139	(1.2-2, **9.7-2**, 6.3-2)
		5(7, 10.4, 1)	414	(**5.3-4**, 1.3-4, 3.0-5)	200	661	(2.1-4, 8.5-4, **9.8-4**)
		7(11, 15.7, 3)	735	(**7.3-7**, 7.4-8, 2.7-8)	388	1282	(2.1-6, 8.8-6, **9.8-6**)
	pobj & Loss	1.30413917 3, (2.07-2, 1.27-1, 1.00, 1.00)			5.48-6, (2.07-2, 1.27-1, 1.00, 1.00)		
ar3	500|61628	1(1, 3.0, 0)	37	(**3.0-1**, 3.1-3, 1.5-2)	34	23	(1.5-2, **9.9-2**, 6.2-2)
		5(7, 8.4, 1)	75	(**1.9-4**, 1.2-4, 2.3-5)	152	103	(3.6-4, **9.9-4**, 9.8-4)
		7(10, 11.0, 3)	121	(**9.4-6**, 6.4-8, 4.9-8)	272	185	(3.4-6, **9.9-6**, 9.2-6)
	pobj & Loss	5.99611444 2, (2.79-2, 1.24-1, 1.00, 0.99)			4.05-6, (2.79-2, 1.24-1, 1.00, 0.99)		
	1000|248253	1(1, 3.0, 0)	180	(**5.3-1**, 4.5-3, 2.1-2)	41	133	(1.0-2, **9.9-2**, 6.2-2)
		5(7, 8.8, 1)	385	(**3.0-4**, 1.2-4, 2.7-5)	171	556	(2.5-4, **9.9-4**, 9.8-4)
		7(11, 12.7, 3)	665	(**7.6-7**, 2.0-7, 5.8-8)	338	1100	(2.4-6, **9.8-6**, 9.4-6)
	pobj & Loss	1.19977822 3, (1.94-2, 1.21-1, 1.00, 1.00)			4.28-6, (1.94-2, 1.21-1, 1.00, 1.00)		
ar4	500|61380	1(1, 3.0, 0)	38	(**2.8-1**, 2.9-3, 1.3-2)	34	23	(1.5-2, **9.9-2**, 5.7-2)
		5(7, 8.4, 1)	75	(**1.4-4**, 9.9-5, 1.4-5)	155	106	(3.6-4, **9.7-4**, 8.1-4)
		7(10, 10.7, 3)	121	(**5.9-6**, 6.0-8, 4.6-8)	271	185	(3.6-6, **9.8-6**, 7.9-6)
	pobj & Loss	5.87213916 2, (2.66-2, 1.23-1, 0.97, 0.97)			3.19-6, (2.66-2, 1.23-1, 0.97, 0.97)		
	1000|247755	1(1, 3.0, 0)	182	(**2.9-1**, 2.2-3, 9.3-3)	32	105	(4.3-2, **9.7-2**, 5.6-2)
		4(6, 10.0, 1)	371	(**3.8-4**, 6.5-5, 1.8-6)	154	508	(2.7-4, **9.9-4**, 8.4-4)
		6(9, 12.0, 3)	618	(**7.4-6**, 5.2-8, 1.0-8)	313	1028	(2.6-6, **9.9-6**, 8.1-6)
	pobj & Loss	1.17889557 3, (1.85-2, 1.20-1, 1.00, 0.99)			2.50-6, (1.85-2, 1.20-1, 1.00, 0.99)		
decay	500|57961	1(1, 3.0, 0)	37	(**4.6-2**, 4.8-4, 1.5-3)	28	27	(5.6-2, **9.0-2**, 5.0-2)
		2(3, 9.5, 1)	59	(1.8-4, **3.1-4**, 1.9-4)	73	66	(5.1-4, **9.9-4**, 6.4-4)
		5(7, 10.4, 4)	136	(**1.4-6**, 4.5-8, 1.7-8)	127	110	(5.0-6, **9.5-6**, 5.9-6)
	pobj & Loss	4.97975030 2, (2.42-2, 1.18-1, 0.62, 0.40)			2.37-6, (2.42-2, 1.18-1, 0.62, 0.40)		
	1000|240836	1(1, 3.0, 0)	182	(**1.1-1**, 9.6-4, 2.8-3)	32	153	(3.7-2, **9.0-2**, 5.0-2)
		3(4, 8.0, 1)	353	(**2.7-4**, 1.3-4, 3.7-5)	94	403	(3.4-4, **9.4-4**, 6.1-4)
		5(7, 10.2, 3)	646	(**6.2-6**, 4.3-8, 1.5-9)	166	687	(3.5-6, **9.4-6**, 5.9-6)
	pobj & Loss	1.00299087 3, (1.66-2, 1.13-1, 0.68, 0.34)			9.94-7, (1.65-2, 1.13-1, 0.68, 0.34)		
circle	500|62125	18(39, 12.3, 17)	324	(2.8-4, 1.1-3, **2.2-2**)	15	10	(1.4-3, **9.1-2**, 7.6-2)
		28(81, 31.4, 27)	735	(1.7-6, 4.5-7, **7.9-4**)	2000	1354	(3.9-6, 2.4-5, **1.9-2**)
		35(109, 47.1, 34)	1144	(**8.0-6**, 4.2-9, 7.5-6)	———		
	pobj & Loss	8.30458362 2, (3.26-1, 1.33-1, 1.00, 1.00)			2.42 0, (2.90-1, 1.17-1, 1.00, 1.00)		
	1000|249250	1(1, 3.0, 0)	179	(1.5-4, 7.0-5, **1.0-1**)	14	45	(3.4-4, **9.6-2**, 6.8-2)
		34(108, 40.1, 33)	5317	(3.0-6, 1.1-7, **5.5-4**)	2000	6563	(3.0-6, 3.9-5, **5.7-2**)
		43(143, 55.9, 42)	8237	(8.1-6, 1.7-9, **8.3-6**)	———		
	pobj & Loss	1.66585999 3, (3.31-1, 1.32-1, 1.00, 1.00)			5.14 1, (2.23-1, 1.26-1, 1.00, 1.00)		

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	1(1, 3.0, 0)	8	(**2.4-3**, 1.5-3)	8.145654 2	45	26	(6.5-2, **9.9-2**)	8.309654 2
		2(4, 8.0, 1)	25	(**2.6-4**, 1.1-4)	8.132650 2	325	189	(8.0-4, **9.9-4**)	8.132632 2
		4(11, 12.5, 3)	61	(**1.8-6**, 2.8-7)	8.132605 2	672	391	(**9.9-6**, 8.0-6)	8.132604 2
ER	692	1(1, 3.0, 0)	52	(**3.4-3**, 3.1-3)	9.281929 2	40	37	(7.3-2, **9.8-2**)	9.493477 2
		2(4, 11.0, 1)	83	(**5.9-4**, 8.3-5)	9.231376 2	422	395	(6.5-4, **9.9-4**)	9.231078 2
		5(12, 16.6, 4)	168	(**1.2-6**, 6.1-8)	9.231045 2	947	892	(7.0-6, **9.9-6**)	9.231042 2
Arabidopsis	834	1(1, 3.0, 0)	83	(7.0-3, **3.0-2**)	1.171698 3	57	85	(5.0-2, **9.9-2**)	1.139305 3
		3(7, 17.0, 1)	173	(**7.1-4**, 2.2-4)	1.109350 3	602	902	(5.0-4, **9.9-4**)	1.109305 3
		6(17, 30.8, 4)	398	(**6.6-6**, 1.5-6)	1.109301 3	1292	1934	(6.3-6, **9.9-6**)	1.109300 3
Leukemia	1255	1(1, 3.0, 0)	122	(5.2-3, **3.5-2**)	1.812662 3	100	473	(3.8-2, **9.9-2**)	1.738725 3
		4(10, 20.5, 1)	542	(**1.6-4**, 8.7-6)	1.697978 3	925	4394	(4.0-4, **9.9-4**)	1.697893 3
		6(18, 34.8, 3)	1154	(**4.0-6**, 1.4-6)	1.697887 3	1935	9183	(5.1-6, **9.9-6**)	1.697887 3
Hereditarybc	1869	2(3, 5.5, 0)	502	(4.1-3, **8.2-2**)	2.726494 3	92	1315	(4.1-2, **9.9-2**)	2.463273 3
		6(15, 22.7, 1)	2320	(**3.3-4**, 1.5-4)	2.372671 3	1554	22256	(2.7-4, **9.9-4**)	2.372595 3
		9(30, 45.2, 4)	6086	(**3.4-6**, 2.1-6)	2.372587 3	2000	28635	(8.8-5, **3.3-4**)	2.372587 3

for all the tested gene data sets. The total number of Newton systems solved, the average PCG steps for solving each Newton system, as well as the total number of outer Newton acceleration steps taken by LGL are also reasonable. 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 iteration, LGL demonstrated superlinear convergence when the iterate is close to the optimal solution. In contrast, though easily implementable and has cheaper cost per iteration, the ADM (5.3) performs very differently for different problem data. In our experiments, the ADM seems to be efficient only for solving some easy 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. Even in cases where ADM obtained solutions of relatively higher accuracy, it takes many iterations and thus its overall efficiency can be inferior to LGL. Based on our extensive experiments, we observed that the performance of ADM is very sensitive to the penalty parameter σ, and in many cases ADM performs poorly no matter how we tune the parameter σ, either manually or adaptively. In comparison, LGL with a unified parameter setting performs efficiently and robustly for all the tested problems. Thus LGL is a promising algorithm for applications to a much wider class of problem scenarios.

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 by a CG based Newton's method to obtain 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 alternating direction method 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, 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, ADM could be faster than the Newton-CG based PPA.

Finally, we note that given the simplicity and the potential superiority of the ADM in certain cases, in practical implementation it is advantageous to incorporate ADM into LGL to provide an initial point. In our experiments, we simply terminated the initialization by ADM via a maximum number of 50 iterations or when the condition "`Res` $< 10^{-2}$" is satisfied (the later condition was met within 50 iterations only for those random problems tested in Section 6.3). 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`), we 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).

Acknowledgement. The authors would like to thank Mr. Caihua Chen from Nanjing University for many helpful discussions.

REFERENCES

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

[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.*, 9:485–516, 2008.

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

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

[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(2):489–509, 2006.

[6] Z. Chan and D. F. Sun. Constraint nondegeneracy, strong regularity, and nonsingularity in semidefinite programming. *SIAM J. Optim.*, 19(1):370–396, 2008.

[7] C. H. Chen, B. S. He, and X. M. Yuan. Matrix completion via alternating direction methods. *IMA J. Numerical Analysis, doi: 10.1093/imanum/drq039*.

[8] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

[9] Frank H. Clarke. *Optimization and nonsmooth analysis*. Canadian Mathematical Society Series of Monographs and Advanced Texts. John Wiley & Sons Inc., New York, 1983. A Wiley-Interscience Publication.

[10] J. Dahl, V. Roychowdhury, and L. Vandenberghe. Maximum likelihood estimation of Gaussian graphical models: Numerical implementation and topology selection. *UCLA Preprint*, 2005.

[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 Review*, 49(3):434–448, 2007.

[12] A. Dempster. Covariance selection. *Biometrics*, 28(1):157–175, 1972.

[13] A. Dobra, C. Hans, B. Jones, J. R. Nevins, G. A. Yao, and M. West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196–212, 2004.

[14] A. Dobra and M. West. Bayesian covariance selection. July 2004.

[15] D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, 2006.

[16] J. Duchi, S. Gould, and D. Koller. Projected Subgradient Methods for Learning Sparse Gaussians. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008.

[17] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programming*, 55(3, Ser. A):293–318, 1992.

[18] E. Esser. Applications of lagrangian-based alternating direction methods and connections to split bregman. *UCLA CAM Report 09-31*, 2009.

[19] J. Fan, Y. Feng, and Y. Wu. Network exploration via the adaptive lasso and SCAD penalties. *Ann. Appl. Stat.*, 3(2):521–541, 2009.

[20] M. Fornasier and H. Rauhut. Recovery algorithms for vector-valued data with joint sparsity constraints. *SIAM J. Numer. Anal.*, 46(2):577–613, 2008.

[21] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[22] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. *Computers and Mathematics with Applications*, 2:17–40, 1976.

[23] Y. Gao and D. F. Sun. Calibrating least squares semidefinite programming with equality and inequality constraints. *SIAM J. Matrix Anal. Appl.*, 31(3):1432–1457, 2009.

[24] 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(R-2):41–76, 1975.

[25] M. R. Hestenes. Multiplier and gradient methods. *J. Optimization Theory Appl.*, 4:303–320, 1969.

[26] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms.*, volume 1 and 2. Springer-Verlag, Berlin, Heidelberg, 1993.

[27] J. Honorio and D. Samaras. Multi-task learning of gaussian graphical models. *Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel*, pages 447–454, 2010.

[28] J. Huang, N. Liu, M. Pourahmadi, and L. Liu. Covariance matrix selection and estimation via penalised normal likelihood. *Biometrika*, 93(1):85–98, 2006.

[29] B. Kummer. Newton's method for nondifferentiable functions. In *Advances in mathematical optimization*, volume 45 of *Math. Res.*, pages 114–125. Akademie-Verlag, Berlin, 1988.

[30] S. L. Lauritzen. *Graphical models*, volume 17 of *Oxford Statistical Science Series*. The Clarendon Press Oxford University Press, New York, 1996. Oxford Science Publications.

[31] C. Lemaréchal and C. Sagastizábal. Practical aspects of the Moreau-Yosida regularization: theoretical preliminaries. *SIAM J. Optim.*, 7(2):367–385, 1997.

[32] H. Li and J. Gui. Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2):302–317, 2006.

[33] L. Li and K.-C. Toh. An inexact interior point method for L_1-regularized sparse covariance selection. *Math. Program. Comput.*, 2(3-4):291–315, 2010.

[34] Z. Lu. Smooth optimization approach for sparse covariance selection. *SIAM J. Optim.*, 19(4):1807–1827, 2009.

[35] Z. Lu. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM J. Matrix Anal. Appl.*, 31(4):2000–2016, 2009/10.

[36] B. M. Marlin and K. P. Murphy. Sparse gaussian graphical models with unknown block structures. *Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada*, pages 705–712, 2009.

[37] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Rev. Française Informat. Recherche Opérationnelle*, 4(Ser. R-3):154–158, 1970.

[38] L. Meier, S. van de Geer, and P. Bühlmann. The group Lasso for logistic regression. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 70(1):53–71, 2008.

[39] N. Meinshausen and P. Buehlmann. High-dimensional graphs and variable selection with the Lasso. *Annals of statistics*, 34(3):1436–1462, 2006.

[40] J.-J. Moreau. Proximitéet dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965.

[41] Yu. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[42] Yu. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1, Ser. A):127–152, 2005.

[43] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In *Optimization (Sympos., Univ. Keele,*

Keele, 1968), pages 283–298. Academic Press, London, 1969.

[44] L. Qi and J. Sun. A nonsmooth version of Newton's method. *Math. Programming*, 58(3, Ser. A):353–367, 1993.

[45] R. T. Rockafellar. *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press.

[46] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.*, 1(2):97–116, 1976.

[47] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.*, 14(5):877–898, 1976.

[48] Nobuko Sagara and Masao Fukushima. A trust region method for nonsmooth convex optimization. *J. Ind. Manag. Optim.*, 1(2):171–180, 2005.

[49] F. Santosa and W. W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.

[50] D. F. Sun, J. Sun, and L. Zhang. The rate of convergence of the augmented Lagrangian method for nonlinear semidefinite programming. *Math. Program.*, 114(2, Ser. A):349–391, 2008.

[51] M. Tao and X. M. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM J. Optim.*, 21(1):57–81, 2011.

[52] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society Series B-Methodological*, 58(1):267–288, 1996.

[53] 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(6):2994–3013, 2010.

[54] B. Wu, C. Ding, D. F. Sun, and K.-C. Toh. On the Moreau-Yosida regularization of the vector k-norm related functions. *Optimization online*, 2011.

[55] J. F. Yang and X. M. Yuan. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of Computation, to appear*.

[56] J. F. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM J. Sci. Comput.*, 33(1):250–278, 2011.

[57] K. Yosida. *Functional analysis*, volume 123 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, sixth edition, 1980.

[58] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(1):49–67, 2006.

[59] M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[60] X. M. Yuan. Alternating direction method of multipliers for covariance selection models. *J. Sci. Comput., to appear*.

[61] S. Yun, P. Tseng, and K.-C. Toh. A block coordinate gradient descent method for regularized convex separable optimization and covariance selection. *Math. Program., to appear*.

[62] P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *Ann. Statist.*, 37(6A):3468–3497, 2009.

[63] X.-Y. Zhao, D. F. Sun, and K.-C. Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM J. Optim.*, 20(4):1737–1765, 2010.